Ein neuer Algorithmus zur Berechnung der Klassenzahl algebraischer Zahlkörper

(Diplomarbeit)

Angefertigt von Erik Scheid
am Fachbereich Mathematik
der Universität des Saarlandes
nach einem Thema von
Prof. Dr. J. Buchmann

Saarbrücken 1993

Inhaltsverzeichnis

Einleitung

Drei Hauptziele der algorithmischen algebraischen Zahlentheorie sind die Berechnung der Klassenzahlen, der Klassengruppen und der Regulatoren von algebraischen Zahlkörpern.

Hafner und McCurley stellten in [HaMcC] einen probabilistischen Algorithmus vom Typ Las-Vegas zur Berechnung der Klassengruppen von imaginär-quadratischen Zahlkörpern vor, der unter Annahme der Verallgemeinerten Riemannschen Vermutung (GRH) die erwartete Laufzeit $\exp(\sqrt{\log|\Delta_{\mathcal{F}}| \log\log|\Delta_{\mathcal{F}}|})^{\sqrt{2}+o(1)}$ besitzt. Dieser war die Grundlage für einen von S. Düllmann [Düll] entwickelten und implementierten Algorithmus zur Bestimmung der Klassengruppe imaginär-quadratischer Zahlkörper, der stets zum Ziel führt.

Ausgehend von der Idee von [HaMcC] schlug J. Buchmann 1989 in [Bu1] einen probabilistischen Algorithmus zur Bestimmung der Klassengruppe und des Regulators von beliebigen Zahlkörpern vor, dessen vermutete Laufzeit unter Annahme der GRH subexponentiell in $\log|\Delta_{\mathcal{F}}|$ ist.

Diese Arbeit beinhaltet einen neuen Algorithmus zur Berechnung der Klassenzahl eines beliebigen Zahlkörpers, der auf diesem Algorithmus von Buchmann basiert. Der Algorithmus ist so angelegt, daß auch die Berechnung der Klassengruppe und des Regulators ermöglicht wird.

Ein weiterer Algorithmus zur Bestimmung der Klassengruppen beliebiger Zahlkörper wurde von Pohst und Zassenhaus in [PoZa] veröffentlicht und von J. Graf von Schmettow im Rahmen seiner Diplomarbeit [Schm] implementiert. In seiner Dissertation [Schm1] hat J. Graf von Schmettow diesen Algorithmus weiterentwickelt.

Die vorliegende Arbeit ist in folgende Abschnitte gegliedert:

Im ersten Abschnitt werden die theoretischen und arithmetischen Grundlagen des Algorithmus behandelt. Eine Einführung in das Themengebiet findet man z.B. in [Bu4], [Coh] und [Stew].

Die Grundversion des Algorithmus wird im zweiten Kapitel beschrieben und von ihrer theoretischen Seite betrachtet.

Die Realisierung des Algorithmus ist Inhalt des dritten Abschnittes. Hierbei wird seine Grundversion erheblich verbessert.

Im vierten Kapitel gehen wir kurz auf die Implementierung des Algorithmus ein und zeigen einige Vorteile der Verwendung der objektorientierten Programmiersprache C++ in der algorithmischen algebraischen Zahlentheorie auf.

Das fünfte Kapitel zeigt die Grenzen des Algorithmus auf. Daraus resultiert eine verbesserte Version des Algorithmus und die Idee zu heuristischen Experimenten.

Eine ausführliche Beispielrechnung findet sich im sechsten Kapitel.

Im letzten Abschnitt werden zunächst Tabellen mit Rechenergebnissen zu Körpern vierten bis siebten Grades und anschließend Ergebnisse von Rechnungen, die unter Annahme einer zusätzlichen Vermutung ausgeführt wurden, angegeben.

Der Algorithmus wurde in C++ implementiert. Zur Realisierung effizienter Datenstrukturen setzen wir dabei die C++ Programmbibliothek LEDA (s. [LEDA]) ein. Als lange $\mathbb{Z}$-Arithmetik verwenden wir die an der Universität Heidelberg von R. Dentzer entwickelte libI.

Bezeichnungen

$\mathcal{F}$	algebraischer Zahlkörper
f	erzeugendes Polynom von $\mathcal{F}$
$\mathcal{O}$	Ring der ganzen Zahlen von $\mathcal{F}$
$\Delta_{\mathcal{F}}$	Diskriminante von $\mathcal{F}$
Δ_f	Polynomdiskriminante von f
$d_{\mathcal{F}}$	Führer von $\mathcal{F}$
$\underline{\alpha}$	Konjungiertenvektor von α
M_α	reguläre Darstellungsmatrix von α
n	Polynomgrad von f
s	Anzahl der reellen Nullstellen von f
t	Anzahl der Paare komplexen Nullstellen von f
(n, s, t)	Signatur von $\mathcal{F}$
$\|\alpha\|_i$	i-te archimedische Bewertung von α
$log\alpha$	Logarithmenabbildung von α
I	Menge der Ideale
H	Menge der Hauptideale
Cl	Klassengruppe von $\mathcal{F}$
h	Klassenzahl
R	Regulator von $\mathcal{F}$
$\mathbb{P}$	Menge der Primzahlen in $\mathbb{N}$

Kapitel 1

Grundlagen

1.1 Algebraische Zahlkörper

Ein *algebraischer Zahlkörper* $\mathcal{F}$ ist ein Teilkörper der komplexen Zahlen, der, als Vektorraum über den rationalen Zahlen betrachtet, endlich–dimensional ist.

Nach dem Satz vom primitiven Element ist $\mathcal{F}$ darstellbar als

$$\mathcal{F} = \mathbb{Q}(\theta)$$

für eine *algebraisch ganze Zahl* $\theta \in \mathbb{C}$, wobei algebraisch ganz bedeutet, daß θ Nullstelle eines $\mathbb{Q}$-irreduziblen Polynoms

$$f(x) = x^n + a_{n-1}x^{n-1} + \cdots + a_0 \ \in \mathbb{Z}[x] \setminus \{0\} \tag{1.1}$$

ist. Ein solches Polynom f nennen wir *erzeugendes Polynom* von $\mathcal{F}$.

Sei $(\omega_1, ..., \omega_n)$, $\omega_i \in \mathcal{F}$, wobei o.B.d.A $\omega_1 = 1$, eine $\mathbb{Q}$-Basis von $\mathcal{F}$, dann nennen wir die eindeutige Darstellung von $\alpha \in \mathcal{F}$ in der Form

$$\alpha = \frac{1}{a_{n+1}} \sum_{i=1}^{n} a_i\omega_i, \text{ mit } \operatorname{ggT}(a_1, ..., a_n, a_{n+1}) = 1, \quad a_i \in \mathbb{Z}, 1 \leq i \leq n, \ a_{n+1} \in \mathbb{N}$$

die *Standarddarstellung* von α bzgl. $(\omega_1, \ldots, \omega_n)$.

Ist $\alpha \in \mathcal{F}$ bzgl. der Potenzbasis $(1,\ \theta,\ \theta^2, \ldots,\ \theta^{n-1})$ in Standarddarstellung gegeben und seien $\rho^{(k)} \in \mathbb{C}$, $k = 1, \ldots, n$ die Nullstellen des erzeugenden Polynoms f, davon s reelle und $2t$ komplexe Nullstellen, die so geordnet sind, daß $\rho^{(s+r)} = \overline{\rho^{(s+t+r)}}$ für $r = 1, \ldots, t$. Dann bezeichnen wir mit

$$\alpha^{(i)} = \frac{1}{a_{n+1}} \sum_{j=1}^{n} a_j(\rho^{(i)})^{j-1} \quad , i = 1, \ldots, n,$$

die *i-te Konjungierte von* α und

$$\underline{\alpha} = (\alpha^{(1)}, \ldots, \alpha^{(s)}, Re\ \alpha^{(s+1)}, \ldots, Re\ \alpha^{(s+t)}, Im\ \alpha^{(s+1)}, \ldots, Im\ \alpha^{(s+t)})$$

als den *Konjungiertenvektor von* α. Das Tripel (n, s, t) nennen wir die *Signatur* des Zahlkörpers.

Weiterhin definieren wir für $\alpha \in \mathcal{F}$

$$|\alpha|_i = \left|\alpha^{(i)}\right|^{e_i}$$

mit

$$e_i = \begin{cases} 1 & falls \quad 1 \leq i \leq s \\ 2 & falls \quad s < i \leq s+t. \end{cases}$$

$||_i$ heißt i-te archimedische Bewertung von $\mathcal{F}$.

Dies erlaubt die Einführung eines Logarithmus für algebraische Zahlen, welcher später im Zusammenhang mit der Einheitengruppe von Bedeutung sein wird, um die Einheitengruppe mit einer multiplikativen Struktur auf ein Gitter mit additiver Struktur abzubilden.

$$\begin{array}{rccl} \mathrm{Log}: & \mathcal{F}^\times & \rightarrow & \mathbb{R}^{s+t} \\ & \alpha & \mapsto & (\log|\alpha|_1, \ldots, \log|\alpha|_{s+t}) \end{array}$$

1.2 Arithmetik algebraischer Zahlen

Seien $\alpha, \beta \in \mathcal{F}$ mit

$$\alpha = \frac{1}{a_{n+1}} \sum_{j=1}^{n} a_j\omega_j\ ,\ \beta = \frac{1}{b_{n+1}} \sum_{k=1}^{n} b_k\omega_k$$

in Standarddarstellung bzgl. $(\omega_1, \ldots, \omega_n)$ gegeben.

Die Addition ist elementar:

$$\alpha + \beta = \frac{1}{\mathrm{kgV}(a_{n+1}, b_{n+1})} \sum_{j=1}^{n} (a'_j + b'_j)\,\omega_j$$

mit

$$a'_j = \frac{\mathrm{kgV}(a_{n+1}, b_{n+1})}{a_{n+1}} a_j \text{ und } b'_j = \frac{\mathrm{kgV}(a_{n+1}, b_{n+1})}{b_{n+1}} b_j.$$

Hierbei liegt $\alpha + \beta$ noch nicht notwendig in Standarddarstellung vor.

Die Multiplikation und Division führen wir wie in [Bu5] beschrieben durch:

Zur Multiplikation sei eine *Multiplikationstabelle* $w_{ijk} \in \mathbb{Z}^{n \times n \times n}$ gegeben mit

$$\omega_j \omega_k = \sum_{i=1}^{n} w_{ijk} \omega_i.$$

Das Produkt von α und β ergibt sich als Standarddarstellung von

$$\begin{aligned} \alpha\beta &= \frac{1}{a_{n+1} b_{n+1}} \sum_{j=1}^{n} \sum_{k=1}^{n} a_j b_k \, \omega_j \omega_k \\ &= \frac{1}{a_{n+1} b_{n+1}} \sum_{i=1}^{n} \left(\sum_{j=1}^{n} \sum_{k=1}^{n} a_j b_k w_{ijk} \right) \omega_i. \end{aligned}$$

Sei $\alpha \in \mathcal{F}$ mit

$$\omega_j \alpha = \sum_{i=1}^{n} m_{ij} \omega_i, \quad m_{ij} \in \mathbb{Q}.$$

Dann nennen wir

$$M_\alpha = (m_{ij}) \ \in \mathbb{Q}^{n \times n}$$

die *reguläre Darstellungsmatrix* von α.

Zur Bestimmung des multiplikativen Inversen einer algebraischen Zahl α berechnen wir zunächst die reguläre Darstellungsmatrix von α. Wegen $\omega_1 = 1$ stehen in der ersten Spalte gerade die Koeffizienten der Standarddarstellung von α. Die Abbildung

$$\begin{aligned} \mathcal{F} &\rightarrow \mathbb{Q}^{n \times n} \\ \alpha &\mapsto M_\alpha \end{aligned}$$

ist ein Ringmonomorphismus [Bu5, Kap. 1.1.3]. Daher können wir wegen

$$E = M_{\alpha\alpha^{-1}} = M_\alpha M_{\alpha^{-1}} \Rightarrow M_{\alpha^{-1}} = {M_\alpha}^{-1}$$

die Koeffizienten des Inversen α^{-1} von α in der ersten Spalte von ${M_\alpha}^{-1}$ ablesen.

1.3 Der Ring der ganzen Zahlen $\mathcal{O}$

In einem algebraischen Zahlkörper bilden die algebraisch ganzen Zahlen (vgl. (1.1)) einen Ring, der als $\mathbb{Z}$-Modul frei vom Rang n ist, den *Ring der ganzen Zahlen* $\mathcal{O}$.

Eine $\mathbb{Z}$-Basis $(1, \omega_2, \ldots, \omega_n)$ von $\mathcal{O}$ nennen wir eine *Ganzheitsbasis* und definieren

$$\Delta = \det \left({\omega_i}^{(j)} \right)^2, \ i, j = 1, \ \ldots, \ n$$

als die *Diskriminante* dieser Ganzheitbasis. Da die Matrix eines Basiswechsels unimodular ist, ist der Wert dieser Diskriminante für alle Ganzheitsbasen gleich. Wir nennen $\Delta_{\mathcal{F}} = \Delta$ die *Diskriminante des Zahlkörpers.*

Sei wieder f das erzeugende Polynom der Zahlkörpers F und $f(\rho) = 0$, dann definieren wir

$$\Delta_f = \det \left(\left(\rho^{i-1} \right)^{(j)} \right)^2, \ i, j = 1, \ \ldots, \ n$$

als die *Diskriminante des erzeugenden Polynoms* f und nennen

$$d_{\mathcal{F}} = \sqrt{\left(\frac{\Delta_f}{\Delta_{\mathcal{F}}} \right)}$$

den *Führer* von $\mathcal{F}$.

Von nun an geben wir algebraische Zahlen stets in ihrer (Standard-)Darstellung bzgl. der Ganzheitsbasis $(\omega_1 = 1, \omega_2, \ldots, \omega_n)$ an.

Bemerkung 1.1 Eine Ganzheitsbasis ist auch eine $\mathbb{Q}$-Basis von $\mathcal{F}$.

1.4 Ideale

Ein *gebrochenes Ideal* $\mathbf{a}$ ist ein Teilmodul ungleich (0) von $\mathcal{F}$, derart daß für ein $d \in \mathbb{Z} \setminus \{0\}$ $d\mathbf{a} \subset \mathcal{O}$ ein Ideal ist. Diese gebrochenen Ideale sind als $\mathbb{Z}$-Module ebenfalls frei vom Rang n. $\mathbf{a}$ besitzt daher eine Darstellung

$$\mathbf{a} = \mathbb{Z}a_1 + \cdots + \mathbb{Z}a_n, \quad a_i \in \mathcal{F}. \tag{1.2}$$

Die *Norm* $N(\mathbf{a})$ von $\mathbf{a}$ ist

$$N(\mathbf{a}) = \left| \det \left(a_i \right)_{i=1,\ldots,n} \right|.$$

In dieser Arbeit werden wir den Begriff Ideal im Sinne von gebrochenem Ideal verwenden. Ferner sei I die Menge der Ideale. Ein Ideal mit $\mathbf{a} \subset \mathcal{O}$ nennen wir Ideal *in* $\mathcal{O}$.

Wir verwenden folgende Bezeichnungen:

- Ein Ideal $\mathbf{a}$ mit

  $$\mathbf{a} = \alpha\mathcal{O} \ \text{ für ein } \alpha \in \mathcal{F},$$

  bezeichnen wir wie üblich als *Hauptideal.* H sei die Menge der Hauptideale.

- Ein Ideal $\mathbf{p}$ in $\mathcal{O}$ wird *Primideal* genannt, falls

$$\mathcal{O}/\mathbf{p}$$

  Integritätsbereich ist.

- Ein Ideal $\mathbf{p}$ in $\mathcal{O}$ wird *maximal* genannt, falls

$$\mathcal{O}/\mathbf{p}$$

  Körper ist.

- Für ein Ideal $\mathbf{a}$ heißt $\alpha \in \mathbf{a}\backslash\{0\}$ *Minimum* in $\mathbf{a}$, falls es kein $\beta \in \mathbf{a}\backslash\{0\}$ gibt mit $|\beta|_i < |\alpha|_i$ für alle $1 \leq i \leq n$.

  Ein Ideal $\mathbf{a}$ heißt *reduziert*, falls $1 \in \mathbf{a}$ und 1 ein Minimum in $\mathbf{a}$ ist.

Die folgenden zentralen Sätze sind z.B. in [Coh], [PoZa1], [Stew] gezeigt:

Satz 1.2 *$\mathcal{O}$ ist ein Dedekind-Ring, d.h ein nullteilerfreier, noetherscher Ring, in dem jedes Primideal ein maximales Ideal ist.*

Satz 1.3 *I ist eine multiplikative, abelsche Gruppe. H ist eine Untergruppe von I.*

Satz 1.4 *Jedes Ideal $\mathbf{a} \in I$ besitzt eine eindeutige Zerlegung der Form*

$$\mathbf{a} = \prod_{i=1}^{g} \mathbf{p}_i^{e_i}, \ \mathbf{p}_i \, Primideal, \ e_i \in \mathbb{Z} \setminus \{0\}, \ g \in \mathbb{N}_0. \tag{1.3}$$

Genau dann, wenn alle $e_i \in \mathbb{N}$, liegt $\mathbf{a}$ in $\mathcal{O}$.

Satz 1.5 *Seien $\mathbf{a}$ und $\mathbf{b}$ zwei von (0) verschiedene Ideale in $\mathcal{O}$, dann gilt:*

$$N(\mathbf{ab}) = N(\mathbf{a})N(\mathbf{b}). \tag{1.4}$$

1.4.1 Hermite-Normalform (HNF)

Die $\mathbb{Z}$-Basis-Darstellung in (1.2) eines Ideals ist nicht eindeutig. Da wir die Gleichheit von Idealen testen wollen, sind wir bemüht eine eindeutige Standarddarstellung für Ideale zu finden.

Eine Matrix $(a_{ji}) \in \mathbb{R}_{\geq 0}^{k \times l}, \ k, l \in \mathbb{N}$ besitzt *obere HNF*, falls

$$a_{ji} \begin{cases} = 0 & \text{falls} \quad j > i \text{ oder } i > k \\ > 0 & \text{falls} \quad j = i \\ < a_{jj} & \text{falls} \quad j < i \leq k. \end{cases} \tag{1.5}$$

Satz 1.6 *Sei $\mathbb{Z}\alpha_1 + \cdots + \mathbb{Z}\alpha_m$ ein Ideal mit*

$$\alpha_i = \frac{1}{d}\sum_{j=1}^{n} a_{ji}\,\omega_j, \quad i = 1,\ldots,m, \ a_{ji} \in \mathbb{Z}, d \in \mathbb{N},$$

bzgl. der Ganzheitsbasis $(\omega_1 = 1, \omega_2, \ldots, \omega_n)$, dann gibt es eine eindeutig bestimmte Matrix $(a'_{ji}) \in \mathbb{N}_0^{n\times m}$ in oberer HNF, so daß das von den

$$\alpha'_i = \frac{1}{d}\sum_{j=1}^{n} a'_{ji}\,\omega_j, \quad i = 1,\ldots,n$$

erzeugte Ideal dasselbe ist.

Beweis: Siehe [Coh]. ■

Eine Basis $(\alpha'_1, \ldots, \alpha'_n)$ von $\mathbf{a}$ bezeichnen wir als *HNF des Ideals.*

Bemerkung 1.7 Die HNF des Ideals ist nicht eindeutig, da sie von dem gemeinsamen Nenner d abhängt und somit die α'_i nicht in Standarddarstellung vorliegen.

Da bei der Berechnung der HNF die Gefahr besteht, daß die Größe der Einträge 'explodiert' – dazu ist ein interessantes Beispiel in [Schm] angegeben – , berechnen wir die HNF möglichst modular. Dies bedeutet, daß wir einen geeigneten Modul D wählen und die während der Berechnung auftretenden Matrixeinträge modulo D reduzieren. Diese Technik geht auf [DoKaTr] zurück. Aus der Literatur (s. z.B. [Coh], [DoKaTr], [Düll]) ist die Verwendung eines Vielfachen der Determinanten der Matrix als Modul bekannt. Wir wählen in einem Fall einen günstigeren Modul und geben den verwendeten Modul jeweils an. Algorithmen zur (modularen) Berechnung der HNF finden sich z.B. in [Düll] und [Coh].

1.4.2 Darstellung von Idealen

Die eindeutige Standarddarstellung des Ideals $\mathbf{a}$ erhalten wir, wie leicht aus Bemerkung 1.7 folgt, aus der HNF des Ideals durch Standarddarstellung der Basiselemente.

Der *größte Eintrag* von $\mathbf{a}$ ist das Maximum der Koeffizienten $(a_{i,j})_{\substack{i=1,\ldots,n \\ j=1,\ldots,n+1}}$ seiner Standarddarstellung.

Weiterhin existiert für jedes Ideal $\mathbf{a}$ in $\mathcal{O}$ eine nicht eindeutig bestimmte 2-Element-Darstellung (s. [PoZa1])

$$\mathbf{a} = \pi\mathcal{O} \, + \, a\mathcal{O}, \quad \pi \in \mathcal{F}, \ a \in \mathbb{N}.$$

Für Primideale $\mathbf{p}$ gibt es eine Darstellung

$$\mathbf{p} = \pi\mathcal{O} \; + \; p\mathcal{O}, \quad \pi \in \mathcal{F},\; p \in \mathbb{P}.$$

Umwandlung: 2-Element-Darstellung → Standarddarstellung

Diese Umwandlung geschieht durch Berechnung der HNF des von $\pi\omega_1, \ldots, \pi\omega_n, p\omega_1, \ldots, p\omega_n$ erzeugten Ideals oder durch die Berechnung der HNF des von $\pi\omega_1, \ldots, \pi\omega_n$ erzeugten Ideals modulo p. Die zweite Möglichkeit ist günstiger, da die modulare HNF–Berechnung schneller ist und zusätzlich von einer Matrix mit nur n Spalten gebildet werden muß.

Umwandlung: Standarddarstellung → 2-Element-Darstellung

Diese Umwandlung ist i.allg. schwierig. Wir benötigen sie jedoch in einer speziellen Situation. Dazu entwickeln wir folgenden

Algorithmus 1.8 Berechnung einer 2-Element-Darstellung

Eingabe: Ein Primideal $\mathbf{p}_i = \mathbb{Z}\pi_1 + \cdots + \mathbb{Z}\pi_n,\; i \in \{1, \ldots, g\},\; g \in \mathbb{N}$ in Standarddarstellung aus der Zerlegung

$$p\mathcal{O} = \prod_{j=1}^{g} \mathbf{p}_j^{e_j}.$$

Ausgabe: $\mathbf{p}_i$ in 2-Element-Darstellung

	Bilde zufällig eine Linearkombination $\pi = \sum_{k=1}^{n} \lambda_k \pi_k \;,\quad \lambda_k \in \mathbb{Z}.$	(1)
WHILE	$\pi = 0$ OR $\pi \in \mathbf{p}_j$ für ein $j \in \{1, \ldots, g\}, j \neq i$	(2)
IF	$e_i = 1$	(3)
THEN	RETURN $p_i = \pi\mathcal{O} + p\mathcal{O}$	(4)
IF	$a\mathcal{O} + p\mathcal{O} \;\neq\; \mathbf{p}_i$	(5)
THEN	GOTO (1)	(6)
ELSE	RETURN $p_i = \pi\mathcal{O} + p\mathcal{O}$	(7)

ad (1) Dabei genügt es meist, $|\lambda_k| \leq 1$ zuzulassen.

Beweis (Korrektheit):

Es ist nur zu zeigen, daß der Algorithmus korrekt ist, falls er in Schritt (4) terminiert:

Falls in Schritt (3) $e_i = 1$ gilt, haben wir an dieser Stelle bereits die gesuchte Darstellung von $\mathbf{p}_i$ gefunden, da

$$\pi\mathcal{O} + p\mathcal{O} \mid p\mathcal{O}$$

und alle anderen Teiler von $p\mathcal{O}$ ungleich $\pi\mathcal{O} + p\mathcal{O}$ sind. ∎

1.4.3 Multiplikation von Idealen in $\mathcal{O}$

Bei der Multiplikation von Idealen in $\mathcal{O}$ unterscheiden wir zwei Fälle:

Falls die Ideale $\mathbf{a}$ und $\mathbf{b}$ beide in Standarddarstellung gegeben sind, werden sie multipliziert, indem man ihre Basen elementweise multipliziert und das von diesen n^2 Produkten erzeugte Ideal auf Standarddarstellung bringt.

Wie oben erwähnt, führen wir dabei die Berechnung der HNF modular durch. Als Modul kann dabei ein Vielfaches der Determinante benutzt werden (s. [Coh], [Düll]). Die Determinante ist wegen der Multiplikativität (Satz 1.5) der Norm durch

$$N(\mathbf{ab}) = N(\mathbf{a})N(\mathbf{b})$$

gegeben. Die Berechnung von $N(\mathbf{ab})$ ist einfach, da beide Ideale in Standarddarstellung vorliegen.

Liegt eines der Ideale in 2-Element-Darstellung vor, so verkleinert sich die Produktanzahl mit einer Technik von J. Buchmann auf $2n$:

Sei

$$\mathbf{a} = \mathbb{Z}a_1 + \cdots + \mathbb{Z}a_n \text{ in Standarddarstellung und } \mathbf{b} = \pi\mathcal{O} + b\mathcal{O},\ \pi\epsilon\mathcal{F},\ b\epsilon\mathbb{N}, \tag{1.6}$$

dann ist

$$\begin{aligned} \mathbf{ba} &= (\pi\mathcal{O} + b\mathcal{O})(\mathbb{Z}a_1 + \cdots + \mathbb{Z}a_n) \\ &= \pi\mathcal{O}(\mathbb{Z}a_1 + \cdots + \mathbb{Z}a_n) + b\mathcal{O}(\mathbb{Z}a_1 + \cdots + \mathbb{Z}a_n) \\ &= \mathbb{Z}\pi a_1 + \cdots + \mathbb{Z}\pi a_n + \mathbb{Z}ba_1 + \cdots + \mathbb{Z}ba_n. \end{aligned}$$

Satz 1.9 *Seien* $\mathbf{a}$ *und* $\mathbf{b}$ *wie in (1.6). Dann kann die HNF von*

$$\mathbf{ba} = \pi\mathcal{O}(\mathbb{Z}a_1 + \cdots + \mathbb{Z}a_n) + b\mathcal{O}(\mathbb{Z}a_1 + \cdots + \mathbb{Z}a_n)$$

modulo ba_{11} *berechnet werden.*

Beweis:

$$\mathbf{a} \text{ in } \mathcal{O} \text{ hat obere HNF} \Rightarrow a_1 = a_{11} \cdot \omega_1 = a_{11}$$

$$ba_1\mathcal{O} = ba_{11}\mathcal{O} \subseteq \mathbf{ba}.$$

D.h. die zu **ba** gehörende Matrix mit den Erzeugenden als Spalten geschrieben läßt sich auffassen als:

$$\begin{pmatrix} * & \cdots & * & ba_{11} & 0 & & \cdots & 0 \\ & & & 0 & ba_{11} & & & \\ \vdots & & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ & & & & & & ba_{11} & 0 \\ * & \cdots & * & 0 & \cdots & & 0 & ba_{11} \end{pmatrix}$$

Daher können wir die HNF modulo ba_{11} berechnen. ■

1.5 Die Klassengruppe

Wir definieren

$$\mathrm{Cl} = {}^{I}\!/_{H}$$

als die *Klassengruppe* von $\mathcal{F}$ und nennen die Kardinalität der Klassengruppe die *Klassenzahl h.*

Da I nach Satz 1.3 eine abelsche Gruppe ist, ist auch Cl abelsch (s. [Hun]). Es gilt

Satz 1.10 *Für die Klassenzahl h eines algebraischen Zahlkörpers gilt:*

$$h < \infty.$$

Beweis: S. [Stew, S.171]. ■

Nach dem Hauptsatz über abelsche Gruppen [Waer2] ist damit Cl als direktes Produkt von zyklischen Gruppen darstellbar. Diese Zerlegung nennen wir die *Struktur der Klassengruppe.*

Bemerkung 1.11 Für $h = 1$ ist $\mathcal{O}$ ein Hauptidealring, und $\mathcal{O}$ ist wegen des Satzes von der Eindeutigkeit der Primfaktorzerlegung in Hauptidealringen (s. [Waer1]) ZPE-Ring.

1.6 Einheiten und Regulator

Nach dem Dirichletschen Einheitensatz [Stew, S.227f] lassen sich mit einem (nicht eindeutig bestimmten) *Fundamentalsystem von Einheiten* in $\mathcal{O}$

$$\varepsilon_1, \ldots, \varepsilon_r \quad r = s + t - 1$$

und einer Einheitswurzel μ die Einheiten von $\mathcal{O}$ eindeutig in der Form

$$\mu^{e_0} \varepsilon_1^{e_1} \cdots \varepsilon_r^{e_r} \;, \quad e_0, \ldots, e_r \in \mathbb{Z}$$

darstellen.

Sei

$$R = \left| \det \Big(\operatorname{Log} (\varepsilon_i)_{i=1,\ldots,r} \Big) \right| .$$

R ist unabhängig von der speziellen Wahl des Fundamentalsystems [PoZa1]. Wir nennen R den *Regulator* des Zahlkörpers.

1.7 Gitter

Ein $\mathbb{Z}$-Modul $L \subset \mathbb{R}^m$, das frei vom Rang m ist, nennen wir ein *Gitter* der Dimension m. Für eine Basis $A = (a_1, \ldots, a_m)$, $a_i = (a_{1i}, \ldots, a_{mi}) \in \mathbb{R}^m$ von L definieren wir

$$\det L = |\det(a_1, \ldots, a_m)|$$

Diese Determinante ist unabhängig von der Wahl der Basis. $\det L$ heißt die *Determinante des Gitters L.*

Wie für Ideale gibt es auch für Gitter eine eindeutige Darstellung in HNF:

Satz 1.12 *Sei $(a_1, \ldots, a_l) \in \mathbb{R}^{m \times l}$ ein Erzeugendensystem des m-dimensionalen Gitters L, dann gibt es eine eindeutig bestimmte Matrix $(a'_{ji}) \in \mathbb{R}^{m \times l}$ in oberer HNF, so daß das von den Spalten $a'_i = (a'_{1i}, \ldots, a'_{mi})$, $i = 1, \ldots, m$ erzeugte Gitter ebenfalls L ist.*

Beweis: s. [Schr, Theorem 4.1, 4.2] ■

Wir nennen (a'_{ji}) die HNF des von $(a_1, \ldots, a_l)$ erzeugten Gitters.

Ein *Teilgitter* L' eines m-dimensionalen Gitters L ist eine Teilmenge von L, die selbst wieder ein m-dimensionales Gitter ist.

Lemma 1.13 *Sei $L' \subset L$ ein Teilgitter, dann gilt:*

$$\det L \,|\, \det L'. \tag{1.7}$$

Falls L' ein echtes Teilgitter ist, so ist $\det L$ *ein echter Teiler von* $\det L'$.

Beweis: s. [PoZa1, Lemma (3.6)] ∎

Kapitel 2

Der Algorithmus

Wir beschreiben in diesem Kapitel die Grundversion des Algorithmus zur Berechnung der Klassenzahl.

2.1 Hintergrund

In diesem Abschnitt skizzieren wir den theoretischen Hintergrund des in [Bu1] vorgeschlagenen Algorithmus zur Bestimmung von Klassengruppe und Regulator.

Für die Klassengruppe eines algebraischen Zahlkörpers $\mathcal{F}$ bilden, wie Bach in [Bach] zeigte, unter Annahme der Verallgemeinerten Riemannschen Vermutung (GRH) die Primideale $\mathbf{p}$ mit

$$N(\mathbf{p}) \leq 12(\log \mid \Delta_{\mathcal{F}} \mid)^2 = c$$

ein Erzeugendensystem $\mathcal{F}_c$.

$$\begin{aligned} \mathcal{F}_c &= \{ \, \mathbf{p} | \, \mathbf{p} \text{ Primideal von } \mathcal{F}, \; N(\mathbf{p}) \leq c \} \\ &= \{ \, \mathbf{p}_1, \ldots, \; \mathbf{p}_{k(c)} \} \end{aligned}$$

Eine *Relation* zwischen den Idealklassen von $\mathcal{F}_c$ ist eine Zahl aus der Menge

$$\mathcal{A}_c = \{ \; \alpha \; \epsilon \; \mathcal{F}^* | \; \alpha\mathcal{O} = \prod_{i=1}^{k(c)} \mathbf{p}_i^{e_i}, \quad e_i \; \epsilon \; \mathbb{Z} \}.$$

Eine Relation stellen wir durch den Exponentenvektor $e = (e_1, \ldots, e_{k(c)})$ oder durch das Paar (e,α) dar.

Für $r = s + t$ und ein $\alpha \; \epsilon \; \mathcal{A}_c$ mit

$$\alpha\mathcal{O} = \prod_{i=1}^{k(c)} \mathbf{p}_i^{e_i}$$

sind die Abbildungen

$$\begin{array}{rccc} \varphi_c : & \mathcal{A}_c & \to & \mathbb{Z}^{k(c)} \times \mathbb{R}^r \\ & \alpha & \mapsto & (\, e_1, \ldots, \, e_{k(c)}, \, \log \, \alpha) \end{array}$$

und

$$\begin{array}{rccc} \varphi_c^{'} : & \mathcal{A}_c & \to & \mathbb{Z}^{k(c)} \\ & \alpha & \mapsto & (e_1, \ldots, \, e_{k(c)}) \end{array}$$

Homomorphismen.

Es gilt:

Satz 2.1 *Sei $c \in \mathbb{R}_{>0}$, $c > 12\ (\log\ \mid \Delta_{\mathcal{F}} \mid)^2$, und es gelte die GRH, dann ist die Menge $L_c = \varphi_c(\mathcal{A}_c)$ ein $(k(c){+}r)$-dimensionales Gitter mit Determinante hR, und die Menge $L_c^{'} = \varphi_c^{'}(\mathcal{A}_c)$ ist ein $k(c)$-dimensionales Gitter mit Determinante h, und es gilt*

$$\mathbb{Z}^{k(c)} \big/ L_c^{'} \;\cong\; Cl.$$

Korollar 2.2 *Die Kenntnis einer Basis des Gitters L_c impliziert die Kenntnis des Regulators und der Klassengruppe.*

Wir nennen $L_c^{'}$ das *Relationengitter* und L_c das *erweiterte Relationengitter* von $\mathcal{F}_c$.

Bemerkung 2.3 Satz 2.1 gilt sinngemäß für jedes Erzeugendensystem der Klassengruppe.

2.2 Die Struktur des Algorithmus

Aus dem letzten Abschnitt ergibt sich folgender Algorithmus zur Bestimmung der Klassengruppe. In Klammern ist der Mehraufwand angegeben, den die zusätzliche Bestimmung des Regulators erfordert.

- Bestimme ein Erzeugendensystem $\mathcal{FB}$ der Klassengruppe.
- Finde Relationen für dieses Erzeugendensystem, die das zugehörige (erweiterte) Relationengitter $L'_{\mathcal{FB}}$ ($L_{\mathcal{FB}}$) erzeugen.
- Bestimme mit Hilfe dieser Relationen eine Basis des (erweiterten) Relationengitters.

In dieser Arbeit beschränken wir uns auf die Berechnung der Klassenzahl, daher genügt es eine Basis des Relationengitters $L'_{\mathcal{FB}}$ zu finden und deren Determinante zu berechnen.

In den nächsten Abschnitten dieses Kapitels werden wir die theoretischen Grundlagen zur Ausführung der einzelnen Schritte des Algorithmus erläutern und im nächsten Kapitel die Realisierung des Algorithmus beschreiben.

2.3 Die Faktorbasis

2.3.1 Berechnung der Schranke

Wie bereits erwähnt, wird unter Annahme der GRH die Klassengruppe eines algebraischen Zahlkörpers von den Primidealen $\mathbf{p}_i$ mit

$$N(\mathbf{p}_i) \leq 12\,(\log \mid \Delta_{\mathcal{F}} \mid)^2$$

erzeugt [Bach]. Diese logarithmische Schranke ist jedoch erst für große Diskriminanten kleiner als die Minkowski Schranke (s. [Min])

$$\frac{n!}{n^n}\left(\frac{4}{\pi}\right)^t \sqrt{\mid \Delta_{\mathcal{F}} \mid}$$

bzw. Verbesserungen dieser Schranke, die durch Zimmert, Markoff, Martinet, Davenport, Noordzij und Godwin für bestimmte Signaturen angegeben wurden (s. [Zimm]). Diese Schranken setzen die Riemannsche Vermutung nicht voraus.

In unserem Algorithmus verwenden wir für jede Signatur das Minimum aller angegebenen Schranken und nennen dieses C. Die *Faktorbasis* $\mathcal{FB}$ ist

$$\begin{aligned} \mathcal{FB} &= \{\, \mathbf{p} \mid \mathbf{p} \text{ Primideal von } \mathcal{F},\ N(\mathbf{p}) \leq C\} \\ &= \{\, \mathbf{p}_1, \ldots,\ \mathbf{p}_{k(C)}\}. \end{aligned}$$

2.3.2 Berechnung der Primideale

Um alle Primideale $\mathbf{p}_i$ mit $N(\mathbf{p}_i) < C$ zu erhalten, müssen wir die Ideale $p\mathcal{O}$ für alle $p \leq C$ zerlegen, d.h. die Darstellung

$$p\mathcal{O} = \prod_{i=1}^{g} \mathbf{p}_i^{e_i},\ \ e_i, g \in \mathbb{N}$$

finden.

Für den Fall, daß $p \not| d_{\mathcal{F}}$, geschieht dies mit [PoZa1, Theorem 2.27] . Gilt nämlich

$$f(x) \equiv \prod_{i=1}^{g} f_i^{e_i}(x) \mod p, \quad f_i \text{ irreduzibel} \bmod p,$$

so besitzt $\mathbf{p}_i$ die (2-Element-) Darstellung

$$\mathbf{p}_i = p\mathcal{O} + f_i(\rho)\mathcal{O}$$

für eine Nullstelle ρ von f.

Die Zerlegung von $p\mathcal{O}$ für $p|d_{\mathcal{F}}$ erfordert einen größeren Aufwand und wird mit einem auf [BuLe] aufbauenden Algorithmus von D. Weber [Web] durchgeführt.

Von den so berechneten Primidealen $\mathbf{p}_i$ bilden diejenigen die Faktorbasis $\mathcal{FB}$ mit $N(\mathbf{p}_i) \leq C$.

2.4 Idealreduktion

Um Relationen zu finden, werden wir eine Technik benötigen, zu einem gegebenen Ideal ein reduziertes Ideal in der gleichen Idealklasse zu finden. Diese *Idealreduktion* führen wir mit einer von J. Buchmann vorgeschlagenen Technik durch:

Satz 2.4 *Für ein Minimum $\alpha \in \mathbf{a}$ gilt: $\frac{1}{\alpha}\mathbf{a}$ ist ein reduziertes Ideal.*

Beweis: Siehe [Bu2]. ■

Sei $\mathbf{a} = \mathbb{Z}\alpha_1 + \cdots + \mathbb{Z}\alpha_n$, $\alpha_1, \ldots, \alpha_n \in \mathcal{F}$, dann bildet $\underline{\mathbf{a}} = \mathbb{Z}\underline{\alpha_1} + \cdots + \mathbb{Z}\underline{\alpha_n}$ ein n-dimensionales Gitter in $\mathbb{R}^n$ und die Abbildung

$$\begin{aligned} \eta : \mathcal{F} &\rightarrow \mathbb{R}^n \\ \alpha &\mapsto \underline{\alpha} \end{aligned}$$

ist ein Algebrenmonomorphismus (s. [Stew, S.158ff]).

Proposition 2.5 *Für einen bzgl. der euklidischen Norm kürzesten Vektor $\underline{\alpha} \in \underline{\mathbf{a}}$ ist das Urbild α unter η ein Minimum von $\mathbf{a}$.*

Beweis: Annahme: $\underline{\alpha} \in \underline{\mathbf{a}}$ sei ein bzgl. der euklidischen Norm kürzester Vektor in $\underline{\mathbf{a}}$, aber sein Urbild α sei kein Minimum von $\mathbf{a}$. Es gibt also ein β in $\mathbf{a}$ mit $|\beta|_i < |\alpha|_i$ für alle $1 \leq i \leq n$, dann gilt:

$$\sqrt{(\beta^{(1)})^2 + \cdots + (\beta^{(s)})^2 + |\beta^{(s+1)}|^2 + \cdots + |\beta^{(s+t)}|^2} < \sqrt{(\alpha^{(1)})^2 + \cdots + (\alpha^{(s)})^2 + |\alpha^{(s+1)}|^2 + \cdots + |\alpha^{(s+t)}|^2}.$$

Dies ist ein Widerspruch zur Annahme. ■

Mit Hilfe des LLL-Algorithmus (s. [Coh, S. 38]) lassen sich — deterministisch, in polynomieller Zeit — kurze Vektoren in einem Gitter finden, welche oft, aber eben nicht immer, kürzeste Vektoren sind. Mit diesen führen wir eine abgeschwächte Idealreduktion durch:

Für das von $\underline{\alpha_1}, \ldots, \underline{\alpha_n}$ aufgespannte Gitter berechnen wir

$$\mathrm{LLL}(\underline{\alpha_1}, \ldots, \underline{\alpha_n}) = (\underline{\alpha_1}, \ldots, \underline{\alpha_n}) \cdot (t_1, \ldots, t_n)$$

mit der Transformationsmatrix $(t_1, \ldots, t_n)$, $t_i = (t_{i1}, \ldots, t_{in})^{trans} \in \mathbb{Z}^n$. Daraus erhalten wir die kurzen Gittervektoren

$$\underline{s_i} = (\underline{\alpha_1}, \ldots, \underline{\alpha_n}) t_i = \underline{\alpha_1} t_{i1} + \cdots + \underline{\alpha_n} t_{in}.$$

Da die Abbildung $\eta : \alpha \mapsto \underline{\alpha}$ ein Algebrenmonomorphismus ist, erhalten wir das Urbild s_i des kurzen Vektors $\underline{s_i}$ einfach als

$$s_i = \alpha_1 t_{i1} + \cdots + \alpha_n t_{in}.$$

s_i ist für den Fall, daß $\underline{s_i}$ ein kürzester Vektor ist, mit Proposition 2.5 ein Minimum und somit $\frac{1}{s_i}\mathbf{a}$ ein reduziertes Ideal. Falls $\underline{s_i}$ kein kürzester Vektor ist, ist es, da die Umkehrung von Proposition 2.5 nicht gilt, trotzdem möglich, daß s_i ein Minimum von $\mathbf{a}$ ist. Das Ideal $\frac{1}{s_i}\mathbf{a}$ liegt dann in der gleichen Idealklasse, wenngleich es nicht unbedingt reduziert ist. Diese abgeschwächte Idealreduktion ist für unsere Anwendung ausreichend.

2.5 Relationensuche

Relationen zwischen den Idealklassen von $\mathcal{FB}$ gewinnen wir folgendermaßen:

Für ein zufällig gewähltes Potenzprodukt von Faktorbasisidealen $\mathbf{p}_i$

$$\prod_{i=1}^{k(C)} \mathbf{p}_i^{e_i}, \; e_i \in \mathbb{N}_0 \tag{2.1}$$

führen wir die oben beschriebene Idealreduktion durch. Daraus erhalten wir ein Ideal

$$\frac{1}{\alpha} \prod_{i=1}^{k(C)} \mathbf{p}_i^{e_i},$$

das in der gleichen Idealklasse wie (2.1) liegt und möglicherweise reduziert ist. Eine Relation erhalten wir, falls wir für zwei verschiedene Potenzprodukte das gleiche reduzierte Ideal erhalten. Seien dazu $\underline{e} = (e_1, \ldots, e_{k(C)})$ und $\underline{e}' = (e'_1, \ldots, e'_{k(C)})$ zwei verschiedene Exponentenvektoren und gelte:

$$\frac{1}{\alpha} \prod_{i=1}^{k(C)} \mathbf{p}_i^{e_i} \;=\; \frac{1}{\beta} \prod_{i=1}^{k(C)} \mathbf{p}_i^{e'_i} \;,\; \alpha, \beta \in \mathcal{F}. \tag{2.2}$$

Dann ist

$$\frac{\alpha}{\beta} = \prod_{i=1}^{k(C)} \mathbf{p}_i^{e_i - e'_i} \tag{2.3}$$

eine Relation, die wir durch $\underline{e} - \underline{e}'$ beschreiben. Im Falle der zusätzlichen Berechnung des Regulators beschreiben wir sie durch das Paar $(\underline{e} - \underline{e}',\ \frac{\alpha}{\beta})$.

Wir nennen eine Menge M von Relationen — als Matrix mit den Relationen als Spalten aufgefaßt — auch *(erweiterte) Relationenmatrix.* Die Relationen in M erzeugen, falls M regulär ist, ein Teilgitter $T' \subset L'_{\mathcal{FB}}$ $(T \subset L_{\mathcal{FB}})$ und es gilt nach Lemma 1.13:

$$|\det M| = \det T' \mid \det L'_{\mathcal{FB}} \qquad (\det T \mid \det L_{\mathcal{FB}}). \tag{2.4}$$

Wir nennen die Relationenmatrix *vollständig*, wenn ihre Spalten $L'_{\mathcal{FB}}$ $(L_{\mathcal{FB}})$ erzeugen. Dann und nur dann gilt

$$|\det M| = \det L'_{\mathcal{FB}} \qquad (|\det M| = \det L_{\mathcal{FB}}). \tag{2.5}$$

Bemerkung 2.6 Gleichung (2.2) legt die Vermutung nahe, daß man Relationen auch erhalten könnte, indem man verschiedene Potenzprodukte von Faktorbasisidealen mit irgendwelchen algebraischen Zahlen in $\mathcal{F}$ multipliziert. Dieser Vermutung ist falsch, da es in jeder Idealklasse von $\mathcal{O}$ unendlich viele Ideale gibt. Erst die Idealreduktion bildet die Ideale auf eine Teilmenge der entsprechenden Idealklasse ab, in der man mit einer positiven Wahrscheinlichkeit auf eine Gleichung der Form (2.2) hoffen kann. Auf diese Wahrscheinlichkeit werden wir im Kapitel 'Grenzen dieser Methode' näher eingehen.

2.6 Berechnung der Klassenzahl

Nach Satz 2.1 können wir h (das Produkt hR) als Determinante des (erweiterten) Relationengitters berechnen. Dazu müssen wir überprüfen können, ob eine gegebene Relationenmatrix M das (erweiterte) Relationengitter L'_{FB} (L_{FB}) erzeugt. Sei M regulär und G' (G) das von M erzeugte Gitter.

Falls wir Klassenzahl und Regulator berechnen wollen, bietet die *analytische Klassenzahlformel* (s. [BuWi]) eine Möglichkeit zu überprüfen, ob G bereits das vollständige erweiterte Relationengitter ist.

Die analytische Klassenzahlformel liefert eine Zahl x mit

$$hR < x < 2hR.$$

M ist also genau dann das erweiterte Relationengitter, wenn

$$\det(G) < x < 2\det(G).$$

Falls wir allerdings nur die Klassenzahl berechnen wollen, ist kein vergleichbares Verfahren bekannt mit dem man überprüfen könnte, ob G' gleich dem Relationengitter $L'_{\mathcal{FB}}$ ist.

Daher entwickeln wir eine heuristische Methode, um zu kontrollieren, ob die Determinante von G' die Klassenzahl oder nur ein Vielfaches der Klassenzahl ist:

- Berechne die Determinante von G'.
- Erzeuge zusätzliche Relationen und berechne die Determinante des neuen G'.
- Iteriere dieses Vorgehen bis it-mal die gleiche Determinante berechnet wurde.

Durch die Anzahl der zusätzlichen Relationen und die Wahl von it ist jede geforderte Wahrscheinlichkeit kleiner 1, daß das berechnete Ergebnis korrekt ist, erreichbar.

Die Struktur der Klassengruppe kann durch die Berechnung der Smith-Normalform der vollständigen Relationenmatrix von $L'_{\mathcal{FB}}$ bestimmt werden (s. dazu [Coh], [Düll]).

Kapitel 3

Die Realisierung des Algorithmus

In diesem Kapitel erläutern wir die Realisierung des Klassenzahlalgorithmus. Zum besseren Verständnis führen wir durch die einzelnen Abschnitte den Zahlkörper $\mathbb{Q}(\rho)$ mit, wobei ρ eine Nullstelle des Polynoms $x^4 + 2x^3 + 24x^2 + 79$ ist.

3.1 Initialisierung des Zahlkörpers

Bevor der Klassenzahlalgorithmus beginnt, sind einige Vorberechnungen notwendig:

Die Eingabe des Zahlkörpers erfolgt durch sein erzeugendes Polynom. Aus dieser Eingabe werden mit Hilfe des in Bordeaux von H. Cohen, C. Batut, D. Bernardi und M. Olivier entwickelten Computeralgebrasystems PARI [PARI] die Wurzeln des Polynoms (somit s und t), die Ganzheitsbasis des Zahlkörpers sowie Körper- und Polynomdiskriminante, damit auch der Führer $d_{\mathcal{F}}$ bestimmt.

Diese Daten stehen jedem der folgenden Algorithmen zur Verfügung.

Beispiel 3.1 In unserem Beispielkörper berechnen wir:

- die Diskriminante des Zahlkörpers 151552336
- die Polynomdiskriminante 151552336
- den Führer 1
- die Potenzbasis bildet eine Ganzheitsbasis
- $s = 0$ und $t = 2$.

3.2 Die Faktorbasis

3.2.1 Berechnung der Schranke

Zur Berechnung der Normschranke werden zunächst, wie bereits im letzten Kapitel erwähnt, drei Werte berechnet:

- die Minkowski-Schranke
$$\frac{n!}{n^n}\left(\frac{4}{\pi}\right)^t \sqrt{\mid \Delta_{\mathcal{F}} \mid}$$
- die Bach-Schranke
$$12\,(\log \mid \Delta_{\mathcal{F}} \mid)^2$$
- die Zimmert-Schranke, bzw. Verbesserungen dieser Schranke
$$\frac{1}{x}\sqrt{\mid \Delta_{\mathcal{F}} \mid},$$
wobei die von der Signatur abhänige Variable x dem Anhang I zu entnehmen ist, welcher uns von J. Graf von Schmettow zur Verfügung gestellt wurde (siehe auch [Zim]).

Unsere Normschranke C setzen wir auf das Minimum dieser drei Werte.

$$C = \min\left\{\frac{n!}{n^n}\left(\frac{4}{\pi}\right)^t \sqrt{\mid \Delta_{\mathcal{F}} \mid}, 12\,(\log \mid \Delta_{\mathcal{F}} \mid)^2, \frac{1}{x}\sqrt{\mid \Delta_{\mathcal{F}} \mid}\right\} \tag{3.1}$$

Beispiel 3.2 Die logarithmische Schranke ist erst von einer von der Signatur abhängigen Diskriminantengröße an günstiger :

Körpergrad	s	Diskriminante	Minkowski	Zimmert	Bach
4	4	169142424	1219	901	4308
4	4	11396184893	10008	7388	6435
4	4	4908982876825	207714	153331	10248
6	2	115061893	268	148	4135
6	2	105933117200	8142	4482	7734
6	2	96424472012957	245662	135200	12442

Beispiel 3.3 Für unseren Beispielkörper haben wir:

$$\begin{aligned} \text{Zimmert} &= 1813 \\ \text{Minkowski} &= 1870 \\ \text{Bach} &= 4258 \\ \text{also } C &= 1813. \end{aligned}$$

3.2.2 Berechnung der Primideale

Die Primideale $\mathbf{p}_i$ mit $N(\mathbf{p}_i) \leq C$ bilden ein Erzeugendensystem der Klassengruppe. Die Berechnung dieser Primideale erfolgt wie im letzten Kapitel beschrieben durch den

Algorithmus 3.4 Berechnung der Primideale

Eingabe: C

Ausgabe: $\{ \mathbf{p} | \ \mathbf{p}$ Primideal von $\mathcal{F}, \ N(\mathbf{p}) \leq C\}$

$\mathcal{FB} = \emptyset$	(1)
FORALL $p \in \mathbb{P}, p \leq C$	(2)
IF $p \not| d_{\mathcal{F}}$	(3)
THEN Zerlege f mod p mit Hilfe von PARI: $$f(x) \equiv \prod_{i=1}^{g} f_i^{e_i}(x) \; mod \; p, \quad f_i \text{ irreduzibel mod } p, .$$ und erhalte, falls $p^{\deg(f_i)} \leq C$, die Primideale $\mathbf{p}_i = p\mathcal{O} + f_i(\rho)\mathcal{O}$, $i = 1, \ldots, g$, mit $f(\rho) = 0$.	(4)
ELSE Berechne mit dem Algorithmus von D. Weber die Zerlegung $$p\mathcal{O} = \prod_{i=1}^{g} \mathbf{p}_i^{e_i}.$$	(5)
FOR $i = 1$ TO g	(6)
IF $\mathrm{N}(\mathbf{p}_i) = p^{\deg(f_i)} \leq C$	(7)
THEN $\mathcal{F}\mathrm{B} = \mathcal{F}\mathrm{B} \cup \{\mathbf{p}_i\}$.	(8)
RETURN $\mathcal{FB}$	(9)

Wir speichern die Primideale nicht in Standarddarstellung ab, sondern durch ihre Erzeuger $f_i(\rho)$ und p, da dies folgende Vorteile bringt:

- bei der Multiplikation von Idealen ist es günstig, wenn ein Ideal in 2-Element-Darstellung vorliegt, da die HNF von einer Matrix mit nur $2n$ Spalten gebildet werden muß (vgl. Kapitel 1).

- das Abspeichern der Erzeuger erspart die HNF Berechnung.

- diese Abspeicherungsform spart Speicherplatz.

Beispiel 3.5 Da in unserem Beispielkörper $d_{\mathcal{F}} = 1$ gilt, können wir alle Primideale mit der ersten Methode berechnen.
Ein Primideal p erhalten z.B. wir durch Faktorisieren von f mod 2 :

$$\begin{array}{lcl} f & \equiv & (x\ +\ 1)^4 \ \bmod\ 2 \\ \alpha & = & (x\ +\ 1)^4 \text{ hat bzgl. der Ganzheitsbasis die Koeffizienten } (1,1,0,0,1) \\ \mathbf{p} & = & \alpha\mathcal{O} + 2\mathcal{O} \end{array}$$

Insgesamt finden wir 285 Primideale $\mathbf{p}_i$ mit $\mathrm{N}(\mathbf{p}_i) \leq 1813 = C$.

3.3 Auffinden von Relationen

Wir wollen wie in Abschnitt 2.5 beschrieben Relationen finden. Dies setzt voraus, daß wir die Exponentenvektoren der Ideale, die algebraische Zahl mit denen diese reduziert werden und die zugehörigen reduzierten Ideale abspeichern und diese Daten daraufhin untersuchen können, ob das gleiche reduzierte Ideal bereits gefunden wurde.

Dazu verwenden wir ein Lexikon (s. [LEDA]). Diese Datenstruktur besteht aus einem sogenannten Schlüssel und einer dazugehörigen Information. Für die Schlüssel wird eine lineare Ordnung definiert. Um das Lexikon zu verwalten sind die folgenden Operationen notwendig:

- Eintragen eines Schlüssels mit der zugehörigen Information in das Lexikon.

- Suchen nach einem gegebenen Schlüssel im Lexikon.

- Lesen der Information, die mit einem im Lexikon vorhandenen Schlüssel verbunden ist.

- Bestimmung der Anzahl der Einträge des Lexikons.

Wir benutzen ein Lexikon mit folgenden Einträgen: als Schlüssel verwenden wir die reduzierten Ideale und verbinden damit als Information den Exponentenvektor des Ausgangsideals und die algebraische Zahl mit der dieses reduziert wurde.

Beispiel 3.6 Schreibweise in diesem wie in den folgenden Beispielen: in den Zeilen stehen die Koeffizienten $a_1, \ldots, a_{n+1}$ einer algebraischen Zahl in Standarddarstellung bzgl. der Ganzheitsbasis. Ideale werden durch ihre vier Erzeuger in Standarddarstellung angegeben.

In unserem von $x^4 + 2x^3 + 24x^2 + 79$ erzeugten Beispielkörper ergibt die Reduktion des Faktorbasisideals

```
2   0   0   0   1
1   1   0   0   1
1   0   1   0   1
1   0   0   1   1
```

mit der algebraischen Zahl

```
2   0   0   0   1
```

das reduzierte Ideal

```
1   0   0   0   1
1   1   0   0   2
1   0   1   0   2
1   0   0   1   2.
```

Als Schlüssel für unser Lexikon dient das reduzierte Ideal, die damit verbundenen Informationen sind das Primideal und die algebraische Zahl.

Der Algorithmus zum Auffinden von Relationen hat folgende Gestalt:

Algorithmus 3.7 Auffinden von Relationen

Eingabe: Ein durch seinen Exponentenvektor $\underline{e} = (e_1, \ldots, e_{k(C)})$ bzgl. $\mathcal{FB}$ gegebenes Ideal, ein Lexikon L von reduzierten Idealen und zugehörigen Informationen, eine Menge M mit bereits gefundenen Relationen.

Ausgabe: M, L.

- Berechne die Standarddarstellung des Ideals $\mathbf{a} = \prod_{i=1}^{k(C)} \mathbf{p}_i^{e_i}$. (1)
- Reduktionsschritt: Erhalte aus der LLL-Reduktion des Konjungiertengitters von $\mathbf{a}$ die ganzzahlige Transformationsmatrix t und daraus die Minimakanidaten s_i, $i = 1, \ldots, n$. (2)
- FOR $i = 1$ TO n (3)
  - Berechne $\mathbf{a}' = \frac{1}{s_i}\mathbf{a}$. (4)
  - IF $\mathbf{a}' = \mathbf{l}$ für einen Schlüssel $\mathbf{l} \in$ L (5)
  - THEN
    - Sei $\underline{e}'$ der zu $\mathbf{l}$ gehörende Exponentenvektor und λ die zugehörige algebraische Zahl. (6)
    - IF $\underline{e} \neq \underline{e}'$ (7)
    - THEN
      - Es gilt die Gleichung: (8)

        $$\frac{1}{s_1} \prod_{i=1}^{k(C)} \mathbf{p}_i^{e_i} = \frac{1}{\lambda} \prod_{i=1}^{k(C)} \mathbf{p}_i^{e_i'}.$$

        Berechne daraus die Relation

        $$\frac{s_1}{\lambda} = \prod_{i=1}^{k(C)} \mathbf{p}_i^{e_i - e_i'}$$

        und fügen diese M hinzu.
      - RETURN L, M (9)
  - ELSE
    - Trage $\mathbf{a}'$ mit den zugehörigen Informationen $\underline{e}$ und s_i in das Lexikon L ein. (10)
- RETURN L, M (11)

ad (1): Die Evaluation der Exponentenvektoren geschieht durch sukzessive Multiplikation der Primideale. Falls dabei Einträge größer 10^{20} in einem Zwischenergebnis auftreten, wird dieses LLL-reduziert bis seine Einträge unterhalb der Schranke liegen. Diese Evaluation ist neben der Idealreduktion der aufwendigste Schritt im Algorithmus. Wie später erläutert, erlaubt Methode 3 der Relationenfindung die Vorberechnung von Potenzen, dadurch reduziert sich die Evaluation auf das Multiplizieren der Primidealpotenzen.

Beispiel 3.8 In unserem Beispielkörper werden bei der Reduktion der ersten 26 Primideale in der Liste der Faktorbasisideale keine Relationen gefunden. Nach diesen Reduktionen hat das Lexikon 104 Einträge. Wir betrachen nun die Reduktion des 27. Faktorbasisideals:

Die HNF-Basis des Ideals $\mathbf{p}_{26}$ ist

```
53   0   0   0   1
13   1   0   0   1
43   0   1   0   1
24   0   0   1   1.
```

Die LLL-Reduktion liefert die kurzen Vektoren

```
16   2   1   0   1

-3  -1  -1   0   1

 1  -4   0   0   1

-1  20   2   1   1.
```

Da das mit Hilfe von $(16, 2, 1, 0, 1)$ berechnete reduzierte Ideal

```
 1   0   0   0   1
 0   1   0   0   1
 0   0   1   0   1
22   8  28   1  93
```

nicht im Lexikon vorhanden ist, tragen wir es dort ein und berechnen mit $\alpha = (-3, -1, -1, 0, 1)$ das nächste reduzierte Ideal:

```
 1   0   0   0   1
 0   1   0   0   1
 0   0   1   0   1
37  14  40   1  45.
```

Dieses Ideal ist im Lexikon vorhanden; es wurde durch Reduktion des Ideals $\mathbf{p}_{10}$

13	0	0	0	1
0	13	0	0	1
4	3	1	0	1
1	8	0	1	1

mit der Zahl $\beta = (17, 3, 1, 0, 1)$ berechnet. Wir erhalten also die Relation:

$$\frac{\beta}{\alpha} = p_{10} \cdot p_{26}^{-1}$$

und tragen diese in die Relationenmenge ein.

3.4 Methoden der Relationensuche

Durch die Auswahl der Exponentenvektoren steuern wir die Relationensuche. Dabei sind folgende Ziele zu verfolgen:

- die Vollständigkeit der Relationenmatrix soll mit möglichst wenigen Relationen erreicht werden.
- die Exponentenvektoren der Ideale sollen wenige Einträge enthalten, damit wenige Idealmultiplikationen notwendig sind und die Relationenmatrix dünn besetzt ist.

Bezeichnungen:

- pos - die Anzahl der besetzten Positionen im Exponentenvektor.
- expo - die maximale Größe der Einträge.
- more - die Anzahl der zusätzlichen Relationen, wobei mindestens $|\mathcal{FB}|$ viele Relationen notwendig sind, um eine reguläre Matrix zu erhalten, und die darüber hinausgehenden Relationen, als zusätzliche Relationen bezeichnet werden. Diese zusätzlichen Relationen sind notwendig, um die Vollständigkeit der Relationenmatrix zu erreichen.

3.4.1 Methode 1

Algorithmus 3.9 Methode 1

Eingabe: Eine Faktorbasis $\mathcal{FB}$

Ausgabe: Eine Menge M von Relationen zwischen den Idealklassen von $\mathcal{FB}$.

L $= M = \emptyset$		(1)
Lege pos, expo und more fest.		(2)
Führe Algorithmus 3.7 für die Primideale von $\mathcal{FB}$ aus		(3)
	Wähle zufällig einen Exponentenvektor $\underline{e}$ mit Einträgen kleiner expo an pos vielen Stellen.	(4)
	Führe Algorithmus 3.7 mit $\underline{e}$ aus.	(5)
WHILE $\vert M\vert \leq \vert\mathcal{FB}\vert +$ more		(6)
RETURN M		(7)

Diese Methode ist für Körper, in denen wir die Schranke C ≤ 100 wählen können, erfolgreich. Die Reduktion der Primideale zu Beginn ist günstig, da für viele Primideale Relationen gefunden werden, ohne daß Idealmultiplikationen nötig sind.

Je größer die Schranke C — und damit die Gitterdimension — wird, um so mehr zusätzliche Relationen müssen berechnet werden, bevor die Vollständigkeit der Matrix erreicht wird. Daher entwickeln wir folgende Methode, um die Regularität der Matrix wahrscheinlicher zu machen:

3.4.2 Methode 2

Zur Kontrolle darüber welche Faktorbasisideale in einer Relation vorhanden sind, wollen wir eine Indexmenge *index* benutzen. Die Faktorbasis besteht aus den Primidealen $\mathbf{p}_i$, $i = 0, \ldots, k(C)$, somit ist die Indexmenge

$$index = \{1, \ldots, k(C)\}.$$

Sobald das Primideal $\mathbf{p}_j$ in einer Relation vertreten ist, wird der Index j aus der Indexmenge gestrichen. Hierdurch wird sichergestellt, daß zumindest jedes Primideal in einer Relation vertreten ist. Dadurch erreichen wir, daß die Regularität der von den Relationen erzeugten Matrix sehr viel wahrscheinlicher wird.

Der Algorithmus ist ähnlich wie Methode 1 aufgebaut. Wir ergänzen zwei Schritte. Zunächst versuchen wir Relationen von einer besonderen Gestalt zu finden, in denen die Primideale vertreten sind, deren Index zu *index* gehören. Am Ende des Algorithmus erzwingen wir dann Relationen zu den Primidealen die noch immer in keiner Relation vertreten sind.

Algorithmus 3.10 Methode 2

Eingabe: Eine Faktorbasis $\mathcal{FB}$

Ausgabe: Eine Menge M von Relationen zwischen den Idealklassen von $\mathcal{FB}$

L $= M = \emptyset$. Lege pos, expo und more fest.				(1)
$index = \{1,\ldots,$ k(C) $\}$. Falls eine Relation gefunden wird, wird jeder in ihr vertretene Index aus $index$ gestrichen.				(2)
Führe Algorithmus 3.7 für die Primideale von $\mathcal{FB}$ aus.				(3)
$index' = index$				(4)
WHILE $index' \neq \emptyset$				(5)
	Wähle ein $x \in index'$.			(6)
	FOR $j = 1$ TO expo			(7)
		FOR $i = 1$ TO Obergrenze		(8)
			Wähle $\underline{e}$, in dem die letzte besetzte Position x mit j besetzt ist, mit höchstens pos Einträge kleiner expo.	(9)
			Führe Algorithmus 3.7 mit $\underline{e}$ aus; streiche dabei die Indizes aller in einer gefundenen Relation vertretenen Primideale von $\mathcal{FB}$ aus $index$ und $index'$.	(10)
			Beende die beiden FOR-Schleife, falls $x \notin index$.	(11)
	Streiche x aus $index'$.			(12)
	Wähle $\underline{e}$ mit pos Einträgen kleiner expo.			(13)
	Führe Algorithmus 3.7 mit $\underline{e}$ aus.			(14)
WHILE $\lvert M\rvert \leq \lvert\mathcal{FB}\rvert +$ more				(15)
WHILE $index \neq \emptyset$				(16)
	Wähle $x \in index$ und $\underline{e}$, in dem x besetzt ist, mit pos Einträgen kleiner expo.			(17)
	Führe Algorithmus 3.7 mit $\underline{e}$ aus.			(18)
RETURN M				(19)

Erläuterungen:

ad (9) Die Anzahl der Versuche, die durch die 'Obergrenze' bestimmt ist, legen wir gewöhnlich auf 4 fest.

Die praktische Erprobung dieser Methode zeigt, daß es auf diese Weise für Zahlkörper mit $C < 1000$ gelingt mit wenigen zusätzlichen Relationen die Vollständigkeit der Relationenmatrix zu erlangen. Dabei setzen wir

$$\text{more} = \begin{cases} 20 & \text{falls} & & & C & < & 100 \\ 40 & \text{falls} & 100 & \leq & C & < & 800 \\ 60 & \text{falls} & 800 & \leq & C & & . \end{cases}$$

Auch die Größe von pos und expo passen wir der Größe von C an. In der Regel lassen wir Werte zwischen 3 und 7 zu.

Diese Methode hat zwei entscheidende Nachteile:

- die Relationenmatrix ist nicht dünn besetzt, und
- die Regularität bzw. Vollständigkeit der Relationenmatrix ist nicht immer gegeben.

3.4.3 Methode 3

Um die Mängel von Methode 2 zu beheben, entwickeln wir eine weitere Methode zur Relationenfindung.

Dazu beobachten wir zunächst, daß die von uns benutzte Schranke C in der Praxis zu schlecht ist, d.h. die Klassengruppe wird schon von den Primidealen mit einer wesentlich kleineren Norm erzeugt. Wir gehen davon aus, daß in dem für uns interessanten Bereich $n \geq 3$ und $\mid \Delta_{\mathcal{F}} \mid > 10^5$ die Klassengruppe von den Idealen $\mathbf{p}_i$ mit $N(\mathbf{p}_i) < \log \mid \Delta_{\mathcal{F}} \mid$ erzeugt wird. Unter dieser Voraussetzung ist es für die übrigen Primideale der Faktorbasis nicht notwendig, in Relationen andere Einträge als 1 oder -1 zu finden.

Dies nützen wir aus, indem wir zunächst eine Menge *basic_part* einführen:

$$\text{basic\_part} = \{i \mid \mathbf{p}_i \in \mathcal{FB} \text{ mit } N(\mathbf{p}_i) < \log \mid \Delta_{\mathcal{F}} \mid\}.$$

Für Primideale, deren Indizes zu basic_part gehören, suchen wir auf ähnliche Art Relationen wie in Methode 1. Für Primideale $\mathbf{p}_i$, deren Index nicht zu basic_part gehört, versuchen wir ausschließlich mit folgendermaßen ausgewählten Exponentenvektoren Relationen zu finden: die i-te Position ist mit 1 besetzt und ansonsten sind nur für Primideale, deren Index zu basic_part gehört, Einträge vorhanden.

Wie schon in Methode 2 benutzen wir ein Indexmenge *index*. Diese Indexmenge besteht aus den Indizes der Faktorbasisideale, die nicht zu basic_part gehören. Aus der Indexmenge wird jedoch im Gegensatz zu Methode 2 nur dann ein Index i gestrichen, wenn eine Relation gefunden wird, in der i der größte nicht zu basic_part gehörende Index ist.

Die Regularität der Relationenmatrix ist erreicht, sobald die Indexmenge leer ist und für die Ideale, deren Index zu basic_part gehört, genügend viele Relationen gefunden sind.

Allerdings gibt es bei dieser Verwaltung der Indexmenge ein Problem:

Sei i der kleinste Index in *index*. Für diesen wird bei der oben angegebenen Wahl des Exponentenvektors nur dann in Algorithmus 3.7 eine Relation der gewünschten Form gefunden, wenn der zweite an der Relation beteiligte Exponentenvektor nur Einträge für Ideale mit Index aus basic_part enthält. Dies stellt wie die Praxis zeigt ein Problem dar, da unter Umständen viele nicht gewünschte Relationen gefunden werden. Entsprechendes gilt für die übrigen kleines Indizes.

Daher nehmen wir Primideale mit Index in *index* in basic_part auf, falls zwar Relationen für diese gefunden werden, diese aber nicht die gewünschte Form haben.

Algorithmus 3.11 Methode 3

Eingabe: Eine Faktorbasis $\mathcal{FB}$

Ausgabe: Eine Menge M von Relationen zwischen den Idealklassen von $\mathcal{FB}$

<table>
<tr><td colspan="4">L $= M = \emptyset$</td><td>(1)</td></tr>
<tr><td colspan="4">Lege pos, expo und more fest.</td><td>(2)</td></tr>
<tr><td colspan="4">Initialisiere basic_part und $index$. Falls eine Relation gefunden wird, wird jeweils nur der größte nicht zu basic_part gehörene Index aus $index$ gestrichen.</td><td>(3)</td></tr>
<tr><td colspan="4">Führe Algorithmus 3.7 für die Primideale von $\mathcal{FB}$ aus.</td><td>(4)</td></tr>
<tr><td colspan="4">WHILE $|\mathrm{M}| < |\mathrm{basic\_part}| + 5$</td><td>(5)</td></tr>
<tr><td rowspan="2"></td><td colspan="3">Wähle $\underline{e}$ mit pos vielen Einträgen kleiner expo für Primideale, deren Index in basic_part liegt.</td><td>(6)</td></tr>
<tr><td colspan="3">Führe Algorithmus 3.7 mit $\underline{e}$ aus.</td><td>(7)</td></tr>
<tr><td colspan="4">WHILE $|\mathrm{M}| < |\mathcal{FB}| + \mathrm{more}$ OR $index \neq \emptyset$</td><td>(8)</td></tr>
<tr><td rowspan="9"></td><td colspan="3">IF $index \neq \emptyset$</td><td>(9)</td></tr>
<tr><td rowspan="6">THEN</td><td colspan="2">Sei i der kleinste Index in $index$ und $oldsize = |\mathrm{M}|$</td><td>(10)</td></tr>
<tr><td colspan="2">WHILE $|\mathrm{M}| < oldsize + z$ AND $i \epsilon index$</td><td>(11)</td></tr>
<tr><td rowspan="2"></td><td>Wählen $\underline{e}$ so, daß die Position i mit einer 1 besetzt ist und pos-1 weitere Einträge kleiner expo für Primideale, deren Index in basic_part liegt, existieren.</td><td>(12)</td></tr>
<tr><td>Führe Algorithmus 3.7 mit $\underline{e}$ aus.</td><td>(13)</td></tr>
<tr><td colspan="2">IF $i \epsilon index$</td><td>(14)</td></tr>
<tr><td colspan="2">THEN Füge i basic_part hinzu und streiche es aus $index$.</td><td></td></tr>
<tr><td rowspan="2">ELSE</td><td colspan="2">Wähle $\underline{e}$ mit pos vielen Einträgen kleiner expo für Primideale, deren Index in basic_part liegt.</td><td>(15)</td></tr>
<tr><td colspan="2">Führe Algorithmus 3.7 mit $\underline{e}$ aus.</td><td>(16)</td></tr>
<tr><td colspan="4">RETURN M</td><td>(17)</td></tr>
</table>

Erläuterungen:

ad (2) Wir setzen

$$\text{pos} = \begin{cases} |\text{basic\_part}| & \text{falls} \quad |\text{basic\_part}| < 3 \\ \max\{3, \lfloor \frac{|\text{basic\_part}|}{4} \rfloor\} & \text{sonst.} \end{cases}$$

Diese Wahl von pos ist groß genug, um die Vollständigkeit der Relationenmatrix erreichen zu können, und trotzdem viel günstiger als bei Methode 2.

ad (11) Wir wählen z abhängig von der Diskriminantengröße zwischen 4 und 10.

Die Praxis zeigt, daß diese Methode die anfangs des Kapitels geforderten Ziele erreicht:

- Mit der Wahl von more = 20 wird in allen bisher getesten Beispielen nach diesen 20 zusätzlichen Relationen die endgültige Determinante der Matrix berechnet, d.h. in der heuristischen Kontrolle wurde dieses Ergebnis bestätigt, was bei den Methoden 1 und 2 nicht immer der Fall war.

- Die Exponentenvektoren enthalten im Vergleich mit den anderen Methoden weniger Einträge. Dadurch sind weniger Idealmultiplikationen notwendig. Wegen der speziellen Form der Exponentenvektoren sind in Relationen höchstens zwei der Indizes nicht in basic_part, es entsteht eine dünn besetzte Matrix.

Zusätzlich erlaubt diese Methode die Vorberechnung von Potenzen der Ideale aus basic_part.

Methode 3 ist im Vergleich mit Methode 2 zuverlässiger und wegen der Vorberechnung der Primidealpotenzen um den Faktor zwei bis drei schneller.

Beispiel 3.12 Für unseren Beispielkörper benutzen wir Methode 3, wobei wir $z = 4$ wählen.

- 9 Primideale $\mathbf{p}_i$ mit $N(\mathbf{p}_i) < \log \mid \Delta_{\mathcal{F}} \mid = 18$ werden in basic_part aufgenommen.

- Nach der Reduktion der Primideale in Schritt (4) sind bereits 140 Relationen gefunden.

- Nachdem 305 Relationen gefunden sind, ist die Abbruchbedingung in Schritt (8) erfüllt. Das Lexikon enthält 728 Einträge.

3.4.4 Vorberechnung von Primidealpotenzen

Diese Vorberechnung ist nur bei Verwendung der Methode 3 sinnvoll und kann auf zwei Arten geschehen:

- Die Primideale sind in 2-Element-Darstellung gegeben; dies erlaubt die Exponentiation durch Potenzieren der Erzeuger (s. [Schm]). Sei
$$\mathbf{a} = \mathbb{Z}\alpha \; + \; \mathbb{Z}p \text{ in 2-Element-Darstellung},$$
dann ist
$$\mathbf{a}^m = \mathbb{Z}\alpha^m \; + \; \mathbb{Z}p^m.$$
- Eine zweite Möglichkeit ist, die Primidealpotenzen in $\mathbb{Z}$-Basis-Darstellung zu berechnen. Die Potenzen werden, falls ihre Einträge größer als 10^6 sind, LLL-reduziert und der Nenner wieder auf 1 gebracht. Diese Division und Multiplikation werden in der *Reduktionszahl* abgespeichert und bei der Relationenbildung berücksichtigt.

Vergleich dieser Arten:

- Das Berechnen der Primidealpotenzen im ersten Fall ist günstig, da dort keine HNF berechnet werden muß, und das Abspeichern dieser Potenzen wenig Platz verbraucht.
- Die Multiplikation der Primidealpotenzen kann im ersten Fall durch die günstigere Multiplikation eines Ideals in $\mathbb{Z}$-Basis-Darstellung mit einem Ideal in 2-Element-Darstellung ausgeführt werden. Dabei wird die HNF einer Matrix mit $2n$ Spalten gebildet. Im zweiten Fall muß die HNF einer Matrix mit n^2 Spalten berechnet werden.
- Die Primidealpotenzen haben im ersten Fall die Form
$$\mathbf{a}^m = \mathbb{Z}\alpha^m \; + \; \mathbb{Z}p^m.$$
Bei der Multiplikation eines Ideals das in oberer HNF bzgl. einer $\mathbb{Z}$-Basis gegeben ist und dessen oberster Diagonaleintrag x ist, kann modulo einer Zahl der Größenordnung $p^m x$ berechnet werden. Im zweiten Fall kann modulo einer Zahl der Größenordnung $10^6 x$ gerechnet werden, was i.allg. günstiger ist.
- Bei der Evaluation der Exponentenvektoren wird nach jeder Multiplikation darauf geachtet, daß die Einträge nicht größer als 10^{20} sind, gegebenenfalls wird das Ergebnis LLL-reduziert. Auch hier ist die Berechnung wegen der Größe der Einträge im ersten Fall langsamer.

Insgesamt ist der zweite Fall günstiger, falls die Diskriminante groß und der Körpergrad klein ist, da dann die aufwendigere Vorberechnung wegen der schnelleren Multiplikation und LLL-Reduktion rentabel wird.

3.5 Berechnung der Klassenzahl

Aus Kapitel 2.5 ergibt sich eine Zweiteilung dieser Aufgabe in die Berechnung der Determinante der Relationenmatrix M und eine heuristische Kontrolle, ob M die vollständige Relationenmatrix ist.

Zunächst zur Berechnung der Determinante:

M ist eine Matrix der Dimension $k(C) \times (k(C) + more)$. Wir wollen die HNF des von den Spalten von M erzeugten Gitters bestimmen. Da $k(C)$ recht groß sein kann, sind bei der Berechnung der HNF schnell die Grenzen der Berechenbarkeit erreicht. Wurde M jedoch mit Methode 3 berechnet, so ist M dünn besetzt. Die HNF können wir daher mit einem Verfahren von S. Düllmann [Düll] berechnen, daß wir hier nur kurz skizzieren:

- Verkleinere die Relationenmatrix unter Ausnutzung ihrer dünnen Besetzung.
- Berechne mit einer modifizierten Form der Gauß-Elimination ein Vielfaches der Determinante der Relationenmatrix.
- Berechne die HNF der verkleinerten Relationenmatrix modulo dieses Vielfachen der Determinante.

Die in Kapitel 2.5 beschriebene heuristische Kontrolle, ob M die vollständige Relationenmatrix ist, wird mit folgendem Algorithmus durchgeführt:

Algorithmus 3.13

Eingabe: M, L, pos, expo, basic_part

Ausgabe: Mit hoher Wahrscheinlichkeit h, ansonsten ein Vielfaches

$olddet = 0$, $counter = 0$				(1)
	Berechne $d = \det M$			(2)
	IF $d = 1$			(3)
	THEN	RETURN d		(4)
	IF $d = olddet$ AND $d \neq 0$			(5)
	THEN	$counter = counter + 1$		(6)
	ELSE	$olddet = d$, $counter = 0$		(7)
	IF $counter < it$			(8)
	THEN	$oldsize = \|M\|$		(9)
			Wähle $\underline{e}$ mit pos vielen Einträgen kleiner expo für Primideale, deren Index in basic_part liegt.	(10)
			Führe Algorithmus 3.7 mit $\underline{e}$ aus.	(11)
		WHILE $oldsize + x \geq \|M\|$		(12)
WHILE $counter < it$				(13)
RETURN D				(14)

Erläuterung:

ad (3) Falls die Determinante 1 ist, ist sicher, daß $h = 1$.

Die Praxis zeigt, daß bei Verwendung von Methode 3 die Wahl $it = 3$ und $x = 10$ genügt, damit die Ausgabe mit hoher Wahrscheinlichkeit die Klassenzahl ist.

Beispiel 3.14 In unserem Beispiel liefert die Berechnung die Determinante 1. Somit ist die Klassenzahl 1.

Kapitel 4

Implementierung in C++

In diesem Kapitel gehen wir auf einige Details der Implementierung des Algorithmus ein. Der Algorithmus wurde auf einer SPARC-station ELC in der Programmiersprache C++ implementiert.

4.1 C++

C++ ist eine auf C aufbauende, objektorientierte Programmiersprache. Eine Einführung in C++ geben z.B.[Pohl], [Eck1], [Eck2]. Wichtig für Algorithmen in der algebraischen Zahlentheorie sind selbstdefinierte Datentypen, die sogenannten Klassen, um die die Programmiersprache erweitert werden kann. Prinzipiell sind diese Erweiterungen auch in C realisierbar — in der Tat wird objektorientiertes Programmieren oft durch C simuliert. Vorteil von C++ ist, daß C++ das objektorientierte Programmieren unterstützt, was sich vor allem auf die Programmierarbeit sowie auf die Lesbarkeit des Programmcodes auswirkt.

Die Verwendung von C++ für Algorithmen der algebraischen Zahlentheorie unterteilt sich in zwei Schritte:

Im ersten Schritt werden die benötigten Datentypen implementiert. Dazu sind eine Speicherverwaltung, die elementare Arithmetik — dazu zählt neben den üblichen Operatoren $+$, $-$, $*$, $/$ auch Übergabefunktionen und Vergleichsfunktionen — und Eingabe und Ausgaberoutinen zu erstellen. Nachdem diese Funktionen implementiert sind, können diese Datentypen genauso verwendet werden wie z.B. der Datentyp long.

Beispiel 4.1 Für die Klasse 'algebraische Zahl' wurden u.a. folgende Funktionen implementiert:

- Speicherverwaltung:
  - Allokieren bzw. Deallokieren von Speicherplatz für algebraische Zahlen.
  - Ein 'copy-constructor' [Eck1].
- Operatoren:
  - $+, -, *, /$ für algebraische Zahlen (s. Kapitel 1).
  - $*, /$ für algebraische Zahlen mit einer Zahl aus $\mathbb{Z}$.
  - $=$ verschieden Übergabeoperatoren
  - $==, >, <$ eine lineare Ordnung für algebraische Zahlen
- Ein- und Ausgabe von algebraischen Zahlen.

Bemerkung 4.2 Als lange $\mathbb{Z}$-Arithmetik verwenden wir, wie bereits in der Einleitung erwähnt, die von R. Denzer entwickelte libI.

Für C++-kundige Leser ein kleines

Beispiel 4.3 Den Multiplikationsoperator $*$ für Ideale in Standarddarstellung (vgl. Kapitel 1) haben wir folgendermaßen überladen (weitere hier benutzte Klassen sind 'ideal' und die libI-Integers 'Int'):

```
ideal ideal::operator *( const ideal& b)
{
      extern long n;
      algebraic_number* num = new algebraic_number[n*n]; // Initial.
                                                         // e. arrays
                                                        // alg. Zahlen
      ideal c;
      int count = 0;

      for ( int i = 1; i <= n; i++)
      for ( int j = 1; j <= n; j++)
      num[count++] = alpha[i] * b.alpha[j]; // * von alg. Zahlen!

      Int modul = *this.norm() * b.norm(); // Berechnung des Moduls

      c = modular_HNF( num, n, n*n, modul); // die Standarddar-
                                            // stellung des Ideals
                                           // wird mod modul bestimmt
      delete[n*n] num;

      return ( c);
}
```

Der zweite Schritt besteht dann aus der eigentlichen Impementierung des Algorithmus. Hierbei ist es nicht mehr notwendig sich z.B. über die Realisierung von algebraischen Zahlen Gedanken zu machen, man kann objektorientiert mit den gewohnten arithmetischen Symbolen arbeiten.

Auf die Implementierung des in den letzten beiden Kapiteln beschriebenen Algorithmus wollen wir nicht näher eingehen. Wir wollen nur einige Details herausstellen:

4.2 Verwendung von LEDA

LEDA ist eine von Stefan Näher am Max-Planck-Institut für Informatik in Saarbrücken entwickelte C++ Programmbibliothek, welche unter anderem besondere Datenstrukturen, wie Mengen oder Lexika zur Verfügung stellt und eine effiziente Handhabung dieser Daten erlaubt (s [LEDA]). Mit Hilfe von LEDA konstruieren wir folgende höhere Datenstrukturen (vgl. letztes Kapitel):

- eine Liste von Primidealen, welche zur Faktorbasis gehören.
- ein Lexikon, das als Einträge reduzierte Ideale enthält. Mit jedem dieser Einträge ist als Information der Exponentenvektor des ursprünglichen Ideals und die algebraische Zahl, mit der dieses reduziert wurde, verknüpft. Aus Speicherplatzgründen werden die reduzierten Ideale z.B. in "unsigned short arrays" (falls nur größte Einträge kleniner 2^{16} zugelassen werden) und die algebraischen Zahlen in Strings umgespeichert. Um die Ideale vergleichen zu können, wird die numerische Ordnung der "unsigned short arrays" als lineare Ordnung verwandt.
- eine Menge von Relationen.
- Indexmengen, die eine Kontrolle darüber erlauben, welche Primideale in Relationen vorkommen.

Kapitel 5

Grenzen dieser Methode

5.1 Dimension der Relationenmatrix

Die Dimension der Relationenmatrix wird durch die Anzahl $\#\mathcal{FB} = k(C)$ der Primideale mit Norm nicht oberhalb C bestimmt. Durch Methode 3 der Relationensuche wird eine dünn besetzte Relationenmatrix erzeugt. Die Determinante dieser Matrix wird berechnet, indem zunächst die Dimension verringert wird. Günstiger wäre es also gleich eine Relationenmatrix einer kleineren Dimension zu finden. Dazu dient der folgende

Satz 5.1 *Am Ende von Methode 3 gilt, falls die verwendete Faktorbasis unter Annahme der GRH ein Erzeugendensystem der Klassengruppe ist:*
Die Ideale, deren Indizes zu basic_part gehören, erzeugen unter der Annahme der GRH die Klassengruppe.

Beweis: Wegen der Abbruchbedingung in Schritt (8) gilt: Für jedes Primideal $\mathbf{p}_i$ der Faktorbasis, dessen Index nicht in basic_part liegt, wird zumindest eine Relation gefunden, die an i-ter Stelle einen Eintrag y und für keinen nicht zu basic_part gehörenden höheren Index einen Eintrag enthält. Dieser Eintrag y ist aufgrund der Wahl der Exponentenvektoren 1 oder -1. Durch Linearkombination dieser Relationen läßt sich daher für jeden nicht in basic_part vertretenen Index i eine Relation

$$\alpha\mathcal{O} = \mathbf{p}_i \cdot \prod_{j \,\epsilon\, basic\_part} \mathbf{p}_j^{m_j}, \;\; \alpha \,\epsilon\, \mathcal{F}, m_j \,\epsilon\, \mathbb{Z}$$

bilden. Hieraus folgt die Behauptung, da es wegen der Endlichkeit der Klassengruppe für alle $p \,\epsilon\, I$ und $m \,\epsilon\, \mathbb{Z}$ ein $x \,\epsilon\, \mathbb{Z} \setminus \mathbb{N}$ und ein $\beta \,\epsilon\, \mathcal{F}$ gibt mit

$$p^m = p^x \cdot \beta\mathcal{O}.$$

■

Korollar 5.2 *Die Berechnung der Klassenzahl kann durchgeführt werden, indem die Determinante eines Relationengitters $L_{\mathcal{FB}'}$ mit basic_part $\subseteq \mathcal{FB}'$ berechnet wird.*

Die Dimension der Relationenmatrix kann daher reduziert werden, indem Methode 3 zur Berechnung der Relationenmatrix $L_{\mathcal{FB}}$ zweigeteilt wird: zunächst werden mit Methode 3 nur Relationen einer Teilmenge $\mathcal{FB}' \subset \mathcal{FB}$, von der im Voraus vermutet wird, daß

$$basic\_part \subseteq \mathcal{FB}',$$

gesucht und die Determinante der Relationenmatrix $L_{\mathcal{FB}'}$ berechnet. Anschließend werden für alle $\mathbf{p}_i \in \mathcal{FB} \setminus \mathcal{FB}'$ Relationen der Form

$$\alpha\mathcal{O} = \mathbf{p}_i \cdot \prod_{j \in \mathcal{FB}} \mathbf{p}_j^{m_j}, \;\alpha \in \mathcal{F}, m_j \in \mathbb{Z} \tag{5.1}$$

gesucht. Aus dem Beweis von Satz 5.1 ergibt sich, daß hierdurch gezeigt wird, daß $\mathcal{FB}'$ ein Erzeugendensystem der Klassengruppe ist.

Algorithmus 5.3 Berechnung der Klassenzahl

Eingabe: $\mathcal{FB}_C$

Ausgabe: Mit hoher Wahrscheinlichkeit h, falls der Beweis in Schritt (5) erfolgreich durchgeführt wird, ansonsten die Zahl 0.

<table>
<tr><td colspan="2">Wähle einen Faktor $l \in \mathbb{C}$ mit $$\frac{1}{\log(\mid \Delta_{\mathcal{F}} \mid)} < l < 12 \log(\mid \Delta_{\mathcal{F}} \mid).$$</td><td>(1)</td></tr>
<tr><td colspan="2">Berechne eine neue Schranke $$C' = \min\{\, C,\ \lceil l \cdot \log(\mid \Delta_{\mathcal{F}} \mid) \rceil \}.$$</td><td>(2)</td></tr>
<tr><td colspan="2">Bilde das Relationengitter $L_{\mathcal{FB}'}$ mit Algorithmus 3.11 für $$\mathcal{FB}' = \{\mathbf{p} \in I \mid N(\mathbf{p}) < C'\}.$$</td><td>(3)</td></tr>
<tr><td colspan="2">Berechne die Determinante det von $L_{\mathcal{FB}'}$ mit Algorithmus 3.13.</td><td>(4)</td></tr>
<tr><td colspan="2">(GRH) Beweise, daß $\mathcal{FB}'$ ein Erzeugendensystem der Klassengruppe ist, indem zu $\mathbf{p}_i \in \mathcal{FB} \setminus \mathcal{FB}'$ Relationen der Form 5.1 gefunden werden.</td><td>(5)</td></tr>
<tr><td colspan="2">IF Beweis ist erfolgreich</td><td>(6)</td></tr>
<tr><td>THEN</td><td>RETURN det</td><td>(7)</td></tr>
<tr><td>ELSE</td><td>RETURN 0</td><td>(8)</td></tr>
</table>

ad (1) Unsere derzeitige Praxis ist es, $l = 3$ zu wählen (s. Kapitel Ergebnisse). Eine sehr kleine Wahl von l führt dazu, daß die Menge basic_part sehr klein ist und möglicherweise eine Ergänzung um einige Primideale notwendig ist, bevor basic_part ein Erzeugendensystem der Klassengruppe wird; zusätzlich schränkt dies die Auswahl bei der Besetzung der Exponentenvektoren ein. Daher sind wir bemüht, l so zu wählen, daß basic_part tatsächlich ein Erzeugendensystem ist.

ad (5) Dazu verwenden wir, da wir hier keine Rücksicht auf die Vollständigkeit der Relationenmatrix nehmen müssen, Exponentenvektoren mit möglichst einfacher Gestalt: wir beginnen mit den Primidealen, für die verbliebenen Ideale verwenden wir zunächst Produkte mit einem Ideal mit Index aus basic_part und anschließend Produkte mit einer Idealpotenz aus der gleichen Menge. Sollte dann für ein Primideal immer noch keine Relation gefunden sein, verwenden wir Produkte dieses Ideals mit Potenzprodukten von Idealen mit Index aus basc_part.

Durch die Verringerung der Dimension der Relationenmatrix wird die Berechnung der Determinante unproblematisch für Zahlkörper mit Diskriminante kleiner 10^{100}.

5.2 Anzahl der reduzierten Ideale

Während es für den imaginär-quadratischen Fall genau ein reduziertes Ideal in jeder Idealklasse gibt, wächst die Anzahl p der reduzierten Ideale pro Idealklasse bei höherem Körpergrad mit der Diskriminante.

Satz 5.4 *Die Anzahl der reduzierten Ideale in $\mathcal{O}_{\mathcal{F}}$ ist*

$$O(\sqrt{\mid \Delta_{\mathcal{F}} \mid} \log(\mid \Delta_{\mathcal{F}} \mid)^{n-1}),$$

wobei die O-Konstante nur von s und t abhängt.

Beweis: Siehe [Bu2]. ∎

Da wir eine abgeschwächte Idealreduktion durchführen, erhalten wir durch die Idealreduktion nicht nur reduzierte Ideale. Zusammen mit der letzten Aussage läßt dies erwarten, daß Probleme bzgl. des Speicherplatzes und der Laufzeit auftreten, wobei wir die Speicherplatzprobleme zuungunsten der Laufzeit begrenzen können:

5.2.1 Speicherplatzbedarf

Der Speicherplatzbedarf hängt im Wesentlichen von der Anzahl der Einträge des in Kapitel 3.3 beschriebenen Lexikons ab. Pro Lexikoneintrag benötigen wir folgenden Speicherplatz:

- Speicherplatz für das reduzierte Ideal:
  - Anzahl der Koeffizienten ungleich 0 in der HNF eines Ideals:

    $$\frac{(n+1)n}{2} + n.$$

  - pro Eintrag 2 Byte, falls die Einträge durch 65500 beschränkt sind und daher in einem "unsigned short" abgespeichert werden können.

- Speicherplatz für eine Information:
  - algebraische Zahl: bei einer durchschnittlichen Länge d von weniger als 12 Dezimalstellen ist es günstig diese als String abzuspeichern; der Nenner ist stets 1.
    Platzbedarf: $(d+1)n+2$ Byte
  - Exponentenvektor: mit durchschnittlicher Länge l, $2l$ shorts.
    Platzbedarf: $2 \cdot 2l$ Byte
- Lexikonverwaltung: ca. 30 Byte

Beispiel 5.5

- $n = 4$:
  - reduzierte Ideale: 28 Byte.
  - Information für $d = 8$ und $l = 5$: $(9 \cdot 4 + 2) + (5 \cdot 2 \cdot 2) = 58$ Byte
  - insgesamt also ca. 116 Byte.
- $n = 10$:
  - reduziertes Ideal: 65 Byte.
  - Information für $d = 8$ und $l = 5$: $(8+1)10 + 2 + (5 \cdot 2 \cdot 2) = 112$ Byte
  - insgesamt also ca. 207 Byte

In diesen Beispielen ist für 100 000 Lexikoneinträge mit einem Speicherplatzbedarf von 12 bzw. 20 MByte zu rechnen.

5.2.2 Laufzeit

Wie bereits oben erwähnt, läßt sich das Speicherplatzproblem zu Lasten der Laufzeit lösen. Dazu geben wir das folgende Beispiel an:

Beispiel 5.6 In diesem Beispiel werden zunächst Körper vierten Grades und dann Körper fünften Grades betrachtet. Die Klassenzahl wurde jeweils zweimal auf einer SPARC-station ELC berechnet. Im ersten Durchgang wurden in das Lexikon nur Ideale mit größtem Eintrag kleiner 2^{16} zugelassen, in das Lexikon wurden **id1** viele Einträge aufgenommen und die Berechnung dauerte **t1** sec. Im zweiten Durchgang wurden insgesamt **red** viele Idealreduktionen durchgeführt, in das Lexikon wurden Ideale mit Einträgen kleiner 2^{31} zugelassen, dabei wurden **id2** viele Ideale eingetragen, und die Rechenzeit betrug **t2** sec.

Diskriminante	red	id1	id2	t1	t2
113657758854949	6504	2803	6460	2481	2954
267064736262929	5605	2121	5555	2724	2775
274227856464997	11030	4589	10985	5998	6194
739181704556800	16191	3417	16043	11406	5315
813831852280788	35717	6800	35651	25951	17898
976392374688308	10214	3887	10126	4188	3043
1179708370997609	2178	2589	2133	1871	1392
1319317583466916	3576	4223	3526	1970	1685
13284552075380092	19370	16590	19307	15714	13640
14706396469180776	17573	15203	17511	14576	12734
-1067649470937739	5237	4344	5175	2819	2266
-1201596373625303	5910	5080	5851	3075	2658
-101287021762383748	19137	6329	19059	25772	19849
-102617516869725227	13406	3768	13326	13437	10638
-103316859286979435	28199	9492	28115	37241	27748

Die **Laufzeit des Algorithmus** hängt im wesentlichen von der Anzahl der Idealreduktionen ab. Dazu die folgenden Aussagen:

- Die Anzahl der reduzierten Ideale ist von der Größenordnung $O(\sqrt{|\Delta_{\mathcal{F}}|})$.
- Nicht alle Ideale, die durch die abgeschwächte Idealreduktion berechnet werden, sind reduzierte Ideale.
- Aus Speicherplatzgründen ist es nicht immer möglich, alle durch Idealreduktion erhaltenen Ideale in das Lexikon aufzunehmen.

Diese Aussagen lassen eine exponentielle Laufzeit erwarten.

Kapitel 6

Beispiel

Einen vollständigen Berechnung der Klassenzahl erläutern wir anhand des Körpers $\mathbb{Q}(\rho)$, wobei ρ eine Nullstelle des Polynomes $x^4 - 11$ ist.

Zunächst wird der Zahlkörper initialisiert. Dabei werden die folgenden Körperdaten berechnet: $\Delta_{\mathcal{F}} = -340736$, $\Delta_f = -340736$ und die Potenzbasis als Ganzheitsbasis.

Die Berechnung der Normschranke ergibt (s. Kapitel 3.2.1):

$C = 62 = \min\{Bachsschr. = 1948, Minkowskischr. = 69, Zimmertschr. = 62\}$.

Anschließend berechnen wir 15 Primideale mit Norm nicht oberhalb 62 und tragen diese in die Liste der Primideale ein $(p_0, \ldots, p_{14})$. Diese Primideale bilden ein Erzeugendensystem der Klassengruppe.

Wir verwenden den in Kapitel 4.1 beschriebenen Algorithmus 5.3 mit Wahl des Parameters $l = 3$. Dies ergibt in Schritt (2):

$$C' = 36 = \min\{\ C,\ \lceil 3 \cdot \log(\mid \Delta_{\mathcal{F}} \mid) \rceil \}.$$

Von den oben berechneten Primidealen haben 12, nämlich $(\mathbf{p}_0, \ldots, \mathbf{p}_{11})$, eine Norm kleiner C'. Diese Primideale bilden im weiteren Verlauf des Algorithmus die Faktorbasis $\mathcal{FB}$. Wir müssen später noch beweisen, daß diese die Klassengruppe erzeugen.

Mit Methode 3 berechnen wir das Relationengitter von $\mathcal{FB}$. In Schritt (3) wird die Menge basic_part aus den Primidealen mit Norm kleiner 12 gebildet,d.h. basic_part $= \{\mathbf{p}_0, \ldots, \mathbf{p}_9\}$. Entsprechend besteht die Indexmenge aus den Indizes 10 und 11.

Zur Darstellung des weiteren Verlaufs des Algorihmus verwenden wir folgende Notationen:

Exponentenvektoren werden in zwei aufeinanderfolgenden Zeilen dargestellt: in der zweiten Zeile steht zunächst die Anzahl der besetzten Positionen, dann folgen die

Indizes der beteiligten Primideale. In der ersten Zeile steht vorne die erlaubte maximale Anzahl von besetzten Relationen und danach die Exponenten der beteiligten Primideale in der Reihenfolge der zweiten Zeile.

Beispiel 6.1 Das Primidealprodukt $\mathbf{p}_0^2 \cdot \mathbf{p}_4 \cdot \mathbf{p}_{12}^3$ wird dargestellt durch:
100 2 1 3
3 0 4 12

Ein Reduktionsschritt hat folgendes Aussehen: Der ausgewählte Exponentenvektor steht zu Beginn eines Reduktionsschrittes, gefolgt von der Standarddarstellung des zugehörigen Ideals **id** (original ideal). In den nächsten vier Zeilen stehen die durch Idealreduktion bestimmten algebraischen Zahlen α_i, $i = 1, \ldots, 4$. Anschließend eventuell mehrere reduzierte Ideale $\frac{1}{\alpha_i}\mathbf{id}$. Falls eine Relation gefunden wird, ist zunächst der Exponentenvektor des zweiten an der Relation beteiligten Ideals angegeben und anschließend die Relation. Wir geben hier die Orginalausgabe des Programmes an, Kommentare zu einzelen Schritten fügen wir in Deutsch hinzu.

Der Algorithmus wird in Schritt (4) von Methode 3 fortgesetzt:

```
100 1
1 0
 the original ideal is
      2    0    0    0    1
      1    1    0    0    1
      1    0    1    0    1
      1    0    0    1    1

    2     0    0    0    1

    1     1    0    0    1

   -1     0    1    0    1

    0    -1    0    1    1

reduced ideal is:
      1    0    0    0    1
      1    1    0    0    2
      1    0    1    0    2
      1    0    0    1    2

items in d :1  /* das Lexikon enthaelt einen Eintrag */
 reduced ideal is:
      1    0    0    0    1
```

```
      0    1    0    0    1
      0    0    1    0    1
      4    1    4    1    5

items in d :2
reduced ideal is:
      1    0    0    0    1
      0    1    0    0    1
      1    0    1    0    5
      1    1    1    1    10

items in d :3
reduced ideal is:
      1     0     0     0     1
      0     1     0     0     1
      1     0     1     0     5
      11    11    11    1     110

items in d :4
items in relations :0
100 1
1 1
 the original ideal is
      3    0    0    0    1
      0    3    0    0    1
      2    2    1    0    1
      2    1    0    1    1

    3     0    0    0    1

   -1    -1    1    0    1

   -1     2    1    0    1

   -1    -2    0    1    1

reduced ideal is:
      1    0    0    0    1
      0    1    0    0    1
      2    2    1    0    3
      2    1    0    1    3

items in d :5
reduced ideal is:
      1    0    0    0    1
      0    1    0    0    1
```

```
      0    0    1    0    1
      2    4    3    1    5

items in d :6
reduced ideal is:
      1     0    0    0    1
      0     1    0    0    1
      1     0    1    0    2
      13    9    3    1    14

items in d :7
reduced ideal is:
      1    0    0    0    1
      0    1    0    0    1
      2    2    1    0    5
      3    5    1    1    10

items in d :8
items in relations :0
100 1
1 2
 the original ideal is
      3    0    0    0    1
      0    3    0    0    1
      2    1    1    0    1
      1    1    0    1    1

    3     0     0    0    1

   -1     1     1    0    1

    1     2    -1    0    1

    1    -2     0    1    1

reduced ideal is:
      1    0    0    0    1
      0    1    0    0    1
      2    1    1    0    3
      1    1    0    1    3

items in d :9
reduced ideal is:
      1    0    0    0    1
      0    1    0    0    1
      0    0    1    0    1
```

```
      3    4    2    1    5

items in d :10
reduced ideal is:
      1    0    0    0    1
      0    1    0    0    1
      1    0    1    0    2
      8    9    4    1    14

items in d :11
reduced ideal is:
      1    0    0    0    1
      0    1    0    0    1
      2    3    1    0    5
      1    1    1    1    10

items in d :12
items in relations :0
100 1
1 3
 the original ideal is
      5    0    0    0    1
      1    1    0    0    1
      4    0    1    0    1
      1    0    0    1    1

    1     1    0    0    1

   -1     0    1    0    1

    4    -1    0    0    1

    0    -1    0    1    1

reduced ideal is:
      1    0    0    0    1
      0    1    0    0    1
      0    0    1    0    1
      1    1    1    1    2

items in d :13
reduced ideal is:
      1    0    0    0    1
      0    1    0    0    1
      1    0    1    0    2
      1    1    1    1    10
```

```
items in d :14
reduced ideal is:
      1     0     0    0    1
      0     1     0    0    1
      0     0     1    0    1
      15    16    4    1    49

items in d :15
reduced ideal is:
      1     0     0     0    1
      0     1     0     0    1
      1     0     1     0    2
      11    11    11    1    110

items in d :16
items in relations :0
100 1
1 4
 the original ideal is
      5    0    0    0    1
      4    1    0    0    1
      4    0    1    0    1
      4    0    0    1    1

   -1     1    0    0    1

   -1     0    1    0    1

    4     1    0    0    1

    0    -1    0    1    1

reduced ideal is:
      1    0    0    0    1
      0    1    0    0    1
      0    0    1    0    1
      1    1    1    1    2

100 1 /* dieses reduzierte Ideal wurde bereits gefunden */
1 3
100 1 -1 /* damit ist die erste Relation gefunden */
2 3 4
items in relations :1
100 1
1 5
```

```
 the original ideal is
      5    0    0    0    1
      2    1    0    0    1
      1    0    1    0    1
      3    0    0    1    1

    2     1    0    0    1

    3    -1    0    0    1

   -1    -1    1    0    1

    1    -1    0    1    1

reduced ideal is:
      1    0    0    0    1
      0    1    0    0    1
      0    0    1    0    1
      0    0    0    1    1

items in d :17
reduced ideal is:
      1     0    0    0    1
      0     1    0    0    1
      0     0    1    0    1
      13    9    3    1    14

items in d :18
reduced ideal is:
      1    0    0    0    1
      0    1    0    0    1
      2    1    1    0    3
      1    1    0    1    3

100 1
1 2
100 1 -1
2 2 5
items in relations :2
100 1
1 6
 the original ideal is
      5    0    0    0    1
      3    1    0    0    1
      1    0    1    0    1
      2    0    0    1    1
```

```
   -2      1     0     0     1

    3      1     0     0     1

   -1      1     1     0     1

   -1     -1     0     1     1

reduced ideal is:
      1     0     0     0     1
      0     1     0     0     1
      0     0     1     0     1
      0     0     0     1     1

100 1
1 5
100 1 -1
2 5 6
items in relations :3
100 1
1 7
 the original ideal is
      7     0     0     0     1
      3     1     0     0     1
      5     0     1     0     1
      6     0     0     1     1

   -2      0     1     0     1

    3      1     0     0     1

    1     -2     0     0     1

    0     -2     0     1     1

reduced ideal is:
      1     0     0     0     1
      0     1     0     0     1
      0     0     1     0     1
      6     2     3     1     7

items in d :19
reduced ideal is:
      1     0     0     0     1
      0     1     0     0     1
```

```
      0    0    1    0    1
      3    9    7    1    10

items in d :20
reduced ideal is:
      1     0     0     0    1
      0     1     0     0    1
      0     0     1     0    1
      22    19    13    1    25

items in d :21
reduced ideal is:
      1     0     0     0    1
      0     1     0     0    1
      0     0     1     0    1
      55    44    66    1    77

items in d :22
items in relations :3
100 1
1 8
 the original ideal is
      7    0    0    0    1
      4    1    0    0    1
      5    0    1    0    1
      1    0    0    1    1

   -2     0    1    0    1

   -3     1    0    0    1

    1     2    0    0    1

    0    -2    0    1    1

reduced ideal is:
      1    0    0    0    1
      0    1    0    0    1
      0    0    1    0    1
      1    2    4    1    7

items in d :23
reduced ideal is:
      1    0    0    0    1
      0    1    0    0    1
      0    0    1    0    1
```

```
      7    9    3    1    10

items in d :24
reduced ideal is:
      1    0     0     0    1
      0    1     0     0    1
      0    0     1     0    1
      3    19    12    1    25

items in d :25
reduced ideal is:
      1     0     0     0    1
      0     1     0     0    1
      0     0     1     0    1
      22    44    11    1    77

items in d :26
items in relations :3
100 1
1 9
 the original ideal is
      11    0    0    0    1
      0     1    0    0    1
      0     0    1    0    1
      0     0    0    1    1

    0     1     0    0    1

    0     0     1    0    1

    0    -1     0    1    1

   11     0    -1    0    1

reduced ideal is:
      1    0    0    0    1
      0    1    0    0    1
      0    0    1    0    1
      0    0    0    1    1

100 1
1 5
100 1 -1
2 5 9
items in relations :4
100 1
```

```
1 10
 the original ideal is
      19    0    0    0    1
      8     1    0    0    1
      12    0    1    0    1
      18    0    0    1    1

    1     1     1    0    1

    3    -2     0    0    1

    4     2    -1    0    1

   -1     0     0    1    1

reduced ideal is:
      1    0    0    0    1
      0    1    0    0    1
      0    0    1    0    1
      1    2    4    1    7

100 1
1 8
100 1 -1
2 8 10
items in relations :5
100 1
1 11
 the original ideal is
      19    0    0    0    1
      11    1    0    0    1
      12    0    1    0    1
      1     0    0    1    1

    1    -1    1    0    1

    3     2    0    0    1

   -4     2    1    0    1

    1     0    0    1    1

reduced ideal is:
      1    0    0    0    1
      0    1    0    0    1
      0    0    1    0    1
```

```
      6     2     3     1     7

100 1
1 7
100 1 -1
2 7 11
items in relations :6
```

Schritt (4) von Methode 3 ist abgeschlossen. Die Indexmenge ist bereits leer. Es folgt Schritt (5).

```
100 4 15 15
3 1 2 3
 the original ideal is
      437893890380859375    0           0     0     1
      81948744658834812     14348907    0     0     1
      401608031212861695    13260591    81    0     1
      206509329735334938    6635682     0     81    1

     3564972    -10768950      2276019    3696516    1

     6880788     -8824869      5550444    -926964    1

   -43977492     -5509053      7494525    2347461    1

    18692127    -22439592    -10061091     101493    1

reduced ideal is:
      1     0     0     0     1
      0     1     0     0     1
      0     0     1     0     1
      2     4     3     1     5

100 1
1 1
100 -3 -15 -15
3 1 2 3
items in relations :7
100 1 12 10
3 1 2 5
 the original ideal is
      5189853515625    0          0     0     1
      4461559860492    531441     0     0     1
      3364714814739    491133     3     0     1
      2974466504901    245766     0     3     1
```

```
   -77862      8217     -6879     8661    1

   -17409     69645     -1338    -1782    1

    97464      9192    -62766    -7323    1

   141006    -36036     -3651    52323    1

reduced ideal is:
      1    0    0    0    1
      0    1    0    0    1
      0    0    1    0    1
      0    0    0    1    1

100 1
1 5
100 -1 -12 -9
3 1 2 5
items in relations :8
100 11 9 5
3 0 2 7
 the original ideal is
      2646497448    0         0    0    1
      2236933584    157464    0    0    1
      2566322752    49976     8    0    1
      748999692     37748     4    4    1

   17052    -1804     -164    -716    1

   -9936    -6200     5672    3008    1

    -760     9048     3704    2128    1

   16036    -8132    -6036    6388    1

reduced ideal is:
      1    0    0    0    1
      0    1    0    0    1
      0    0    1    0    1
      3    4    2    1    5

100 1
1 2
100 -11 -8 -5
3 0 2 7
```

```
items in relations :9
100 11 7 1
3 4 5 7
 the original ideal is
      341796875    0        0    0    1
      182343750    78125    0    0    1
      56453558     58796    1    0    1
      299025707    30817    0    1    1

   -2958     -423      125      -94    1

    1924    -2535     -548      219    1

     906      360      563    -1064    1

    2409     1924    -2535     -548    1

reduced ideal is:
      1    0    0    0    1
      0    1    0    0    1
      0    0    1    0    1
      1    1    1    1    2

100 1
1 3
100 1 -11 -7 -1
4 3 4 5 7
items in relations :10
100 12 6 1
3 0 2 3
 the original ideal is
      29160    0       0    0    1
      5832     5832    0    0    1
      9472     3320    8    0    1
      5720     2192    0    8    1

   -256    112    -128      24    1

    640     72    -128     -40    1

     -8    144      16     104    1

   -440    640      72    -128    1

reduced ideal is:
      1    0    0    0    1
```

```
      0    1    0    0    1
      0    0    1    0    1
      0    0    0    1    1

100 1
1 5
100 -12 -6 -1 1
4 0 2 3 5
items in relations :11
100 10 13 6
3 3 7 9
    2430406     316657    -215347     172172    1

     989054      42713     -20229    -421443    1

   -1216413    1348424    -192863    -269500    1

    -679899    -765325    -724867     118019    1

Hier wurde eine Zwischenreduktion durchgefuehrt.

 the original ideal is
      2    0    0    0    1
      0    2    0    0    1
      0    0    2    0    1
      1    1    1    1    1

     2     0     0    0    1

     0     2     0    0    1

    -2     0     2    0    1

     1    -1    -1    1    1

reduced ideal is:
      1    0    0    0    1
      0    1    0    0    1
      0    0    1    0    1
      1    1    1    1    2

100 1
1 3
100 -9 -13 -6
3 3 7 9
items in relations :12
```

```
100 1 12 10
3 2 6 8
 the original ideal is
      206891051513671875    0    0    0    1
      116430304955283114    3    0    0    1
      64802053366064219     1    1    0    1
      185916494878374160    1    0    1    1

     7500     17654    -12317    4825    1

    64508     -7271    -14720    2958    1

   -28971    -14429     15011    9095    1

   -60575    -25154     -5337    7492    1

reduced ideal is:
      1    0    0    0    1
      0    1    0    0    1
      0    0    1    0    1
      3    9    7    1    10

100 1
1 7
100 -1 -12 1 -10
4 2 6 7 8
items in relations :13
100 8 12 9
3 5 6 8
 the original ideal is
      9851954833984375    0         0    0    1
      7859183999218750    390625    0    0    1
      6288042046334431    0         1    0    1
      8133566115625000    240681    0    1    1

    423088    227207     -75927      2997    1

   -154416    330385      57064    -52415    1

   -340860    -19912      84315     83573    1

   -187383    -92703    -170143     23512    1

reduced ideal is:
      1     0    0     0    1
      0     1    0     0    1
```

```
      0     0    1     0    1
      21    6    16    1    25

items in d :27
reduced ideal is:
      1    0    0    0    1
      0    1    0    0    1
      1    0    1    0    2
      8    9    2    1    10

items in d :28
reduced ideal is:
      1    0    0     0    1
      0    1    0     0    1
      0    0    1     0    1
      1    7    11    1    38

items in d :29
reduced ideal is:
      1     0     0     0    1
      0     1     0     0    1
      0     0     1     0    1
      34    16    24    1    35

items in d :30
items in relations :13
100 14 7 12
3 3 4 6
 the original ideal is
      6103515625    0            0        0    1
      3906250000    244140625    0        0    1
      4672500000    45234375     78125    0    1
      2299972497    46009319     42288    1    1

    -239656      71838     364101      45927    1

    -751373    -474796      66933     135041    1

   -1245795     679535     110695    -112860    1

    -500942     368041    -181568     205314    1

reduced ideal is:
      1     0    0    0    1
      0     1    0    0    1
      0     0    1    0    1
```

```
      13    9    3    1    14

100 1
1 5
100 -14 -7 1 -12
4 3 4 5 6
items in relations :14
100 1 7 1
3 0 5 8
 the original ideal is
      1093750    0    0    0    1
      35837      1    0    0    1
      865681     0    1    0    1
      808753     0    0    1    1

    68     -1     -6     -3    1

    30    -22    -17      7    1

   -26    -36     -9     -5    1

     9     31    -16    -14    1

reduced ideal is:
      1    0    0     0    1
      0    1    0     0    1
      0    0    1     0    1
      1    9    11    1    14

items in d :31
reduced ideal is:
      1    0    0    0    1
      0    1    0    0    1
      2    2    1    0    3
      2    1    0    1    3

100 1
1 1
100 -1 1 -7 -1
4 0 1 5 8
items in relations :15
```

Schritt (5) von Methode 3 ist abgeschlossen.

```
 100 7 14 1  3 2 7 11
```

```
 the original ideal is
      28182203346094497    0       0    0    1
      25266085327017495    2187    0    0    1
      15507135588114068    1873    1    0    1
      11243870629021588    1003    0    1    1

    61817     63485    -31782    -3973    1

    63771     63628     15923     5975    1

    45657    -61674    -15780    41730    1

   335396    -81313      1890    -7834    1

reduced ideal is:
      1    0    0    0    1
      0    1    0    0    1
      0    0    1    0    1
      3    4    2    1    5

100 1
1 2
100 -6 -14 -1
3 2 7 11
items in relations :16
100 1 3 2
3 2 4 8
 the original ideal is
      18375    0    0    0    1
      6627     3    0    0    1
      16403    1    1    0    1
      7663     1    0    1    1

   -16      0    -7    -2    1

    -4    -21    -1     1    1

    25      1    -6    -5    1

    43     -4    -7     3    1

reduced ideal is:
      1    0    0    0    1
      0    1    0    0    1
      0    0    1    0    1
      3    4    2    1    5
```

```
100 1
1 2
100 -3 -2
2 4 8
items in relations :17
reduction_steps 48
```

Methode 3 ist abgeschlossen. Die Berechnung der Determinanten des Relationengitters ergibt:

det = 1.

Es muß noch bewiesen werden, daß die Faktorbasis ein Erzeugendensystm der Klassengruppe ist.

```
 proofer: size of big_primeideals = 3 /* Der Beweis muss noch fuer 3
Primideale erbracht werden */
 the original ideal is
      7    0    0    0    1
      0    7    0    0    1
      2    0    1    0    1
      0    2    0    1    1

reduced ideal is:
      1    0    0    0    1
      0    1    0    0    1
      2    0    1    0    7
      0    2    0    1    7

   2    0     1     0    1

   5    0    -1     0    1

   0    2     0     1    1

   0    5     0    -1    1

reduced ideal is:
      1    0    0    0    1
      0    1    0    0    1
      0    0    1    0    1
      0    0    0    1    1
ok./* Fuer dieses Primideal ist eine Relation der gewueschten Form gefunden */
```

```
 the original ideal is
      43     0     0     0     1
      8      1     0     0     1
      22     0     1     0     1
      39     0     0     1     1

reduced ideal is:
      1      0     0     0     1
      8      1     0     0     43
      22     0     1     0     43
      39     0     0     1     43

   -5      2      1     0     1

    1     -2     -1     1     1

   -2     -3      1     0     1

    1      0      2     0     1

reduced ideal is:
      1     0     0     0     1
      0     1     0     0     1
      1     0     1     0     2
      7     9     3     1     10

reduced ideal is:
      1     0     0     0     1
      0     1     0     0     1
      0     0     1     0     1
      0     0     0     1     1
ok.

 the original ideal is
      43     0     0     0     1
      35     1     0     0     1
      22     0     1     0     1
      4      0     0     1     1

reduced ideal is:
      1      0     0     0     1
      35     1     0     0     43
      22     0     1     0     43
      4      0     0     1     43

   -5     -2      1     0     1
```

```
   -1    -2     1    1    1

   -2     3     1    0    1

   -1     0    -2    0    1

reduced ideal is:
        1     0     0     0     1
        0     1     0     0     1
        1     0     1     0     2
        8     9     2     1    10
ok.

 after first round = 0
```

Da der Beweis erfolgreich beendet wurde, ist die Klassenzahl $h = 1$.

Für diese Berechnung wurden 744 1/100 sec. benötigt; davon

- 69 1/100 sec. für die die Initialisierung des Zahlkörpers und das Berechnen der Primideale.
- 211 1/100 sec. für LLL-Reduktionen.
- 145 1/100 sec. für die Evaluation der Exponentenvektoren.
- 75 1/100 sec. zum Invertieren algebraischer Zahlen.
- 54 1/100 sec. zur Berechnung von HNF'en.
- 52 1/100 sec. für den Beweis.
- 23 1/100 sec. für Schnittstellen zu PARI.
- 22 1/100 sec. zur Herstellung der Standarddarstellung alg. Zahlen.

Kapitel 7

Ergebnisse

Der Algorithmus läßt die Wahl von Parametern zu, welche auf die Laufzeit und die Speicherplatzausnutzung Einfluß haben:

- Der Faktor l kann wie in Kapitel 4.1 beschrieben variiert werden.
- Die Vorberechnung der Primidealpotenzen kann mit Primidealen in Standarddarstellung oder in 2-Element-Darstellung ausgeführt werden.
- Die Präzision, mit der die Idealreduktion ausgeführt wird, kann verändert werden.
- Der maximal zugelassene größte Eintrag der Ideale, die in das Lexikon aufgenommen werden, ist ebenfalls variabel.

7.1 Generierung der erzeugenden Polynome

Die in den Beispielen verwendeten körpererzeugenden Polynome werden folgendermaßen generiert:

Für die Koeffizienten des Polynomes wählen wir zufällig Werte zwischen Schranken, die so abgestimmt sind, daß, falls das entstehende Polynom irreduzibel ist, die Diskriminante des von ihm erzeugten Zahlkörpers meist von einer bestimmten Größenordnung ist.

Beispiel 7.1 Wir suchen Polynome siebten Grades, die Zahlkörper erzeugen, für deren Diskriminanten $\Delta_{\mathcal{F}}$ gilt:

$$10^{15} < |\Delta_{\mathcal{F}}| < 10^{18}.$$

Dazu lassen wir für die Koeffizienten von x^6 bis x^4 Werte zwischen -7 und 7 und für die übrigen zwischen -51 und 51 zu.

7.2 Beispiele für Körper vierten bis siebten Grades

Diese Berechnungen wurden mit dem in Kapitel 5.1 beschriebenen Algorithmus durchgeführt; dabei wurde der Parameter $l = 3$ gewählt. Die Vorberechnung der Primideale wurde mit Primidealen in 2-Element-Darstellung ausgeführt.

Bemerkung 7.2 In allen Beispielen konnte gezeigt werden, daß die Primideale mit Norm kleiner C' die Klassengruppe bereits erzeugen. Für diesen Beweis wurde meistens mehr als 75% der Rechenzeit benötigt. In einem Experiment für die Signatur (4, 2, 1) konnte darüberhinaus gezeigt werden, daß schon eine Wahl von $l = 2$ genügt hätte. Die Laufzeiten in diesem Experiment waren durchschnittlich um 10% kürzer als die hier angegebenen.

Notation: In jeder Zeile steht zunächst die Diskriminante des Zahlkörpers. Anschließend die Koeffizienten des erzeugenden Polynoms $f = x^n + a_{n-1}x^{n-1} + \cdots + a_0$. Es folgen die Normschranke C, die Anzahl der Primideale p_i, die Größe des Lexikons dic, die Determinante der Relationenmatrix det und die auf einer SPARC station ELC benötigte Zeit.

Tabelle7.1: n = 4, t = 1

Diskriminante	Polynom	C	p_i	dic	det	sec
-9256916816	8 -5 112 -93	6320	25	580	4	1046
-9327429567	-8 -11 -111 -111	6324	30	1219	1	1426
-9422716191	2 -31 89 145	6330	15	565	1	1116
-9981683839	34 19 -95 -87	6362	27	1016	4	1303
-9989889863	49 -18 -81 52	6362	16	561	1	997
-10143562731	-15 -28 -24 57	6371	22	663	1	1225
-10448546327	17 -1 -100 117	6387	23	766	1	1327
-10471914887	-4 -43 41 -102	6388	18	565	1	962
-10669378531	-2 48 15 -91	6399	22	460	1	1075
-10852340480	-4 -26 -136 -25	6408	24	859	2	1138
-39967761700	15 -24 139 137	7151	25	1081	1	1769
-39985389007	-33 -17 22 39	7152	31	1555	1	2012
-40033660243	-26 40 -79 91	7152	24	887	1	1607
-43210088495	-38 49 -59 -7	7197	20	1130	2	1687
-44109043231	-2 45 115 -147	7209	22	1538	1	1851
-45249479400	-40 -47 -60 26	7224	18	1949	2	3050
-45449504311	-14 3 -147 -16	7227	17	555	8	1027
-45651934400	14 -32 148 -126	7230	28	1671	4	2129
-47039365888	40 12 92 130	7247	24	1047	4	1837
-48859596788	-17 -45 -48 -122	7270	25	1976	1	1437

Diskriminante	Polynom	C	p_i	dic	det	sec
-49646852255	-22 23 -117 52	7279	24	1334	1	2645
-52293153952	28 -39 -92 -84	7310	22	1988	1	3206
-52301221547	41 10 -53 -63	7310	22	1846	2	3344
-53348610207	37 -41 66 -76	7322	22	1015	1	2243
-59195662700	31 15 41 54	7383	27	1775	2	2636
-59681475471	19 -46 41 84	7388	26	1845	1	3016
-61027279575	30 47 75 -24	7401	19	1624	4	2523
-61920523575	-37 -39 80 -35	7410	29	1500	2	2138
-62004454827	-20 14 39 -147	7411	24	746	1	2157
-62760945987	-41 3 -90 -24	7418	27	1862	1	3657
-63348238971	22 24 -73 104	7424	29	2116	1	1994
-63847549264	24 -17 40 67	7429	19	663	4	1354
-66807366691	-22 -48 -123 -25	7456	23	534	1	1922
-68444506796	-27 -41 5 -52	7470	24	1795	1	3286
-68848428628	33 18 102 -52	7474	23	2489	2	4386
-69584158303	21 -8 -95 -128	7480	13	423	1	2062
-70857903040	34 -19 76 106	7491	18	1760	2	3606
-72024307376	18 8 98 -122	7501	20	890	2	2016
-72144911795	-39 27 43 -53	7502	23	765	1	2213
-73293753227	-36 -30 -79 -23	7511	27	637	1	3038
-76018874663	-18 -1 -137 34	7533	30	1203	1	2581
-76122790023	-24 7 -9 96	7534	23	1294	1	2827
-76167431168	-28 8 96 -110	7534	23	1023	2	2242
-77426244332	-42 -20 -86 -52	7544	25	1129	4	2917
-77444074075	-29 -18 10 63	7544	28	999	1	3296
-86657595619	-34 -24 57 -43	7612	17	417	1	2073
-90063282824	-37 -3 31 -84	7635	26	2597	3	4087
-93199338671	-20 39 -137 -70	7656	26	891	3	1807
-93781278063	-29 15 -68 73	7660	23	743	2	1989
-95312778048	40 16 132 -105	7670	23	1254	8	2620
-95691666759	-49 -40 -33 -30	7672	24	1333	1	2798
-96768276691	30 -32 -29 -51	7679	17	684	2	2647

Tabelle7.2: n = 4, t = 2

Diskriminante	Polynom	C	p_i	dic	det	sec
135030068	6 30 -16 35	1711	22	426	1	422
149794677	-1 18 46 47	1802	13	169	1	258
259601140	0 42 62 40	2373	17	529	2	485
280172821	-6 6 21 85	2465	18	222	4	426
326704436	-2 42 102 68	2662	16	385	1	612
344574989	-5 1 79 115	2734	12	99	4	318
364400209	1 20 -15 148	2811	22	477	3	482
368816192	10 41 -90 63	2828	22	522	2	590
707712876	9 45 29 120	3917	14	839	1	1392
1035660389	0 30 41 146	4739	19	581	2	967
1154885845	1 26 78 137	5004	22	301	5	786
1371479793	-4 47 -41 39	5312	21	991	2	1166
1511570432	12 28 -104 94	5361	15	310	4	1085
1819365500	-2 49 22 96	5456	19	1541	32	2508
2229739988	-9 36 30 62	5560	25	1380	1	2099
2914802937	3 20 -81 136	5700	24	34	24	1049

Tabelle7.3: n = 5, t = 1

Diskriminante	Polynom	C	p_i	dic	det	sec
-148751289435	0 -2 2 -108 57	7942	24	986	1	3176
-157823682380	1 1 -49 -78 16	7979	26	1569	1	4021
-171420713804	1 0 62 -126 36	8030	23	1460	1	5001
-197833582243	1 -2 -4 -58 21	8119	19	418	1	1559
-223903585592	1 -1 81 90 16	8197	24	1609	1	3933
-276497572635	-37 -36 4 0 -5	8330	27	631	2	1716
-3067508740676	-2 1 -13 -80 122	9921	28	765	1	3114
-3181097057968	35 36 -2 1 -5	9946	30	1589	1	5250
-3327425625575	-1 -1 -27 -60 113	9977	30	787	2	2802
-3599387496187	2 0 106 118 -87	10031	20	763	1	4272
-3989125000324	1 0 -127 -32 16	10103	18	599	3	3505
-4096412499371	0 2 -76 94 -5	10121	15	618	1	4800
-4168052987028	-2 0 57 -13 -133	10133	21	1336	2	6289
-73828752228131	0 1 -88 -115 49	12237	22	1214	1	6803
-76052149669079	-2 1 -134 -22 15	12260	29	1422	1	8159
-78744633746375	-1 0 -49 -137 149	12286	30	954	1	5765
-78864637836608	2 1 94 -35 -76	12288	24	1315	1	8138
-81790793109883	0 0 60 -138 -131	12316	27	1841	1	10263
-97566024128479	0 -2 147 90 1	12452	22	1039	1	6719

Tabelle7.4: n = 5 , t = 2

130369552865	1 1 -1 104 -69	7816	27	641	1	2158
137431301825	1 1 16 -86 99	7893	29	1490	1	3233
150217952268	1 0 33 68 -40	7984	28	1670	1	3427
159450973165	-1 -1 29 75 -65	7985	19	618	1	2368
178636915804	2 -2 -17 46 34	8056	21	980	1	3562
188637771008	-1 2 -18 82 -94	8089	23	1187	1	3194
202879363664	0 -2 56 94 42	8135	19	746	1	3004
1043520151829	2 -2 52 -46 59	9190	25	1221	6	3677
1048493209177	1 -2 -65 29 -32	9194	19	1162	1	4964
1156714368877	0 2 -29 49 141	9259	32	2111	2	4493
1253030583493	2 -1 -37 94 -148	9312	21	743	1	3735
1332377436393	43 40 -1 -1 5	9353	16	732	1	4406
1376291621712	0 2 26 78 138	9375	26	1484	3	4215
1416689847541	0 2 40 -89 121	9395	23	1362	2	3442
1588788260429	-2 1 41 62 132	9472	28	1319	1	4388
1729102822908	0 -1 71 -94 75	9529	29	4261	1	11432
1730476157001	0 -2 -7 92 11	9529	30	990	2	3375
1773808470476	1 1 -147 16 -60	9546	26	2881	2	7179
2021633568064	-1 -2 -6 139 -143	9635	31	3225	1	8269
9312793452036	-2 -1 65 -74 129	10702	30	2722	1	8097
11420584460317	2 1 -8 123 125	10848	28	2261	3	8656
11508401067989	0 0 72 -147 119	10854	24	1696	1	7280
11759002537445	-2 0 -20 141 11	10869	27	1634	1	6859
11985216947557	-1 0 -72 136 125	10883	30	2902	1	10217
12410178838364	-2 2 59 34 98	10908	25	2906	5	8420
12502449606109	2 -2 -55 101 80	10914	24	1531	1	6980
13001343518485	2 2 -58 35 -105	10942	30	2745	2	7677

Tabelle7.5: n = 6 , t = 1

Diskriminante	Polynom	C	p_i	dic	det	sec
-103380790936	2 2 1 -17 -2 6	2782	13	136	1	733
-107461833512	2 -2 1 -11 -40 -24	2836	20	161	1	776
-140425690032	0 0 -2 -42 -30 8	3242	26	352	1	1378
-148364065219	2 0 0 -27 -8 29	3333	24	180	1	851
-173284042324	1 2 1 -45 -44 32	3602	22	397	1	1142
-192791098835	-2 0 0 -44 -17 -1	3799	20	158	1	975
-239940287920	2 -1 0 -23 0 5	4238	24	340	1	1310
-2029968379648	1 -2 0 -39 25 -2	9638	22	705	1	4105
-2232446007232	-2 -1 -2 -31 -2 2	9702	27	417	1	3186
-2350044052703	1 1 0 -26 -6 31	9738	21	293	1	2168
-2381947502111	1 -2 -1 -21 -7 16	9747	25	271	1	2786
-2399520472320	2 -1 2 -43 24 -3	9752	25	498	1	3190
-10927578184064	2 2 0 -42 24 -1	10817	20	675	1	5293
-11320447096339	0 -1 2 -33 -37 -4	10824	23	823	1	5956
-12151587454848	0 2 2 -45 -36 18	10893	28	867	1	6096
-12716903749696	0 1 0 -38 24 1	10926	27	427	1	3478
-13225185144743	1 0 -1 -31 -11 44	10955	19	605	1	5106
-13690919414536	-1 -2 0 -26 7 47	10980	23	333	1	4107

Tabelle7.6: n = 6 , t = 2

Diskriminante	Polynom	C	p_i	dic	det	sec
105933117200	-2 0 2 -4 40 -5	4482	25	521	1	1861
109938523149	0 0 0 0 27 -44	4566	22	362	1	1604
110509871017	0 2 1 -20 -39 40	4578	21	271	1	1483
110923730748	0 2 1 -15 0 -6	4586	18	193	1	1138
113285979968	-2 1 2 -31 22 -15	4635	27	514	1	1859
974997625077	1 0 0 12 -42 -21	9145	27	474	1	3216
998719602317	2 0 1 -18 -48 40	9161	24	619	1	3019
1018613341533	2 1 -2 -23 9 -6	9174	24	1148	1	5318
1020152911941	1 0 0 -16 18 -19	9175	18	228	1	2507
1063012799389	-1 2 -1 24 38 -15	9203	25	854	1	4945
1067824824052	1 -2 2 19 37 -42	9206	30	1579	1	6923
9189275309192	1 -1 0 31 -36 -18	10692	23	992	1	7467
9493969464912	0 0 0 -34 46 -32	10715	28	1230	1	8861
9964811036416	-2 2 -2 -26 -36 -26	10750	26	809	1	5785
10241165737217	-2 0 1 -21 -16 27	10770	23	890	1	7004
10295068735689	0 -2 0 5 -43 37	10774	25	951	1	6145
10325509577605	1 0 0 8 -46 29	10776	37	547	2	4393
10356804091472	-1 0 -1 -44 23 -25	10778	21	478	1	3760

Tabelle7.7: n = 6 , t = 3

Diskriminante	Polynom	C	p_i	dic	det	sec
-103643948864	0 1 2 12 -4 17	6887	18	192	2	1543
-105784501731	0 0 0 -16 7 40	6958	15	514	1	2900
-120224797463	0 1 -1 35 -40 37	7417	34	533	1	2530
-123479444743	-1 0 1 -1 3 19	7517	23	308	1	2072
-132336806156	0 2 0 -13 6 32	7782	20	891	1	5053
-146921921875	2 0 0 15 -5 40	7935	20	427	1	3475
-887814935531	0 0 2 7 11 28	9083	30	820	2	4199
-914126660287	0 0 1 -5 -4 29	9103	22	523	1	3124
-973624509212	2 -1 1 20 -43 22	9144	21	744	1	5880
-991130615728	-1 -2 0 11 15 50	9156	23	928	2	4876
-1053050862592	2 0 0 22 -4 10	9196	26	946	1	3266
-1117200792980	1 1 -2 35 28 12	9236	22	1294	3	5095
-1213756780300	2 0 1 28 -31 33	9291	19	955	1	5558
-1328349530579	-1 -2 -2 12 -16 29	9351	16	357	1	3120
-8445378178036	1 1 -2 16 1 40	10632	18	1028	4	7184
-9299930740992	0 -2 2 25 -14 22	10701	27	1746	8	7229
-9603459094651	0 0 2 5 9 46	10724	26	1820	1	7843
-9654467569007	1 1 -2 48 0 19	10728	25	1013	1	7921
-10257801281787	0 0 1 26 -26 41	10771	23	845	1	6179
-10374805439419	1 0 0 10 -26 43	10779	27	706	1	3915
-10412688097764	1 -1 0 36 -49 46	10782	34	2157	2	9167
-10510628267696	1 0 2 26 28 38	10789	28	2171	1	9504
-11103650422019	1 0 0 35 13 7	10828	21	809	6	5095
-66064907014859	2 -2 1 40 -32 33	12152	20	1042	2	8308
-70391135841124	-2 0 1 42 -21 35	12200	24	2731	2	22666
-72438356744431	2 0 -1 43 39 19	12222	23	987	2	8083
-79743028760423	2 -2 -2 33 19 31	12296	27	1510	1	11355

Tabelle7.8: n = 7 , t = 1

Diskriminante	Polynom	C	p_i	dic	det	sec
-155197211916	-3 2 -3 0 12 -5 -2	888	24	99	1	398
-163095412855	-4 -2 3 1 17 -19 4	911	23	88	1	412
-365905242796	1 -4 1 0 -10 1 4	1364	24	172	1	698
-656778860672	1 -3 1 -4 -18 0 2	1827	41	237	1	843
-2954967756427	-4 -1 -2 3 11 -4 -3	3875	23	192	1	1448
-3168001287775	-4 1 1 3 17 -17 -3	4012	23	173	1	1457
-26749692981183	-2 -4 0 1 18 -8 -3	11471	21	176	1	3494

Tabelle7.9: n = 7 , t = 2

Diskriminante	Polynom	C	p_i	dic	det	sec
114419627536	-3 1 4 -4 -8 -4 16	1258	23	216	1	601
125789371456	-1 0 -1 0 4 -16 10	1319	17	163	1	580
179740465852	4 -3 3 4 -4 8 -9	1577	20	63	1	617
211805977248	-4 -1 1 -2 11 0 8	1711	22	218	1	804
225808613412	-4 2 -1 -1 0 6 -2	1767	25	184	1	881
289787873624	4 -4 -2 4 -1 5 -3	2002	24	274	1	1065
936606136644	-4 -2 4 -2 -5 -9 -1	3598	27	265	1	1845
960411875344	1 -3 4 -4 18 -12 2	3644	18	125	1	1131
965332770768	3 3 -2 3 6 -12 -3	3653	20	191	1	1400
1007186218069	-1 0 0 0 2 -11 -2	3731	24	172	1	1234
1017860636329	2 -1 4 0 13 16 -16	3751	20	188	1	1600
9330695006821	-2 -3 4 4 -10 -11 9	10703	23	310	1	3134
9398217785616	-2 2 -3 -3 2 -8 17	10708	25	348	1	4218
9613495389425	0 -4 -1 2 16 -2 -9	10724	22	354	1	4062
10027936009585	1 -2 3 4 3 -10 -9	10755	11	201	1	3230
10120814501212	-3 -3 0 -1 16 6 4	10761	30	748	1	8296
10376901733144	0 -2 0 -3 -11 -6 10	10779	21	376	1	4730
10840357575000	-2 0 0 0 15 -15 -15	10811	23	533	1	6604
10866002054164	4 1 3 4 6 -2 -8	10812	27	548	1	5788
91388814506697	1 -1 -1 1 25 17 -16	12401	19	728	2	6931
97463023437357	-2 -3 1 0 -5 -18 -1	12451	26	335	1	5495
98225781103032	0 -1 -1 -1 -11 -6 30	12457	30	678	1	7351
99467656868032	0 1 0 0 24 -3 -32	12466	24	533	1	6612
99537807148273	-4 0 -2 -4 2 -18 -9	12467	24	625	1	6579

Tabelle7.10: n = 7 , t = 3

Diskriminante	Polynom	C	p_i	dic	det	sec
-105827534579	1 1 -3 -1 8 0 -9	1955	19	122	1	761
-110183715675	2 -1 3 4 6 0 5	1995	26	153	1	793
-121296396399	0 3 -3 3 14 9 2	2093	20	125	1	714
-137220611455	0 0 2 -2 10 16 13	2226	29	175	1	829
-142026420779	2 1 0 0 -7 5 2	2265	20	163	4	1277
-142718474851	-2 1 -3 0 1 -3 -5	2271	26	139	1	796
-896325816715	0 2 -1 0 7 -2 11	5689	26	346	1	2026
-945698773507	0 0 0 -3 18 -7 10	5844	18	168	1	1869
-963030861312	0 0 1 1 -4 0 13	5897	20	382	1	3103
-1033507869743	1 1 0 0 -15 -18 -11	6109	23	170	1	1933
-1035167487467	-1 0 4 3 -8 -11 -8	6114	19	654	1	4170
-1170515537787	3 4 -4 -1 1 9 -6	6502	22	352	1	3006
-1174841090567	-1 0 1 -2 -13 -15 -11	6514	23	300	1	2565
-1252373757363	0 2 -1 -1 -12 15 -9	6725	26	339	1	3295
-9390572912947	0 -4 4 2 19 17 10	10708	18	331	1	4711
-9681431268547	3 2 -4 1 -11 -11 -19	10730	21	320	1	4508
-9833288125267	4 1 1 0 -3 -4 7	10741	30	410	1	5505
-9947777719167	0 0 0 0 18 -11 3	10749	19	251	2	3522
-10153675736324	0 0 1 4 4 -15 -17	10764	24	429	2	4856
-10201892207439	-1 -3 -1 -1 13 19 -18	10767	21	495	1	6105
-10449032497684	2 0 -4 3 -15 16 -1	10784	20	418	2	5316
-10717431866087	2 3 -1 -1 5 -7 17	10803	33	393	1	5185
-10777597781184	1 0 0 -1 -1 -11 15	10807	27	585	1	5199
-93575417722291	3 4 -4 3 3 16 -9	12419	22	526	1	6412
-96196507554264	1 -1 0 1 -12 13 -29	12441	29	1035	1	11270
-96522678760244	0 1 -1 1 38 -31 23	12443	35	1080	1	10236
-96951074579379	-3 -1 2 1 -8 15 -19	12447	22	1022	1	9850
-97909639200151	1 4 -4 2 0 -14 15	12454	22	792	1	9352

7.3 Berechnungen unter Annahme einer zusätzlichen Vermutung

Wie bereits in 3.4 und 4.1 erwähnt und durch die Beispiele in 7.2 bestätigt, ist die Wahl der Schranke

$$C = 12(\log \mid \Delta_{\mathcal{F}} \mid)^2$$

in der Praxis vermutlich viel zu schlecht. Daher gehen wir bei der Berechnung in Tab.7.11-14 zunächst davon aus, daß die Klassengruppe für diese Beispiele von den Primidealen p_i mit

$$N(p_i) \leq \min\{C, 1000\}$$

erzeugt wird. In den Beispielen in Tab. 7.15-24 benutzen wir die verschärfte Vermutung, daß die Klassengruppe von den Primidealen p_i mit

$$N(p_i) \leq 3 \log \mid \Delta_{\mathcal{F}} \mid$$

erzeugt wird. In allen Beispielen des letzten Abschnittes wurde diese Schranke bestätigt. Zur Wahl dieser Schranke siehe auch Erläuterungen zu Algorithmus 5.3.

Tabelle7.11: n = 4, t = 1

Diskriminante	Polynom	C	p_i	dic	det	sec
-501054548495692	59 -122 476 -1346	13789	190	7578	1	5472
-998110183613935	101 -1237 -188 -211	14314	178	10588	1	18665
-1258258077025467	-382 46 663 513	14507	167	5706	1	6396
-1279877621893167	357 987 1316 -825	14521	185	18099	1	27477
-1345730608139195	53 1327 -631 -1231	14563	163	6006	1	9725
-1536228343312123	233 333 -1 99	14674	160	6517	2	10333
-1918107726190700	67 -807 929 -480	14861	167	13800	1	24777
-2079770100058103	176 -889 157 -1376	14929	175	4308	1	6747
-2935469813341683	99 440 -32 683	15222	172	4486	1	9568
-4057430037194075	97 -552 1032 481	15500	161	3188	12	7129
-4504360369814523	-51 -1126 810 -239	15590	169	3740	1	6761
-4537819194641516	-327 -745 1235 -1284	15597	193	13531	1	21541
-5756480583003212	219 -930 542 -158	15803	180	7956	2	21016
-5906531540575288	214 439 544 -1086	15826	184	15461	2	24113
-7929120086078144	-46 1104 886 -363	16083	153	7536	1	18955
-34469886468050803	-6 1214 1425 -544	17400	174	16526	1	69418
-37848961195768731	-356 -1054 -1037 -629	17486	172	6303	2	31848
-50696748740429416	-321 724 -187 -587	17755	195	7118	1	24586
-70767033280707371	-292 558 -905 -111	18064	176	6835	4	47289
-81855277283853751	364 -1431 1427 -683	18200	170	6032	8	60064

Tabelle7.12: n = 4, t = 2

Diskriminante	Polynom	C'	p_i	dic	det	sec
113657758854949	-56 -808 -19 718	96	24	2803	1	2481
267064736262929	-184 1153 -1112 233	99	30	2121	1	2724
274227856464997	96 -44 -577 518	99	24	4589	1	5998
739181704556800	-60 1406 244 42	102	30	3190	24	11406
813831852280788	27 996 88 66	103	29	6800	2	25951
976392374688308	348 268 -1196 -1160	104	31	3887	1	4188
1145883046837628	33 -245 -1444 282	105	39	5549	1	9788
1333310284479348	212 1374 -1034 -916	105	30	4933	1	4716
2010335809464144	-2 1042 1152 1269	106	30	3830	16	14571
2362035770127905	7 696 -1175 1074	107	38	7625	4	32623
2675755459615485	-101 1311 -597 -567	107	29	3654	1	5256
2834662762777748	17 768 -44 562	107	30	3644	4	17837
3216042589075325	-117 -1482 -1215 -187	108	31	8077	1	17627
3273664040903472	-332 185 636 -333	108	30	1951	2	4120
3360459558279212	37 -746 505 481	108	31	2934	1	7172

Tabelle7.13: n = 5, t = 0

Diskriminante	Polynom	C	p_i	dic	det	sec
1179708370997609	-15 15 202 81 -145	14453	180	2589	1	1871
1319317583466916	-14 -7 386 -456 46	14546	178	4223	1	1970
1453662273916273	17 10 -199 -139 290	14627	153	5003	1	2981
8731537371729741	-17 -18 259 -15 -474	16168	185	19992	1	17127
11185904632252085	13 -16 -355 190 47	16387	163	4482	1	3259
13284552075380092	-18 0 231 -194 -150	16540	193	16590	1	15714
14706396469180776	12 -16 -307 164 294	16631	175	15203	1	14576
52875170883290857	-18 5 274 -74 -287	17794	163	3560	1	5578
59970645040594353	17 -1 -304 -273 312	17910	176	8881	1	15250
74258263652740557	17 -5 -488 480 183	18109	178	4769	1	6626
78729391190840209	19 1 -377 -328 -5	18163	166	2593	1	5915
113726436034776305	18 -1 -447 125 60	18509	186	9529	1	21925
184679806182811181	16 -15 -366 293 417	18968	180	3660	1	6016
224361093996160673	-19 -15 478 82 -255	19154	185	9915	1	26458
298814557995625481	18 -11 -431 421 121	19430	178	3436	1	13614
355669416723683976	-19 -19 356 246 -399	19599	186	9018	1	24630

Tabelle7.14: n = 6, t = 1

Diskriminante	Polynom	C	p_i	dic	det	sec
-1000925867776875	7 6 5 -5 -90 -35	14316	168	3989	2	2418
-1093228012809504	-7 6 18 -86 0 92	14390	180	6755	1	5508
-1143198699213528	-6 5 1 -72 -5 28	14427	168	9864	1	7922
-9844414232659751	2 7 87 48 -76 -19	16274	164	3285	1	3966
-10563606149444111	7 1 6 -8 -74 21	16336	164	3605	1	4356
-10716608739837983	-4 -8 33 -49 -82 13	16349	174	3442	1	2929
-10930407072344215	0 -7 -60 12 59 -18	16367	159	3709	1	2980
-78341299631237399	7 5 -78 -54 94 11	18159	170	4398	1	7632
-94561203584546511	0 -6 -15 -89 85 39	18335	193	7249	1	11868
-158848093871772119	-7 2 -71 -63 47 6	18825	196	4047	2	6486
-164835824111457544	5 -7 11 -91 -64 23	18860	182	3478	1	6139
-175040926656347328	8 -3 -42 94 70 -71	18917	182	5824	2	14117
-180419374280327867	-8 8 96 -60 -73 26	18946	170	7515	1	12612
-184028047633224303	-4 8 -87 64 81 6	18965	207	6072	1	11067
-334603357517060288	-6 -9 -22 86 74 -65	19540	173	3972	1	9325
-351185298715911167	0 3 -89 77 93 -4	19587	171	4601	1	10430
-398691138765017887	6 9 -93 -75 76 3	19710	169	4079	2	6254
-487358490887050924	6 0 53 58 -99 -41	19906	157	3335	2	5762

Tabelle7.15: n = 8, t = 2

Diskriminante	Polynom	C	p_i	dic	det	sec
10844271408328941289	5 8 -1 2 23 -60 -71 86	23054	193	5260	4	14232
18104475050224245509	-3 -4 6 -9 62 16 3 29	23596	166	3762	1	13973
20881574665099249741	-3 3 -5 6 12 -65 6 13	23748	172	3870	1	11618
69957310293379550021	2 -5 -7 -2 62 -39 -16 9	25056	170	2527	1	8269
109043261179609633209	6 0 -6 5 -90 36 -75 25	25545	176	7540	1	57262
321579698793005584748	-5 4 -6 -5 69 93 -68 -12	26757	180	5911	1	55731
388681255189824172673	3 -5 -4 1 -50 96 -8 -21	26972	176	4460	2	32605
438664384063177571241	2 -2 9 6 52 23 -31 -12	27110	183	13880	1	56756
611465378445730577152	-4 1 0 7 44 -47 10 38	27490	183	8290	1	36171
818241561200159219212	-5 6 -8 -4 44 -66 99 -49	27826	182	7660	1	68867
915266833349661738181	-7 -8 -7 -9 70 -89 97 -26	27955	161	8887	1	63876

Tabelle7.16: n = 6, t = 2

Diskriminante	Polynom	C'	p_i	dic	det	sec
1180812036306933	3 2 -22 -98 -73 -26	102	27	2039	1	1309
1259621319604404	-2 2 57 54 -51 -11	102	31	2011	1	1094
7696197430489185	-7 9 -39 42 -19 -42	108	35	4955	1	4957
7711049700554036	9 8 -31 94 -46 50	108	32	2587	1	2226
7781930831223232	4 -7 22 0 50 61	108	31	1729	2	2250
8096045035485164	-3 9 23 -13 96 70	108	30	2206	1	2073
9787170505428477	-2 0 -68 -78 45 75	108	41	11652	1	12744
84960613589175524	-3 9 78 -19 -25 -37	114	30	5425	4	7000
90869156526685645	8 5 64 3 43 -45	117	35	1628	1	2209
95828349503198945	-7 8 -41 -47 25 60	117	34	3365	1	4607
102940315755944512	7 4 92 -59 47 16	117	30	4670	1	6358
103938306062887613	-8 -8 -68 -5 1 -22	117	32	2674	1	3287
106532267610236457	-9 5 91 -10 -99 -88	117	34	2502	2	3801
699250819307145961	5 -7 -78 59 -15 -74	123	38	2444	2	11485
709826694001256353	2 -5 84 76 -21 34	123	30	2063	1	7147
750537331434654068	-5 2 85 -4 84 89	123	31	3421	1	6109
842754237340466000	-3 -6 92 -59 -35 -52	123	31	5714	1	12185
914513439031614637	6 5 -90 -23 59 -56	123	41	6550	3	24986
959128567972040925	5 -6 70 -42 -24 93	123	28	1710	1	3875
973537856278838877	0 -6 86 76 39 -39	123	31	2143	1	6736
993990104767291321	5 9 -92 24 28 -53	123	33	2404	2	9187
998079111908728409	-3 4 81 -15 -49 -66	123	28	2774	1	9145

Tabelle7.17: n = 6, t = 3

Diskriminante	Polynom	C'	p_i	dic	det	sec
-1017587891782019	-2 5 -24 57 -9 47	102	33	2405	1	2139
-1246338595449799	3 -5 21 86 -99 46	102	25	1612	1	1630
-2374117412388932	-3 -1 -6 89 -42 14	105	27	5335	1	5054
-2430359774936092	0 0 -51 68 46 98	105	39	6578	1	4044
-6508722880910000	3 8 -64 65 -5 90	108	33	5663	2	7174
-9049867105287232	-2 7 -12 84 -28 75	108	37	3394	1	3903
-14950027695860272	5 4 -42 6 50 66	111	26	5171	1	6064
-27975749525236319	1 -9 -41 92 47 54	111	28	3590	1	6456
-33460447739964352	0 0 -44 98 2 34	114	29	2591	1	4928
-36313299831581428	2 0 -7 85 28 62	114	33	4647	2	8810
-41259945241024455	5 0 -27 79 55 40	114	35	4468	1	10146
-47655845258690883	3 2 -56 94 -33 48	114	34	7281	1	8950
-69624022325967623	3 3 -1 98 91 80	114	35	3957	1	8679
-204946412202701511	9 8 -85 43 39 69	117	38	5550	5	22954

Tabelle7.18: n = 7, t = 1

Diskriminante	Polynom	C'	p_i	dic	det	sec
-1386270770816451	3 5 -1 -49 0 47 -3	102	35	763	1	965
-1396513067644627	2 -5 0 -2 -27 -3 8	102	28	1043	1	789
-2142571484792743	-3 -4 1 34 -10 -41 19	105	25	441	1	402
-9324179179867584	0 -6 0 -33 12 46 -6	108	27	1619	1	1302
-10460793635496772	-2 -5 -5 -23 19 23 -14	108	24	1265	1	1270
-12620821556479483	-1 0 4 -36 23 19 -7	111	27	682	1	731
-16143808476247859	0 -3 6 -31 12 28 -5	111	31	881	1	964
-23651510084787912	-4 1 0 -32 35 36 -33	111	22	1271	1	827
-36341740819302448	6 -3 6 -30 -30 20 -2	114	30	2289	1	3612
-45458942747137627	-6 2 -1 18 26 -43 12	114	33	1496	1	1740
-53569584035379479	2 5 -4 -48 -21 49 11	114	31	748	1	1126
-85573609003896075	-5 0 2 -39 27 43 -4	114	31	1764	1	3176
-103527031378082375	-5 3 4 -2 44 -23 -42	117	38	3317	1	7215
-107738242898163996	3 -5 -4 0 -44 48 3	117	33	1568	1	2071
-110291779666523428	2 -4 -2 -13 -39 40 42	117	30	3375	1	4242
-233330799399733568	-3 5 -3 -44 24 47 5	117	31	1384	1	2065
-246028439183223744	-6 3 -1 -22 42 18 -2	120	31	2818	1	6910

Tabelle7.19: n = 7, t = 2

Diskriminante	Polynom	C'	p_i	dic	det	sec
8878794084057613	5 -3 0 41 41 8 -29	108	22	1135	1	1627
9299423980709737	4 0 0 -19 6 32 11	108	24	754	1	1007
9381251719735101	-1 -6 4 -34 9 6 27	108	36	4145	1	7647
9746570436047233	-1 2 4 21 -40 -47 -11	108	40	1090	1	2131
10038811737149200	-5 -4 2 17 -49 -28 16	108	25	1428	1	1295
10124162474314752	1 -5 -4 -39 -45 -33 -18	108	30	3120	1	4587
10675394876749685	2 6 -2 -48 -5 36 25	108	21	1215	1	1383
91774877847778065	3 -6 -1 -11 -11 -11 24	117	39	6601	1	18167
98769497949065836	-3 4 3 -8 -45 -28 24	117	42	4803	1	11492
99165768152571297	2 0 -4 7 -30 -34 25	117	27	1489	2	3991
99294507547325892	5 1 -4 -31 -13 11 -18	117	34	10146	1	12291
102034482240175181	2 0 3 9 -43 -25 27	117	19	1280	4	2706
104195587389147328	5 4 4 -6 -4 -37 -29	117	27	2156	1	4785
744884926185688973	-6 -1 0 -8 23 -19 36	123	27	3965	1	4635
810421771468421580	-6 -6 -3 14 28 25 -15	123	34	3683	1	7702
817149536570522433	0 -5 5 -35 -38 30 33	123	24	2171	1	4157
834528137162536669	4 0 5 -5 28 23 -37	123	28	1332	1	3150
875562287086552412	4 0 3 -4 30 -11 -41	123	24	2045	1	4412
964357542942253373	6 3 5 48 0 17 35	123	38	2522	1	13087
964610100799403309	1 2 -3 -31 10 -4 -49	123	32	1353	1	3431

Tabelle7.20: n = 7, t = 3

Diskriminante	Polynom	C'	p_i	dic	det	sec
-1009894897370704	2 3 0 19 -28 4 24	102	25	1164	1	1273
-1146263278096636	0 3 -2 0 17 -12 31	102	33	1394	1	2343
-1176588721058220	0 3 1 -12 11 -2 28	102	28	3223	1	4816
-1178356687368596	3 6 -2 -1 -32 40 13	102	25	1211	1	1877
-8569662687486203	-1 -6 -6 34 -19 26 7	108	32	2104	1	2870
-9470719389054492	0 -2 0 20 11 41 13	108	30	2887	1	3917
-9477873101165859	-3 1 3 -35 -28 -19 -49	108	38	2236	4	4591
-9551851031566052	0 1 -3 25 31 4 12	108	40	3185	1	5034
-11234505097419328	3 1 5 32 -44 -4 -2	108	32	3321	2	7155
-11452066847487227	0 -6 3 43 15 -11 -7	108	22	1175	2	2523
-11591260729645747	1 1 4 -11 -42 3 -4	108	29	1426	1	2068
-11708462812129036	3 0 1 24 18 -37 28	108	34	2906	1	4775
-74949410358384263	-2 6 0 -11 -2 49 23	114	33	2287	1	6022
-76491905156365667	6 5 -6 43 8 -6 11	114	33	2794	2	9912
-87396916945310215	0 0 0 0 42 12 41	117	30	3457	2	12653
-88578569230432803	2 5 -6 -39 39 -12 39	117	35	2450	1	6061
-96645296865709936	-2 -3 -5 -24 -38 32 -28	117	26	1612	1	2629
-100668081917533123	-2 -5 3 40 -25 -48 -41	117	42	2348	1	6518
-105777953615933883	-2 3 0 29 0 42 15	117	37	2311	1	4997
-708623163328970151	-3 3 -4 37 45 -3 -19	123	30	2538	1	14504
-739056965245712036	0 5 6 47 -21 -4 -24	123	45	7887	2	30672
-744406604403644416	0 1 5 30 -8 -2 -50	123	30	3253	1	11597
-761206828983034800	1 5 -2 35 20 -10 -29	123	38	4079	1	24465

Tabelle7.21: n = 9, t = 1, .., 4

Diskriminante	Polynom	C'	p_i	dic	det	sec
-25156205962915538672	3 -4 -7 5 -41 -16 68 4 -20	132	31	1110	1	3773
327449714798985864193 6	1 -3 -7 8 -57 78 61 -66 -18	147	26	3804	1	12421
3798503845377851062020	6 5 -5 0 8 68 52 -54 9	147	38	3700	1	24129
-1477048927836975068720	0 -4 -3 -6 45 -70 62 -72 72	144	39	4719	1	24667
-2869031634535963209031	-2 -2 4 1 10 -16 -26 -9 -31	147	40	2876	1	22747
-3556714353668269465824	-5 0 1 4 -29 5 17 -24 12	147	29	6618	1	50764
-4380142996935593946027	7 3 2 -5 -17 45 7 -25 12	147	33	6319	1	46842
-5506318053079005225712	3 -8 2 9 -45 -20 32 -39 -59	150	38	4107	1	43223
-7078658284263796128272	3 7 0 -9 12 46 14 -64 -32	150	37	7883	1	53208
-9380180315509762184000	3 -5 0 -6 9 5 -25 -50 45	150	48	6879	1	73709
213046870059239548629	0 1 -8 -2 19 -43 56 -21 6	138	44	3768	1	28075
1662243442552682312380	0 2 -8 -5 50 5 -80 94 -52	144	40	4189	1	45965
4469099320978568190352	3 0 0 8 -49 42 8 -9 -10	147	40	12996	1	12947

Tabelle7.22: n = 10, t = 3

Diskriminante	Polynom	C'	p_i	dic	det	sec
-9838247980148956	2 -1 0 2 -2 0 -1 4 10 2	108	31	156	1	810
-775851026363386048	1 -1 2 0 -2 0 0 9 -7 -11	123	25	481	1	1391
-1597614817453799680	0 -2 0 -2 0 1 0 -7 -2 16	123	39	882	1	3088
-1646159693749184883	-2 0 1 0 0 0 -1 -10 28 -16	123	38	714	1	2647
-2446691219680362127	0 -2 2 -2 1 0 -2 15 -16 1	126	35	665	1	2207
-31041727960075825388	2 0 0 0 2 -2 1 -12 2 2	132	26	1539	1	4609
-33869251397801031747	-1 0 2 -2 1 -2 2 -8 -17 -3	132	34	768	1	3016
-34558829278542331772	-2 0 -1 -1 0 0 0 1 11 -5	132	39	1065	1	4111
-34829606588998886784	-2 0 1 2 -1 1 -1 -16 -2 4	132	33	1477	1	4470
-89711519634453896363	1 -1 2 2 0 -1 0 -22 -20 3	135	36	843	1	3627

Tabelle7.23: n = 10, t = 4

Diskriminante	Polynom	C'	p_i	dic	det	sec
8680623892288144	-1 -2 2 -1 0 1 2 8 -8 -4	108	30	275	2	1916
11172589001675933	1 0 -1 0 -2 0 -2 9 -4 -1	108	26	144	1	683
23146554427558729	-2 0 2 -2 -1 0 2 0 -1 -4	111	36	220	1	6655
26385999853118200	2 1 1 -2 0 1 1 16 -26 9	111	26	242	1	822
29451106995159132	0 -2 -1 2 1 -1 0 -15 -2 18	111	27	336	1	1503
47634346094543104	-2 0 2 0 -2 2 2 -18 28 -16	114	29	686	1	1410
182430876435622449	0 -1 0 0 0 0 2 -1 -9 -4	117	24	473	1	958
263379257278601505	0 -2 0 -2 -1 0 2 -4 -27 -24	120	26	616	1	1462
446434141309507584	0 -2 0 0 0 0 0 -4 6 5	120	34	990	1	3240
703076362452344432	0 -2 -1 -1 0 0 0 11 7 -4	123	37	711	1	2489
826070376822497025	-1 -2 -2 2 0 1 1 6 0 5	123	33	639	1	2621
1304738583962978704	0 -1 0 0 -1 1 0 1 11 4	123	28	1069	1	3161
1962191094119120896	0 -2 0 0 2 -1 0 11 10 1	126	29	760	1	3576
1966525522476344848	-2 1 -2 -1 1 0 2 0 2 -8	126	23	1189	1	3611

Tabelle7.24: n = 10, t = 5

Diskriminante	Polynom	C'	p_i	dic	det	sec
-124046572051732080	1 1 0 1 -2 1 0 2 -6 6	117	26	657	1	2074
-225044626879533808	1 -2 -2 0 1 1 0 -3 -4 11	117	31	855	1	3067
-3056620718288261419	0 2 1 0 2 0 2 14 -9 4	126	39	1032	2	4833
-3875315312211377091	1 1 -2 2 0 -1 -1 4 1 6	126	40	2704	1	10599
-5463841164076575671	0 -1 2 1 1 -2 -1 5 -14 9	129	26	1018	10	3600
-8110250369065788487	0 1 2 1 -2 -1 0 20 4 7	129	28	1467	1	5953
-8199500440631771723	0 1 0 -2 2 0 -1 11 -4 15	129	33	1029	1	3181
-11070890315667213312	0 0 -2 0 0 -2 2 13 16 20	129	33	6396	1	27460
-13626624925214747448	-2 0 0 2 0 0 1 10 -6 12	132	37	5313	1	22405
-19389453993704041475	0 1 0 -2 0 -1 0 7 19 23	132	28	1809	1	5059
-22768427695111815707	0 -1 2 1 0 2 -2 13 3 7	132	34	1962	1	9382
-26156263193079313875	-1 -1 2 -1 0 2 -1 14 5 10	132	39	4358	1	21315

Anhang A

Explizite Normschranken

Explizite Zimmert-Schranken bzw. Verbesserungen dieser Schranke

$$\frac{1}{x}\sqrt{|\ \Delta_{\mathcal{F}}\ |}$$

für verschiedene Signaturen, die uns von Johannes Graf von Schmettow zur Verfügung gestellt wurden.

```
if( n == 2 && t == 0) x = 1.760;
if( n == 2 && t == 1) x = 1.4;
if( n == 3 && t == 0) x = 4.636;
if( n == 3 && t == 1) x = 3.355;
if( n == 4 && t == 0) x = 14.45;
if( n == 4 && t == 1) x = 9.479;
if( n == 4 && t == 2) x = 6.792;
if( n == 5 && t == 1) x = 32.1271;
if( n == 5 && t == 2) x = 21.1190;
if( n == 6 && t == 0) x = 188.124;
if( n == 6 && t == 1) x = 115.594;
if( n == 6 && t == 2) x = 72.6303;
if( n == 6 && t == 3) x = 46.7493;
if( n == 7 && t == 0) x = 745.668;
if( n == 7 && t == 1) x = 443.722;
if( n == 7 && t == 2) x = 269.005;
if( n == 7 && t == 3) x = 166.419;
if( n == 8 && t == 0) x = 3088.20;
if( n == 8 && t == 1) x = 1790.34;
if( n == 8 && t == 2) x = 1054.39;
if( n == 8 && t == 3) x = 631.686;
if( n == 8 && t == 4) x = 385.531;
if( n == 9 && t == 0) x = 13250.4;
```

```
if( n == 9 && t == 1) x = 7516.57;
if( n == 9 && t == 2) x = 4321.83;
if( n == 9 && t == 3) x = 2521.6;
if( n == 9 && t == 4) x = 1494.78;
if( n == 10 && t == 0) x = 58542.0;
if( n == 10 && t == 1) x = 32602.2;
if( n == 10 && t == 2) x = 18370.1;
if( n == 10 && t == 3) x = 10482.8;
if( n == 10 && t == 4) x = 6064.55;
if( n == 10 && t == 5) x = 3560.79;
if( n == 11 && t == 0) x = 265123.0;
if( n == 11 && t == 1) x = 145319.0;
if( n == 11 && t == 2) x = 80475.0;
if( n == 11 && t == 3) x = 45062.2;
if( n == 11 && t == 4) x = 25536.1;
if( n == 11 && t == 5) x = 14658.6;
```

Literaturverzeichnis

[Bach] Erik Bach. Explicit bounds for primality testing and related problems, Math. Comp. **55** (1990), 355-380.

[Bu1] Johannes Buchmann. A subexponential algorithm for the determination of class groups and regulators of algebraic number fields. Séminaire de Théorie des Nombres, Paris 1988-1989. Birkhäuser, 1990, 27-41

[Bu2] Johannes Buchmann. On the computation of units and class numbers by a generalization of Lagrange's algorithm. J. Number Theory **26** (1987), 8-30.

[Bu3] Johannes Buchmann. On the period length of the generalized Lagrange algorithm. J. Number Theory **26** (1987), 31-37.

[Bu4] Johannes Buchmann. Zur Komplexität der Berechnung von Einheiten und Klassenzahlen algebraischer Zahlkörper. Habilitationsschrift, Düsseldorf, 1987.

[Bu5] Johannes Buchmann. Algebraic Number Theory. Saarbrücken, Skript zur Vorlesung WS 89/90.

[BuLe] Johannes Buchmann und Hendrik Lenstra. Computing maximal orders and decomposing primes in number fields, 1991

[BuWi] Johannes Buchmann und H.C. Williams. On the Computation of the Class Number of an Algebraic Number Field. Math. Comp. **53** (1989), 679-688.

[Coh] Henri Cohen. A course in computational algebraic number theory. Buch-Manuskript, Bordeaux, 1991.

[DoKaTr] P.D. Domich, R.Kannan und L.E. Trotter. Hermite normal form computation using modulo determinant arithmetic. Math. Op. Res. **12** (1987), 57-59.

[Düll] Stephan Düllmann. Ein Algorithmus zur Bestimmung der Klassengruppe positiv definiter binärer quadratischer Formen. Dissertation, Saarbrücken, 1991.

[Eck1] Bruce Eckel. Using C++. McGraw-Hill, 1989.

[Eck2] Bruce Eckel. C++ Inside Out. McGraw-Hill, 1993.

[Ga] C.F. Gauss. Disquisitiones Arithmethicae. Fleischer, Leipzig, 1801.

[HaMcC] J.L. Hafner und K.S. McCurley. A rigorous subexponential algorithm for computation of class groups, Journal AMS, to appear.

[Hua] Hua Loo Keng. Introduction to Number Theory. Springer-Verlag, New York 1982.

[Hun] Thomas W. Hungerford. Algebra. Springer-Verlag, New York, 1974.

[JoSte] Robin Jones und Ian Stewart. The art of C programming. Springer-Verlag, New York, 1987.

[KerRi] Kernighan/Ritchie. The C programming language. Prentice-Hall, 1978.

[Knuth] Donald E. Knuth. The Art of Computer Programming II. Addison-Wesley, 1981.

[Kob] Neal Koblitz. A Course in Number Theory and Cryptography. Springer Verlag, New York, 1987.

[LEDA] Stefan Näher. LEDA User Manual, Version 2.0.1. Saarbrücken, 1991.

[Lang] Serge Lang. Algebraic Number Theory. Springer-Verlag, New York, 1986.

[LePo] H.W. Lenstra und M. Pohst. Computational algebraic number theory. Birkhäuser, 1991.

[Mar] D. A. Marcus. Number Fields. Springer-Verlag, New York, 1977.

[Min] H. Minkowski. Diophantische Approximation. Teubner, 1907.

[Mol] Richard A. Mollin. Number Theory and Applications. ASI Series, 1989.

[Nar] W. Narkiewicz. Elementary and analytic theory of algebraic numbers. Warszawa, 1971.

[PARI] Batut/Bernardi/Cohen/Olivier. User's Guide to PARI-GP Version 1.37. Bordeaux, 1992.

[PoZa1] M. Pohst und H. Zassenhaus. Algorithmic Algebraic Number Theory. Cambridge University Press, 1989.

[PoZa] M. Pohst und H. Zassenhaus. Über die Berechnung von Klassenzahlen und Klassengruppen algebraischer Zahlkörper. J. Reine Angew. Math. **361** (1985), 50-72.

[Pohl] Ira Pohl. C++ for C programmers. Benjamin/Cummings, 1989.

[Schm] Johannes Graf von Schmettow. Zur Berechnung der Klassengruppe algebraischer Zahlkörper. Diplomarbeit, Düsseldorf, 1987.

[Schm1] Johannes Graf von Schmettow. Beiträge zur Klassengruppenberechnung Dissertation, Düsseldorf, 1991.

[Schr] Alexander Schrijver. Theory of linear and integer programming. John Wiley and Sons, Chichester 1987.

[Stew] Ian Stewart/David Tall. Algebraic Number Theory. Chapman and Hall, Cambridge 1987.

[Waer1] B.L. van der Waerden. Algebra I. Springer-Verlag, Berlin, 1971.

[Waer2] B.L. van der Waerden. Algebra II. Springer-Verlag, Berlin, 1967.

[Web] Damian Weber. Zerlegung von Primzahlen in Primideale. Diplomarbeit, Saarbrücken, 1993.

[Zim] R. Zimmert. Ideale kleiner Norm in Idealklassen und eine Regulatorabschätzung. Invent. math. **62** (1981), 367-380.