Modulare lineare Algebra über algebraischen Zahlkörpern

Wissenschaftliche Hausarbeit
zur Ersten Wissenschaftlichen Staatsprüfung
für das Amt des Studienrats

Vorgelegt von:
Harald Bartel
Cecilienstraße 14
12307 Berlin

Berlin, den 1. Juli 1997

Inhaltsverzeichnis

KAPITEL 1

Einleitung

Zunächst wollen wir den Begriff „modulares Verfahren der linearen Algebra“ festlegen. Bei dem modularen Verfahren wird die Berechnung eines Wertes über einer algebraischen Struktur zurückgeführt auf Berechnungen in einem oder mehreren homomorphen Bildern dieser Struktur.

Das modulare Verfahren besteht entsprechend aus drei Hauptschritten. Erstens wird das Problem auf die homomorphen Bilder abgebildet (Hintransformation), zweitens findet die Berechnung in dem bzw. den homomorphen Bildern statt. Rechnet man in mehreren homomorphen Bildern, so werden vor der Rücktransformation die Ergebnisse mit Hilfe des chinesischen Restsatzes zu einem Ergebnis in einem geeigneten homomorphen Bild zusammengesetzt. Drittens wird unter Ausnutzung der sogenannten Schrankenbedingung der ursprünglich gesuchte Wert in der ursprünglichen Struktur berechnet (Rücktransformation).

Trotz der für die Hin- und Rücktransformation benötigten Zeit kann das modulare Verfahren eine Zeitersparnis gegenüber der direkten Berechnung liefern, da das Rechnen in den homomorphen Bilder oft schneller durchgeführt werden kann als in der ursprünglichen Struktur.

Wie im ersten Abschnitt von Kapitel 2 dargestellt wird, ist ein homomorphes Bild nichts anderes als eine Faktorstruktur nach einer Kongruenz. Historisch wurde die Bedeutung des Kongruenzbegriffs für die Zahlentheorie zuerst von

Gauss (1777–1855) voll erkannt, der ihn ganz an den Anfang seiner *„Disquisitiones Arithmeticae“* [Gau] stellte. Daran anschließend entwickelte er eine sehr weitgehende und reichhaltige Theorie der Kongruenzen, die sofort allgemein akzeptiert und zu einem bleibenden Bestandteil der Zahlentheorie wurde.

Abgesehen vom Kongruenzbegriff der Geometrie wurde hier, historisch erstmalig, rein formal mit einer Äquivalenzrelation operiert. Da Kongruenzrelationen verträglich bezüglich der Verknüpfungen der algebraischen Struktur sind, kann man mit ihnen weitgehend so rechnen, wie man es mit Gleichungen gewohnt ist. Wegen dieser starken Analogie hat Gauss das Zeichen „$\equiv$“ für „kongruent“ in Anlehnung an das Gleichheitszeichen gewählt.

Es soll jedoch nicht unerwähnt bleiben, daß sich der zahlentheoretische Begriff der Kongruenz vor Gauss bereits ab 1730 in Briefen findet, die Goldbach (1690–1764) an Euler (1707–1783) geschrieben hat. Goldbach verwendet anstelle des Symbols „$\equiv$“ das Symbol „$\mp$“, allerdings blieb bei ihm im Vergleich zu Gauss der Kongruenzkalkül noch ganz in den Anfängen stecken. Eine Gleichung der Form $a \equiv b \pmod{m}$ liest man als “a ist kongruent b *modulo* m“, eine Sprechweise, der das modulare Verfahren seinen Namen verdankt. Diese Gleichung ist äquivalent mit der Gleichung $a + (m) = b + (m)$ im homomorphen Bild. Lipson bezeichnet das modulare Verfahren als „*Methode der homomorphen Bilder*“ [Lip81].

Mit Hilfe des chinesischen Restsatzes kann man simultane lineare Kongruenzen lösen. Da Fragestellungen, die auf Probleme dieses Typs rekurrieren, nach heutigem Wissen erstmals im Suan-ching (Handbuch der Arithmetik) des Chinesen Sun-Tsu ca. im ersten Jahrhundert nach Christus Erwähnung finden [Dic52, S. 57], spricht man vom „chinesischen Restsatz“. Dicson [Dic52] erwähnt weitere Mathematiker, u.a. Euler, im Zusammenhang mit dem Problem der simultanen Kongruenzen. Jedoch ist es wiederum Gauss, der zweifelsohne dieses Problem am prägnantesten in algorithmischer Darstellung in *Disquisitiones Arithmeticae, Art. 32, 36* behandelt hat.

Ziel dieser Arbeit ist es, die Verwendbarkeit modularer Algorithmen bei Problemen der linearen Algebra über algebraischen Zahlkörpern zu untersuchen, deren Korrektheit zu zeigen und das Laufzeitverhalten der modularen Algorithmen mit

dem der nichtmodularen zu vergleichen. Dies geschieht exemplarisch anhand der Determinantenberechnung über algebraischen Zahlkörpern und Multiplikation algebraischer Zahlen.

Ein Schwerpunkt dieser Arbeit ist die Implementierung der neuen modularen Algorithmen in dem Computeralgebrasystem KANT-V4 [DFK+96], welches bereits die entsprechenden nichtmodularen Algorithmen enthält. Beim Laufzeitvergleich dienen die nichtmodularen Algorithmen von KANT-V4 als Referenz. Die Beispielrechnungen für den Laufzeitvergleich wurden mit dem Computeralgebrasystem KANT-V4 in der Oberfläche KASH ausgeführt.

In Kapitel 2 wird zunächst die Grundidee des allgemeinen modularen Verfahrens sowie der chinesische Restsatz thematisiert. Im zweiten Abschnitt wird das modulare Verfahren für spezielle Ringe dargestellt. Diese Ringe sind $\mathbb{Z}$ und kommutative freie Algebren über $\mathbb{Z}$. Der dritte Abschnitt stellt Grundlagen über algebraische Zahlkörper und Ordnungen zusammen. Für das Rechnen in algebraischen Zahlkörpern sind Ordnungen wichtige Strukturen. Es wird gezeigt, daß Ordnungen geeignete Strukturen für das modulare Verfahren sind. Im letzten Abschnitt des Kapitels wird darauf eingegangen, wie wir das Laufzeitverhalten von Algorithmen beschreiben können.

Kapitel 3 teilt sich in zwei Abschnitte. Jeder Abschnitt befaßt sich mit einer Aufgabenstellung der linearen Algebra über algebraischen Zahlkörpern. Diese beiden Abschnitte sind gleich gegliedert. Zunächst wird der nichtmodulare Algorithmus dargestellt. Anschließend wird der neue modulare Algorithmus beschrieben und seine Korrektheit bewiesen. Im dritten Unterabschnitt werden jeweils die Laufzeiten der Varianten verglichen.

Bei der Nummerierung von Sätzen, Definitionen, Abbildungen und Tabellen wird die Kapitelnummer vorangestellt. Innerhalb der Kapitel ist die Nummerierung fortlaufend.

Das Literaturverzeichnis und eine Liste der verwendeten Bezeichnungen mit deren Bedeutung befinden sich am Ende der Arbeit.

KAPITEL 2

Grundlagen

1. Homomorphe Bilder und der chinesische Restsatz

Eine algebraische Struktur schreiben wir als Paar (A, Ω), wobei A eine Menge und Ω eine Menge von inneren Verknüpfungen auf A ist. Die Menge der Ω-Terme bezeichnen wir mit $T(\Omega)$.

Zunächst werden einige elementare jedoch für das modulare Verfahren grundlegende Sätze und Definitionen aus [Mey80a], [Lip81] aufgeführt.

DEFINITION 2.1. *Seien (A, Ω) und (H, Ω) algebraische Strukturen. Dann heißt eine Funkion $\Phi : A \to H$ Homomorphismus von (A, Ω) nach (H, Ω), falls für alle $\omega \in \Omega$ und alle $a_1, \ldots, a_d \in A$*

$$\Phi(\omega(a_1, \ldots, a_d)) = \omega(\Phi(a_1), \ldots, \Phi(a_d))$$

gilt. Dabei sei d die Stelligkeit von ω.

Obwohl die Verknüpfungen der Strukturen (A, Ω) und (H, Ω) gleich bezeichnet werden, können keine Zweideutigkeiten auftreten, da die Argumente die Bedeutung eindeutig festlegen.

BEZEICHNUNG 2.2. *Für einen Homomorphismus $\Phi : A \to H$ von (A, Ω) nach (H, Ω) ist die Menge $\Phi(A)$ zusammen mit der Menge $\Omega \downarrow_{\Phi(A)}$, der auf $\Phi(A)$ einge-*

schränkten Vernüpfungen aus Ω*, bildet eine algebraische Struktur* $(\Phi(A), \Omega\downarrow_{\Phi(A)})$*. Diese nennen wir homomorphes Bild von* (A, Ω) *unter* Φ*.*

Ist $\Phi : A \to H$ ein surjektiver Homomorphismus von (A, Ω) nach (H, Ω), so ist nach Bezeichnung 2.2 (H, Ω) homomorphes Bild von (A, Ω).

Für einen Homomorphismus Φ gilt die obige Verträglichkeitsbedingung nicht nur für alle Verknüpfungen $\omega \in \Omega$, sondern sogar für alle Ω-$Terme$.

Satz 2.3. *(Homomorphismus-Theorem) Sei* $\Phi : A \to H$ *ein Homomorphismus von* (A, Ω) *nach* (H, Ω)*. Sei* $t(\mathbf{x}) := t(x_1, \ldots, x_d)$ *ein* Ω-$Term$*. Dann gilt für alle* $\mathbf{a} := (a_1, \ldots, a_d) \in A^d$

$$\Phi(t(\mathbf{a})) = t(\Phi(\mathbf{a})).$$

Dabei ist $\Phi(\mathbf{a}) := (\Phi(a_1), \ldots, \Phi(a_d))$*.*

Dies entspricht folgendem kommutativen Diagramm:

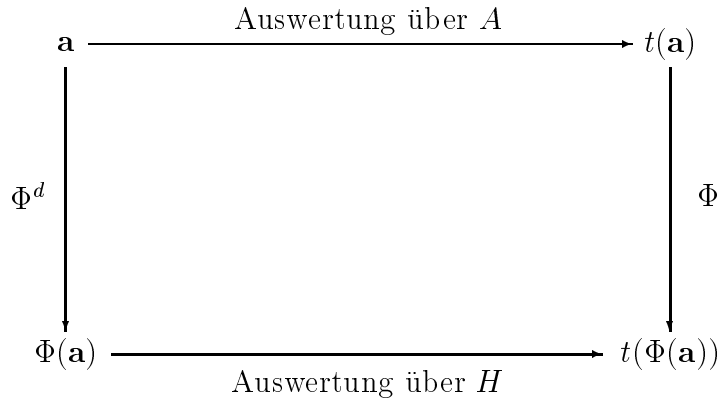

Abbildung 2.1. Kommutatives Diagramm zum Homomorphismus-Theorem (Satz 2.3)

Die Grundidee des modularen Verfahrens besteht darin, Rechenoperationen in einer algebraischen Struktur (A, Ω) durch Rechenoperationen in einfacheren Strukturen zu ersetzen. Die hierfür geeigneten Strukturen sind nach Satz 2.3 die homomorphen Bilder von (A, Ω).

Nun ist es interessant zu wissen, welche Gestalt homomorphe Bilder annehmen können. Eine Charakterisierung der homomorphen Bilder bis auf Isomorphie liefern die beiden folgenden Sätze.

SATZ 2.4. *Sei (A,Ω) eine algebraische Struktur und E eine Kongruenzrelation auf (A,Ω). Sei $A/E := \{[a]_E \mid a \in A\}$ die Menge der Äquivalenzklassen von A und seien für alle $\omega \in \Omega$ Verknüpfungen auf A/E (bezeichnet durch dasselbe Symbol) definiert durch*

$$\omega([a_1]_E, \ldots, [a_d]_E) := [\omega(a_1, \ldots, a_d)]_E.$$

Dann ist $(A/E,\Omega)$ eine algebraische Struktur und heißt Quotientenstruktur von (A,Ω) nach E.

Quotientenstrukturen nach Kongruenzrelationen sind somit homomorphe Bilder. Satz 2.7 besagt, daß jedes homomorphe Bild isomorph zu einer Quotientenstruktur nach einer Kongruenzrelation ist.

BEZEICHNUNG 2.5. *Sei $\Phi : A \to H$ ein Homomorphismus von (A,Ω) nach (H,Ω). Dann bezeichne E_Φ die von Φ mittels*

$$a_1 E_\Phi a_2 \Leftrightarrow \Phi(a_1) = \Phi(a_2)$$

induzierte Relation auf A.

SATZ 2.6. *Sei $\Phi : A \to H$ ein Homomorphismusvon (A,Ω) nach (H,Ω). Dann ist E_Φ eine Kongruenzrelation auf A.*

SATZ 2.7. *(Isomorphie-Satz) Sei (A,Ω) algebraische Struktur und (H,Ω) ein homomorphes Bild von (A,Ω) unter dem Homomorphismus Φ. Dann gilt $H \cong A/E_\Phi$.*

Nach den Sätzen 2.4, 2.7 sind die homomorphen Bilder einer algebraischen Struktur bis auf Isomorphie die Quotientenstrukturen nach Kongruenzrelationen.

Im folgenden werden wir als spezielle algebraische Strukturen Ringe $(R, \{+, \cdot\})$ betrachten. Die Anwendbarkeit des modularen Verfahrens auf Gruppen, Körper und Vektorräume wird am Ende des Abschnitts diskutiert.

Generalvoraussetzung: $(R, \{+, \cdot\})$ bezeichne für den Rest des Kapitels stets einen kommutativen Ring mit Einselement 1.

BEZEICHNUNG 2.8. *Sei $\mathfrak{a}$ ein Ideal von R. Dann bezeichne $E_\mathfrak{a}$ die von dem Ideal $\mathfrak{a}$ mittels*

$$r_1 E_\mathfrak{a} r_2 \Leftrightarrow r_1 - r_2 \in \mathfrak{a}$$

induzierte Relation auf R.

SATZ 2.9. *Sei $\mathfrak{a}$ ein Ideal von R. Dann ist $E_\mathfrak{a}$ eine Kongruenzrelation auf R.*

BEZEICHNUNG 2.10. *Statt $R/E_\mathfrak{a}$ schreiben wir kurz $R/\mathfrak{a}$.*

SATZ 2.11. *Sei $\mathfrak{a}$ ein Ideal von R. Dann gilt*

(1) *Die Quotientenstruktur $R/\mathfrak{a}$ ist ein Ring, genannt Restklassenring. Die Elemente $[r] \in R/\mathfrak{a}$ haben die Form $[r] = r + \mathfrak{a}, r \in R$.*

(2) *$R/\mathfrak{a}$ ist homomorphes Bild von R unter $\Phi : R \to R/\mathfrak{a}, r \mapsto r + \mathfrak{a}$.*

SATZ 2.12. *Sei $\Phi : R \to H$ ein Homomorphismus von (R, Ω) nach (H, Ω) und sei $\tilde{R}$ homomorphes Bild von R unter Φ. Dann ist $\mathrm{Ker}(\Phi)$ ein Ideal in R und $\tilde{R} \cong R/\mathrm{Ker}(\Phi)$.*

Nach den Sätzen 2.11,2.12 sind die homomorphen Bilder von R bis auf Isomorphie genau die Restklassenringe von R.

BEZEICHNUNG 2.13. *Sei $\mathfrak{a}$ ein Ideal von R und $r_1, r_2 \in R$. Dann schreiben wir für die Gleichung $r_1 + \mathfrak{a} = r_2 + \mathfrak{a}$ in $R/\mathfrak{a}$ auch $r_1 \equiv r_2 \pmod{\mathfrak{a}}$.*

Für einen kommutativen Ring mit Einselement ist es möglich, das kommutative Diagramm aus Abb. 1 auf mehrere homomorphe Bilder zu verallgemeinern. Zentrales Hilfsmittel hierfür ist der chinesische Restsatz.

DEFINITION 2.14. *Seien $\mathfrak{a}_1, \mathfrak{a}_2$ Ideale von R. Dann heißen $\mathfrak{a}_1$ und $\mathfrak{a}_2$ komaximal genau dann, wenn $\mathfrak{a}_1 + \mathfrak{a}_2 = R$.*

SATZ 2.15. *(Chinesischer Restsatz) Seien $\mathfrak{a}_1, \ldots, \mathfrak{a}_s$ paarweise komaximale Ideale von R. Dann ist*

$$\begin{array}{rccl} \Psi: & R/\prod_{j=1}^{s} \mathfrak{a}_j & \to & R/\mathfrak{a}_1 \times \cdots \times R/\mathfrak{a}_s \\ & r + \prod_{j=1}^{s} \mathfrak{a}_j & \mapsto & (r + \mathfrak{a}_1, \ldots, r + \mathfrak{a}_s) \end{array}$$

ein Isomorphismus.

BEZEICHNUNG 2.16. *Sei $\mathfrak{a}$ ein Ideal von R. Dann bezeichnen wir den offensichtlich surjektiven Homomorphismus Φ von dem Ring R auf den Restklassenring $R/\mathfrak{a}$*

$$\begin{array}{rccl} \Phi & : & R & \to & R/\mathfrak{a} \\ & & r & \mapsto & r + \mathfrak{a}. \end{array}$$

als den kanonischen Epimorphismus zu $\mathfrak{a}$.

SATZ 2.17. *Ψ bezeichne den Isomorphismus aus Satz 2.15. Seien $\mathfrak{a}_1, \ldots, \mathfrak{a}_s$ paarweise komaximale Ideale von R und $\mathbf{r} := (r_1, \ldots, r_d) \in R^d$. Für $1 \leq i \leq s$ sei Φ_i der kanonische Epimorphismus zu $\mathfrak{a}_i$. Weiterhin sei $\Phi_i(\mathbf{r}) := (\Phi_i(r_1), \ldots, \Phi_i(r_s))$ für $1 \leq i \leq s$ und t ein $\{+, -, \cdot\}$-Term der Stelligkeit d. Dann gilt folgendes kommutatives Diagramm:*

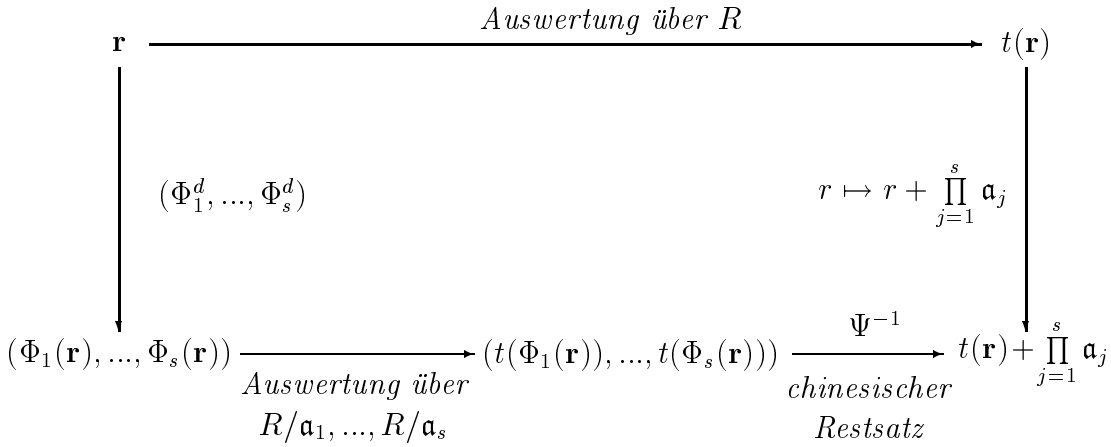

ABBILDUNG 2.2. Kommutatives Diagramm für mehrere homomorphe Bilder

Beweis: Ψ^{-1} ist bijektiv. Daher genügt es, die folgende Aussage zu zeigen:

$$(t(\Phi_1(\mathbf{r})) + \mathfrak{a}_1, \ldots, t(\Phi_s(\mathbf{r})) + \mathfrak{a}_s) = \Psi\left(t(\mathbf{r}) + \prod_{j=1}^{s} \mathfrak{a}_j\right). \tag{2-1}$$

Nach Satz 2.3 (Homomorphismus-Theorem) gilt für $1 \leq j \leq s$:

$$t(\Phi_j(\mathbf{r})) + \mathfrak{a}_j = t(\mathbf{r}) + \mathfrak{a}_j.$$

Somit folgt (2-1) nach Definition von Ψ in Satz 2.15. □

Sei $\mathfrak{a} = \prod_{j=1}^{a} \mathfrak{a}_j$ und Φ der kanonische Epimorphismus zu $\mathfrak{a}$. Um das modulare Verfahren anwenden zu können, fehlt uns nun nur noch die Möglichkeit, aus der berechneten Nebenklasse $t(\Phi(\mathbf{r})) = t(\mathbf{r}) + \mathfrak{a}$ den gesuchten Wert $t(\mathbf{r})$ zu berechnen. Diese Berechnung nennen wir Rücktransformation. Im ersten Teil des nächsten Abschnitts wird gezeigt, wie die Rücktransformation für den Ring der ganzen Zahlen $\mathbb{Z}$ durchzuführen ist.

Für Gruppen und Vektorräume gibt es kein Analogon zum chinesischen Restsatz. Das modulare Verfahren für mehrere homomorphe Bilder ist somit weder für

Gruppen noch für Vektorräume durchführbar.
Körper F sind einfache Ringe, besitzen somit nur die trivialen Ideale $\{0\}$ bzw. F. Nach Satz 2.12 sind $F/\{0\} \cong F$ bzw. $F/F \cong \{0\}$ bis auf Isomorphie die einzigen homomorphen Bilder von F. Somit ist das modulare Verfahren für Körper nicht geeignet.

2. Das modulare Verfahren für spezielle Ringe

Zunächst wollen wir das modulare Verfahren für ein homomorphes Bild in Form eines kommutativen Diagrammes darstellen.

Sei $\mathfrak{a}$ ein Ideal von R, Φ der kanonische Epimorphismus zu $\mathfrak{a}$ und t ein $\{+,-,\cdot\}$-Term der Stelligkeit d. Weiterhin sei $\mathbf{r} := (r_1, \ldots, r_d) \in R^d$ und $\Phi(\mathbf{r}) := (\Phi(r_1), \ldots, \Phi(r_s))$. Dann gilt folgendes kommutatives Diagramm:

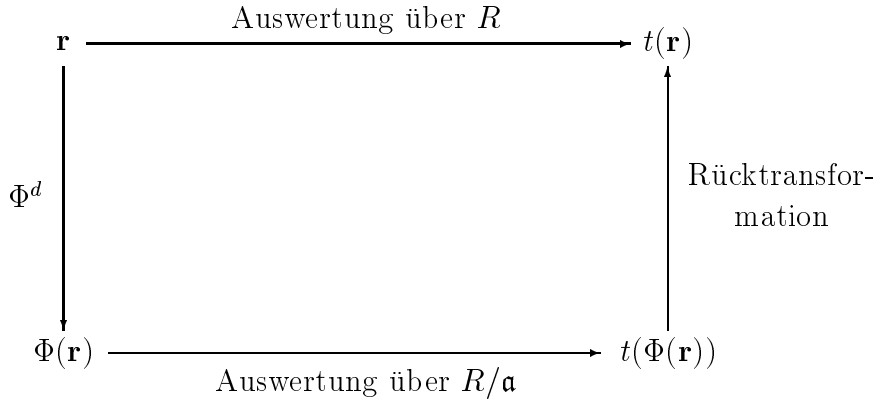

Abbildung 2.3. Modulares Verfahren für *ein* homomorphes Bild

Φ ist zwar surjektiv, jedoch für $\mathfrak{a} \neq \{0\}$ nicht injektiv. Deshalb wird für die Rücktransformation zusätzliche Information benötigt. Wir zeigen nun, daß diese Information für den Ring $\mathbb{Z}$[1] eine geeignete Schranke für den Absolutbetrag von

[1]Die Betrachtungen in Unterabschnitt 2.1 und die Verallgemeinerung in Unterabschnitt 2.2 sind analog zum euklidischen Ring $\mathbb{Z}$ auch für den euklidischen Ring $F[x]$ durchführbar [Lip81, Kapitel 8]. Dabei sei F ein Körper.

$t(\Phi(\mathbf{r}))$ kann.

2.1. Der Ring $\mathbb{Z}$. $\mathbb{Z}$ ist ein euklidischer Ring. Mit Hilfe der Division mit Rest kann man in jeder Nebenklasse $l + (m)$ $(l, m \in \mathbb{Z})$ eines Ideals $(m) \neq \{0\}$ ein Element $r \in \mathbb{Z}$ auszeichnen.

SATZ 2.18. *Sei $l \in \mathbb{Z}$ und $m \in \mathbb{N} \setminus \{0\}$. Dann gilt*

(1) *Es gibt eindeutig bestimmte $r, s \in \mathbb{Z}$ mit $l = sm + r$ und $0 \leq r < m$.*
(2) *Für m ungerade gibt es eindeutig bestimmte $r, s \in \mathbb{Z}$ mit $l = sm + r$ und $|r| < \frac{m}{2}$.*

KOROLLAR 2.19. *Seien $l \in \mathbb{Z}$ und $m \in \mathbb{N} \setminus \{0\}$. Dann gilt*

(1) *In jeder Nebenklasse $l + (m)$ gibt es genau ein Element r mit $0 \leq r < m$.*
(2) *Für m ungerade gibt es in jeder Nebenklasse $l + (m)$ genau ein Element r mit $|r| < \frac{m}{2}$.*

Beweis: Direkte Folgerung aus Satz 2.18. □

BEZEICHNUNG 2.20. *Das nach Korollar 2.19 Teil (1) eindeutig bestimmte Element bezeichnen wir auch als $r = l \bmod m$ und nennen es den positiven kanonischen Repräsentanten von $l + (m)$.*
Das nach Korollar 2.19 Teil (2) eindeutig bestimmte Element bezeichnen wir auch als $r = l \operatorname{mods} m$ und nennen es den symmetrischen kanonischen Repräsentanten von $l + (m)$.

Sei $\mathbf{r} := (r_1, \ldots, r_d) \in R^d$ und $\{+, -, \cdot\}$-Term der Stelligkeit d. Wenn wir eine Schranke $S \in \mathbb{Z}$ mit $0 \leq t(\mathbf{r}) < S$ bzw. $|t(\mathbf{r})| < S$ kennen, so können wir ein Ideal (m) von $\mathbb{Z}$ derart wählen, daß $m \geq S$ bzw. $m \geq 2S$ für m ungerade. Haben wir dann im Restklassenring $\mathbb{Z}/(m)$ den Wert $t(\mathbf{r}) + (m)$ berechnet, so besteht die Rücktransformation nur in der Bestimmung des positiven bzw. symmetrischen

Wir beschränken uns auf den Ring $\mathbb{Z}$ und die freien kommutativen Algebren über $\mathbb{Z}$, da wir nur diese für die Anwendungen in Kapitel 3 benötigen.

kanonischen Repräsentanten von $t(\mathbf{r}) + (m)$. Dies wird sofort klar, wenn man beachtet, daß $t(\mathbf{r}) \in t(\mathbf{r}) + (m)$ die Bedingung $0 \leq t(\mathbf{r}) < S < m$ bzw. $|t(\mathbf{r})| < S < \frac{m}{2}$ erfüllt und somit nach Satz 2.19 Teil (1) bzw. Teil (2) der positive bzw. symmetrische kanonische Repräsentant von $t(\mathbf{r}) + (m)$ ist.

Beim modularen Verfahren für mehrere homomorphe Bilder $R/(m_1), \ldots, R/(m_s)$ kann man die Rücktransformation bei gegebener Schranke S mit $0 \leq t(\mathbf{r}) < S$ bzw. $|t(\mathbf{r})| < S$ natürlich auf die Restklasse $t(\mathbf{r}) + \prod_{j=1}^{s}(m_j) = t(\mathbf{r}) + (\prod_{j=1}^{s} m_j)$ anwenden, falls $\prod_{j=1}^{s} m_j > S$ bzw. $\prod_{j=1}^{s} m_j > 2S$ erfüllt ist.

Seien $\mathfrak{a}_1, \ldots, \mathfrak{a}_s$ paarweise komaximale Ideale von $\mathbb{Z}$ und $\mathbf{r} := (r_1, \ldots, r_d) \in \mathbb{Z}^d$. Für $1 \leq i \leq s$ sei Φ_i der kanonische Epimorphismus zu $\mathfrak{a}_i$. Weiterhin sei $\Phi_i(\mathbf{r}) := (\Phi_i(r_1), \ldots, \Phi_i(r_s))$ für $1 \leq i \leq s$ und t ein $\{+, -, \cdot\}$-Term der Stelligkeit d. Dann gilt folgendes kommutatives Diagramm:

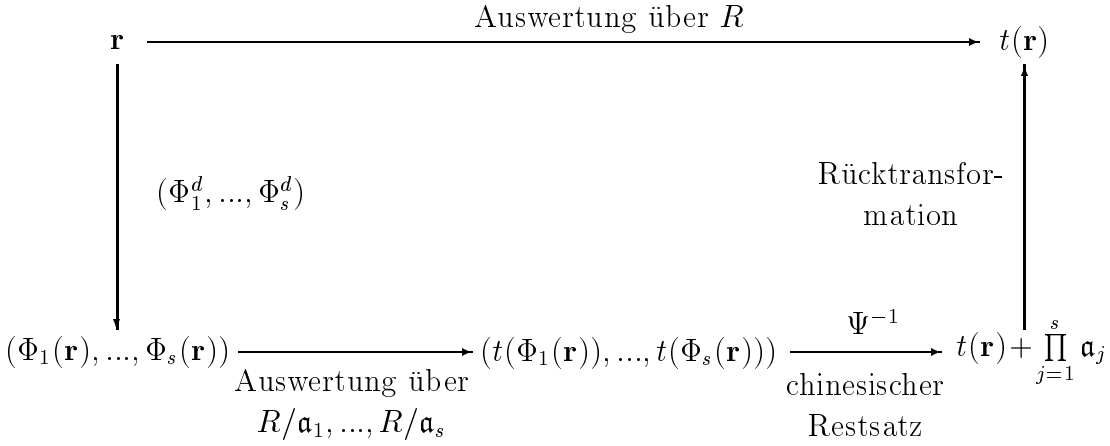

ABBILDUNG 2.4. Modulares Verfahren für *mehrere* homomorphe Bilder

Die Erzeuger m_j der Ideale $\mathfrak{a}_j := (m_j)$, $1 \leq j \leq s$ werden wir auch als Moduli bezeichnen.

Der Algorithmus des modularen Verfahrens für $\mathbb{Z}$ wird im folgenden beschrieben [Lip81].

ALGORITHMUS 2.21. (modulares Verfahren für $\mathbb{Z}$)

Input: $\{+,-,\cdot\}$-*Term* $t(x_1,\ldots,x_d) := t(\mathbf{x})$ *und und Argumente* $(r_1,\ldots,r_d) := \mathbf{r} \in \mathbb{Z}^n$.

Output: $v = t(\mathbf{r}) \in \mathbb{Z}$.

Schritt 1: *(Bestimmung der Schranke und Wahl der Moduli)*
VERSION 1: *Bestimme eine Schranke* S *mit* $0 \leq t(\mathbf{r}) < S$ *und paarweise teilerfremde positive* $m_1,\ldots,m_s$ *mit* $\prod_{j=1}^{s} m_j \geq S$.
VERSION 2: *Bestimme eine Schranke* S *mit* $|t(\mathbf{r})| < S$ *und paarweise teilerfremde ungerade positive* $m_1,\ldots,m_s$ *mit* $\prod_{j=1}^{s} m_j \geq 2S$.

Schritt 2: *(Abbilden in die Restklassenringe)*
Für $j = 1,\ldots,s$*:*
Berechne $w_i^{(j)} = r_i \bmod m_j$, $1 \leq i \leq d$.

Schritt 3: *(Auswerten in den Restklassenringen)*
Für $j = 1,\ldots,s$*:*
Berechne $\beta_j = t(w_1^{(j)},\ldots,(w_d^{(j)})$ *(über* $\mathbb{Z}/(m_j)$*)*.

Schritt 4: *(Chinesischer Restsatz)*
Berechne mit Hilfe des chinesischen Restsatzes ein $u \in \mathbb{Z}$ *mit* $u \equiv \beta_j \bmod m_j$ *für* $1 \leq j \leq s$.

Schritt 5: *(Rücktransformation)*
VERSION 1: *Bestimme den positiven kanonischen Repräsentanten* v *von* $u + (\prod_{j=1}^{s} m_j)$ *und gib* v *zurück. ENDE.*
VERSION 2: *Bestimme den symmetrischen kanonischen Repräsentanten* v *von* $u + (\prod_{j=1}^{s} m_j)$ *und gib* v *zurück. ENDE.*

SATZ 2.22. *(Korrektheit von Algorithmus* 2.21*)*
Der Algorithmus 2.21 *berechnet* $v = t(\mathbf{r})$.

Beweis: Wir verwenden die Bezeichnungen aus dem Algorithmus 2.21. Nach Satz 2.17 gilt

$$u + \left(\prod_{j=1}^{s} m_j\right) = t(\mathbf{r}) + \left(\prod_{j=1}^{s} m_j\right).$$

Hieraus folgt

$$t(\mathbf{r}) \in u + \left(\prod_{j=1}^{s} m_j\right). \tag{2-2}$$

Version 1: In Schritt 5 wird der positive kanonische Repräsentant v von $u + (\prod_{j=1}^{s} m_j)$ bestimmt. Nach Bezeichnung 2.20 gilt somit

$$v \in u + (\prod_{j=1}^{s} m_j) \text{ und } 0 \leq v < \prod_{j=1}^{s} m_j. \tag{2-3}$$

Weiterhin gilt nach Voraussetzung

$$0 \leq t(\mathbf{r}) < S \leq \prod_{j=1}^{s} m_j. \tag{2-4}$$

Mit (2-2), (2-3) und (2-4) folgt aus Korollar 2.19 Teil (1) $v = t(\mathbf{r})$.

Version 2: In Schritt 5 wird der symmetrische kanonische Repräsentant v von $u + (\prod_{j=1}^{s} m_j)$ bestimmt. Nach Bezeichnung 2.20 gilt somit

$$v \in u + (\prod_{j=1}^{s} m_j) \text{ und } |v| < \frac{1}{2} \prod_{j=1}^{s} m_j. \tag{2-5}$$

Weiterhin gilt nach Voraussetzung

$$|t(\mathbf{r})| < S \leq \frac{1}{2} \prod_{j=1}^{s} m_j. \tag{2-6}$$

Mit (2-2), (2-5) und (2-6) folgt aus Korollar 2.19 Teil (2) $v = t(\mathbf{r})$. □

Die homomorphen Bilder $\mathbb{Z}/(m_j) = \mathbb{Z}_{m_j}$ sind endliche Ringe mit m_j Elementen. Die Ideale (m_j) sind genau dann komaximal, wenn die Moduli m_j teilerfremd sind. Sind die Moduli Primzahlen, so sind die Restklassenringe endliche Körper. Rechnungen in Restklassenringen werden durchgeführt, indem man mit Repräsentanten der Nebenklassen rechnet. Wählt man als Repräsentanten einer Nebenklasse die kanonischen Repräsentanten, so sind während der Auswertung eines Termes die Repräsentanten stets beschränkt und man hat nur Operationen

in $\mathbb{Z}$ mit beschränkten ganzen Zahlen auszuführen. Dies kann die erwünschte Zeitersparnis bringen.

2.2. Verallgemeinerung auf freie kommutative Algebren über $\mathbb{Z}$. In diesem Abschnitt verallgemeinern wir das modulare Verfahren für den Ring $\mathbb{Z}$ auf auf freie kommutative Algebren über $\mathbb{Z}$. In Abschnitt 3.1 dieses Kapitels werden wir zeigen, daß die für uns wichtigen Ordnungen von algebraischen Zahlkörpern freie kommutative Algebren über $\mathbb{Z}$ sind.

DEFINITION 2.23. *Unter einer Algebra über R verstehen wir einen Ring $(A, \{+, \cdot\})$ mit folgenden Eigenschaften:*

(1) *$(A, \{+\})$ ist unitärer R-Modul.*
(2) *Für $r \in R$ und $a_1, a_2 \in A$ gilt $r(a_1 \cdot a_2) = a_1(ra_2) = (ra_1)a_2$*

Dabei heißt eine Algebra A kommutativ, falls $(A, \{+, \cdot\})$ als Ring kommutativ ist. Eine Algebra A über R heißt frei, falls $(A, \{+\})$ als R-Modul frei ist.

SATZ 2.24. *Sei A eine freie kommutative Algebra über R. A habe als R-Modul die Basis $\{b_1, \ldots, b_n\}$. Dann gelten:*

(1) *Seien $\mathfrak{a}_1, \ldots, \mathfrak{a}_s$ verschiedene nichttriviale, paarweise komaximale Ideale in R, so sind $(\mathfrak{a}_1 \cdot A), \ldots, (\mathfrak{a}_s \cdot A)$ verschiedene nichttriviale paarweise komaximale Ideale in A.*
(2) *Sei $R = \mathbb{Z}, \mathfrak{a} = (m)$ Ideal in $\mathbb{Z}, m \in \mathbb{Z}$. Dann gibt es in jeder Nebenklasse $a + (\mathfrak{a} \cdot A) \in A/(\mathfrak{a} \cdot A)$ genau ein Element $k = \sum_{i=1}^{n} z_i b_i$ mit $0 \leq z_i < m$ für $i \in \{1, \ldots, n\}$.*
(3) *Sei $R = \mathbb{Z}, \mathfrak{a} = (m)$ Ideal in $\mathbb{Z}$ und $m \in \mathbb{Z}$ ungerade. Dann gibt es in jeder Nebenklasse $a + (\mathfrak{a} \cdot A) \in A/(\mathfrak{a} \cdot A)$ genau ein Element $k = \sum_{i=1}^{n} z_i b_i$ mit $|z_i| < \frac{m}{2}$ für $i \in \{1, \ldots, n\}$.*

Beweis: (1) Aus der Darstellung $(\mathfrak{a}_i \cdot A) = \{\sum_{j=1}^{n} r_j b_j | r_j \in \mathfrak{a}_i\}$ folgt sofort, daß die Ideale nichttrivial und paarweise verschieden sind.
Seien l, k mit $1 \leq k, l \leq n$ und $k \neq l$ fest. Wir setzen $\mathfrak{a}_k + \mathfrak{a}_l = R$ voraus und

zeigen $(\mathfrak{a}_k \cdot A) + (\mathfrak{a}_l \cdot A) = A$.
Sei $a \in A$ beliebig. a habe die Darstellung

$$a = \sum_{i=1}^{n} r_i b_i \tag{2-7}$$

mit $r_i \in R, 1 \leq i \leq n$. Nach der Voraussetzung $\mathfrak{a}_k + \mathfrak{a}_l = R$ folgt, daß es $a_{i,k} \in \mathfrak{a}_k$ und $a_{i,l} \in \mathfrak{a}_l$ gibt mit

$$r_i = a_{i,k} + a_{i,l} \text{ für } 1 \leq i \leq n. \tag{2-8}$$

Aus (2-7) und (2-8) folgt:

$$a = \sum_{i=1}^{n}(a_{i,k} + a_{i,l})b_i = \left(\sum_{i=1}^{n} a_{i,k} b_i\right) + \left(\sum_{i=1}^{n} a_{i,l} b_i\right) \in (\mathfrak{a}_k \cdot A) + (\mathfrak{a}_l \cdot A)$$

Es folgt $(\mathfrak{a}_k \cdot A) + (\mathfrak{a}_l \cdot A) = A$. Somit sind die Ideale $(\mathfrak{a}_i \cdot A)$, $1 \leq i \leq n$ paarweise komaximal.
Aussagen (2) und (3) folgen aus $(\mathfrak{a} \cdot A) = \{\sum_{i=1}^{n} r_i b_i | r_i \in \mathfrak{a}\}$ und Satz 2.19. □

Bezeichnung 2.25. *Das nach Satz 2.24 Teil (2) eindeutige Element bezeichnen wir auch als $k = a \bmod m$ und nennen es den positiven kanonischen Repräsentanten von $a + (m \cdot A)$.*
Das nach Satz 2.24 Teil (3) eindeutige Element bezeichnen wir auch als $k = a \operatorname{mods} m$ und nennen es den symmetrischen kanonischen Repräsentanten von $a + (m \cdot A)$.

Bezeichnung 2.26. *Sei A eine freie kommutative Algebra über $\mathbb{Z}$ mit $\mathbb{Z}$-Basis $b_1, \ldots, b_n$ und sei $\alpha = \sum_{i=1}^{n} a_i b_i \in A$. Dann bezeichnen wir das Maximum der Beträge der Koeffizienten von α bezüglich der Basis von A als*

$$\mathrm{MAX}(\alpha) := \max_{1 \leq i \leq n}(|a_i|)$$

Nach Satz 2.24 können wir das modulare Verfahren von $\mathbb{Z}$ sofort auf freie kommutative Algebren über $\mathbb{Z}$ erweitern. Wir müssen dazu nur eine *gemeinsame* Schranke S für die Koeffizienten von $t(\mathbf{r})$ bzw. bezüglich der Basis $b_1, \ldots, b_n$ bestimmen. Damit erhalten wir für eine freie kommutative Algebra über $\mathbb{Z}$ mit

der Basis $b_1, \ldots, b_n$ den folgenden Algorithmus, der den Kern der in Kapitel 3 beschriebenen Programme darstellt.

ALGORITHMUS 2.27. (modulares Verfahren für eine freie kommutative Algebra A mit Basis $\{b_1, \ldots, b_n\}$ über $\mathbb{Z}$)

Input: $\{+, -, \cdot\}$-*Term* $t(x_1, \ldots, x_d) := t(\mathbf{x})$ *und Argumente* $(r_1, \ldots, r_d) := \mathbf{r} \in A^d$.

Output: $v = t(\mathbf{r}) \in A$.

Schritt 1: *(Bestimmung der Schranke und Wahl der Moduli)*
Sei $t(\mathbf{r}) = \sum_{k=1}^{n} z_k b_k, z_k \in \mathbb{Z}$ *für* $k \in \{1, \ldots, n\}$.
VERSION 1: *Bestimme eine Schranke* S *mit* $0 \leq z_k < S$ *für* $k \in \{1, \ldots, n\}$ *und paarweise teilerfremde positive* $m_1, \ldots, m_s$ *mit* $\prod_{j=1}^{s} m_j \geq S$.
VERSION 2: *Bestimme eine Schranke* S *mit* $|z_k| < S$ *für* $k \in \{1, \ldots, n\}$ *und paarweise teilerfremde positive ungerade* $m_1, \ldots, m_s$ *mit* $\prod_{j=1}^{s} m_j \geq 2S$.

Schritt 2: *(Abbilden in die Restklassenringe)*
Sei $r_i = \sum_{k=1}^{n} r_{i,k} b_k$ *für* $i \in \{1, \ldots, d\}$.
Für $j = 1, \ldots, s$:
Berechne $w_i^{(j)} = r_i \bmod m_j$, $1 \leq i \leq d$ *(Bezeichnung 2.25).*

Schritt 3: *(Auswerten in den Restklassenringen)*
Für $j = 1, \ldots, s$:
Berechne $\beta_j = t(w_1^{(j)}, \ldots, (w_d^{(j)})$ *(über* $A/(m_j \cdot A)$*).*

Schritt 4: *(Chinesischer Restsatz)*
Berechne $u = \Psi^{-1}(\beta_1, \ldots, \beta_s)$.

Schritt 5: *(Rücktransformation)*
VERSION 1: *Bestimme den positiven kanonischen Repräsentanten* $v = \sum_{k=1}^{n} v_k b_k$ *von* $u + (A(\prod_{j=1}^{s} m_j))$ *und gib* v *zurück. ENDE.*
VERSION 2: *Bestimme den symmetrischen kanonischen Repräsentanten* $v = \sum_{k=1}^{n} v_k b_k$ *von* $u + (A(\prod_{j=1}^{s} m_j))$ *und gib* v *zurück. ENDE.*

SATZ 2.28. *(Korrektheit von Algorithmus 2.27) Der Algorithmus 2.21 berechnet* $v = t(\mathbf{r})$.

Beweis: Wir verwenden die Bezeichnungen aus dem Algorithmus 2.27. Im folgenden verwenden wir die Identität $\prod_{j=1}^{s}(Am_j) = \left(A(\prod_{j=1}^{s} m_j)\right)$. Nach Satz 2.17 gilt

$$u + \left(A\left(\prod_{j=1}^{s} m_j\right)\right) = t(\mathbf{r}) + \left(A\left(\prod_{j=1}^{s} m_j\right)\right).$$

Hieraus folgt

$$t(\mathbf{r}) \in u + \left(A\left(\prod_{j=1}^{s} m_j\right)\right). \tag{2-9}$$

Version 1: In Schritt 5 wird der positive kanonische Repräsentant v von $u + (A(\prod_{j=1}^{s} m_j))$ bestimmt. Nach Bezeichnung 2.25 gilt somit

$$v \in u + \left(A\left(\prod_{j=1}^{s} m_j\right)\right) \text{ und } 0 \leq v_k < \prod_{j=1}^{s} m_j \text{ für } 1 \leq k \leq n. \tag{2-10}$$

Weiterhin gilt nach Voraussetzung für die Koeffizienten z_k von $\mathrm{t}(\mathbf{r})$

$$0 \leq z_k < S \leq \prod_{j=1}^{s} m_j \text{ für } 1 \leq k \leq n. \tag{2-11}$$

Mit (2-9), (2-10) und (2-11) folgt aus Korollar 2.19 Teil (1) $v = t(\mathbf{r})$.

Version 2: In Schritt 5 wird der symmetrische kanonische Repräsentant v von $u + (A(\prod_{j=1}^{s} m_j))$ bestimmt. Nach Bezeichnung 2.25 gilt somit

$$v \in u + \left(A\left(\prod_{j=1}^{s} m_j\right)\right) \text{ und } |v_k| < \frac{1}{2}\prod_{j=1}^{s} m_j \text{ für } 1 \leq k \leq n. \tag{2-12}$$

Weiterhin gilt nach Voraussetzung für die Koeffizienten z_k von $\mathrm{t}(\mathbf{r})$

$$|z_k| < S \leq \frac{1}{2}\prod_{j=1}^{s} m_j \text{ für } 1 \leq k \leq n. \tag{2-13}$$

Mit (2-9), (2-12) und (2-13) folgt aus Korollar 2.19 Teil (2) $v = t(\mathbf{r})$. □

Insgesamt halten wir als Ergebnis dieses Kapitels:

SATZ 2.29. *Die freien kommutativen Algebren über $\mathbb{Z}$ sind prinzipiell für das modulare Verfahren geeignet. Der Algorithmus 2.27 ist anwendbar.*

3. Algebraische Zahlkörper und Ordnungen

BEZEICHNUNG 2.30. *Sei F_2/F_1 eine Körpererweiterung. Dann können wir F_2 als F_1-Vektorraum auffassen. Eine Vektorraumbasis des F_1-Vektorraums F_2 bezeichnen wir auch einfach als F_1-Basis des Körpers F_2. Die Anzahl n der Elemente einer solchen Basis nennen wir den Grad $n = [F_2 : F_1]$ der Körpererweiterung F_2/F_1.*

Zunächst führen wir den Begriff des algebraischen Zahlkörpers ein.

DEFINITION 2.31. *Ein algebraischer Zahlkörper K ist eine Körpererweiterung von $\mathbb{Q}$, dessen Grad $n = [K : \mathbb{Q}]$ endlich ist.*

SATZ 2.32. *Sei K ein algebraischer Zahlkörper vom Grad n. Dann gilt:*

(1) *Es gibt ein $\varrho \in \mathbb{C}$ und ein normiertes (über $\mathbb{Q}$) irreduzibles Polynom $f(x) \in \mathbb{Z}[x]$ mit $f(\varrho) = 0$, so daß $K = \mathbb{Q}(\varrho) = \mathbb{Q}[\varrho] \cong \mathbb{Q}[x]/(f(x))$. ϱ heißt primitives Element, f(x) heißt erzeugendes Polynom und es gilt $\deg(f(x)) = n$.*
(2) *Für das Element ϱ aus (1) gilt: $\{1, \varrho, \varrho^2, \ldots, \varrho^{n-1}\}$ ist eine $\mathbb{Q}$-Basis von K.*

Die Elemente von K nennen wir algebraische Zahlen.

Zunächst werden für das weitere Verständnis wichtige Ergebnisse aus [Mar95], [Poh93] und [PZ89] zusammengefaßt.

3.1. Absolute Erweiterungen. Zunächst betrachten wir den Fall, daß der algebraische Zahlkörper K als Körpererweiterung $K/\mathbb{Q}$ gegeben ist, d.h. durch ein erzeugendes Polynom $f(x) \in \mathbb{Z}[x]$. In diesem Fall bezeichnen wir die Körpererweiterung $K/\mathbb{Q}$ als absolute Erweiterung.

Generalvoraussetzung: K bezeichne stets einen algebraischen Zahlkörper $K = \mathbb{Q}[\varrho]$ einer absoluten Erweiterung $K/\mathbb{Q}$, $f(x) \in \mathbb{Z}$ das erzeugende Polynom von K, ϱ ein primitives Element und $n := [K : \mathbb{Q}]$ den Grad der Körpererweiterung.

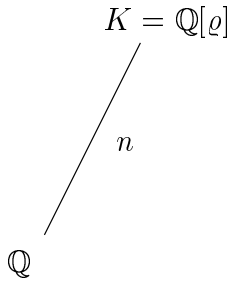

ABBILDUNG 2.5. Absolute Erweiterung

Für das Rechnen in einem algebraischen Zahlkörper benötigen wir die Verallgemeinerung der ganzen Zahlen aus $\mathbb{Q}$.

DEFINITION 2.33. *$\alpha \in \mathbb{C}$ heißt ganze algebraische Zahl genau dann, wenn es ein normiertes Polynom $f(x) \in \mathbb{Z}[x]$ gibt mit $f(\alpha) = 0$. Die Menge der ganzen algebraischen Zahlen bezeichnen wir mit $\mathbb{A}$.*

SATZ 2.34. *Die Menge $\mathfrak{o}_K := \mathbb{A} \cap K$, der ganzen algebraischen Zahlen eines algebraischen Zahlkörpers K, ist ein Ring. Weiterhin ist $\mathfrak{o}_K$ ein freier $\mathbb{Z}$-Modul von Rang $n = [K : \mathbb{Q}]$.*

BEZEICHNUNG 2.35. *Wir nennen $\mathfrak{o}_K := \mathbb{A} \cap K$ den Ring der ganzen Zahlen von K oder Maximalordnung von K. Eine $\mathbb{Z}$-Basis von $\mathfrak{o}_K$ nennen wir Ganzheitsbasis von K.*

DEFINITION 2.36. *Sei K ein algebraischer Zahlkörper vom Grad n und $\mathfrak{o}_K$ die Maximalordnung von K. Einen unitären Teilring $\mathfrak{o}$ von $\mathfrak{o}_K$, der gleichzeitig ein freier $\mathbb{Z}$-Modul vom Rang n ist, heißt eine Ordnung des Zahlkörpers K.*

Mit Hilfe des folgenden Satzes können wir zeigen, daß das modulare Verfahren prinzipiell auf Ordnungen anwendbar ist.

SATZ 2.37. *Sei K ein algebraischer Zahlkörper und $\mathfrak{o}$ eine Ordnung von K. Dann ist $\mathfrak{o}$ eine freie kommutative Algebra über $\mathbb{Z}$.*

Beweis: $\mathbb{Z}$ ist eine Teilmenge von $\mathfrak{o}$. Somit gilt (2) von Definition 2.23. Der Rest folgt aus der Definition 2.36. □

Nach Satz 2.29 sind Ordnungen eines algebraischen Zahlkörpers prinzipiell für das modulare Verfahren geeignete Strukturen. Algorithmus 2.27 aus dem vorherigen Abschnitt ist anwendbar.
Da das modulare Verfahren auf Körper direkt nicht anwendbar ist, werden wir Probleme über Zahlkörpern auf Probleme über Ordnungen zurückführen.
Ob die Verwendung des modularen Verfahrens bei konkreten Problemen die gewünschte Beschleunigung gegenüber dem nichtmodularen Verfahren bringt, wird exemplarisch in Kapitel 3 untersucht.

Voraussetzung: Wir betrachten in K eine feste Ordnung $\mathfrak{o}$ mit $\mathbb{Z}$-Basis $\theta_1, \ldots, \theta_n$.

Jede Ordnung $\mathfrak{o}$ von K besitzt per Definition eine $\mathbb{Z}$-Basis $\theta_1, \ldots, \theta_n$. Dabei ist $\theta_1, \ldots, \theta_n$ auch eine $\mathbb{Q}$-Basis von K. Jedes Element $k \in K$ läßt sich eindeutig schreiben als

$$k = \frac{\sum_{i=1}^{n} z_i \theta_i}{d} \tag{2-14}$$

mit $d \in \mathbb{N}, z_i \in \mathbb{Z} (1 \leq i \leq n)$ und $ggT(z_1, \ldots, z_n, d) = 1$. Der Zähler $\sum_{i=1}^{n} z_i \theta_i$ ist offensichtlich aus $\mathfrak{o}$. Es ist 2-14 die von uns verwendete Standarddarstellung von algebraischen Zahlen.

Die Multiplikationstabelle von $K/\mathbb{Q}$ bezüglich der Basis $\theta_1, \ldots, \theta_n$ wird festgelegt mittels

$$\theta_i \theta_j = \sum_{k=1}^{n} \Gamma_{i,j,k}^{K} \theta_k .$$

Da wir jedoch in K eine feste Ordnung mit einer festen $\mathbb{Z}$-Basis betrachten, sprechen wir auch kurz von der Multiplikationstabelle von $K/\mathbb{Q}$ und bezeichnen sie mit $(\Gamma_{i,j,k}^{K})_{i,j,k \in \{1,\ldots,n\}}$.

BEMERKUNG 2.38. *Es gilt*

$$(\Gamma_{i,j,k}^K)_{i,j,k\in\{1,\dots,n\}} \in \mathbb{Z}^{n\times n\times n},$$

da $\theta_i\theta_j \in \mathfrak{o} = \mathbb{Z}\theta_1 \bigoplus \cdots \bigoplus \mathbb{Z}\theta_n$.

DEFINITION 2.39. *Ein Dedekindring R ist ein Integritätsring mit folgenden Eigenschaften:*

(1) *R ist noethersch.*
(2) *R ist ganz abgeschlossen (in seinem Quotientenkörper).*
(3) *Jedes vom Nullideal verschiedene Primideal von R ist maximal.*

SATZ 2.40. *Sei R ein Dedekindring. Dann ist jedes Ideal $\mathfrak{a}$ ($\mathfrak{a} \notin \{\{0\}, R\}$) ein bis auf die Reihenfolge eindeutiges Produkt von Primidealen:*

$$\mathfrak{a} = \mathfrak{p}_1 \dots \mathfrak{p}_r, \tag{2-15}$$

$\mathfrak{p}_i$ Primideal von R, $(1 \le i \le r)$.

BEZEICHNUNG 2.41. *Es gelten die Bezeichnungen aus Satz 2.40. Wir sagen, daß das Primideal $\mathfrak{p}$ ein Teiler von $\mathfrak{a}$ ist genau dann, wenn es in der Primidealzerlegung (2-15) von $\mathfrak{a}$ vorkommt und schreiben dafür $\mathfrak{p}|\mathfrak{a}$.*

SATZ 2.42. *Die Maximalordung $\mathfrak{o}_K$ eines algebraischen Zahlkörpers K ist ein Dedekindring.*

Für beliebige Ordnungen ist eine Zerlegung eines Ideals in ein Produkt von Primidealen im allgemeinen nicht möglich.

Nun werden wir einige wichtige Eigenschaften von Primidealen in Ringen ganzer Zahlen betrachten.
Sei im folgenden $p \in \mathbb{Z}$ eine Primzahl.

SATZ 2.43. *Sei $\mathfrak{p}$ ein Primideal von $\mathfrak{o}_K$. Dann sind die folgenden Bedingungen äquivalent:*

(1) $\mathfrak{p}|p\mathfrak{o}_K$
(2) $\mathfrak{p} \supseteq p\mathfrak{o}_K$

(3) $\mathfrak{p} \supseteq p\mathbb{Z}$
(4) $\mathfrak{p} \cap \mathfrak{o}_K = p\mathbb{Z}$
(5) $\mathfrak{p} \cap K = p\mathbb{Z}$

Falls eine der Bedingungen $(1), \ldots, (5)$ erfüllt ist, so sagen wir auch $\mathfrak{p}$ liegt über $p\mathbb{Z}$ oder $p\mathbb{Z}$ liegt unter $\mathfrak{p}$. Diejenigen Primideale von $\mathfrak{o}_K$, die über $p\mathbb{Z}$ liegen sind genau die Primideale, welche in der Primidealzerlegung von $p\mathfrak{o}_K$ in $\mathfrak{o}_K$ auftreten.

Für die Betrachtung von Primidealzerlegungen benötigen wir noch zwei wichtige Begriffe.

DEFINITION 2.44. *Für ein Primideal $p\mathbb{Z}$ in $\mathbb{Z}$ betrachten wir die Primidealzerlegung von $p\mathfrak{o}_K$ in $\mathfrak{o}_K$ in paarweise verschiedene Primideale*

$$p\mathfrak{o}_K = \mathfrak{p}_1^{e_1} \ldots \mathfrak{p}_g^{e_g}$$

Wir definieren $e(\mathfrak{p}_i/p\mathbb{Z}) := e_i$, $(1 \leq i \leq g)$ als den Verzweigungsindex von $\mathfrak{p}_i$ über $p\mathbb{Z}$. Ist $e(\mathfrak{p}_i/p\mathbb{Z}) > 1$ für mindestens ein i, so nennen wir $p\mathbb{Z}$ verzweigt in $\mathfrak{o}_K$, sonst unverzweigt in $\mathfrak{o}_K$.

$p\mathbb{Z}$ ist ein maximales Ideal in $\mathbb{Z}$. Nach Satz 2.42 und Definition 2.39 ist ein Primideal $\mathfrak{p}$ von $\mathfrak{o}_K$ ein maximales Ideal in $\mathfrak{o}_K$. Somit sind $\mathbb{Z}/p\mathbb{Z}$ und $\mathfrak{o}_K/\mathfrak{p}_i$ Körper. Nach [Lip81, S. 64] sind $\mathbb{Z}/p\mathbb{Z}$ und $\mathfrak{o}_K/\mathfrak{p}_i$ sogar endliche Körper. Mit Hilfe des injektiven Homomorphismus

$$\begin{array}{rccc} \Phi & : & \mathbb{Z}/p\mathbb{Z} & \to & \mathfrak{o}_K/\mathfrak{p}_i \\ & & z + p\mathbb{Z} & \mapsto & z + \mathfrak{p}_i \end{array}$$

identifiziert man die Körper $\mathbb{Z}/p\mathbb{Z}$ und $\Phi(\mathbb{Z}/p\mathbb{Z})$ und erhält somit $\mathbb{Z}/p\mathbb{Z}$ als Teilkörper von $\mathfrak{o}_K/\mathfrak{p}_i$ [Lip81, S. 64].

DEFINITION 2.45. *Es gelten die Bezeichnungen aus 2.44. Als den Trägheitsgrad $f(\mathfrak{p}_i/p\mathbb{Z})$ von $\mathfrak{p}_i$ über $p\mathbb{Z}$ bezeichnen wir den Körpergrad $[(\mathfrak{o}_K/\mathfrak{p}_i) : (\mathbb{Z}/p\mathbb{Z})]$.*

KOROLLAR 2.46. *Sei $\mathfrak{p}$ ein Primideal in $\mathfrak{o}_K$, welches über $p\mathbb{Z}$ liegt und $f := f(\mathfrak{p}/p\mathbb{Z})$. Dann ist $\mathfrak{o}_K/\mathfrak{p} \cong \mathbb{F}_q$, wobei $\mathbb{F}_q$ ein Körper mit $q = p^f$ Elementen ist.*

Für die unten angegebenen Sätze stellen wir grundlegende Ergebnisse aus [Mar95, Anhang 2] zusammen.

Nach Satz 2.32 gilt $\deg(f(x)) = [K : \mathbb{Q}]$. Im Zerfällungskörper hat $f(x)$ n paarweise verschiedene Nullstellen. Diese nennt man die Konjugierten von ϱ und bezeichnet sie mit $\varrho^{(1)}, \cdots \varrho^{(n)}$, wobei o.B.d.A $\varrho^{(1)} = \varrho$. Diese Konjugierten induzieren Abbildungen $\sigma_1, \ldots, \sigma_n$:

$$\begin{array}{rccc} \sigma_i : & K & \to & \mathbb{C} \\ & \alpha = \sum_{j=1}^{n} q_j \varrho^{j-1} & \mapsto & \sum_{j=1}^{n} q_j \varrho^{(i)j-1} \quad 1 \leq i \leq n. \end{array}$$

Die Abbildungen $\sigma_1, \ldots, \sigma_n$ sind genau die Einbettungen von K in $\mathbb{C}$, die $\mathbb{Q}$ punktweise festlassen.

Bezeichnung 2.47. *Die Einbettungen von K in $\mathbb{C}$, die $\mathbb{Q}$ punktweise festlassen bezeichnen wir mit $\sigma_1, \ldots, \sigma_n$. Sei $\alpha \in K$. Für $\sigma_i(\alpha)$ schreiben wir auch $\alpha^{(i)}$ für $1 \leq i \leq n$.*

Schließlich können wir die Diskriminante eines algebraischen Zahlkörpers einführen und ein Kriterium angeben, wann ein Primideal $p\mathbb{Z}$ in $\mathfrak{o}_K$ verzweigt.

Definition 2.48. *Für $\alpha_1, \ldots, \alpha_n \in K$ definiere die Diskriminante als*

$$\operatorname{disc}(\alpha_1, \ldots, \alpha_n) := \begin{vmatrix} \sigma_1(\alpha_1) & \sigma_1(\alpha_2) & \cdots & \sigma_1(\alpha_n) \\ \vdots & \vdots & & \vdots \\ \sigma_n(\alpha_1) & \sigma_n(\alpha_2) & \cdots & \sigma_n(\alpha_n) \end{vmatrix}^2$$

Für $\alpha_1, \ldots, \alpha_n \in \mathfrak{o}_K$ gilt $\operatorname{disc}(\alpha_1, \ldots, \alpha_n) \in \mathbb{Z}$.

Bezeichnung 2.49. *Sei $\gamma_1, \ldots, \gamma_n$ eine $\mathbb{Z}$-Basis von $\mathfrak{o}_K$, so definieren wir $\mathfrak{d}_K := \operatorname{disc}(\gamma_1, \ldots, \gamma_n)$ als die Diskriminante von $\mathfrak{o}_K$ bzw. K.*

$\mathfrak{d}_K$ ist wohldefiniert, da je zwei $\mathbb{Z}$-Basen von $\mathfrak{o}_K$ dieselbe Diskriminante besitzen.

Satz 2.50. *Bezeichne $\mathfrak{o}_K$ die Maximalordnung und und $\mathfrak{d}_K$ die Diskriminante von K. Sei $p \in \mathbb{Z}$ eine Primzahl. Dann sind folgende Aussagen äquivalent:*

(1) *$p\mathbb{Z}$ ist verzweigt in $\mathfrak{o}_K$.*

(2) $p|\mathfrak{d}_K$ *(in* $\mathbb{Z}$*).*

Insbesondere sind stets nur endlich viele Primideale in einer Maximalordnung verzweigt.

DEFINITION 2.51. *Für eine von den Elementen* $\alpha_1, \ldots, \alpha_n \in K$ *erzeugte additive Untergruppe* G *von* K *definiert man*

$$\operatorname{disc}(G) := \operatorname{disc}(\alpha_1, \ldots, \alpha_n).$$

Nach [Mar95, S. 45] gilt

SATZ 2.52. *Seien* G *und* H *zwei freie abelsche Untergruppen vom Rang* n *in* K *mit* $H \subseteq G$*. Dann gelten*

(1) G/H *ist eine endliche Gruppe, insbesondere ist der Gruppenindex* $|G : H|$ *endlich und*
(2) $\operatorname{disc}(H) = |G : H|^2 \operatorname{disc}(G)$.

DEFINITION 2.53. *Für* $\alpha \in K$ *definieren wir die absolute Norm* $T_{K:\mathbb{Q}}(\alpha)$ *und die absolute Spur* $N_{K:\mathbb{Q}}(\alpha)$

$$T_{K:\mathbb{Q}}(\alpha) = \sum_{i=1}^{n} \sigma_i(\alpha) \text{ und } N_{K:\mathbb{Q}}(\alpha) = \prod_{i=1}^{n} \sigma_i(\alpha).$$

Wir schreiben auch T für $T_{K:\mathbb{Q}}$ und N für $N_{K:\mathbb{Q}}$.

SATZ 2.54. *Für* $\alpha, \beta \in K$ *gelten:*

(1) $T(\alpha), N(\alpha) \in \mathbb{Q}$,
(2) $T(\alpha + \beta) = T(\alpha) + T(\beta)$,
(3) $N(\alpha\beta) = N(\alpha)N(\beta)$ *und*
(4) $T(r\alpha) = rT(\alpha), N(r\alpha) = r^n\alpha$ *für* $r \in \mathbb{Q}$.

Nun geben wir noch eine Formel zur Bestimmung der Diskriminante eines n-Tupels an, falls dieses Tupel aus Potenzen von ϱ besteht.

SATZ 2.55. *Sei $K = \mathbb{Q}[\varrho]$ und seien $\sigma_1(\varrho), \ldots, \sigma_n(\varrho)$ die Konjugierten von ϱ über $\mathbb{Q}$. Dann gilt:*

$$\operatorname{disc}(1, \varrho, \ldots, \varrho^{n-1}) = \prod_{1 \le r < s \le n} (\varrho_r - \varrho_s)^2 = \pm N_{\mathbb{Q}}^K(f'(\varrho)).$$

Das Zeichen + gilt genau dann, wenn $n \equiv 0$ oder $1 \pmod 4$.

Für die Bestimmung von Schranken bei der Determinantenberechnung benötigen wir den Begriff der Dualbasis.

SATZ 2.56. *Sei $\alpha_1, \ldots, \alpha_n$ eine Basis von K über $\mathbb{Q}$. Dann gibt es $\beta_1, \ldots, \beta_n \in K$, so daß $T(\alpha_{i\beta_j}) = \delta_{ij}$, $i, j \in \{1, \ldots, n\}$.*

$\beta_1, \ldots, \beta_n$ ist dann ebenfalls eine Basis von K über $\mathbb{Q}$ und wird Dualbasis zu $\alpha_1, \ldots, \alpha_n$ genannt.

3.2. Relative Erweiterungen. Wir haben oben den Fall behandelt, daß der betrachtete algebraische Zahlkörper K als absolute Erweiterung $K/\mathbb{Q}$ gegeben ist. Nun betrachten wir den Fall, daß ein algebraischer Zahlkörper K_1 als relative Erweiterung K_1/K_0 gegeben ist.

Nach [Mey80b] und [Fri97, S. 14] gilt der folgende Satz.

SATZ 2.57. *Seien $K_0/\mathbb{Q}$ und K_1/K_0 endliche Körpererweiterungen mit $[K_0 : \mathbb{Q}] = n_0, [K_1 : K_0] = n_1$, dann ist $K_1 : \mathbb{Q}$ eine endliche Körpererweiterung vom Grad $[K_1 : \mathbb{Q}] = n_0 n_1$ und K_1 somit ein algebraischer Zahlkörper. Analog zum absoluten Fall existiert ein normiertes über K_0 irreduzibles Polynom $f_1(x)$ vom Grad n_1 mit Koeffizienten aus der Maximalordnung von K_0, so daß für eine Nullstelle ψ von $f_1(x)$ gilt:*

$$K_1 = K_0[\psi]$$

BEZEICHNUNG 2.58. *Die Körpererweiterung K_1/K_0 nennen wir eine relative Erweiterung, falls der Grundkörper K_0 von $\mathbb{Q}$ verschieden ist.*

Generalvoraussetzung: K_0 bezeichne stets einen algebraischen Zahlkörper der absoluten Erweiterung $K_0/\mathbb{Q}$ mit $n_0 = [K_0 : \mathbb{Q}]$. $\mathfrak{o}_0$ sei eine Ordnung von K_0 mit $\mathbb{Z}$-Basis $b_1^{(0)}, \ldots, b_{n_o}^{(0)}$.
Weiterhin bezeichne K_1 einen algebraischen Zahlkörper der relativen Erweiterung K_1/K_0 mit $n_1 = [K_1 : K_0]$.

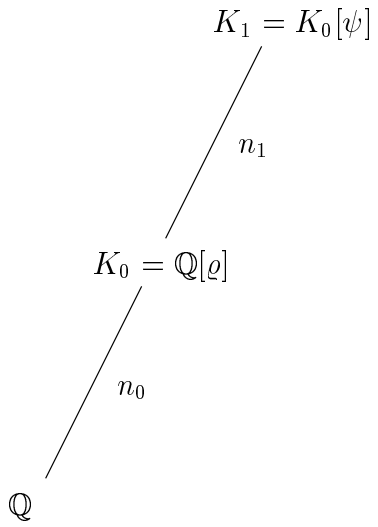

Abbildung 2.6. Einfach relative Erweiterung

Satz 2.59. *Sei* $b_1^{(1)}, \ldots, b_{n_1}^{(1)}$ *eine* K_0*-Basis von* K_1*. Dann ist* $\{b_i^{(0)} b_j^{(1)}\}_{\substack{1 \le i \le n_0 \\ 1 \le j \le n_1}}$ *eine* $\mathbb{Q}$*-Basis von* K_1*.*

Die Multiplikationstabelle einer relativen Erweiterung K_1/K_0 bezüglich einer K_0-Basis $b_1^{(1)}, \ldots, b_{n_1}^{(1)}$ wird festgelegt mittels

$$b_i^{(1)} b_j^{(1)} = \sum_{k=1}^{n_1} \Gamma_{i,j,k}^{K_1} b_k^{(1)}.$$

Da wir jedoch in K_1 stets eine feste K_0-Basis $b_1^{(1)}, \ldots, b_{n_1}^{(1)}$ von K_1 betrachten, sprechen wir auch kurz von der Multiplikationstabelle von K_1/K_0 und bezeichnen sie mit $(\Gamma_{i,j,k}^{K_1})_{i,j,k \in \{1,\ldots,n_1\}}$.

Die oben erwähnte relative Erweiterung K_1/K_0 bezeichnen wir auch als einfache relative Erweiterung, da K_0 eine absolute Erweiterung ist. Unter einer mehrfachen relativen Erweiterung verstehen wir die relative Erweiterung einer relativen Erweiterung.

Es seien K_i, $0 \leq i \leq r$ algebraische Zahlkörper und die Erweiterung K_i/K_{i-1} habe den Grad n_i, $1 \leq i \leq r$. In jedem algebraischen Zahlkörper K_i, $1 \leq i \leq r$ betrachten wir eine feste K_{i-1}-Basis $b_1^{(i)}, \ldots, b_{n_1}^{(i)}$ von K_i für , $1 \leq i \leq r$.

DEFINITION 2.60. *Als Produktbasis von K_r bezeichnen wir:*

$$\{b_{i_0}^{(0)} \cdot \cdots \cdot b_{i_r}^{(r)} \mid i_j \in \{1, \ldots, n_j\} \text{ für } 0 \leq j \leq r\}. \tag{2-16}$$

BEMERKUNG 2.61. *Durch eine Induktion über r erhalten wir aus Satz 2.59, daß die Produktbasis (2-16) von K_r eine $\mathbb{Q}$-Basis von K_r.*

Der von der Produktbasis erzeugte $\mathbb{Z}$-Modul zusammen mit der Multiplikation in K_r ist im allgemeinen nicht multiplikativ abgeschlossen und somit kein Ring und erst recht keine Algebra. Auf diese Struktur ist das modulare Verfahren nicht anwendbar.

Unser Ziel ist es, eine freie kommutative Algebra anzugeben, auf die wir im relativen Fall das modulare Verfahren anwenden können. Diese Struktur werden wir bei der Multiplikation von algebraischen Zahlen in Kapitel 3.2 verwenden.

Die Basis (2-16) bezeichnen wir im folgenden mit $b_1, \ldots, b_t$. Jedes Element $\alpha \in K_r$ kann man bezüglich der Produktbasis $b_1, \ldots, b_t$ von K_r in der Form

$$\alpha = \frac{\sum_{i=1}^{t} z_i b_i}{d}$$

darstellen, wobei $z_i \in \mathbb{Z}$ für $1 \leq i \leq t$, $d \in \mathbb{N}$ und $\mathrm{ggT}(z_1, \ldots, z_t, d) = 1$.

DEFINITION 2.62. *Den Nenner d in obiger Darstellung bezeichnen wir auch als* $d = \mathrm{Nen}(\alpha)$

DEFINITION 2.63. *Als den Nenner $\mathcal{N}(K_r)$ von K_r bezeichnen wir das kleinste gemeinsame Vielfache der Nenner der Einträge der Multiplikationstabelle von*

K_r:

$$\mathcal{N}(K_r) := \mathrm{kgV}(\{\mathrm{Nen}(\Gamma_{i,j,k}^{K_r}) \mid i,j,k \in \{1,\dots,n_r\}\})$$

Schließlich definieren wir noch zu K_r die Größe $\mathfrak{N}(K_r)$, mit deren Hilfe die gewünschte Algebra definiert werden kann.

DEFINITION 2.64. *Mit den obigen Bezeichnungen definieren wir*

$$\mathfrak{N}(K_r) := \prod_{i=0}^{r} \mathcal{N}(K_i)^{2^{r-i}} \in \mathbb{N}.$$

BEMERKUNG 2.65. *Für eine absolute Erweiterung $K_0/\mathbb{Q}$ gilt nach* 2.38 $\mathfrak{N}(K_0) = 1$.

DEFINITION 2.66. *Den von der Produktbasis $b_1,\dots,b_t$ (Definition* 2.60*) von K_r erzeugten $\mathbb{Z}$-Modul $(A,+)$*

$$A = \bigoplus_{i=1}^{t} \mathbb{Z} b_i$$

versehen wir mit der Multiplikation $*$:

$$\begin{array}{rccc} * : & M \times M & \to & M \\ & x * y & \mapsto & \mathfrak{N}(K_r) \cdot x \cdot y \end{array}$$

Auf der rechten Seite steht dabei die normale Multiplikation in K_r. Die Struktur $(A,+,*)$ nennen wir die zu K_r gehörige Algebra und bezeichnen sie mit $A(K_r)$. $b_1,\dots,b_t$ nennen wir die zu $A(K_r)$ gehörige $\mathbb{Z}$-Basis. (Hier wird $b_1,\dots,b_t$ als $\mathbb{Z}$-Basis des $\mathbb{Z}$-Moduls $A(K_r)$ aufgefaßt.)

SATZ 2.67. *Die Struktur $(A,+,*) := A(K_r)$ ist eine freie kommutative Algebra über $\mathbb{Z}$.*

Beweis: Wir führen eine Induktion über r durch. Für $r = 0$ liegt der Fall einer absoluten Erweiterung vor und mit Bemerkung 2.65 erhalten wir, daß $(A,+,*)$ mit der Struktur $(\mathfrak{o}_0,+,\cdot)$ übereinstimmt und somit eine Ordnung ist. Nach Satz 2.37 ist $A(K_0)$ eine freie kommutative Algebra über $\mathbb{Z}$. Dies ist die Behauptung für r = 0.

Nun führen wir den Induktionsschritt durch. Es ist zu zeigen, daß $A(K_{r+1})$ bezüglich $*$ abgeschlossen ist. Dann ist $A(K_{r+1})$ offensichtlich ein Ring und sogar eine freie kommutative Algebra über $\mathbb{Z}$.
Seien $\alpha, \beta \in A(K_{r+1})$ mit

$$\alpha = \sum_{i=1}^{n_{r+1}} \alpha_i b_i^{(r+1)} \text{ und } \beta = \sum_{i=1}^{n_{r+1}} \beta_i b_i^{(r+1)}.$$

Dabei sind $\alpha_i, \beta_i \in A(K_r)$ für $1 \leq i \leq n_{r+1}$. Dann gilt:

$$\begin{aligned}
\alpha * \beta &= \mathfrak{N}(K_{r+1})\alpha\beta \\
&= \mathfrak{N}(K_{r+1}) \sum_{k=1}^{n_{r+1}} \left(\sum_{i,j=1}^{n_{r+1}} \alpha_i \beta_j \Gamma_{i,j,k}^{K_{r+1}} \right) b_k^{(r+1)} \\
&= \frac{\mathfrak{N}(K_{r+1})}{\mathfrak{N}^2(K_r)} \sum_{k=1}^{n_{r+1}} \left(\sum_{i,j=1}^{n_{r+1}} (\alpha_i * \beta_j) * \Gamma_{i,j,k}^{K_{r+1}} \right) b_k^{(r+1)} \\
&= \mathcal{N}(K_{r+1}) \sum_{k=1}^{n_{r+1}} \left(\sum_{i,j=1}^{n_{r+1}} (\alpha_i * \beta_j) * \Gamma_{i,j,k}^{K_{r+1}} \right) b_k^{(r+1)} \\
&= \sum_{k=1}^{n_{r+1}} \left(\sum_{i,j=1}^{n_{r+1}} (\alpha_i * \beta_j) * \left(\mathcal{N}(K_{r+1}) \Gamma_{i,j,k}^{K_{r+1}} \right) \right) b_k^{(r+1)}.
\end{aligned}$$

Dabei ist nach Definition 2.62

$$\mathcal{N}(K_{r+1})\Gamma_{i,j,k}^{K_{r+1}} \in A(K_r) \text{ für } 1 \leq i, j, k \leq n_{r+1}.$$

Weiterhin gilt nach Induktionsvoraussetzung

$$(\alpha_i * \beta_j) * \left(\mathcal{N}(K_{r+1}) \Gamma_{i,j,k}^{K_{r+1}} \right) \in A(K_r) \text{ für } 1 \leq i, j, k \leq n_{r+1}.$$

Da die $\mathbb{Z}$-Basis von $A(K_{r+1})$ nach Definition 2.66 die Produktbasis von K_{r+1} ist, folgt die Behauptung

$$\alpha * \beta \in A(K_{r+1}).$$

□

4. Zeitaufwand und Komplexität

In diesem Abschnitt legen wir die Grundlagen dafür, den Zeitbedarf eines Algorithmus abzuschätzen. Die für die Ausführung des Algorithmus auf einem Computersystem benötigte Zeit ist vom Computersystem abhängig. Deshalb werden wir stattdessen nur die Anzahl der auszuführenden zeitaufwendigsten Grundoperationen des Algorithmus angeben. Die Anzahl wird in der Regel von den Eingabedaten abhängen, nicht jedoch von dem verwendeten Computersystem und ist somit charakteristisch für den verwendeten Algorithmus. Oft werden wir sogar nur eine asymptotische obere Schranke für diese Anzahlfunktion angeben. Dies geschieht mit Hilfe des Landau-Symbols O.

DEFINITION 2.68. *Sei* $g : \mathbb{N}^r \supseteq D_g \to \mathbb{R}, (n_1, \ldots, n_r) \mapsto g(n_1, \ldots, n_r)$, *so daß es ein* $n \in \mathbb{N}$ *derart gibt, für das* $g(n_1, \ldots, n_r)$ *definiert und positiv ist für alle* $n_i > n$. *Dann definieren wir:*

$$O(g) = \{f : \mathbb{N}^r \supseteq D_f \to \mathbb{R} \mid \text{es existieren } n \in \mathbb{N} \text{ und } c \in \mathbb{R}, \text{ so daß } f(n_1, \ldots, n_r),\ g(n_1, \ldots, n_r) \text{ definiert und positiv und } f(n_1, \ldots, n_r) \leq c \cdot g(n_1, \ldots, n_r) \text{ für alle } n_i > n\}.$$

Wenn $f \in O(g)$, *so sagen wir auch, daß* f *ist durch* g *beschränkt ist.*

BEZEICHNUNG 2.69. *Anstatt* $f \in O(g)$ *werden wir die übliche Schreibweise* $f = O(g)$ *verwenden.*

Mit Hilfe des Landau-Symbols O geben wir für eine Funktion f unter Ignorieren konstanter Faktoren eine asymptotische obere Schranke an.
Für $f_1 = O(g_1)$ und $f_2 = O(g_2)$ gilt $f_1 + f_2 = O(g_1 + g_2)$ und $f_1 \cdot f_2 = O(g_1 \cdot g_2)$.

Wir schätzen den Zeitaufwand ab, indem wir die Anzahl der benötigten Bitoperationen [Kob87] mit Hilfe des Landau-Symbols beschreiben.

Als erstes Beispiel betrachten wir Addition, Subtraktion, Multiplikation und Division von ganzen Zahlen in Binärdarstellung mit Hilfe der Schulmethode.
Die Anzahl der Bitoperationen für die Addition und Subtraktion zweier n-Bit

Zahlen ist jeweils $O(n)$. Für die Multiplikation zweier n-Bit Zahlen werden $O(n^2)$ Bitoperationen, für die Division mit Rest von einer n-Bit Zahl durch eine m-Bit Zahl $O(nm)$ Bitoperationen benötigt [Kob87].

In den in Kapitel 3 verwendeten Ringen ist in der von uns verwendeten Darstellung der Ringelemente die Multiplikation ebenfalls zeitaufwendiger als die Addition.

Bei den endlichen Körpern $\mathbb{F}_q$ sind Addition, Subtraktion, Multiplikation und Division nach [Coh93, S. 5] in nur $O(\log_2^2(q))$ Bitoperation durchführbar. Bei der Determinantenberechnung werden wir endliche Körper als homomorphe Bilder wählen.

KAPITEL 3

Anwendung des modularen Verfahrens

In diesem Kapitel werden zwei Anwendungen des modularen Verfahrens für algebraische Zahlkörper vorgestellt und mit nichtmodularen Verfahren verglichen. Diese Anwendungen sind die Berechnung von Determinanten über algebraischen Zahlkörpern und die Multiplikation von algebraischen Zahlen.

1. Determinantenberechnung

Generalvoraussetzung: In diesem Kapitel sei $K/\mathbb{Q}$ stets ein algebraischer Zahlkörper vom Grad n. $\mathfrak{o}_K$ sei die Maximalordnung von K und $\{\omega_1, \ldots, \omega_n\}$ eine $\mathbb{Z}$-Basis von $\mathfrak{o}_K$. Weiterhin bezeichne $\mathfrak{d}_K$ die Diskriminante von K.

Mit $m \in \mathbb{Z}, \alpha \in \mathfrak{o}_K$ schreiben wir entsprechend Bezeichnung 2.25 $\alpha \bmod p$ für den positiven kanonischen Repräsentanten von $\alpha + (m\mathfrak{o}_K)$ und $\alpha \operatorname{mods} p$ für den symmetrischen kanonischen Repräsentanten von $\alpha + (m\mathfrak{o}_K)$.

1.1. Der nichtmodulare Algorithmus. Sei $M = (m_{ij})_{1 \le i,j \le r} \in K^{r \times r}$ eine quadratische Matrix mit Einträgen aus dem algebraischen Zahlkörper K. Die Berechnung von $\mathrm{Det}(M)$ geschieht mit Hilfe der Gauss-Elimination. Die Matrix M wird durch die beiden elementaren Zeilenumformungen

(1) Subtraktion des Vielfachen einer Zeile von einer anderen,

(2) Vertauschen zweier Zeilen

in obere Dreiecksgestalt gebracht. Dabei wird berücksichtigt, daß durch das die elementare Zeilenumformung (2) das Vorzeichen der Determinante geändert wird. Durch die elementare Zeilenumformung (1) ändert sich die Determinante nicht. Die Determinante einer Matrix in Dreiecksgestalt erhält man als Produkt der Hauptdiagonalelemente.

Der folgende Algorithmus arbeitet nach diesem Verfahren und ist für beliebige Körper gültig [Coh93, S. 49].

ALGORITHMUS 3.1. (DetGauss: Determinantenberechnung mit Gauss-Elimination)

Input: $M = (m_{i,j})$, $1 \leq i, j \leq r$
Output: $d = \det(M)$

Schritt 1: *(Initialisierung)*
(1) $j \leftarrow 0$
(2) $d \leftarrow 1$
Schritt 2: *(Fertig?)*
(3) $j \leftarrow j + 1$
(4) *Wenn* $j > r$ *dann gib* d *aus. ENDE.*
Schritt 3: *(Finde Element ungleich Null)*
(5) *Wenn* $m_{i,j} = 0$ *für alle* $i \geq j$ *dann gib 0 aus. ENDE.*
(6) *Sonst sei* i *ein Index* $i \geq j$ *mit* $m_{i,j} \neq 0$.
Schritt 4: *(Vertauschen)*
(7) *Wenn* $i > j$
(8) *Für* $l = j, \ldots, r$*:*
(9) *Vertausche* $m_{i,l}$ *mit* $m_{j,l}$.
(10) $d \leftarrow -d$
Schritt 5: *(Elimination)*
(11) $\tilde{c} \leftarrow m_{j,j}^{-1}$
(12) *Für* $k = j + 1, \ldots, r$*:*
(13) $c_k \leftarrow \tilde{c} m_{k,j}$
(14) *Für* $l = j + 1, \ldots, r$*:*

(15) $m_{k,l} \leftarrow m_{k,l} - c_k m_{j,l}$

(16) $d \leftarrow d m_{j,j}$

(17) *Gehe zu Schritt 2*

SATZ 3.2. *Für die Berechnung der Determinante einer $r \times r$-Matrix mit Algorithmus* 3.1 *werden $\frac{1}{3}r^3 + O(r^2)$ Multiplikationen bzw. Divisionen im Körper benötigt.*

Beweis: Zeile (11) wird höchstens r mal durchlaufen, Zeile (13) höchstens $\sum_{j=1}^{r} j - 1 = \frac{(r-1)r}{2}$ mal, Zeile (15) höchstens $\sum_{j=1}^{r}(j-1)^2 = \frac{(r-1)r(2r-1)}{6}$ mal und Zeile (16) höchstens r mal. In jeder dieser Zeilen wird eine Multiplikation oder Invertierung (Division) im Körper durchgeführt. Insgesamt sind dies $\frac{1}{3}r^3+\frac{5}{3}r$ Multiplikationen bzw. Divisionen. □

Diese Abschätzung gilt für alle Körper. Dennoch ist das Laufzeitverhalten dieses Algorithmus stark von dem verwendeten Körper abhängig. Dieses werden wir uns zunutze machen.
Bei der nichtmodularen Methode wird der Algorithmus 3.1 direkt auf den algebraischen Zahlkörper K angewendet. In diesem Fall tritt durch die Multiplikationen und Invertierungen in Schritt 5 (Elimination) eine Koeffizientenexplosion auf. Um ein Element

$$k \in K, k = \frac{1}{d_k}\sum_{i=1}^{n} k_i\omega_i \text{ mit } k_1, \ldots, k_n, d_k \in \mathbb{Z}$$

in die gekürzte Darstellung

$$k = \frac{1}{\tilde{d}_k}\sum_{i=1}^{n} \tilde{k}_i\omega_i \text{ mit } \mathrm{ggT}(\tilde{k}_1, \ldots, \tilde{k}_n, \tilde{d}_k) = 1$$

zu überführen, werden zeitaufwendige ggT-Berechnungen notwendig. Diese verlangsamen den Algorithmus weiter.

1.2. Der modulare Algorithmus. Bei der modularen Methode werden wir in endlichen Körpern isomorph zu $\mathbb{F}_q, q = p^f$, wobei p Primzahl und $f \in \mathbb{N}$, rechnen. Dort treten die oben geschilderten Probleme nicht auf. Nach Kapitel 1.4 sind Addition, Subtraktion, Multiplikation und Division in $\mathbb{F}_q$ schnell ausführbar.

Zu Beginn müssen wir das Problem der Determinantenberechnung einer Matrix $M = (m_{i,j}) \in K^{r\times r}$ mit $m_{i,j} = \frac{1}{d_{m_{i,j}}} \sum_{k=1}^{n} \mu_{i,j,k}\omega_k$ in gekürzter Darstellung auf Berechnungen über $\mathfrak{o}_K$ zurückführen.

Sei $t := \mathrm{kgV}_{1\le i,j\le r}(d_{m_{i,j}})$. Wir wenden das modulare Verfahren auf die Matrix

$$A := t \cdot M \in \mathfrak{o}_K^{r\times r}$$

an. Sei im folgenden $A = (\alpha_{i,j})_{1\le i,j\le r}$. Die Maximalordnung ist nach Satz 2.29 für das modulare Verfahren geeignet. Das gesuchte Ergebnis $d = \mathrm{Det}(M)$ erhält man mittels

$$d = \frac{\mathrm{Det}(A)}{t^r}$$

wegen der Linearität der Determinante in den Zeilen bzw. Spalten.

1.2.1. Bestimmung der Schranke

Sei $\omega_1^*, \ldots, \omega_n^*$ die Dualbasis von $\omega_1, \ldots, \omega_n$.

Ziel ist die Berechnung einer obere Schranke $S(A)$ für den Betrag der Koeffizienten von $\delta := \mathrm{Det}(A)$, wobei $A \in \mathfrak{o}_K^{r\times r}$ bezüglich der Ganzheitsbasis $\omega_1, \ldots, \omega_n$. Sei

$$\delta = \sum_{i=1}^{n} \delta_i\omega_i,\ \delta_i \in \mathbb{Z} \text{ für } 1 \le i \le n, \tag{3-1}$$

dann ist eine Schranke $S(A)$ mit

$$\max_{1\le i\le n} |\delta_i| < S(A)$$

gesucht.

Folgende Ungleichung ist in [Hup90, S. 185] bewiesen.

Satz 3.3. *(Ungleichung von Hadamard)*
Sei $M = (m_{i,j})_{1\le i,j\le n} \in \mathbb{C}^{n\times n}$. Dann gilt

$$|\mathrm{Det}(M)| \le \sqrt{\prod_{j=1}^{n}\left(\sum_{i=1}^{n} |m_{i,j}|^2\right)}.$$

Nach der Definition der Dualbasis und der Spur $T = T_{K/\mathbb{Q}}$ gilt:

$$T(\delta\omega_j^*) = \sum_{k=1}^{n} \delta^{(k)}\omega_j^{*(k)} = \sum_{i,k=1}^{n} \delta_i^{(k)}\omega_i^{(k)}\omega_j^{*(k)} = \sum_{i=1}^{n} \delta_i T(\omega_i\omega_j^*) = \delta_j. \tag{3-2}$$

Dabei sind $\delta^{(k)}$ bzw. $\omega^{*(k)}$, $1 \le k \le n$ die Konjugierten von δ bzw. ω^*. Mit Hilfe der Dreiecksungleichung und der Ungleichung von Cauchy-Schwarz ergibt sich aus (3-2) und (3-1):

$$\begin{aligned} |\delta_j| &= |T(\delta\omega_j^*)| = |\sum_{k=1}^{n} \delta^{(k)}\omega_j^{*(k)}| \le \sum_{k=1}^{n} |\delta^{(k)}\omega^{*(k)}_j| \\ &\le \sqrt{\sum_{k=1}^{n} |\delta^{(k)}|^2}\sqrt{\sum_{k=1}^{n} |\omega_j^{*(k)}|^2} \end{aligned} \tag{3-3}$$

Für den Betrag $|\delta^{(k)}|$ der Konjugierten von δ erhalten wir mit der Ungleichung von Hadamard (Satz 3.3) folgende Abschätzung:

$$|\delta^{(k)}| = |\left(\mathrm{Det}(A)\right)^{(k)}| \le \sqrt{\prod_{j=1}^{n}\left(\sum_{i=1}^{n} |\alpha_{i,j}^{(k)}|\right)} \tag{3-4}$$

Insgesamt erhalten wir aus (3-3) und (3-4) folgenden Satz:

Satz 3.4. *Für $A = (\alpha_{i,j}) \in \mathfrak{o}_K^{r\times r}, \delta = \mathrm{Det}(A)$, wobei $\delta = \sum_{i=1}^{n} \delta_i\omega_i$ ist, gilt:*

$$|\delta_j| \le \sqrt{\left(\sum_{k=1}^{n}\left(\prod_{j=1}^{n}\left(\sum_{i=1}^{n} |\alpha_{i,j}^{(k)}|^2\right)\right)\right)\left(\sum_{k=1}^{n} |\omega_j^{*(k)}|^2\right)}.$$

Korollar 3.5. *Für $A = (\alpha_{i,j}) \in \mathfrak{o}_K^{r\times r}, \delta = \mathrm{Det}(A)$, wobei $\delta = \sum_{i=1}^{n} \delta_i\omega_i$ ist, gilt:*

$$S(A) := \sqrt{\left(\sum_{k=1}^{n}\left(\prod_{j=1}^{n}\left(\sum_{i=1}^{n} |m_{i,j}^{(k)}|^2\right)\right)\right)\max_{1\le j\le n}\left(\sum_{k=1}^{n} |\omega_j^{*(k)}|^2\right)} + 1$$

ist eine obere Schranke für die Beträge der Koeffizienten von δ, d.h.

$$\max_{1\le i\le n}(|\delta_i|) < S(A).$$

1.2.2. Wahl der Moduli

Um das modulare Verfahren korrekt anwenden zu können, wählen wir, bei der Primzahl 2 beginnend, s aufeinander folgende geeignete Primzahlen $p_1, \ldots, p_s$ mit

$$\prod_{i=1}^{s} p_i \geq 2S(A).$$

Dabei ist eine Primzahl p genau dann für uns geeignet, wenn p kein Diskriminantenteiler ist, d.h. es soll

$$p \not| \mathfrak{d}_K \tag{3-5}$$

gelten. Dahinter steckt die folgende Überlegung:
Für eine Primzahl p ist $(p\mathfrak{o}_K)$, das von p in $\mathfrak{o}_K$ erzeugte Ideal, i.a. kein Primideal und somit erst recht kein maximales Ideal in $\mathfrak{o}_K$. Da $\mathfrak{o}_K$ ein Dedekindring ist, besitzt das Ideal $(p\mathfrak{o}_K)$ nach Satz 2.40 eine bis auf Reihenfolge eindeutige Zerlegung in paarweise verschiedene Primideale:

$$(p\mathfrak{o}_K) = \mathfrak{p}^{e_1} \cdots \mathfrak{p}_r^{e_r}$$

Die Restklassenringe $\mathfrak{o}_K/\mathfrak{p}_j$, $1 \leq j \leq r$ sind sogar Körper, da in Dedekindringen jedes Primideal auch maximales Ideal ist (Definition 2.39). Wenn $e_1 = \cdots = e_r = 1$ gelten würde, d.h. $p\mathbb{Z}$ in $\mathfrak{o}_K$ unverzweigt wäre, so könnte man nach der Berechnung von $\delta_i = \delta \bmod \mathfrak{p}_j$ in den Körpern $\mathfrak{o}_K/\mathfrak{p}_j$, $1 \leq j \leq r$ mit Hilfe des chinesischen Restsatzes $\delta \bmod (\prod_{j=1}^{r} \mathfrak{p}_j) = \delta \bmod p\mathfrak{o}_K$ berechnen. Falls $p\mathbb{Z}$ verzweigt, so ist die Berechnung von $\delta \bmod (p\mathfrak{o}_K)$ nicht möglich, da $\prod_{j=1}^{r} \mathfrak{p}_j \neq (p\mathfrak{o}_K)$. Nach Satz 2.50 ist eine Primzahl genau dann für uns geeignet, wenn $p \not| \mathfrak{d}_K$.

Falls die Diskriminante $\mathfrak{d}_K$ nicht bereits berechnet ist, so können wir das folgende hinreiche Kriterium für die Unverzweigtheit von $p\mathbb{Z}$ verwenden, falls ein $\varrho \in \mathfrak{o}_K$ mit $K = \mathbb{Q}[\varrho]$ bekannt ist.

SATZ 3.6. *Sei p eine Primzahl und $K = \mathbb{Q}[\varrho]$ ein algebraischer Zahlkörper. Dann gilt:*

$$p \not| \operatorname{disc}(\mathbb{Z}[\varrho]) \Rightarrow p\mathfrak{o}_k \text{ unverzweigt.}$$

Dabei gilt $\mathbb{Z}[\varrho] = \mathbb{Z} \oplus \mathbb{Z}\varrho \oplus \cdots \oplus \mathbb{Z}\varrho^{n-1}$.

Beweis: Nach Voraussetzung gilt $\varrho \in \mathfrak{o}_K$. Somit gilt für die beiden freien abelschen Untergruppen $\mathbb{Z}[\varrho]$ und $\mathfrak{o}_K$ von K vom Rang n die Inklusion $\mathbb{Z}[\varrho] \subseteq \mathfrak{o}_K$. Aus Satz 2.52 folgt die Behauptung mit Satz 2.50. □

In KANT ist meistens ein $\varrho \in \mathfrak{o}_K$ mit $K = \mathbb{Q}[\varrho]$ bekannt. Daher ist $\mathrm{disc}(\mathbb{Z}[\varrho])$ mit Hilfe von Satz 2.55 schneller zu berechnen als $\mathfrak{d}_K$.

Im folgenden werden wir $\mathfrak{d}_K$ als bekannt voraussetzen und das günstigere Kriterium (3-5) verwenden.

Bei der Wahl von p nach Satz 3.6 sind wir nicht stark eingeschränkt, da natürlich nur endlich viele Primzahlen $\mathfrak{d}_K$ teilen.

Wir wählen, mit 2 beginnend, aufeinanderfolgende Primzahlen $p_1, \ldots, p_s$, so daß $p_i \nmid \mathfrak{d}_K$, $1 \leq i \leq s$ und $\prod_{i=1}^{s} \geq 2S$.

Der folgende Satz nach [Mü94, S. 55] gibt uns erstens eine gemeinsame obere Schranke für die Primzahlen p_i, $1 \leq i \leq s$ und zweitens eine obere Schranke für die Anzahl s der Primzahlen.

Satz 3.7. *Für $m, C \in \mathbb{N}$ mit $C \geq 2$ sei $P \in \mathbb{N}$ definiert als*

$$P := \max\{43, \lceil 2\log_2(mC)\rceil\}.$$

Dann gilt die Ungleichung

$$\prod_{\substack{p \leq P,\, p \nmid m \\ p \text{ Primzahl}}} p \geq C.$$

Korollar 3.8. *Sei*

$$P := \max\{43, \lceil 2\log_2(\mathfrak{d}_K 2S(A))\rceil\}.$$

Dann gilt mit obigen Bezeichnungen:

(1) $p_s \leq P$ *und*
(2) $s \leq \pi(P) < 1,26\frac{P}{\log(P)}$.

Dabei ist $\pi(P)$ die Anzahl der Primzahlen kleiner oder gleich P.

Beweis: Teil (1) folgt aus Satz 3.7. Teil (2) folgt aus der Definition von π und aus [RS62, S.69 Korollar 1]. □

1.2.3. Abbilden in die Restklassenringe

Sei p eine der Primzahlen $p_1, \ldots, p_s$.

$p\mathbb{Z}$ habe folgende Primidealzerlegung in $\mathfrak{o}_K$

$$(p\mathfrak{o}_K) = \mathfrak{p}_1, \ldots, \mathfrak{p}.$$

Die Einträge der Matrix $A = (\alpha_{k,l})_{1 \le k,l \le r}$ werden mit Hilfe des kanonischen Epimorphismus

$$\begin{array}{rccc} \Phi_j & : & \mathfrak{o}_K & \to & \mathfrak{o}_K/\mathfrak{p}_j \\ & & \alpha & \mapsto & \alpha + \mathfrak{p}_j \end{array}$$

abgebildet. Die zugehörige Matrix und ihre Determinante bezeichnen wir mit

$$A_j := (\Phi_j(\alpha_{k,l}))_{1 \le k,l \le r} \text{ und } \delta_j := \mathrm{Det}(A_j).$$

1.2.4. Auswerten in den Restklassenringen

Nach Korollar 2.46 ist $\mathfrak{o}_K/\mathfrak{p}_j$ ein endlicher Körper mit p^f Elementen, wobei $f = f(\mathfrak{p}_j/p\mathbb{Z})$ und somit isomorph zu $\mathbb{F}_q, q = p^f$. Die Berechnung von $\mathrm{Det}(A_j)$ erfolgt mit Algorithmus 3.1 über dem endlichen Körper $\mathfrak{o}_K/\mathfrak{p}_j$.
Die Verwendung von endlichen Körpern ist ein großer Vorteil des modularen Verfahrens gegenüber der direkten Berechnung im algebraischen Zahlkörper. In endlichen Körpern kann keine Koeffizientenexplosion auftreten, und die zeitaufwendige ggT-Berechnung entfällt ebenfalls. Nach Kapitel 1.4 sind Addition, Subtraktion, Multiplikation und Division in $\mathbb{F}_q$ in $O(\log_2^2(q))$ Bitoperationen ausführbar.

1.2.5. Chinesischer Restsatz

In KANT steht uns ein chinesischer Restsatz „ChinRem“ zur Verfügung, der für $\alpha_1, \alpha_2 \in \mathfrak{o}_K$ und zwei Ideale $\mathfrak{a}_1, \mathfrak{a}_2$ von $\mathfrak{o}_K$ als Rückgabewert

$$\alpha = \mathrm{ChinRem}(\alpha_1, \alpha_2, \mathfrak{a}_1, \mathfrak{a}_2)$$

liefert, wobei

$$\alpha \equiv \alpha_1 \bmod \mathfrak{a}_1 \text{ und } \alpha \equiv \alpha_2 \bmod \mathfrak{a}_2.$$

In Schritt 3 von Algorithmus 3.9 wird ChinRem innerhalb einer geschachtelten Schleife aufgerufen. Die äußere durchläuft die Primzahlen $p_1, \ldots, p_s$. Für jedes $p \in \{p_1, \ldots, p_s\}$ wird in der inneren Schleife für alle über $p\mathbb{Z}$ liegende Primideale von $\mathfrak{o}_K$ ChinRem einmal derart aufgerufen, daß
nach dem ersten vollständigen Durchlauf der inneren Schleife
δ_0 den Wert $\mathrm{Det}((\alpha_{i,j})_{1\le i,j\le r}) \bmod p_1$,
nach dem zweiten vollständigen Durchlauf der inneren Schleife
δ_0 den Wert $\mathrm{Det}((\alpha_{i,j})_{1\le i,j\le r}) \bmod p_1p_2$
$\vdots$
und am Ende von Schritt 3
δ_0 den Wert $\mathrm{Det}((\alpha_{i,j})_{1\le i,j\le r}) \bmod \prod_{i=1}^{s} p_i$ besitzt.
Natürlich würde man dieses Ergebnis auch bei einer anderen Reihenfolge der Primideale erhalten. Jedoch hat sich herausgestellt, daß der Zeitaufwand am geringsten ist, falls man wie oben beschrieben vorgeht, d.h. zuerst die über einer Primzahl liegenden Primideale sukzessive zusammensetzt und anschließend die von den Primzahlen $p_1, \ldots, p_s$ in $\mathfrak{o}_K$ erzeugten Ideale.

1.2.6. Rücktransformation

Die Rücktransformation entspricht der Bestimmung des symmetrischen kanonischen Repräsentanten modulo $\prod_{i=1}^{s} p_i$.

1.2.7. Der Algorithmus MatDetMod

ALGORITHMUS 3.9. (MatDetMod)

Input: $M = (m_{i,j})$, $1 \le i,j \le r$. *Dabei sei* $m_{i,j} = \frac{1}{d_{m_{i,j}}} \sum_{k=1}^{r} \mu_{i,j,k}\omega_k$, *vollständig gekürzt und* $d_{m_{ij}} \in \mathbb{N}$.

Output: $d = \mathrm{Det}(M)$

Schritt 1: $t \leftarrow \mathrm{kgV}_{1\le i,j\le r}(d_{m_{i,j}})$
$\alpha_{i,j} \leftarrow tm_{i,j}$ *für* $1 \le i,j \le r$
$\mathrm{den} \leftarrow t^r$

Schritt 2: *(Bestimmung der Schranke und Wahl der Moduli)*
$S \leftarrow \sqrt{\left(\sum_{k=1}^{n}\left(\prod_{j=1}^{n}\left(\sum_{i=1}^{n}|m_{i,j}^{(k)}|^2\right)\right)\right)\max_{1\le j\le n}\left(\sum_{k=1}^{n}|\omega_j^{*(k)}|^2\right)}+1$ *(Korollar 3-6)*
Bestimme s *aufeinander folgende Primzahlen* $p_1,\dots,p_s$,
so daß $p_i \not| \mathfrak{d}_K$ *für* $1 \le i \le s$ *und* $\prod_{i=1}^{s} \ge 2S$.

Schritt 3: *(Hauptteil)*
Für $i = 1,\dots,s$:
Bestimme alle Primidealfaktoren $\mathfrak{p}_1,\dots\mathfrak{p}_{r_i}$ *von* $(p_i\mathfrak{o}_K)$.
Für $j = 1,\dots,r_i$:
$\beta_{k,l} \leftarrow \Phi_j(\alpha_{k,l})$ *für* $1 \le k,l \le r$ *(Φ_j ist kanonischer Epimorphismus zu $\mathfrak{p}_j$)*
$\delta_j \leftarrow \mathrm{DetGauss}\left((\beta_{k,l})_{1\le k,l\le r}\right)$ *(Algorithmus 3.1 über dem Körper $\mathfrak{o}_K/\mathfrak{p}_j$)*
Für $j = r_i,\dots,1$:
Wenn $i > 1$ *oder* $j > 1$
$\mathfrak{p}_j \leftarrow \mathfrak{p}_j\mathfrak{p}_{j-1}$
$\delta_{j-1} \leftarrow \mathrm{ChinRem}(\delta_j,\delta_{j-1},\mathfrak{p}_j,\mathfrak{p}_{j-1})$
Sonst
$\mathfrak{p}_0 \leftarrow \mathfrak{p}_1$
$\delta_0 \leftarrow \delta_1$

Schritt 4: *(Rücktransformation)*
$\delta \leftarrow \delta_0 \bmod \prod_{i=1}^{s} p_i$

Schritt 5:
$d \leftarrow \frac{\delta}{den}$
Gib d aus. ENDE.

1.2.8. Korrektheit

Da der obige Algorithmus 3.9 etwas von der Vorgehensweise von Algorithmus 2.27 abweicht, gehen wir nun auf diese Abweichungen ein, um die Korrektheit von Algorithmus 3.9 zu zeigen.
Die Abweichung besteht darin, daß wir in Algorithmus 3.9 nicht den $\{+,-,\cdot\}$-Term

$$\sum_{\pi\in S_r} \operatorname{sign}(\pi) x_{1,\pi(1)},\ldots,x_{r,\pi(r)}$$

über dem Restklassenring $\mathfrak{o}_K/(p_i\mathfrak{o}_K)$, $1 \leq i \leq s$ auswerten und somit direkt den Wert $\operatorname{Det}((\alpha_{i,j})_{1\leq i,j\leq r}) \bmod p_i$ erhalten, sondern stattdessen den Algorithmus DetGauss auf bestimmte endliche Körper anwenden. Diese Körper sind die Restklassenringe, die durch Faktorisierung nach den über $p_i\mathbb{Z}$ liegenden Primidealen von $\mathfrak{o}_K$ entstehen. Da wir erstens nach Voraussetzung nur unverzweigte Primzahlen verwenden und zweitens der Algorithmus DetGauss korrekt ist, erhalten wir nach Anwendung des chinesischen Restsatzes (nach vollständigem Durchlaufen der inneren Schleife) gerade den Wert $\operatorname{Det}((\alpha_{i,j})_{1\leq i,j\leq r}) \bmod p_i$ in der Variablen δ_1. Die restlichen Schritte entsprechen dem Schema 2.27. Der Algorithmus 3.9 ist somit nach Satz 2.28 korrekt.

1.3. Vergleich. In diesem Abschnitt werden einige Beispielrechnungen aufgeführt, die mit dem Computeralgebrasystem KANT V4 in der Oberfläche KASH berechnet wurden. Alle Rechnungen wurden auf einem HP9000/735s mit 160MB Speicher unter dem Betriebssystem HP–UX 9.04 ausgeführt. In den Tabellenköpfen werden die folgenden Bezeichnungen verwendet:

N für die maximale Größe der Beträge der Koeffizienten der Matrixeinträge (Koeffizienten bzgl. der fest gewählten Ganzheitsbasis) und

r für die Anzahl der Zeilen und Spalten der Matrix

In der oberen Hälfte jedes Tabellenfeldes steht die von Algorithmus MatDetMod (Algorithmus 3.9) für die Berechnung der Determinante benötigte Zeit. Diese Zeilen sind mit dem Hinweis „modular“ in der Spalte „Algorithmus“ gekennzeichnet.
In der unteren Hälfte steht die von dem nichtmodularen Algorithmus DetGauss (Algorithmus 3.1) benötigte Zeit.

Die Berechnung der Determinante mit der nichtmodularen Methode ist bei großen Beispielen sehr zeitintensiv. Deshalb ist teilweise nur die von dem modularen Algorithmus MatDetMod benötigte Zeit angegeben.

Sei ϱ eine Nullstelle des Polynoms
$f(x) = x^2 + 1$.
Wir betrachten den Zahlkörper $K := \mathbb{Q}[\varrho]$.

r \ N	10	10^4	10^8	10^{16}	Algorithmus
4	0,26 s 0,00 s	0,42 s 0,00 s	1,20 s 0,02 s	3,02 s 0,03 s	modular
8	0,49 s 0,05 s	1,15 s 0,74 s	3,91 s 2,88 s	6,89 s 4,22 s	modular
12	1,19 s 6,36 s	3,39 s 180 s	7,78 s 12,6 min	9,82 s 19,1 min	modular
16	2,40 s 20,7 min	8,66 s	14,1 s	19.9 s	modular

Sei ϱ eine Nullstelle des Polynoms
$f(x) = x^4 - 3x^3 + 2x + 1$.
Wir betrachten den Zahlkörper $K := \mathbb{Q}[\varrho]$.

r \ N	10	10^4	10^8	10^{16}	Algorithmus
4	0,60 s 0,03 s	1.02 s 0,15 s	2,32 s 0,43 s	6,54 s 1,67 s	modular
8	1,73 s 3,68 s	3,05 s 60,2 s	8,08 s 209 s	16,0 s 266 s	modular
12	4,18 s 24,3 min	8,95 s 302 min	13,8 s	20.2 s	modular
16	6,47 s	15,7 s	22,1 s	32,9 s	modular

Sei ϱ eine Nullstelle des Polynoms
$f(x) = x^{24} + 18x^{20} + 6x^{19} + 56x^{18} - 49x^{17} - 94x^{16} - 1292x^{15} - 1005x^{14} + 865x^{13} + 1583x^{12} + 4309x^{11} + 5423x^{10} + 2630x^9 - 1015x^8 - 3463x^7 - 1747x^6 - 651x^5 + 789x^4 + 303x^3 + 40x^2 - 7x + 1.$
Wir betrachten den Zahlkörper $K := \mathbb{Q}[\varrho]$.

r \ N	10	10^8	Algorithmus
2	29,4 min 0,4 min	38,3 min 1,3 min	modular
4	39,6 min 2,5 min	49,6 min 10,1 min	modular
6	52,2 min 12,3 min	62,5 min 51,9 min	modular
8	56,1 min 92,7 min	67,2 min 408 min	modular
10	75,9 min 21,53 h	87,5 min	modular

Bei kleinen Beispielen benötigt der Algorithmus MatDetMod einen großen Teil der Gesamtlaufzeit für die Berechnung der Schranke. Deshalb ist in diesem Fall der nichtmodulare Algorithmus schneller.
Ab etwa $r = 8$ ist der modulare Algorithmus MatDetMod schneller. Schon ab $r = 12$ ist der modulare Algorithmus um ein Vielfaches effizienter als der nichtmodulare. Teilweise liegen die Laufzeiten des modularen Algorithmus im Sekundenbereich, während die entsprechenden Laufzeiten des nichtmodularen Algorithmus im Minuten- und Stundenbereich liegen.
Bei Fehlen von Vergleichswerten wurde die Berechnung mit dem nichtmodularen Algorithmus nach frühestens zwei Stunden abgebrochen.

2. Multiplikation algebraischer Zahlen

Generalvoraussetzung: Es seien K_i, $0 \leq i \leq r$ algebraische Zahlkörper und die Erweiterung K_i/K_{i-1} habe den Grad n_i, $1 \leq i \leq r$. In jedem algebraischen Zahlkörper K_i, $1 \leq i \leq r$ betrachten wir eine feste K_{i-1}-Basis $b_1^{(i)}, \ldots, b_{n_i}^{(i)}$ von K_i für $1 \leq i \leq r$.
K_0 sei eine absolute Erweiterung vom Grad n_0 und $\mathfrak{o}_0$ sei eine Ordnung von K_0 mit $\mathbb{Z}$-Basis $b_1^{(0)}, \ldots, b_{n_0}^{(0)}$.
Weiterhin sei $A(K_r)$ die nach Definition 2.66 zu K_r gehörige Algebra und $b_1, \ldots, b_t$ die zu $A(K_r)$ gehörende $\mathbb{Z}$-Basis. Nach Definition von $A(K_r)$ gilt dabei $t = \prod_{i=0}^{r} n_i$.

2.1. Der nichtmodulare Algorithmus. Hier sind zwei Algorithmen angegeben, die das Produkt zweier Elemente aus K_r berechnen. Der erst Algorithmus gilt allgemein. Der zweite Algorithmus gilt, falls die Basis $b_1^{(r)}, \ldots, b_{n_r}^{(r)}$ eine Potenzbasis ist.

Der erste Algorithmus benutzt die Multiplikationstabelle $(\Gamma_{i,j,k}^{K_r})_{i,j,k \in \{1,\ldots,n\}}$. Seien $\alpha, \beta \in K_r$. Das Produkt $\gamma = \alpha\beta \in K_r$, wobei

$$\alpha = \sum_{i=1}^{n_r} \alpha_i b_i^{(r)} \text{ und } \beta = \sum_{i=1}^{n_r} \beta_i b_i^{(r)}$$

ist, wird mit Hilfe der Multiplikationstabelle berechnet:

$$\gamma = \sum_{i,j,k=1}^{n_r} \alpha_i \beta_j \Gamma_{i,j,k}^{\mathfrak{o}_r} b_k^{(r)}$$

Sei $\gamma = \sum_{i=1}^{n_r} \gamma_i b_i = \alpha\beta$. Der folgende Algorithmus berechnet γ. Dabei nutzen wir die Tatsache aus, daß $\Gamma_{i,j,k}^{K_r} = \Gamma_{j,i,k}^{K_r}$ gilt.

ALGORITHMUS 3.10. (Mult: Multiplikation zweier algebraischer Zahlen)

Input: $\alpha_1, \ldots, \alpha_{n_r}$ *und* $\beta_1, \ldots, \beta_{n_r}$, *wobei* $\alpha = \sum_{i=1}^{n_r} \alpha_i b_i$ $\beta = \sum_{i=1}^{n_r} \beta_i b_i$
Output: $\gamma_1, \ldots, \gamma_{n_r}$ *mit* $\sum_{k=1}^{n_r} \gamma_k b_k = \alpha\beta$

(1) *Für* $i = 1, \ldots, n_r$:
(2) $\quad \gamma_i \leftarrow 0$

(3) *Für* $i = 1, \ldots, n_r$:
(4) *Für* $j = 1, \ldots, i-1$:
(5) $\text{temp}_1 \leftarrow \alpha_i \beta_j$
(6) $\text{temp}_2 \leftarrow \alpha_j \beta_i$
(7) $\text{temp}_3 \leftarrow temp_1 + temp_2$
(8) *Für* $k = 1, \ldots, n_r$:
(9) $\gamma_k \leftarrow \gamma_k + \text{temp}_3 \Gamma^{\mathfrak{o}_r}_{i,j,k}$
(10) $\text{temp}_4 \leftarrow \alpha_i \beta_i$
(11) *Für* $k = 1, \ldots, n_r$:
(12) $\gamma_k \leftarrow \gamma_k + \text{temp}_4 \Gamma^{\mathfrak{o}_r}_{i,i,k}$
(13) *Gib* $\gamma = \sum_{i=1}^{n_r} \gamma_i b_i$ *aus. ENDE.*

SATZ 3.11. *Algorithmus* 3.10 *benötigt für die Multiplikation zweier Elemente aus* K_0 $\frac{1}{2}n_r^3 + \frac{3}{2}n_r^2 = \frac{n_r^3}{2} + O(n_r^2)$ *Multiplikationen in* $\mathbb{Q}$ *und für die Multiplikation zweier Elemente aus* K_r *die gleiche Anzahl Multiplikationen in* K_{r-1} *für* $r \geq 1$.

Beweis: Die Zeilen (5) und (6) werden jeweils $\frac{n_r(n_r-1)}{2}$, Zeile (9) wird $\frac{n_r^2(n_r-1)}{2}$, Zeile (10) wird n_r mal und Zeile (11) wird n_r^2 mal durchlaufen. □

Ist die Basis $b_1^{(r)}, \ldots, b_{n_r}^{(r)}$ eine Potenzbasis, d.h. $b_i^{(r)} = \varrho^{i-1}$ für $1 \leq i \leq n_r$, so sollte die Multiplikation mit Hilfe des unten angegebenen Algorithmus ausgeführt werden. Er ist schneller als Algorithmus 3.10, da wir die einfache Gestalt der Multiplikationstabelle ausnutzen. Dies zeigt auch ein Vergleich von Satz 3.11 und Satz 3.13. Die Multiplikation mit Algorithmus 3.12 entspricht einer Multiplikation von Polynomen mit einer Reduktion modulo dem Minimalpolynom von ϱ.

ALGORITHMUS 3.12. (Multiplikation zweier algebraischer Zahlen aus K_r. Dabei ist $1, \varrho^1, \ldots, \varrho^{n_r-1}$ eine K_{r-1}-Basis von K_r und das $p(x) = \sum_{i=0}^{n_r} p_i x^i$ Minimalpolynom von ϱ)

Input: $\alpha_0, \ldots, \alpha_{n_r-1}$ *und* $\beta_0, \ldots, \beta_{n_r-1}$, *wobei* $\alpha = \sum_{i=0}^{n_r-1} \alpha_i \varrho^i$ $\beta = \sum_{i=0}^{n_r-1} \beta_i \varrho^i$
Output: $\gamma_0, \ldots, \gamma_{n_r-1}$ *mit* $\sum_{k=0}^{n_r-1} \gamma_k \varrho^k = \alpha\beta$

(1) *Für* $i = 0, \ldots, n_r - 1$:
(2) $c_i \leftarrow 0$

(3) $l \leftarrow n_r - 1$
(4) bound $\leftarrow n_r$
(5) *Für* $k = 2n_r - 2, \ldots, n_r$:
(6) bound $\leftarrow$ bound $- 1$
(7) *Für* $i = n_r - 1, \ldots,$ bound :
(8) $\gamma_l \leftarrow \gamma_l + \alpha_i \beta_{k-i}$
(9) mult $\leftarrow -\gamma_l$
(10) $m \leftarrow n_r - 1$
(11) *Für* $i = n_r - 1, \ldots, 1$:
(12) $\gamma_l \leftarrow \gamma_{l-1} + p_m$mult
(13) $l \leftarrow l - 1$
(14) $m \leftarrow m - 1$
(15) $c_l \leftarrow p_m$mult
(16) $l \leftarrow l + n_r - 1$
(17) *Für* $k = n_r - 1, \ldots, 0$:
(18) *Für* $i = k, \ldots, 0$:
(19) $\gamma_l \leftarrow \gamma_l + \alpha_i \beta_{k-i}$
(20) $l \leftarrow l - 1$
(21) *Gib* $\gamma = \sum_{i=0}^{n_r-1} \gamma_i \varrho_i$ *aus. ENDE.*

SATZ 3.13. *Algorithmus* 3.12 *benötigt für die Multiplikation zweier Elemente aus* K_0 $2n_r^2 - n_r = 2n_r^2 + O(n_r)$ *Multiplikationen in* $\mathbb{Q}$ *und für die Multiplikation zweier Elemente aus* K_r *die gleiche Anzahl Multiplikationen in* K_{r-1} *für* $r \geq 1$.

Beweis: Zeile (8) wird $\frac{n_r(n_r-1)}{2}$ mal, Zeile (12) wird $(n_r-1)^2$ mal, Zeile (15) wird $n_r - 1$ mal und Zeile (19) wird $\frac{n_r(n_r+1)}{2}$ mal durchlaufen. □

2.2. Der modulare Algorithmus. Der Kern des hier vorgestellten modularen Algorithmus entspricht dem Schema von Algorithmus 2.27 in Version 2. Dabei wird das modulare Verfahren auf die zu K_r gehörige Algebra angewendet (siehe Definition 2.66).

A. Schönhage beschäftigte sich in [Sch66] mit der Multiplikation von großen ganzen Zahlen aus $\mathbb{Z}$. Den dort beschriebenen Algorithmus werden wir auf alge-

braische Zahlen übertragen. Ebenfalls werden wir die rekursive Schachtelung des modularen Verfahren übernehmen.

Voraussetzung:
Die zu K_r gehörende Algebra $A(K_r)$ schreiben wir auch als $(A, +, *)$.

BEZEICHNUNG 3.14. *Wir sagen, daß $\alpha \in A(K_r)$ ein l-Bit Element ist, falls* $\mathrm{MAX}(\alpha) < 2^l$.

Seien $a, b \in K_r$ zwei Elemente, welche wir multiplizieren wollen. Das modulare Verfahren wenden wir auf $(A, +, *)$ an.

SATZ 3.15. *Seien $d_a = \mathrm{Nen}(a), d_b = \mathrm{Nen}(b) \in \mathbb{N}$ die nach 2.62 definierten Nenner von a bzw. b. Dann gilt für $\alpha := d_a a, \beta := d_b b$:*

(1) $\alpha, \beta \in A(K_r)$ *und*
(2) $ab = \dfrac{\alpha * \beta}{d_a d_b \mathfrak{N}(K_r)}$.

Zur Definition von $\mathfrak{N}(K_r)$ siehe Definition 2.64.

Beweis: Aussage (1) ist klar nach der Definition von Nen in Definition 2.62 und der Definition von $A(K_r)$ in Definition 2.66.
Aussage (2) gilt nach der Definition von $*$ in Definition 2.66:

$$ab = \frac{\alpha\beta}{d_a d_b} = \frac{\alpha * \beta}{d_a d_b \mathfrak{N}(K_r)}$$

□

Nun werden die einzelnen Teilschritte des Verfahrens beschrieben. Diese entsprechen den Teilschritten im Schema 2.27.

Das Hauptproblem besteht darin, eine Variante des chinesischen Restsatzes zu finden, die mit möglichst wenig Multiplikationen auskommt. Sonst wird die Zeitersparnis, die durch die Berechnung in den Restklassenringen erzielt wurde, durch die Anwendung des chinesischen Restsatzes zunichte gemacht. Durch eine geschickte Wahl der Moduli können wir die einzelnen Teilschritte schnell ausführen.

2.2.1. Bestimmung der Schranke

Eine gemeinsame obere Schranke S für den Betrag der Koeffizienten c_k von $\gamma = \alpha\beta$ bezüglich der Basis $b_1, \ldots, b_t$ von A erhalten wir aus dem folgendem Satz.

Voraussetzung: Für den Rest des Kapitels sei

$$c_l := \max\left\{\mathrm{MAX}\left(\mathcal{N}(K_l)\Gamma_{i,j,k}^{\mathfrak{o}_l}\right) \mid 1 \leq i,j,k \leq n_l\right\} \in \mathbb{Z} \text{ für } 0 \leq l \leq r.$$

Zur Definition von $\mathcal{N}(K_l)$ siehe Definition 2.63.

Satz 3.16. *Seien $\alpha, \beta \in A = A(K_r)$. Dann ist*

$$S_r(\alpha,\beta) := \mathrm{MAX}(\alpha)\mathrm{MAX}(\beta)\prod_{l=0}^{r}(c_l n_l^2)^{2^{r-l}}$$

eine obere Schranke für die Koeffizienten von $\gamma = \alpha * \beta$ bezüglich der Basis von $A(K_r)$, d.h.

$$\mathrm{MAX}(\alpha * \beta) \leq S_r(\alpha,\beta)$$

Beweis: Der Beweis erfolgt mittels Induktion über r. Für $r = 0$ liegt der Fall einer absoluten Erweiterung vor und es gilt $A(\mathfrak{o}_0) = \mathfrak{o}_0$, und die zu A gehörende Basis ist die $\mathbb{Z}$-Basis $b_1^{(0)}, \ldots, b_n^{(0)}$ von $\mathfrak{o}_0$. Nach Definition 2.66, Bemerkung 2.65 und der Definition der Multiplikationstabelle gilt

$$\alpha * \beta = \mathfrak{N}(K_0)\alpha\beta = \alpha\beta = \sum_{k=1}^{n_0}\Big(\sum_{i,j=1}^{n_0}\alpha_i\beta_j\Gamma_{i,j,k}^{K_0}\Big)b_k.$$

Daraus folgt

$$\mathrm{MAX}(\alpha * \beta) \leq \mathrm{MAX}(\alpha)\mathrm{MAX}(\beta)c_0 n_0^2$$

Dies ist die Behauptung für r = 0.

Nun führen wir den Induktionsschritt durch. Seien $\alpha, \beta \in A(K_{r+1})$ mit

$$\alpha = \sum_{i=1}^{n_{r+1}}\alpha_i b_i^{(r+1)} \text{ und } \beta = \sum_{i=1}^{n_{r+1}}\beta_i b_i^{(r+1)}.$$

Dabei sind $\alpha_i, \beta_i \in A(K_r)$ für $1 \leq i \leq n_{r+1}$. Dann gilt:

$$\begin{aligned}
\alpha * \beta &= \mathfrak{N}(K_{r+1})\alpha\beta \\
&= \mathfrak{N}(K_{r+1}) \sum_{k=1}^{n_{r+1}} \left(\sum_{i,j=1}^{n_{r+1}} \alpha_i \beta_j \Gamma_{i,j,k}^{K_{r+1}} \right) b_k^{(r+1)} \\
&= \frac{\mathfrak{N}(K_{r+1})}{\mathfrak{N}^2(K_r)} \sum_{k=1}^{n_{r+1}} \left(\sum_{i,j=1}^{n_{r+1}} (\alpha_i * \beta_j) * \Gamma_{i,j,k}^{K_{r+1}} \right) b_k^{(r+1)} \\
&= \mathcal{N}(K_{r+1}) \sum_{k=1}^{n_{r+1}} \left(\sum_{i,j=1}^{n_{r+1}} (\alpha_i * \beta_j) * \Gamma_{i,j,k}^{K_{r+1}} \right) b_k^{(r+1)} \\
&= \sum_{k=1}^{n_{r+1}} \left(\sum_{i,j=1}^{n_{r+1}} (\alpha_i * \beta_j) * \left(\mathcal{N}(K_{r+1}) \Gamma_{i,j,k}^{K_{r+1}} \right) \right) b_k^{(r+1)}
\end{aligned}$$

Hieraus folgt mit Hilfe der Induktionsvoraussetzung:

$$\begin{aligned}
\mathrm{MAX}(\alpha * \beta) &\leq \max \left\{ S_r \left(\alpha_i * \beta_j, \mathcal{N}(K_{r+1}) \Gamma_{i,j,k}^{K_{r+1}} \right) \mid 1 \leq i,j,k \leq n_{r+1} \right\} n_{r+1}^2 \\
&\leq \max \left\{ \mathrm{MAX}(\alpha_i * \beta_j) \mathrm{MAX} \left(\mathcal{N}(K_{r+1}) \Gamma_{i,j,k}^{K_{r+1}} \right) \mid 1 \leq i,j,k \leq n_{r+1} \right\} n_{r+1}^2 \\
&\quad \prod_{l=0}^{r} (c_l n_l^2)^{2^{r-l}} \\
&\leq \max \left\{ \mathrm{MAX}(\alpha_i * \beta_j) \mid 1 \leq i,j \leq n_{r+1} \right\} \\
&\quad \max \left\{ \mathrm{MAX} \left(\mathcal{N}(K_{r+1}) \Gamma_{i,j,k}^{K_{r+1}} \right) \mid 1 \leq i,j,k \leq n_{r+1} \right\} n_{r+1}^2 \prod_{l=0}^{r} (c_l n_l^2)^{2^{r-l}} \\
&\leq \max \left\{ S_r(\alpha_i, \beta_j) \mid 1 \leq i,j \leq n_{r+1} \right\} c_{r+1} n_{r+1}^2 \prod_{l=0}^{r} (c_l n_l^2)^{2^{r-l}} \\
&\leq \max \left\{ \mathrm{MAX}(\alpha_i) \mathrm{MAX}(\beta_j) \mid 1 \leq i,j \leq n_{r+1} \right\} c_{r+1} n_{r+1}^2 \left(\prod_{l=0}^{r} (c_l n_l^2)^{2^{r-l}} \right)^2 \\
&\leq \mathrm{MAX}(\alpha) \mathrm{MAX}(\beta) \prod_{l=0}^{r+1} (c_l n_l^2)^{2^{r-l}} = S_{r+1}(\alpha, \beta).
\end{aligned}$$

□

2.2.2. Wahl der Moduli

Für die Implementierung auf einem Computer bieten sich als Moduli Ideale von

A der Form $(m_1A), \ldots, (m_sA)$ an, wobei

$$m_j = 2^{e_j} - 1,\ e_j \in \mathbb{N},\ j \in \{1, \ldots, s\}$$

gilt. Wir benötigen zunächst ein Kriterium für die Komaximalität der Ideale $(m_1), \ldots, (m_s)$ in $\mathbb{Z}$, also ein Kriterium für die Teilerfremdheit der $m_1, \ldots, m_s \in \mathbb{N}$.

SATZ 3.17. *Seien $e, f, g \in \mathbb{Z}$. Dann gilt $2^e \equiv 2^g \bmod(2^f - 1) \Leftrightarrow e \equiv g \bmod f$.*

Beweis: „$\Rightarrow$“: Wenn $2^e \equiv 2^g \bmod(2^f - 1)$, dann $2^{e \bmod f} \equiv 2^{g \bmod f} \bmod(2^f - 1)$. Dabei gilt $0 < 2^{e \bmod f}, 2^{g \bmod f} < 2^f - 1$. Dies geht nur für $e \bmod f = g \bmod f$.
„$\Leftarrow$“: Aus $e = g + kf$ folgt $2^e = 2^g(2^f)^k \equiv 2^g 1^k \bmod(2^f - 1)$. □

Als Folgerung aus Satz 3.17 erhalten wir das folgende Korollar:

KOROLLAR 3.18. *Seien $e, f \in \mathbb{Z}$. Dann gilt*

$$(2^e - 1) \bmod(2^f - 1) = 2^{e \bmod f} - 1.$$

SATZ 3.19. *Seien $e, f \in \mathbb{Z}$. Dann gilt* $\mathrm{ggT}(2^e - 1, 2^f - 1) = 2^{\mathrm{ggT}(e,f)} - 1$.

Beweis: Die Aussage erhalten wir mit dem Euklidischen Algorithmus und Satz 3.18. □

KOROLLAR 3.20. *Seien $e, f \in \mathbb{Z}$. Dann gilt*

$$\mathrm{ggT}(2^e - 1, 2^f - 1) = 1 \Leftrightarrow \mathrm{ggT}(e, f) = 1$$

Beweis: Direkte Konsequenz aus Satz 3.19. □

Wir werden als Exponenten e_j paarweise verschiedene Primzahlen p_j nehmen, welche aufeinanderfolgen:

$$m_j = 2^{p_j} - 1,\ j \in \{1, \ldots, s\}$$

Dann sind nach dem Korollar 3.20 die Ideale $(m_1), \ldots, (m_s)$ in $\mathbb{Z}$ paarweise komaximal und somit nach Satz 2.24 die Ideale $(m_1A), \ldots, (m_sA)$ in A nichttrivial und paarweise komaximal.

Dabei halten wir die Anzahl s der Moduli fest. Die Wahl des Parameters s wird unten besprochen.

Voraussetzung:
Für den Rest des Kapitels gelte $m_j := 2^{p_j} - 1$ für $1 \leq j \leq s$.

Nach Algorithmus 2.27 Version 2 genügt es, die Bedingung

$$\prod_{i=1}^{s} m_i > 2S_r(\alpha, \beta) \tag{3-6}$$

zu erfüllen. Nach Satz 3.16 gilt

$$S_r(\alpha, \beta) = \mathrm{MAX}(\alpha)\mathrm{MAX}(\beta) \prod_{l=0}^{r} (c_l n_l^2)^{2^{r-l}}. \tag{3-7}$$

Wir wählen s aufeinanderfolgende Primzahlen $p_1, \ldots, p_s$ minimal, so daß

$$\sum_{i=1}^{s} p_i \geq \sharp\mathrm{bit}(\mathrm{MAX}(\alpha)) + \sharp\mathrm{bit}(\mathrm{MAX}(\beta)) + 1 + (\sum_{l=0}^{r} 2^{r-l}\, \sharp\mathrm{bit}(c_l n_l^2)) + s \tag{3-8}$$

erfüllt ist.
Dabei bezeichnet $\sharp\mathrm{bit}(z)$ die Anzahl der für die Darstellung von $z \in \mathbb{N}$ in Binärdarstellung benötigten (Binär-) Stellen. Es gilt $\lceil \log_2(z) \rceil \leq \sharp\mathrm{bit}(z) = \lfloor \log_2(z) \rfloor + 1$.
Nach dem folgenden Lemma haben wir somit die Schrankenbedingung (3-6) erfüllt.

LEMMA 3.21. *Es gilt: Ungleichung* (3-8) *impliziert Ungleichung* (3-6).

Beweis: Mit (3-8) sowie $m_i = 2^{p_i} - 1$, $1 \leq i \leq s$ erhalten wir

$$\begin{aligned}
\prod_{i=1}^{s} m_i &= \prod_{i=1}^{s}(2^{p_i}-1) = \prod_{i=1}^{s} 2^{p_i}\frac{2^{p_i}-1}{2^{p_i}} > 2^{\sum_{i=1}^{s} p_i}\ 2^{-s} = 2^{\left(\sum_{i=1}^{s} p_i\right) - s} \\
&\geq 2^{\sharp\mathrm{bit}(\mathrm{MAX}(\alpha)) + \sharp\mathrm{bit}(\mathrm{MAX}(\beta)) + 1 + (\sum_{l=0}^{r} 2^{r-l}\,\sharp\mathrm{bit}(c_l n_l^2))} \\
&\geq 2^{\lceil \log_2 2S_r(\alpha,\beta)\rceil} \\
&\geq 2S_r(\alpha,\beta)
\end{aligned}$$

□

(3-8) ist die Bedingung, die beim ersten Aufruf von MultMod erfüllt wird. Bei dem von uns verwendeten rekursiven Verfahren ruft MultMod sich jedoch selber auf. Betrachten wir den Rekursionsschritt k. Wir haben bereits die Primzahlen $p_1^{(k)}, \ldots, p_s^{(k)}$ gewählt und rufen dann im folgenden Rekursionsschritt $k-1$ erneut für jedes $p_i^{(k)}$, $1 \leq i \leq j$ einmal MultMod auf. Dabei werden nun Elemente α, β übergeben, die vorher modulo $p_i^{(k)}$ für ein $p_i^{(k)} \in \{p_1^{(k)}, \ldots, p_s^{(k)}\}$ reduziert wurden. Von diesen Elementen wissen wir, daß es p_s-Bit Elemente sind, d.h. $\sharp\mathrm{bit}(\mathrm{MAX}(\alpha)) + \sharp\mathrm{bit}(\mathrm{MAX}(\beta)) \leq 2p_s$. Somit genügt es, im Falle eines rekursiven Aufrufes s aufeinanderfolgende Primzahlen $p_1, \ldots, p_s$ minimal zu wählen , so daß

$$\sum_{i=1}^{s} p_i^{(k-1)} \geq 2p_s^{(k)} + 1 + (\sum_{l=0}^{r} 2^{r-l}\,\sharp\mathrm{bit}(c_l n_l^2)) + s. \tag{3-9}$$

erfüllt ist.

Bei diesem rekursiven Verfahren erhalten wir eine Folge $p_s^{(0)}, p_s^{(1)}, p_s^{(2)}, \ldots$, wobei $p_s^{(k)}$ die größte Primzahl aus Rekursionsschritt k ist.
Für die Laufzeitabschätzung stellen wir bereits hier zwei Aussagen über das Wachstum der Folge $(p_s^{(k)})_{k\in\mathbb{N}_0}$ zusammen.

SATZ 3.22. *Es gilt* $p_s^{(k)} = O\left(\left(\frac{s}{2}\right)^k\right)$.

Beweis: Mit

$$\sum_{i=1}^{s} p_i^{(k-1)} \leq s p_s^{(k-1)}$$

folgt aus (3-9)

$$\begin{aligned} p_s^{(k)} &\leq \frac{s}{2} p_s^{(k-1)} - \frac{1}{2}(1 + (\sum_{l=0}^{r} 2^{r-l} \sharp\mathrm{bit}(c_l n_l^2)) + s) \\ &\leq \frac{s}{2} p_s^{(k-1)}. \end{aligned} \tag{3-10}$$

Hieraus folgt die Behauptung. □

Nachdem wir in Satz 3.22 eine obere Abschätzung von $p_s^{(k)}$ angegeben haben, geben wir nun eine untere Abschätzung an.

Satz 3.23. *Für alle $\epsilon > 0$ gibt es ein $d(= d(\epsilon)) > 0$ mit*

$$p_s^{(k)} > d\Big(\frac{s}{2} - \epsilon\Big)^k.$$

Beim Beweis benutzen wir die folgende Aussage über Primzahlen aus [Bun92, S. 283]:

Satz 3.24. *Sei p eine Primzahl und $\hat{p}$ die kleinste Primzahl echt größer als p ($\hat{p}$ folgt auf p). Dann gilt*

$$\hat{p} - p = O(p^{\frac{5}{8}}).$$

Korollar 3.25. *Es gibt eine positive Konstante $C \in \mathbb{R}$, so daß für alle Primzahlen p die Ungleichung*

$$\hat{p} - p < C\hat{p}^{\frac{5}{8}}$$

gilt.

Hier nun der Beweis von Satz 3.23:
Da in (3-9) $p_1^{(k)}, \ldots, p_s^{(k)}$ minimal gewählt wurden, gilt

$$\sum_{i=0}^{s-1} p_i^{(k-1)} \leq 2p_s^{(k)} + 1 + (\sum_{l=0}^{r} 2^{r-l} \sharp\mathrm{bit}(c_l n_l^2)) + s. \tag{3-11}$$

Dabei ist $p_0^{(k-1)}$ die größte Primzahl kleiner als $p_1^{(k-1)}$. Korollar 3.25 liefert

$$p_i^{(k-1)} - p_0^{(k-1)} \leq Cs\left(p_s^{(k)}\right)^{\frac{5}{8}} \text{ für } 0 \leq i \leq s-1. \tag{3-12}$$

Aus (3-11) und (3-12) erhalten wir

$$p_s^{(k)} \geq \frac{1}{2}\left(sp_s^{(k-1)} - Cs^2\left(p_s^{(k)}\right)^{\frac{5}{8}} - 1 - (\sum_{l=0}^{r} 2^{r-l}\,\sharp\mathrm{bit}(c_l n_l^2)) - s\right). \tag{3-13}$$

Es gilt $p_s^{(k)} > p_s^{(k-1)}$, da andernfalls das rekursive Verfahren keinen weiteren Rekursionsschritt eingeleitet hätte. Aus (3-13) erhalten wir somit

$$\lim_{k\to\infty} \frac{p_s^{(k+1)}}{p_s^{(k)}} = \frac{s}{2}.$$

Hieraus folgt die Behauptung. □

2.2.3. Abbilden in die Restklassenringe

Das Ideal $(m_j A)$ hat die Gestalt $(m_j A) = \{\sum_{i=1}^{t} z_i b_i | z_i \in (m_j)\}$. Das Abbilden von $\alpha = \sum_{i=1}^{t} \alpha_i b_i \in A$ nach $A/(m_j A)$ entspricht der Bestimmung des symmetrischen kanonischen Repräsentanten $\alpha \bmod s\, m_j$. Hierbei wurde Bezeichnung 2.25 verwendet.
Sei $\alpha = \sum_{i=1}^{t}, \alpha_i b_i\, \alpha_i \in \mathbb{Z}$ für $1 \leq i \leq t$. Die Bestimmung des symmetrischen kanonischen Repräsentanten von α können wir zurück führen auf die Bestimmung von t symmetrischen kanonischen Repräsentanten der ganzen Zahlen α_i.

Für die Bestimmung des symmetrischen kanonischen Repräsentanten $\alpha_i \bmod s\, m_j$ einer b-Bit Zahl α_i werden $O(b)$ Bitoperationen benötigt [Sch66, S. 190, Hilfssatz 3]. Dabei wird ein Verfahren benutzt, welches Schönhage als zyklische Addition bezeichnet. Es nutzt die Form der Moduli $m_j = 2^{p_j} - 1$ aus beruht auf dem folgenden Satz.

Satz 3.26. *$z \in \mathbb{Z}$ habe die Gestalt*

$$z = a_t B^t + a_{t-1} B^{t-1} + \cdots + a_1 B + a_0$$

mit $B = 2^{p_j}$ und $0 \leq a_i < 2^{p_j}$ für $i \in \{1, \ldots, t\}$. Dann gilt

$$z \equiv a_t + a_{t-1} + \cdots + a_1 + a_0 \bmod (2^{p_j} - 1).$$

Beweis: Es gilt $B = 2^{p_j} \equiv 1 \bmod (2^{p_j} - 1)$ □

Wir verwenden den Algorithmus Mods zur Berechnung des symmetrischen kanonischen Repräsentanten eines b-Bit Elementes $\alpha \in A$. Mods geht nach dem oben beschriebenen Verfahren vor und benötigt somit $O(tb)$ Bitoperationen zur Bestimmung von $\alpha \operatorname{mods} m_j$.

Betrachten wir nun den Rekursionsschritt k des rekursiven Algorithmus MultMod. Sei α ein $p_j^{(k+1)}$-Bit Element. Die Bestimmung von $\alpha \operatorname{mods} m_i$ für ein i mit $1 \leq i \leq s$ benötigt $O(p_j^{(k+1)}t)$ Bitoperationen. Nach (3-10) gilt $p_j^{(k+1)} \leq p_s^{(k+1)} \leq \frac{s}{2} p_s^{(k)}$. Somit erhalten wir den folgenden Satz.

BEMERKUNG 3.27. *Sei $\alpha \in A(K_r)$ ein $p_j^{(k+1)}$-Bit Element.* Mods *benötigt $O(p_s^{(k)}t)$ Bitoperationen für die Bestimmung von $\alpha \operatorname{mods} m_j$.*

2.2.4. Auswerten in den Restklassenringen

Seien $\alpha, \beta \in A(K_r)$. Zur Berechnung von $\alpha * \beta \operatorname{mods} m_j$ wird der Algorithmus MultRR benutzt. MultRR(α, β, m_j) liefert das Ergebnis $\alpha * \beta \operatorname{mods} m_j$ für $1 \leq j \leq s$ und benutzt dabei Varianten der Algorithmen 3.10 bzw. 3.12. Diese Varianten unterscheiden sich von Algorithmus 3.10 bzw. 3.12 nur dadurch, daß

(1) nach jeder erfolgten Multiplikation das Ergebnis mittels Mods$(., m_j)$ reduziert wird und
(2) statt der Multiplikationstabelle $(\Gamma_{i,j,k}^{K_r})_{1 \leq i,j,k \leq n_r}$ die Tabelle $(\mathfrak{N}(K_r)\Gamma_{i,j,k}^{K_r})_{1 \leq i,j,k, \leq n_r}$ verwendet wird.

Durch die Anwendung von Mods werden die Koeffizienten betragsmäßig klein gehalten. Somit sind die nachfolgenden Multiplikationen schneller ausführbar.

Sei $b_i^{(r)} = \varrho^{i-1}$ für $1 \leq i \leq n_r$. Die Variante von Algorithmus 3.12 wird nur verwendet, falls die Koeffizienten des Minimalpolynoms von ϱ Elemente aus $A(K_{r-1})$ sind.

Erstens sind die Koeffizienten von α, β bezüglich $b_1^{(r)}, \ldots, b_{n_r}^{(r)}$ Elemente aus $A(K_{r-1}), r \geq 1$. Zweitens ist $\mathfrak{N}(K_r)$ so gewählt, daß die Koeffizienten der

Einträge der Tabelle $(\mathfrak{N}(K_r)\Gamma_{i,j,k}^{K_r})_{1\leq i,j,k,\leq n_r}$ Elemente aus $A(K_{r-1})$ sind für $r \geq 1$. Wir erhalten somit aus Satz 3.11 den folgenden Satz.

SATZ 3.28. *Der Algorithmus* MultRR *benötigt für die Multiplikation zweier Elemente aus* $A(K_0)$ $\frac{1}{2}n_r^3 + \frac{3}{2}n_r^2 = \frac{n_r^3}{2} + O(n_r^2)$ *Multiplikationen in* $\mathbb{Z}$ *und für die Multiplikation zweier Elemente aus* $A(K_r)$ *die gleiche Anzahl Multiplikationen in* $A(K_{r-1})$ *für* $r \geq 1$.

KOROLLAR 3.29. *Der Algorithmus* MultRR *benötigt für die Multiplikation zweier Elemente aus* $A(K_r)$ $O(t^3)$ *Multiplikationen in* $\mathbb{Z}$.

Beweis: Mit Hilfe einer Induktion über r folgt die Behauptung aus Satz 3.28 und $t = \prod_{i=1}^{r} n_i$. □

2.2.5. Chinesischer Restsatz

Der hier aufgeführte chinesische Restsatz wurde in seiner ursprünglichen Form 1958 von H.L. Garner [Gar58] für ganze Zahlen aus $\mathbb{Z}$ vorgeschlagen und auch von A. Schönhage in [Sch66] für die schnelle Multiplikation von ganzen Zahlen verwendet. Hier wird die verallgemeinerte Version für $A(K_r)$ vorgestellt und anschließend die Korrektheit bewiesen. Dabei verwenden wir die Schreibweise aus Bezeichnung 2.25. Sei im folgenden $m := \prod_{j=1}^{s} m_j$.

ALGORITHMUS 3.30. (CR: Chinesischer Restsatz)

Input: $\alpha_1, \ldots, \alpha_s \in A(K_r)$ *und paarweise teilerfremde* $m_1, \ldots, m_s \in \mathbb{Z}$
Output: α *mit* $\alpha \equiv \alpha_i \bmod m_i$ *für* $i \in \{1, .., s\}$ *und* $\mathrm{MAX}(\alpha) < \frac{m}{2}$

Schritt 1: *(Bestimmung der* β_j*)*
(1) $\beta_1 \leftarrow \alpha_1$
(2) *Für* $j = 1, \ldots, s$:
(3) $\quad$ *Für* $i = 2, \ldots, j$:
(4) $\quad\quad \beta_j \leftarrow \mathrm{Multc}((\alpha_j - \beta_{i-1}), i-1, j)$
(5) $\quad \beta_j \leftarrow \mathrm{Mods}(\beta_j, m_j)$
Schritt 2: *(Zusammensetzen)*
(6) $\alpha \leftarrow \beta_s$
(7) *Für* $j = s-1, \ldots, 1$:

(8) $\alpha \leftarrow \mathrm{Multm}(\alpha, j) + \beta_j$
(9) *Gib* α *aus. ENDE.*

Dabei liefert $\mathrm{Multc}(\alpha, i, j)$ als Ergebnis $\alpha c_{i,j} \bmod m_j$, wobei c_{ij} das Inverse von m_i modulo m_j ist. Das Inverse $c_{i,j}$ existiert, da die m_i, $i \in \{1, \ldots, s\}$ nach Voraussetzung paarweise teilerfremd sind.
$\mathrm{Multm}(\alpha, j)$ liefert als Ergebnis das Produkt von α mit der ganzen Zahl m_j. Diese beiden unten beschriebenen Prozeduren nutzen die spezielle Form der Moduli m_j aus, um eine Beschleunigung bezüglich der normalen Multiplikation zu erzielen. Wir beweisen nun die Korrektheit von Algorithmus 3.30 unter der Voraussetzung, daß die Algorithmen Multc und Multm korrekt sind.

SATZ 3.31. *(Korrektheit von Algorithmus* 3.30*)*
Das Ergebnis α *von Algorithmus* 3.30 *erfüllt*

(1) $\alpha \equiv \alpha_j \bmod m_j$ *für* $1 \le j \le s$ *und*
(2) $\mathrm{MAX}(\alpha) < \frac{1}{2}m$.

Beweis: Wir betrachten die Werte von β_j zu Beginn von Schritt 2.
(1) Es gilt:

$$\begin{aligned} m_{j-1} \cdot \cdots \cdot m_1 \beta_j &\equiv m_{j-1} \cdot \cdots \cdot m_1 (\cdots((\alpha_j - \beta_1)c_{1,j} - \beta_2)c_{2,j} - \cdots - \beta_{j-1}) \cdot \\ &\quad c_{j-1,j} \\ &\equiv m_{j-2} \cdot \cdots \cdot m_1 (\cdots(\alpha_i - \beta_1)c_{1,i} - \cdots - \beta_{j-2})c_{j-2,j} \\ &\quad -\beta_{j-1} m_{j-2} \cdot \cdots \cdot m_1 \\ &\vdots \\ &\equiv \alpha_j - \beta_1 - \beta_2 m_1 - \cdots - \beta_{j-1} m_{j-2} \cdot \cdots \cdot m_1 (\bmod m_j) \end{aligned}$$

Dabei haben wir $m_i c_{ij} \equiv 1 \bmod m_j$ verwendet. Es folgt

$$\alpha_i \equiv \beta_1 + \beta_2 m_1 + \beta_3 m_2 m_1 + \cdots + \beta_i m_{i-1} \cdot \cdots \cdot m_1 \equiv \alpha (\bmod m_j).$$

(2) Es gilt: $\mathrm{MAX}(\beta_j) \leq \frac{m_j-1}{2}$, da zuvor $\beta_j \leftarrow \mathrm{Mods}(\beta_j, m_j)$ ausgeführt wurde. Wir erhalten

$$\begin{aligned}\mathrm{MAX}(\alpha) &= \mathrm{MAX}(\sum_{j=1}^{s} \beta_j \prod_{i=1}^{j-1} m_i) \leq \sum_{j=1}^{s} \frac{m_j-1}{2} \prod_{i=1}^{j-1} m_i = \frac{1}{2} \sum_{j=1}^{s} \left[(\prod_{i=1}^{j} m_i) - (\prod_{i=1}^{j-1} m_i) \right] \\ &= \frac{1}{2} \prod_{i=1}^{s} -1 < \frac{1}{2} m.\end{aligned}$$

□

Sei $m_j = 2^{p_j} - 1$ für $j \in \{1, .., s\}$ mit $p_i < p_j$ für $i < j$. Würden wir in Zeile (4) von Algorithmus 3.30 eine normale Multiplikation eines p_s-Bit Elementes mit einer p_s-Bit Zahl ausführen, so würden wir $O(p_s^2 n_r)$ Bitoperationen benötigen. Wir zeigen unten, daß Multc mit $O(p_s log(p_s) n_r)$ Bitoperationen auskommt. Zeile (4) wird $\frac{s(s-1)}{2} = O(s^2)$ mal durchlaufen. Für die Ausführung von Schritt 1 benötigt Algorithmus 3.30 somit $O(s^2 p_s log(p_s) n_r)$ Bitoperationen.
Zeile (8) wird $s-1$ mal durchlaufen. Der Algorithmus Multm liefert das Ergebnis $\mathrm{Multm}(\alpha, j) = \alpha m_j = \alpha(2^{p_j} - 1) = \alpha 2^{p_j} - \alpha$. Die Multiplikation von α mit 2^{p_j} geschieht, indem wir in der Binärdarstellung der n Koeffizienten von α eine Verschiebung um p_j Bit nach links vornehmen. Wir können damit den Aufwand für Teil 2 großzügig mit $O(s^2 p_s n_r)$ Bitoperationen abschätzen. Damit erhalten wir:

SATZ 3.32. *Sei $m_j = 2^{p_j} - 1$ für $j \in \{1, .., s\}$ mit $p_i < p_j$ für $i < j$. Dann benötigt Algorithmus* 3.30 *$O(s^2 p_s log(p_s) n_r)$ Bitoperationen.*

Der Kern von Algorithmus Multc ist der folgende Satz.

SATZ 3.33. *Sei $c \in \mathbb{Z}$ das Inverse von $2^e - 1$ modulo $2^f - 1$. Dann gilt*

$$c = \left(\sum_{0 \leq i < b} 2^{ie} \right) mod(2^f - 1).$$

Dabei ist $b \in \mathbb{Z}$ die kleinste positive Zahl mit $be \equiv 1 \bmod f$.

Beweis: Sei $b \in \mathbb{Z}$ die kleinste positive Zahl mit $be \equiv 1 \bmod f$. Es gilt:

$$(1 + 2^e + \cdots + 2^{(b-1)e})(2^e - 1) \equiv 2^{eb} - 1 \equiv 2^1 - 1 = 1 \bmod (2^f - 1).$$

Das erste Kongruenzzeichen gilt nach der Formel für die geometrische Reihe, das zweite nach Satz 3.17. □

ALGORITHMUS 3.34. (Multc)

Input: $\alpha \in A(K_r), e, f \in \mathbb{Z}$
Output: $\beta = \alpha c_{ef}$, *wobei* $c_{ef}(2^e - 1) \equiv 1 \bmod (2^f - 1), 0 < c_{ef} < 2^e - 1$

(1) *Bestimme minimales* $b > 0$ *mit* $be \equiv 1 \bmod f$. *Sei* $b = (b_t \ldots b_1 b_0)_2$ *die Binärdarstellung.*
(2) $a_0 \leftarrow e$
(3) $d_0 \leftarrow b_0 e$
(4) $\alpha_0 \leftarrow \alpha$
(5) $\beta_0 \leftarrow b_0 \alpha$
(6) *Für* $i = 1, \ldots, t$
(7) $\quad a_i \leftarrow 2a_{i-1} \bmod f$
(8) $\quad d_i \leftarrow (d_{i-1} + b_i a_i) \bmod f$
(9) $\quad \alpha_i \leftarrow (\alpha_{i-1} + 2^{a_{i-1}} \alpha_{i-1}) \bmod (2^f - 1)$
(10) $\quad \beta_i \leftarrow (\beta_{i-1} + b_i 2^{d_{i-1}} \alpha_i) \bmod (2^f - 1)$
(11) $\beta \leftarrow \beta_t$
(12) *Gib* β *aus. ENDE.*

Mit Hilfe von Satz 3.33 folgt die Korrektheit mittels Induktion über t [Knu81, S. 289].

Die Bestimmung von b benötigt $O(\log_2(f)^2)$ Bitoperationen [Coh93, S. 13]. Die Schleife wird $t = \log_2(b) = O(\log_2(f))$ mal durchlaufen. Ein Schleifendurchlauf benötigt für Zeile (7) $O(\log_2(f))$, für Zeile (8) $O(\log_2(f))$, für Zeile (9) $O(n_r f)$, für Zeile (10) $O(n_r f)$ und somit insgesamt $O(n_r f)$ Bitoperationen. Mit $O((\log_2(f))^2) + O(n_r f)O(\log_2(f)) = O(f \log_2(f) n_r)$. erhalten wir:

SATZ 3.35. *Algorithmus* 3.34 *benötigt* $O(f \log_2(f) n_r)$ *Bitoperationen.*

2.2.6. Rücktransformation

Die Rücktransformation entspricht der Bestimmung des symmetrischen kanonischen Repräsentanten. Da der chinesische Restsatz nach Satz 3.31 bereits den symmetrischen kanonischen Repräsentanten zurückliefert, ist in diesem Falle nichts mehr zu tun.

2.2.7. Der Algorithmus MultMod

Nun geben wir den rekursiven Algorithmus an. Ein rekursiver Aufruf findet in Schritt 3 nur dann statt, wenn die in Schritt 1 berechnete Schranke S die Bedingung $S >$ bound erfüllt. Der rekursive Aufruf von MultMod soll nur dann erfolgen, wenn MultMod schneller als die nichtmodulare Multiplikation ist. In den Beispielrechnungen im nächsten Abschnitt wurde stets der Wert bound $= 4000$ gewählt.

ALGORITHMUS 3.36. (MultMod: Multiplikation von algebraischen Zahlen mit Hilfe der modularen Methode)

Input: a, b, *wobei* $a = \frac{1}{d_a}\sum_{j=1}^{t}\alpha_j b_j, b = \frac{1}{d_b}\sum_{j=1}^{t}\beta_j b_j$
Bei einem rekursiven Aufruf sind a, b $\tilde{p}_s$*-Bit Elemente und* $\tilde{p}_s$ *wird übergeben.*

Output: $c = ab$

Schritt 0: $\alpha \leftarrow \sum_{j=1}^{t}\alpha_j b_j, \beta \leftarrow \sum_{j=1}^{t}\beta_j b_j$
$d_c \leftarrow d_a d_b \mathfrak{N}(o_r)$

Schritt 1: *(Bestimmung der Schranke und Wahl der Moduli)*
Beim ersten Aufruf:
$B \leftarrow \sharp\mathrm{bit}(\mathrm{MAX}(\alpha)) + \sharp\mathrm{bit}(\mathrm{MAX}(\beta)) + 1 + (\sum_{l=0}^{r} 2^{r-l}\,\sharp\mathrm{bit}(c_l n_l^2)) + s$
Sonst (rekursiver Aufruf):
$B \leftarrow 2\tilde{p_s} + 1 + (\sum_{l=0}^{r} 2^{r-l}\,\sharp\mathrm{bit}(c_l n_l^2)) + s$
Wähle s *aufeinander folgende ungerade Primzahlen* $p_1, \ldots, p_s$, *so daß* $\sum_{i=1}^{s} p_i \geq B$.
Für $j = 1, \ldots, s$:
$m_j \leftarrow 2^{p_j} - 1$

Schritt 2: *(Abbilden in die Restklassenringe)*

Für $j = 1, \ldots, s$*:*
$w_{1,j} \leftarrow \mathrm{Mods}(\alpha, m_j)$
$w_{2,j} \leftarrow \mathrm{Mods}(\beta, m_j)$

Schritt 3: *(Auswerten in den Restklassenringen)*
Für $j = 1, \ldots, s$*:*
wenn $B >$ bound*, dann*
$\gamma_j \leftarrow \mathrm{MultMod}(w_{1,j}, w_{2,j}, p_s)$ *(Bei rekursivem Aufruf wird aktuelles* p_s *als dritter Parameter übergeben)*
$\gamma_j \leftarrow \mathrm{Mods}(\gamma_j, m_j)$
sonst
$\gamma_j \leftarrow \mathrm{MultRR}(w_{1,j}, w_{2,j}, m_j)$

Schritte 4,5: *(Chinesischer Restsatz, Rücktransformation)*
Berechne $\gamma \leftarrow \mathrm{CR}(\gamma_1, \ldots, \gamma_s, m_1, \ldots, m_s)$.

Schritt 6: $c \leftarrow \frac{\gamma}{d_c}$
Gib c *aus. ENDE.*

2.2.8. Korrektheit

Der Algorithmus 3.36 entspricht der Vorgehensweise von Algorithmus 2.27 und ist somit nach Satz 2.28 korrekt.

2.2.9. Laufzeitverhalten

Wir bezeichnen mit $time(b)$ die Anzahl der Bitoperationen, die höchstens notwendig sind, um mit MultMod zwei b-Bit-Elemente aus K_r zu multiplizieren.
Wir nehmen an, daß wir insgesamt $\tilde{k}$ Rekursionsschritte benötigen. Der Algorithmus MultMod ruft in Schritt 3 sich selbst s mal auf, falls $S >$ bound. Diese Aufrufe benötigen $s \cdot time(p_s^{(\tilde{k})})$ Bitoperationen. Für Schritt 4 und 5 reichen nach Satz 3.32 $O(p_s^{(\tilde{k})} \log_2(p_s^{(\tilde{k})}) s^2 t)$ Bitoperationen. Schritt 2 benötigt nach Bemerkung 3.27 $O(s p_s^{(\tilde{k})})$ Bitoperationen. Mit Hilfe von 3.22 erhalten wir mit

geeignetem positivem $c \in \mathbb{R}$:

$$time(b) \leq s \cdot time(p_s^{(\tilde{k})}) + O(p_s^{(\tilde{k})} \log_2(p_s^{(\tilde{k})}) s^2 t) \leq s \cdot time(p_s^{(\tilde{k})}) + c \left(\frac{s}{2}\right)^{\tilde{k}} \tilde{k} s^2 t.$$

Es folgt

$$\frac{time(b)}{s^{\tilde{k}}} \leq \frac{time(p_s^{(\tilde{k})})}{s^{(\tilde{k}-1)}} + \frac{c\tilde{k}s^2 t}{2^{\tilde{k}}} \leq \cdots \leq time(p_s^{(0)}) + cs^2 t \sum_{k=1}^{\infty} \frac{k}{2^k} =: \tilde{c}(t).$$

Dabei ist die unendliche Summe konvergent. Nach Korollar 3.29 gilt $\tilde{c}(t) = O(t^3)$. Satz 3.23 liefert uns $b^{\log_{\frac{s}{2}-\epsilon} s} \geq d^{\log_{\frac{s}{2}-\epsilon} s} s^{\tilde{k}}$. Unter Verwendung dieser Ungleichung erhalten wir

$$time(b) \leq \tilde{c}(t) s^{\tilde{k}} \leq \frac{\tilde{c}(t)}{d^{\log_{\frac{s}{2}-\epsilon} s}} b^{\log_{\frac{s}{2}-\epsilon} s}.$$

Wir erhalten somit als Ergebnis:

SATZ 3.37. *Für die Multiplikation zweier b-Bit Elemente aus K_r benötigt Algorithmus* MultMod *(Algorithmus* 3.36*) bei Verwendung von s Moduli*

$$time(b) = O(t^3 b^{\log_{\frac{s}{2}-\epsilon} s})$$

Bitoperationen. Dabei ist $\epsilon > 0$ beliebig.

Asymptotisch wäre eine große Anzahl s von Moduli wünschenswert. In der Praxis jedoch steigt mit s nach Satz 3.32 die für den chinesischen Restsatz benötigte Zeit an. Dies hat zur Folge, daß das modulare Verfahren erst bei sehr großen Koeffizienten schneller als das nichtmodulare Verfahren wird. In MultMod wurde $s = 10$ fest gewählt. Mit dieser Wahl sind auch die im nächsten Abschnitt aufgeführten Beispielrechnungen durchgeführt. Für $s = 10$ gilt $\log_{\frac{s}{2}} s \approx 1,43$.

2.3. Vergleich. In diesem Abschnitt werden einige Beispielrechnungen aufgeführt, die mit dem Computeralgebrasystem KANT V4 in der Oberfläche KASH berechnet wurden. Alle Rechnungen wurden auf einem HP9000/735s mit 160MB Speicher unter dem Betriebssystem HP–UX 9.04 ausgeführt.

In den Tabellenköpfen werden die folgenden Bezeichnungen verwendet:

N	für die maximale Anzahl der Dezimalstellen der Koeffizienten der beiden Faktoren,
T(MultMod)	für die von MultMod benötigte Zeit zur Multiplikation der beiden Faktoren und
T(Mult)	für die von den (nichtmodularen) in KANT implementierten Algorithmus 3.10 bzw. 3.12 benötigte Zeit zur Multiplikation der beiden Faktoren.

Sei ϱ eine Nullstelle des Polynoms
$f_0(x) = x^4 + 95147171x^3 + 73416722x^2 + 58392756x + 28295709$.
Wir betrachten den Zahlkörper $K_0 := \mathbb{Q}[\varrho]$ mit der $\mathbb{Q}$-Basis $1, \varrho, \ldots, \varrho^3$.

N	1187	1472	1672	2157	2642	3369	3853	4580
T(MultMod) in s	0,16	0,19	0,24	0,35	0,46	0,67	0,79	1,02
T(Mult) in s	0,16	0,23	0,32	0,52	0,78	1,28	1,76	2,39

5550	7004	9428	11851				
1,43	2,21	3,30	4,19				
3,45	5,48	9,94	15,95				

Sei ϱ eine Nullstelle des Polynoms
$f_0(x) = x^6 + 91943121x^5 + 92603883x^4 + 57293488x^3 + 2945097x^2 + 42904848x + 3091119$.
Wir betrachten den Zahlkörper $K_0 := \mathbb{Q}[\varrho]$ mit der $\mathbb{Q}$-Basis $1, \varrho, \ldots, \varrho^5$.

N	761	1162	1563	2366	3569	4773	5977	7180
T(MultMod) in s	0,16	0,29	0,43	0,76	1,51	2,37	3,14	3,92
T(Mult) in s	0,16	0,37	0,66	1,49	3,83	5,97	9,41	13,51

8785	10390	11994	14000	16006	18012		
5,06	6,32	7,69	9,61	11,52	13,17		
20,25	28,30	37,74	51,35	67,47	84,98		

Sei ϱ eine Nullstelle des Polynoms
$f_0(x) = x^8 + 35546019x^7 + 37900717x^6 + 80836557x^5 + 89917871x^4 + 73888743x^3 + 19567104x^2 + 74437689x + 94466888$.
Wir betrachten den Zahlkörper $K_0 := \mathbb{Q}[\varrho]$ mit der $\mathbb{Q}$-Basis $1, \varrho, \ldots, \varrho^7$.

N	479	745	1011	1542	2074	2605	3137	3669
T(MultMod) in s	0,12	0,26	0,37	0,71	1,03	1,48	2,05	2,69
T(Mult) in s	0,12	0,28	0,50	1,15	2,10	3,29	4,73	6,54

4200	4997	5795	6592	7389	8453	9516	10579
3,02	3,92	4,76	5,76	6,39	7,53	8,93	10,31
8,47	11,96	16,17	20,96	26,04	34,09	43,29	53,75

Sei ϱ eine Nullstelle des Polynoms

$f_0(x) = x^{10} + 32062788x^9 + 53232120x^8 + 25063586x^7 + 76311955x^6 + 36966596x^5 + 63305947x^4 + 91303332x^3 + 29329255x^2 + 21453819x + 58218522.$

Wir betrachten den Zahlkörper $K_0 := \mathbb{Q}[\varrho]$ mit der $\mathbb{Q}$-Basis $1, \varrho, \ldots, \varrho^9$.

N	476	611	747	1018	1290	1561	1968	2375
T(MultMod) in s	0,19	0,29	0,36	0,57	0,80	1,09	1,41	1,93
T(Mult) in s	0,18	0,30	0,44	0,81	1,29	1,87	2,96	4,38

2783	3190	5795					
2,52	3,17	4,76					
5,89	7,75	16,17					

Sei ϱ eine Nullstelle des Polynoms

$f_0(x) = x^{20} + 75761676x^{19} + 53902747x^{18} + 19078359x^{17} - 4286451x^{16} + 72831383x^{15} + 89381336x^{14} + 88199690x^{13} + 7606977x^{12} + 53330311x^{11} + 5967468x^{10} + 39897517x^9 + 45295734x^8 + 52125328x^7 + 20901972x^6 + 70952768x^5 + 46849766x^4 + 16566973x^3 + 61837857x^2 + 67981289x + 33943048.$

Wir betrachten den Zahlkörper $K_0 := \mathbb{Q}[\varrho]$ mit der $\mathbb{Q}$-Basis $1, \varrho, \ldots, \varrho^{19}$.

N	450	600	750	1050	1500	1950	2400	2850
T(MultMod) in s	0,66	1,01	1,39	2,08	3,72	4,68	6,93	9,44
T(Mult) in s	0,66	1,12	1,72	3,37	6,69	11,29	17,03	24,02

Sei ϱ eine Nullstelle des Polynoms

$$\begin{aligned} f_0(x) =& x^{40} + 11861152x^{39} + 32590607x^{38} + 74838005x^{37} + 49725069x^{36} + \\ & 40901085x^{35} + 49049917x^{34} - 7504920x^{33} + 64106306x^{32} + 1127091x^{31} + \\ & 96422312x^{30} + 17453796x^{29} - 449903x^{28} + 45934193x^{27} + 73260956x^{26} + \\ & 52842614x^{25} + 98200583x^{24} + 65216447x^{23} + 66447992x^{22} + 8853485x^{21} + \\ & 27686323x^{20} + 6484844x^{19} + 35726599x^{18} + 97548393x^{17} + 8380737x^{16} + \\ & 6328040x^{15} + 74653396x^{14} + 28070125x^{13} + 84788656x^{12} + 17213040x^{11} + \\ & 96016513x^{10} + 65801028x^9 + 55409637x^8 + 40456346x^7 + 17876130x^6 + \\ & 85672318x^5 + 85311774x^4 - 163061x^3 - 7660685x^2 + 58556164x + 71031965. \end{aligned}$$

Wir betrachten den Zahlkörper $K_0 := \mathbb{Q}[\varrho]$ mit der $\mathbb{Q}$-Basis $1, \varrho, \ldots, \varrho^{39}$.

N	553	830	1106	1658	2211	2763	3315	3868
T(MultMod) in s	4,00	6,16	9,34	18,01	24,02	35,75	51,89	60,20
T(Mult) in s	3,95	8,59	15,01	33,08	59,09	91,05	131,3	178,7

Sei ϱ eine Nullstelle des Polynoms

$$\begin{aligned} f_0(x) =& x^{80} + 2x^{79} + 2x^{78} - 4x^{77} + 9x^{76} - 4x^{75} + 8x^{74} + 5x^{72} - x^{71} + 8x^{69} - 8x^{68} + \\ & 8x^{67} + 4x^{66} - 9x^{65} + 6x^{64} + 7x^{63} - 10x^{62} - 2x^{61} - 4x^{60} + 3x^{59} - 10x^{58} - \\ & 4x^{57} + 7x^{56} + x^{55} + 7x^{54} - 2x^{53} - 2x^{52} - 9x^{51} - x^{50} - x^{49} - 2x^{48} - 8x^{47} + \\ & 4x^{46} - 6x^{45} - 5x^{44} - 4x^{43} + 3x^{42} - 5x^{41} + 9x^{40} + 2x^{39} - 3x^{38} - 5x^{37} + 2x^{36} + \\ & 2x^{35} + 5x^{34} + 8x^{33} - 4x^{32} + 6x^{31} + 4x^{30} - 5x^{29} - 9x^{27} - 9x^{26} - 6x^{25} - 10x^{24} + \\ & 4x^{23} - 10x^{22} - 8x^{21} + x^{20} - 7x^{19} - 8x^{18} + 10x^{17} - 3x^{16} - 8x^{15} - 6x^{14} - 3x^{13} - \\ & 5x^{11} + 7x^{10} + 4x^9 - 8x^8 + 7x^7 - 8x^6 + 2x^5 - 8x^4 - 8x^3 - 3x^2 - 10x + 1. \end{aligned}$$

Wir betrachten den Zahlkörper $K_0 := \mathbb{Q}[\varrho]$ mit der $\mathbb{Q}$-Basis $1, \varrho, \ldots, \varrho^{79}$.

N	156	344	532	719	907	1095	1283	1471
T(MultMod) in s	1,66	5,17	10,00	16,50	20,80	31,21	37,7	45,11
T(Mult) in s	1,65	6,54	14,47	26.03	41,09	59,11	81,05	105,5

1658	1846						
51,72	56,60						
134,3	165,7						

Sei ϱ eine Nullstelle des Polynoms
$f_0(x) = x^4 + 95147171x^3 + 73416722x^2 + 58392756x + 28295709$.
Wir betrachten den Zahlkörper $K_0 := \mathbb{Q}[\varrho]$ mit einer in KANT berechneten Ganzheitsbasis von K_0.

N	1187	1472	1672	2157	2642	3369	3853	4580
T(MultMod) in s	0,16	0,19	0,22	0,33	0,46	0,64	0,73	0,97
T(Mult) in s	0,16	0,22	0,30	0,50	0,74	1,24	1,57	2,22

5550	7004	9428	11851				
1,32	1,98	3,16	3,95				
3,29	5,20	9,35	14,91				

Sei ϱ eine Nullstelle des Polynoms
$f_0(x) = x^{10} - 3x^9 + 2x^8 + 4x^7 + 5x^6 + 4x^5 - 4x^4 + 12x^2 - 2x + 5$.
In $K_0 := \mathbb{Q}[\varrho]$ betrachten wir die $\mathbb{Q}$-Basis $1, \varrho, \ldots, \varrho^9$.
Sei ψ eine Nullstelle des Polynoms

$$\begin{aligned} f_1(x) =& x^{10} + (10\varrho^0 - 8\varrho^1 - 9\varrho^2 - 5\varrho^3 - 20\varrho^4 - 6\varrho^5 - 4\varrho^6 + 4\varrho^7 - 6\varrho^8 - 6\varrho^9)x^9 + \\ & (-14\varrho^0 - 7\varrho^1 + 7\varrho^3 + 18\varrho^4 - 7\varrho^5 - 3\varrho^6 + 11\varrho^7 + 2\varrho^8 - 11\varrho^9)x^8 + (-14\varrho^0 - \\ & 10\varrho^1 - 14\varrho^2 3\varrho^3 + \varrho^4 + 9\varrho^5 + 20\varrho^6 + 15\varrho^7 + 8\varrho^8 + 10\varrho^9)x^7 + (-8\varrho^0 + 10\varrho^1 + \\ & 6\varrho^3 + \varrho^4 - 8\varrho^5 + \varrho^6 - 8\varrho^7 + 8\varrho^8 + 12\varrho^9)x^6 + (-19\varrho^1 - 10\varrho^2 - 14\varrho^3 + 2\varrho^4 - \\ & 18\varrho^5 + 20\varrho^6 + 14\varrho^8 - 4\varrho^9)x^5 + (8\varrho^0 + 2\varrho^1 + 10\varrho^2 + 14\varrho^3 - 11\varrho^4 + 7\varrho^5 + 9\varrho^6 + \\ & 12\varrho^7 + 4\varrho^8 - 10\varrho^9)x^4 + (-1\varrho^0 + 20\varrho^1 + 18\varrho^2 + 13\varrho^3 - 1\varrho^4 - 16\varrho^6 + 13\varrho^7 + 2\varrho^8 + \\ & 10\varrho^9)x^3 + (13\varrho^0 - 12\varrho^1 - 5\varrho^2 + 14\varrho^3 + 16\varrho^4 - 16\varrho^5 - 12\varrho^6 + 15\varrho^7 - 13\varrho^8 + \\ & 9\varrho^9)x^2 + (-14\varrho^0 + 16\varrho^1 + 7\varrho^2 + 8\varrho^3 - 17\varrho^4 - 2\varrho^5 - 18\varrho^6 - 8\varrho^7 + 13\varrho^8 + 14\varrho^9)x + \\ & (-13\varrho^0 + 19\varrho^1 - 13\varrho^2 + 20\varrho^3 + 16\varrho^4 - 11\varrho^5 - 3\varrho^6 + 14\varrho^7 - 12\varrho^8 + 17\varrho^9). \end{aligned}$$

Wir betrachten den Zahlkörper $K_1 := K_0[\psi]$ mit der K_1-Basis $1, \psi, \ldots, \psi^9$.

N	223	357	490	622	757	890	1024	1157
T(MultMod) in s	5,77	10,24	15,10	20,57	27,44	33,49	42,47	51,84
T(Mult) in s	5,81	12,13	21,27	33,08	47,25	64,05	85,37	106,6

Sei ϱ eine Nullstelle des Polynoms
$f_0(x) = x^3 + x^2 - 13x + 36$.
In $K_0 := \mathbb{Q}[\varrho]$ betrachten wir die $\mathbb{Q}$-Basis $1, \varrho, \varrho^2$.
Sei ψ eine Nullstelle des Polynoms

$$\begin{aligned} f_1(x) =& x^{12} + (-108\varrho^0 + 270\varrho^1 + 24\varrho^2)x^{10} + (-14624\varrho^0 - 24744\varrho^1 - 4372\varrho^2)x^9 + \\ & (-631344\varrho^0 - 421176\varrho^1 - 48369\varrho^2)x^8 + (-174101760\varrho^0 - 3142080\varrho^1 + \\ & 6298560\varrho^2)x^7 + (-1051170392\varrho^0 + 732746246\varrho^1 + 186322312\varrho^2)x^6 + \\ & (-85808920416\varrho^0 + 10746990552\varrho^1 + 5544877344\varrho^2)x^5 + \\ & (4717709430588\varrho^0 + 2145887127201\varrho^1 + 240342270672\varrho^2)x^4 + \\ & (160143514883280\varrho^0 + 14458229480636\varrho^1 - 3476213893676\varrho^2)x^3 + \\ & (10741252484644056\varrho^0 - 1619403263824086\varrho^1 + 749027789533920\varrho^2)x^2 + \\ & (-62344580730746400\varrho^0 - 18875348023315080\varrho^1 - 1285873363213020\varrho^2)x^1 + \\ & (-179237747831041548\varrho^0 + 81710865349088363\varrho^1 + 23394835280694410\varrho^2). \end{aligned}$$

Wir betrachten den Zahlkörper $K_1 := K_0[\psi]$ mit der K_1-Basis $1, \psi, \ldots, \psi^{11}$.

N	328	411	494	576	824	991	1156	1405
T(MultMod) in s	1,34	2,06	2,41	3,01	5,21	6,12	7,93	10,84
T(Mult) in s	1,33	1,98	2,78	3,69	7,45	10,53	14,24	20,93

1736	2067	2398	2728	3060			
15,91	17,76	23,09	28,92	33,14			
31,59	44,62	59,84	77,55	96,85			

Der modulare Algorithmus MultMod zur Multiplikation algebraischer Zahlen liefert erst ab einigen tausend Dezimalstellen der Koeffizienten der Faktoren eine deutliche Laufzeitverbesserung gegenüber dem nichtmodularen Algorithmus.
Bei absoluten Erweiterungen von großem Grad und bei relativen Erweiterungen ist der modulare Algorithmus teilweise schon bei einigen hundert Dezimalstellen der Koeffizienten der Faktoren schneller als der nichtmodulare, da in diesen Fällen der Anteil der vom chinesischen Restsatz verwendeten Zeit an der Gesamtlaufzeit gering ist.

Bezeichnungen

In der vorliegenden Arbeit gelten die folgenden Bezeichnungen:

R	Ring
Ω	Menge von inneren Verknüpfungen
$T(\Omega)$	Menge der Ω-Terme
$(A,\Omega),(H,\Omega)$	algebraische Strukturen
Φ	Homomorphismus
E_Φ	die von Φ induzierte Kongruenzrelation (Bezeichnung 2.5)
$E_\mathfrak{a}$	die von dem Ideal $\mathfrak{a}$ induzierte Kongruenzrelation (Bezeichnung 2.8)
$R/\mathfrak{a}$	Restklassenring von R nach dem Ideal $\mathfrak{a}$
. mod .	positiver kanonischer Repräsentant (Bezeichnungen 2.20, 2.25)
. mods .	symmetrischer kanonischer Repräsentant (Bezeichnungen 2.20, 2.25)
MAX(.)	Maximum der Beträge der Koeffizienten (Bezeichnung 2.26)
$\mathfrak{a}$	Ideal
$\mathfrak{p}$	Primideal
p	Primzahl
ggT	größter gemeinsamer Teiler
kgV	kleinstes gemeinsamen Vielfaches
O	Landau-Symbol

$\mathbb{A}$	Menge der ganzen algebraischen Zahlen
K, K_0	algebraische Zahlkörper, gegeben als absolute Erweiterungen
K_i	algebraische Zahlkörper, gegeben als reative Erweiterungen für $i \geq 1$
$\mathfrak{o}_K$	Maximalordnung von K
$\mathfrak{o}_0$	Ordnung von K_0
n	Grad der Körpererweiterung $K/\mathbb{Q}$
n_i	Grad der Körpererweiterung K_i/K_{i-1} für $i \geq 1$
$b_1^{(i)}, ..., b_{n_i}^{(i)}$	eine K_{i-1}-Basis von K_i für $i \geq 1$
$A(K_r)$	die zu K_r gehörende Algebra (Definition 2.66)
$b_1, ..., b_t$	Basis von $A(K_r)$ (Definition 2.66)
Nen(.)	Nenner eines Elementes aus $A(K_r)$ bzgl. fester Basis (Definition 2.62)
$\mathcal{N}(.)$	Nenner einer Ordnung (Definition 2.63)
$\mathfrak{N}$	siehe Definition 2.64
$e(\mathfrak{p}/p\mathbb{Z})$	Verzweigungsindex von $\mathfrak{p}$ über $p\mathbb{Z}$
$f(\mathfrak{p}/p\mathbb{Z})$	Trägheitsgrad von $\mathfrak{p}$ über $p\mathbb{Z}$
σ_i	Konjugierten-Abbildungen
$\mathfrak{d}_K$	Diskriminante von K
$T_{K:\mathbb{Q}}$	absolute Spur
$N_{K:\mathbb{Q}}$	absolute Norm

Literaturverzeichnis

[Bun92] P. Bundschuh. *Einführung in die Zahlentheorie.* Springer, Berlin–Heidelberg–New York, 2 Auflage, 1992.

[Coh93] H. Cohen. *A Course in Computational Algebraic Number Theory.* Springer, 1993.

[DFK+96] Mario Daberkow, Claus Fieker, Jürgen Klüners, Michael Pohst, Katherine Roegner, und Klaus Wildanger. KANT V4. to appear in J. Symb. Comput., 1996.

[Dic52] L.E. Dickson. *History of the Theory of Numbers.* Chealsea, New York, 1952.

[Fri97] C. Friedrichs. Berechnungen relativer Ganzheitsbasen mit dem Round-2-Algorithmus. Diplomarbeit, Technische Universität Berlin, 1997.

[Gar58] H.L. Garner. The residue number system. *IRE Transactions*, **EC–8** (1958), 140–147.

[Gau] C.F. Gauss. *Werke*, Band 1 (Disquisitiones Arithmeticae). Deutsche Übersetzung der Disquisitiones Arithmeticae, G. Fleischer Jun., Leipzig 1801; hiervon Nachdruck: Springer, Berlin, 1986.

[Hup90] B. Huppert. *Angewandte lineare Algebra.* de Gruyter, Berlin–New York, 1990.

[Knu81] D.E. Knuth. *The Art of Computer Programming*, Band 2. Addison–Wesley Publishing Company, 1981.

[Kob87] N. Koblitz. *A Course in Number Theory and Cryptography.* Springer, New York, 1987.

[Lip81] J.D. Lipson. *Elements of Algebra and Algebraic Computing.* Benjamin/Cummings Publishing Company, Menlo Park, California–Reading, Massachusetts–Don Mills, Ontario, 1981.

[Mar95] D. A. Marcus. *Number Fields.* Springer–Verlag, New York–Berlin–Heidelberg, 1995.

[Mey80a] K. Meyberg. *Algebra 1.* Hanser, München–Wien, 1980.

[Mey80b] K. Meyberg. *Algebra 2.* Hanser, München–Wien, 1980.

[Mü94] A. Müller. Effiziente Algorithmen für Probleme der linearen Algebra über Z. Diplomarbeit, Universität des Saarlandes, 1994.

[Poh93] M.E. Pohst. *Computational Algebraic Number Theory.* Birkhäuser, 1993.

[PZ89] M.E. Pohst und H. Zassenhaus. *Algorithmic Algebraic Number Theory.* Encyclopaedia of mathematics and its applications. Cambridge University Press, 1989.

[RS62] J.B. Rosser und L. Schoenfeld. Approximate formulas for some functions of prime numbers. *Illinois Journal of Mathematics*, **6** (1962).

[Sch66] A. Schönhage. Multiplikation großer Zahlen. *Computing*, **1** (1966), 182–196.

Erklärung

Ich versichere, daß ich die vorliegende Arbeit über das Thema 'Modulare lineare Algebra über algebraischen Zahlkörpern' in der gesetzten Frist selbständig verfaßt und keine anderen Hilfsmittel als die angegebenen verwendet habe. Alle Stellen der Arbeit, die anderen Werken wörtlich oder sinngemäß entnommen sind, sind unter Angabe der Quelle als Entlehnung kenntlich gemacht. Die Abbildungen

sind von mir verfaßt, soweit nicht als Entlehnung gekennzeichnet.

Berlin, 1. Juli 1997 Harald Bartel