Einführung in das symbolische Rechnen

Vorlesungsmanuskript, Sommersemester 2018

Janko Böhm

14. Juli 2018

Inhaltsverzeichnis

Abbildungsverzeichnis

Symbolverzeichnis

1

Was ist symbolisches Rechnen?

Der Begriff symbolisches Rechnen bezieht sich im Allgemeinen auf alle Arten von exakten Berechnungen, die man mit Hilfe eines Computers durchführen kann. In letzter Konsequenz werden alle Rechnungen auf Rechnungen mit ganzen Zahlen zurückgeführt. Im Gegensatz zu numerischen Methoden, wird die floating point Einheit der CPU nicht benutzt. Falls Ihr Programm korrekt ist und auch die CPU fehlerfrei arbeitet, erhalten Sie mathematisch korrekte und präzise Antworten. Wenn Sie dagegen mit Floats arbeiten, erhalten Sie nur eine Approximation der Wahrheit, eventuell eine sehr schlechte, falls das Problem numerisch instabil ist. Dies passiert etwa dann, wenn in dem Algorithmus durch Zahlen nahe bei 0 dividiert wird:

$$\frac{1}{0.001} = 1000, \text{ aber } \frac{1}{-0.001} = -1000,$$

ein kleiner Fehler von 0.002 im Input kann also zu drastisch unterschiedlichen Outputwerten führen. Es gibt auch hybride Methoden, die symbolische und numerische Methoden kombinieren, um die Geschwindigkeit von numerischen Verfahren mit der Korrektheit von symbolischen Verfahren zu vereinigen.
Symbolisches Rechnen wird in praktisch allen Bereichen der Mathematik verwendet, zentrale Beispiele sind:

1) Die Zahlentheorie, die die Beziehung von Addition und Multplikation untersucht, und wiederum Anwendungen z.B. im RSA-Kryptosystem hat. Hier führen wir z.B. Rechnungen durch wie

$$2^3 \equiv 3 \bmod 5.$$

2) Die kommutative Algebra und algebraische Geometrie, die polynomiale Gleichungssysteme löst und Anwendung findet z.B. in der Robotik, Statistik und theoretischen Physik. Gegeben zwei Roboterarme der Länge 3 in der Ebene mit Kugelgelenk am Punkt $(0, -2)$ bzw. $(0, 2)$, kann man sich z.B. fragen, wie an welchen Punkten sich die Spitzen der Arme treffen können, siehe

Abbildung 1.1. Dazu müssen wir (nach dem Satz von Pythagoras) das Gleichungssystem

$$(x-2)^2 + y^2 - 3^2 = 0$$
$$(x+2)^2 + y^2 - 3^2 = 0$$

lösen. Die Lösungsmenge ist gegeben durch $(x, y) = (0, \pm\sqrt{5})$.

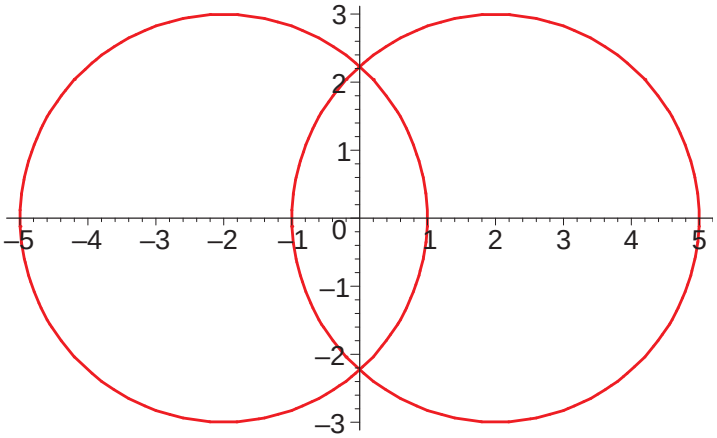


Abbildung 1.1: Die Treffpunkte von zwei Roboterarmen.

3) Die Gruppentheorie, die z.B. mit Permutationen rechnet, und zur Anwendung kommt wann immer ein mathematisches Problem Symmetrien besitzt. Sei etwa $\mathrm{Sym}\,(T)$ die Menge aller Symmetrien eines Tetraeders, wobei eine Symmetrie eine Bewegungen (d.h. eine abstandserhaltende Abbildung) ist, die den Tetraeder in sich selbst überführt. Eine solche Abbildung ist durch ihre Wirkung auf den Ecken des Tetraeders festgelegt, wir können $\mathrm{Sym}\,(T)$ also als Untergruppe von S_4 auffassen. Tatsächlich besitzt $\mathrm{Sym}\,(T)$ mit der Komposition als Verknüpfung die Struktur einer Gruppe: Das Hintereinanderausführen von zwei Symmetrien wieder eine Symmetrie ist und wir jede Symmetrie durch eine andere wieder rückgängig machen können. Zum Beispiel ist in $\mathrm{Sym}\,(T)$ die Drehsymmetrie um 120° gleich dem Produkt von zwei Spiegelungen, siehe Abbildung 1.2. entsprechend der Rechnung

$$(2,3) \circ (3,4) = (2,3,4)$$

in Zykelnotation für die Permutationen.

Die Spiegelung an der Ebene, aufgespannt durch eine Kante und den Mittelpunkt der gegenüberliegenden Seite, entspricht einer Transposition, z.B. die Spiegelung an der in Abbildung 1.3 eingezeichneten Ebene entspricht $(2,3)$. Da die S_4 von den Trans-

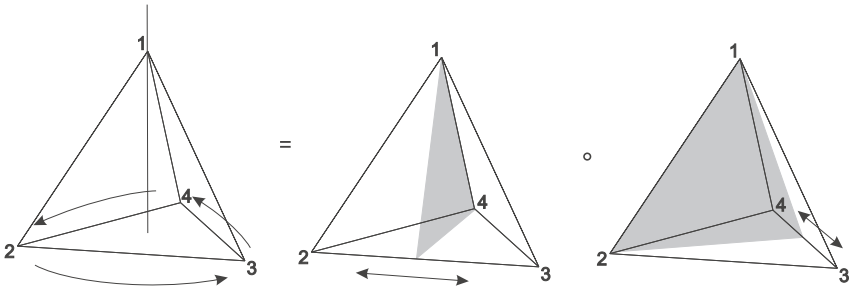


Abbildung 1.2: Komposition von zwei Symmetrien des Tetraeders

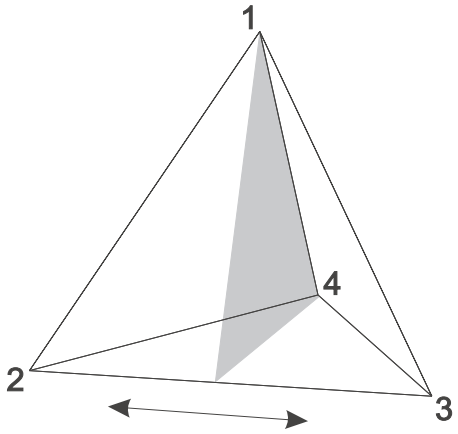


Abbildung 1.3: Spiegelsymmetrie $(2,3)$ des Tetraeders

positionen erzeugt wird, folgt:

$$S_4 = \langle (1,2),(1,3),(1,4),(2,3),(2,4),(3,4) \rangle \subset G \subset S_4$$

also

$$\mathrm{Sym}\,(T) = S_4.$$

Im Allgemeinen haben wir nicht soviel Glück, etwa ist die Symmetriegruppe $\mathrm{Sym}\,(O)$ des Oktaeders (Abbildung 1.4)) eine Untergruppe der S_6 (wieder durch Nummerieren der Ecken), jedoch sicher nicht die ganze S_6. Können Sie eine Symmetrie angeben, die nicht in $\mathrm{Sym}\,(O)$ liegt? Vom Standpunkt des symbolischen Rechnens stellt sich also z.B. die Frage, wie man für gegebene Elemente der S_n die von diesen Elementen erzeugte Untergruppe bestimmt.

4) Die konvexe Geometrie, die z.B. konvexe Hüllen von Punkten in einem Gitter untersucht, und Anwendungen hat z.B. in der Optimierung und der algebraischen Geometrie. Wieviele ganzzahlige

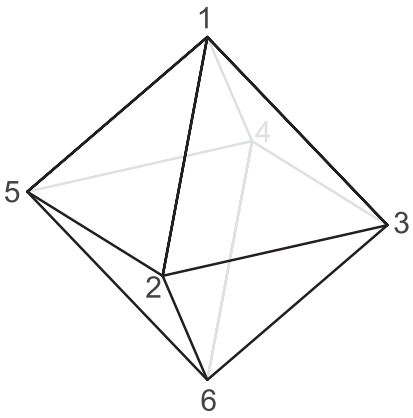


Abbildung 1.4: Oktaeder

Lösungen hat etwa das Ungleichungssystem

$$\begin{aligned} x &\geq -1 \\ y &\geq -1 \\ x + y &\leq 1 \end{aligned}$$

(siehe Abbildung 1.5)?

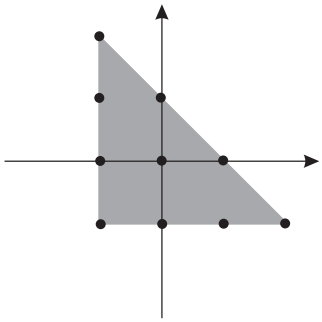


Abbildung 1.5: Polyeder

Beispiele von Open Source Computeralgebrasystemen: SINGULAR [5] und MACAULAY2 [7] sind die führenden Systeme im Bereich kommutative Algebra und algebraischen Geometrie, PARI/GP [14] in der Zahlentheorie, GAP [13] in der Gruppentheorie, und POLYMAKE [15] in der konvexen Geometrie. SINGULAR wird primär an der TU Kaiserslautern entwickelt. Ein neues Open Source Computeralgebrasystemen, das gerade, unter anderem in Kaiserslautern, entwickent wird ist OSCAR. Hier werden SINGULAR, GAP, POLYMAKE und Zahltentheoriekompententen in einem gemeinsamen System integriert. Die Programmiersprache JULIA [3] bildet sowohl die Entwicklungsumgebung

als auch das Userinterface. Sie zeichnet sich insbesondere durch eine Just-In-Time Compiler aus, der Usercode während der Laufzeit des Systems kompilieren kann. Wird also z.B. eine neue Funktion im Userinterface eingegeben, dann wird diese beim ersten Aufruf in Maschinencode übersetzt und läuft damit potentiell deutlich schneller als zeilenweise interpretierter Code. Der arithmetische Teil der Zahlentheoriekomponente von OSCAR wird in dem Paket NEMO entwickelt. Die Sprache JULIA und NEMO werden wir auch für Codebeispiele und Programmieraufgaben verwenden.

Es gibt aber viele weitere Computeralgebrasysteme, auch kommerzielle, wie MAGMA [4], MAPLE [11], und MATHEMATICA [16]. Die letzteren beiden sind sogenannte general purpose Computeralgebrasysteme, die eine breiten Bereich an mathematischen Themen abdecken, von der symbolischen Integration bis hin zu numerischen Methoden.

Wir werden uns hauptsächlich mit dem Lösen von allen Arten von Systemen aus polynomialen Gleichungen befassen, insbesondere von linearen Systemen über den ganzen Zahlen und Systemen aus Gleichungen höheren Grades über Körpern. In diesem Kontext müssen wir uns auch mit der Faktorisierung von ganzen Zahlen und Polynomen beschäftigen. Wir werden aber auch einen kurzen Blick auf andere Themenbereiche wie Gruppentheorie werfen.

Ein Spezialfall von polynomialen Gleichungssystemen sind lineare Gleichungssysteme. Der Gauß-Algorithmus transformiert ein multivariates lineares Gleichungssystem in Zeilenstufenform

$$\begin{array}{rcrcr} 2x & + & y & = & 1 \\ x & + & 2y & = & -1 \end{array} \quad \Longleftrightarrow \quad \begin{array}{rcrcr} 2x & + & y & = & 1 \\ -3x & & & = & -3 \end{array}$$

(siehe Abbildung 1.6), von dem wir dann direkt die Lösung ablesen können:

$$(x, y) = (1, -1).$$

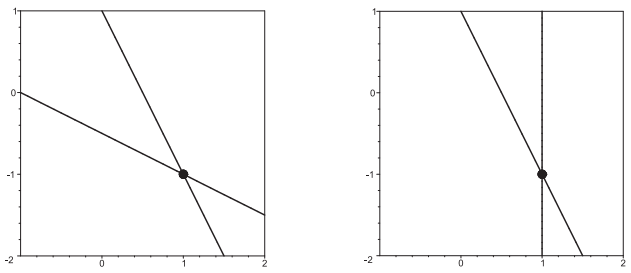


Abbildung 1.6: Gauß-Elimination für den Durchschnitt von zwei Geraden

Wie können wir diesen Ansatz auf Gleichungen höheren Grades erweitern? Das grundlegende Hilfsmittel ist hier der Buchberger-Algorithmus

zur Berechnung von Gröbnerbasen, der sowohl den Gauß-Algorithmus als auch den Euklidischen Algorithmus zur Berechnung des größten gemeinsamen Teilers verallgemeinert. Zum Beispiel transformiert er

$$\begin{matrix} 2x^2 - xy + 2y^2 - 2 = 0 \\ 2x^2 - 3xy + 3y^2 - 2 = 0 \end{matrix} \quad \Longleftrightarrow \quad \begin{matrix} 3y + 8x^3 - 8x = 0 \\ 4x^4 - 5x^2 + 1 = 0 \end{matrix}$$

Für die meisten Rechnungen werden wir hier direkt oder via JULIA das System SINGULAR verwenden, das an der TU Kaiserslautern entwickelt wird. Auf der Webseite

https://www.singular.uni-kl.de/

von SINGULAR gibt es auch ein leicht zu bedienendes Online-Interface, in dem man SINGULAR ohne Installation verwenden kann.

Beispiel 1.1 Die obige Gröbnerbasis-Rechnung können wir in SINGULAR mit dem folgenden Code durchführen:

```
ring R=0,(y,x),lp;
ideal I = 2*x^2-x*y+2*y^2-2, 2*x^2-3*x*y+3*y^2-2;
groebner(I);
_[1]=4x4-5x2+1
_[2]=3y+8x3-8x
```

Die `ring`-Definition spezifiziert den Primkörper über die Charakteristik (also 0 entspricht $\mathbb{Q}$), die Variablen, und eine Ordnung der Monome (lp). In Analogie zum Gauß-Algorithmus, kann man durch die Ordnung dem System mitteilen, in welcher Reihenfolge die Variablen eliminiert werden sollen (zum Beispiel, ob man eine rechte oder linke obere Dreiecksmatrix als Zeilenstufenform erhalten will). Ein Ideal repräsentiert ein System von polynomialen Gleichungen, und `std` steht für den Begriff **Standardbasis**, der in unserem Setup ein Synonym für Gröbnerbasen darstellt.

In unserem Beispiel transformiert der Gröbnerbasisalgorithmus den Durchschnitt von zwei Ellipsen in den Durchschnitt von 4 Geraden und einem kubischen Funktionengraphen, siehe Abbildung 1.7. Aus dem resultierenden System können wir die Lösungen $(x, y) = (\pm 1, 0), (\pm\frac{1}{2}, \pm 1)$ ablesen.

Beispiel 1.2 Diese Plots wurden mit dem universellen, kommerziellen Computeralgebrasystem MAPLE [11] erstellt:

```
with(plots):
p1:=implicitplot(2*x^2-x*y+2*y^2-2,x=-2..2,y=-2..2):
p2:=implicitplot(2*x^2-3*x*y+3*y^2-2,x=-2..2,y=-2..2):
display(p1,p2,view=[-2..2,-2..2],thickness=2);
p3:=implicitplot(4*x^4-5*x^2+1,x=-2..2,y=-2..2):
p4:=implicitplot(2*x^2-3*x*y+3*y^2-2,x=-2..2,y=-2..2):
display(p3,p4,view=[-2..2,-2..2],thickness=2);
```

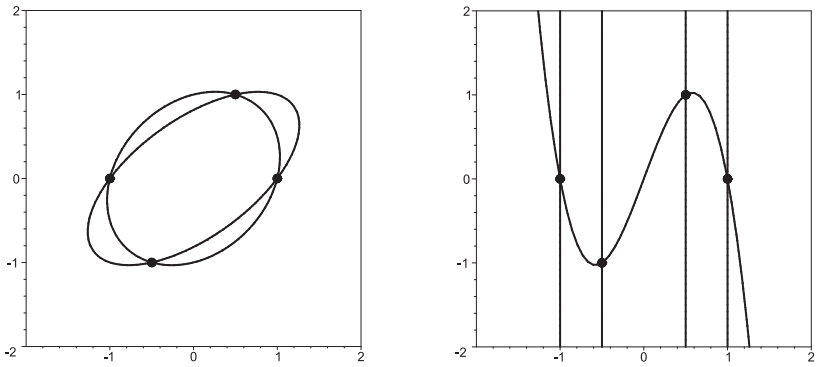


Abbildung 1.7: Buchberger-Algorithmus für den Schnitt von zwei Ellipsen

Natürlich muss die Lösungsmenge eines polynomialen Gleichungssystems nicht endlich sein. In diesem Fall ist es interessant, analog zu Vektorräumen einen Dimensionsbegriff zu entwickeln. Zum Beispiel die **Kummerquartik** in Abbildung 1.8 ist eine zweidimensionale Menge (eine **Fläche**) gegeben durch eine einzige Gleichung vom Grad 4 in 3 Variablen. Auf ähnliche Weise kann man die **Togliattiquintik** und die **Barthsextik** in den Abbildungen 1.9 und 1.10 konstruieren.

Diese Plots wurden mit dem Visualisierungsprogramm SURFER [8] erstellt, das im Rahmen des IMAGINARY Projekts entwickelt wurde, siehe

www.imaginary.de.

SURFER [8] ist ein leicht zu benutzendes Frontend für Raytracingprogramm SURF [6] zur Visualisierung von Flächen im 3-Raum. Mit diesem Programm können Sie leicht interessante Flächen im 3-Raum konstruieren, indem Sie die Gleichung angeben, probieren Sie es aus!

Abbildung 1.8: Kummerquartik

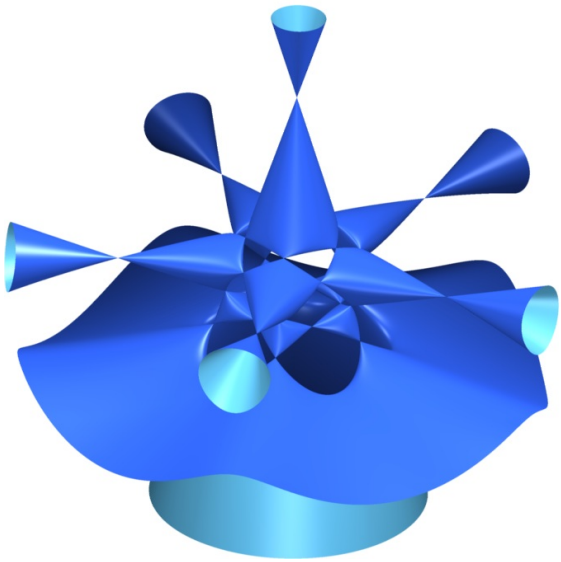


Abbildung 1.9: Togliattiquintik

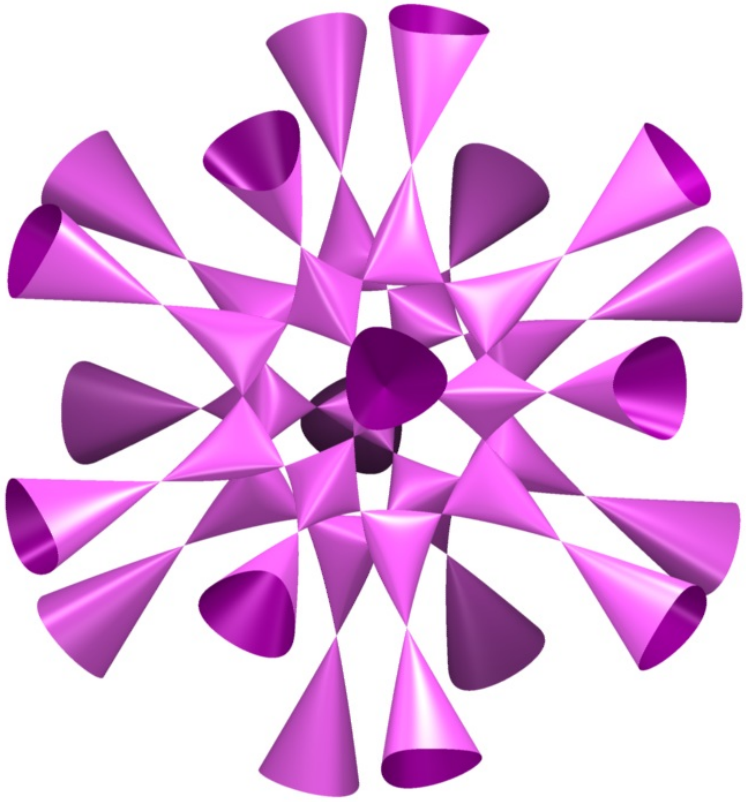


Abbildung 1.10: Barthsextik

2

Symbolisches Rechnen mit ganzen Zahlen

2.1 Die ganzen Zahlen

Die **natürlichen Zahlen** sind

$$\mathbb{N} = \{1, 2, 3, ...\}$$

die **ganzen Zahlen**

$$\mathbb{Z} = \{..., -3, -2, -1, 0, 1, 2, 3, ...\}$$

und wir schreiben $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$.

Auf den natürlichen Zahlen $\mathbb{N}_0 = \{0, 1, 2, 3, ...\}$ gibt es Verknüpfungen + und $\cdot$, die dem Assoziativgesetz

$$a + (b + c) = (a + b) + c$$
$$a \cdot (b \cdot c) = (a \cdot b) \cdot c$$

Kommutativgesetz

$$a + b = b + a$$
$$a \cdot b = b \cdot a$$

und Distributivgesetz

$$a \cdot (b + c) = a \cdot b + a \cdot c$$

gehorchen für alle $a, b, c \in \mathbb{N}_0$. Auf die axiomatische Definition der natürlichen Zahlen wollen wir hier nicht weiter eingehen. Als Übungsaufgabe informiere man sich in Buch oder Suchmaschine der Wahl über die Peano-Axiome.

In $\mathbb{N}_0$ gibt es keine Zahl a mit

$$1 + a = 0.$$

Anschaulich heißt das: Wir können zwar Guthaben auf einem Konto darstellen aber keine Schulden.

Aus den natürlichen Zahlen konstruiert man deshalb die ganzen Zahlen $\mathbb{Z}$ wie folgt:

Bemerkung 2.1.1 *Die Grundidee zur Konstruktion ist: Den Wert eines Kontos kann man als Differenz von Guthaben und Schulden schreiben. Verschiedene Tupel (Guthaben, Schulden) führen zu demselben Wert des Kontos, z.B.*

$$5 - 1 = 1000006 - 1000002$$

d.h. der Wert eines Kontos mit $5 €$ *Guthaben und* $1 €$ *Schulden entspricht einem Konto mit* $1000006 €$ *Guthaben und* $1000002 €$ *Schulden. Um den Wert zu repräsentieren, müssen wir also Äquivalenzklassen bezüglich einer geeigneten Äquivalenzrelation betrachten. Die beiden Konten in dem Beispiel haben denselben Wert, da*

$$5 + 1000002 = 1000006 + 1.$$

Man definiert also

$$\mathbb{Z} := (\mathbb{N}_0 \times \mathbb{N}_0) / \sim$$

mit der Äquivalenzrelation

$$(a, b) \sim (c, d) \Leftrightarrow a + d = b + c,$$

und die Äquivalenzklasse

$$[(a, b)] = \{(c, d) \mid (c, d) \sim (a, b)\}.$$

Wir stellen uns unter $[(a, b)]$ *die ganze Zahl* $a - b$ *vor. Dies motiviert die folgenden wohldefinierten Verknüpfungen* $+$ *und* $\cdot$ *auf* $\mathbb{Z}$

$$\begin{aligned} [(a, b)] + [(c, d)] &:= [(a + c, b + d)] \\ [(a, b)] \cdot [(c, d)] &:= [(a \cdot c + b \cdot d, a \cdot d + b \cdot c)], \end{aligned}$$

die dem Assoziativ-, Kommutativ- und Distributivgesetz gehorchen (siehe auch Übung 2.2). Es gilt dann

$$[(a, b)] + [(b, a)] = [(0, 0)]$$

für alle $[(a, b)] \in \mathbb{Z}$*, insbesondere*

$$[(1, 0)] + [(0, 1)] = [(0, 0)].$$

Weiter ist

$$\begin{aligned} [(0, 0)] + [(a, b)] &= [(a, b)] \\ [(1, 0)] \cdot [(a, b)] &= [(a, b)]. \end{aligned}$$

Eine Menge mit solchen Verknüpfungen nennt man kommutativen Ring mit 1*. Des Weiteren sind die ganzen Zahlen angeordnet durch die Totalordnung* $\leq$*.*

Siehe auch Übung 2.2.

Notation 2.1.2 *Jedes Element von $\mathbb{Z}$ hat einen Repräsentanten der Form $(a,0)$ oder $(0,a)$. Wir schreiben kurz*

$$a = [(a,0)]$$

Damit ist

$$-a = [(0,a)].$$

Wir werden später sehen, dass es durchaus üblich ist, bei der Darstellung von $\mathbb{Z}$ im Computer Repräsentanten zu wählen, die nicht von dieser Form sind.

Bemerkung 2.1.3 *Auf ähnliche Weise lässt sich wiederum $\mathbb{Q}$ aus $\mathbb{Z}$ konstruieren als*

$$\mathbb{Q} = (\mathbb{Z} \times \mathbb{Z}\backslash\{0\}) / \sim$$

mit der Äquivalenzrelation

$$(a,b) \sim (c,d) \Leftrightarrow ad = bc,$$

wobei wir die Äquivalenzklassen schreiben als

$$\frac{a}{b} := [(a,b)].$$

Die Verknüpfungen sind

$$\frac{a}{b} + \frac{c}{d} = \frac{ad+cb}{bd} \quad \text{und} \quad \frac{a}{b} \cdot \frac{c}{d} = \frac{ac}{bd}.$$

*Prüfen Sie, dass diese Verknüpfungen wohldefiniert (d.h. unabhängig von der Wahl der Repräsentanten der Äquivalenzklassen) sind. Mit dieser Konstruktion haben wir erreicht, dass jedes Element $\neq 0$ von $\mathbb{Q}$ bezüglich der Multiplikation ein Inverses besitzt, $\mathbb{Q}$ ist also ein **Körper**.*

Bemerkung 2.1.4 *Ist R allgemein ein kommutativer Ring mit 1, dann heißt $a \in R$ **Nullteiler** von R, wenn ein $0 \neq b \in R$ existiert mit*

$$a \cdot b = 0.$$

*Einen (kommutativen) Ring ohne nicht-triviale Nullteiler nennt man auch **Integritätsring**. Man kann dann genau wie in Bemerkung 2.1.3 durch Einführen von Brüchen jedes Element außer 0 bezüglich der Multiplikation invertierbar machen und erhält damit den Quotientenkörper $Q(R)$. Man beachte, dass wir dazu in den Formeln für die Verknüpfungen $+$ und $\cdot$ die Eigenschaft $b, d \neq 0 \Rightarrow b \cdot d \neq 0$ brauchen. Siehe dazu auch Aufgabe 2.3.*

In $\mathbb{Q}$ lässt sich also jede Zahl a durch jede Zahl $b \neq 0$ teilen. In vielen Problemen des täglichen Lebens und der Mathematik macht dies allerdings keinen Sinn, da die kleinste sinnvolle Einheit 1 ist. Wollen wir etwa 1000 Passagiere gleichmäßig auf 3 Flugzeuge verteilen, so ist $\frac{1000}{3}$ keine sinnvolle Lösung, sondern vielmehr

$$1000 = 3 \cdot 333 + 1.$$

Dies bezeichnet man als Division mit Rest (1 Passagier bleibt übrig).

Lemma 2.1.5 (Division mit Rest) *Sind $a, b \in \mathbb{Z}$, $b \neq 0$, dann gibt es $q, r \in \mathbb{Z}$ mit*

$$a = b \cdot q + r$$

mit $0 \leq r < |b|$.

Beispiel 2.1.6 *Teilen wir $a = 1000$ durch $b = 3$ erhalten wir*

$$1000 = 3 \cdot 333 + 1.$$

Zum Beweis von Lemma 2.1.5:

Beweis. Existenz: Ohne Einschränkung ist $b > 0$. Die Menge

$$\{w \in \mathbb{Z} \mid b \cdot w > a\} \neq \varnothing$$

hat ein kleinstes Element w. Setze dann

$$q := w - 1 \qquad r := a - qb.$$

Offenbar gilt dann $a = qb + r$, außerdem $qb + b > a$ also

$$r < b$$

und da w minimal gewählt war auch $bq \leq a$ also

$$r \geq 0.$$

Eindeutigkeit: Haben wir zwei solcher Darstellungen

$$b \cdot q_1 + r_1 = a = b \cdot q_2 + r_2$$

und ist OE $r_2 \leq r_1$, dann gilt

$$0 \leq \underbrace{r_1 - r_2}_{b \cdot (q_2 - q_1)} < |b|,$$

also $q_1 = q_2$ und somit $r_1 = r_2$. ∎

Der Beweis liefert einen expliziten (aber sehr ineffizienten) Algorithmus für die Division mit Rest. Praktisch geht man wie folgt vor:

Bemerkung 2.1.7 *Schulbuchdivision ohne Nachkommastellen bestimmt schrittweise die Dezimalstellen von q (beginnend mit der höchsten Dezimalstelle), gibt also einen Algorithmus zur Division mit Rest.*

Beispiel 2.1.8 *Für $a = 2225$ und $b = 7$ schreiben wir*

$$\begin{array}{rl} 2225 & = 7\cdot 300 + 7\cdot 10 + 7\cdot 7 + 6 = 7\cdot 317 + 6 \\ -2100 & \\ \hline 125 & \\ -70 & \\ \hline 55 & \\ -49 & \\ \hline 6 & \end{array}$$

also $q = 317$ und $r = 6$.
In JULIA *können wir die Division mit Rest durchführen mit:*

```
using Nemo
(q,r)=divrem(2225,7)
(317,6)
q*7+r
2225
```

Mit Hilfe der Division mit Rest können wir insbesondere Teilbarkeit algorithmisch entscheiden.

Definition 2.1.9 *Seien $a, b \in \mathbb{Z}$. Man sagt b **teilt** a*

$$b \mid a$$

wenn es ein $q \in \mathbb{Z}$ gibt mit $a = b \cdot q$.

Wie repräsentiert man Zahlen im Computer? Man rechnet typischerweise mit der Binärentwicklung, d.h. der Darstellung der Zahl durch eine Summe von Potenzen von der Basis $B = 2$. Allgemeiner hat man für beliebige Basis $B \geq 2$:

Definition und Satz 2.1.10 *Für jedes $B \in \mathbb{Z}, B \geq 2$ ist die Abbildung*

$$\begin{array}{rccc} \phi_{B,r}: & \{0,\ldots,B-1\}^r & \to & \{0,\ldots,B^r-1\} \\ & (a_{r-1},\ldots,a_0) & \mapsto & \sum_{i=0}^{r-1} a_i B^i \end{array}$$

*bijektiv. Für $n \in \{0,\ldots,B^r-1\}$ heißt $\phi_{B,r}^{-1}(n)$ die **B-adische Entwicklung mit r Stellen** von n.*

Beispiel 2.1.11 *Im Dezimalsystem ($B = 10$) gilt*

$$\phi_{10,3}((0,2,3)) = 0\cdot 10^2 + 2\cdot 10 + 3\cdot 10^0 = 23$$

und im Binärsystem ($B = 2$)

$$\phi_{2,8}((0,0,0,1,0,1,1,1)) = 1\cdot 2^4 + 0\cdot 2^3 + 1\cdot 2^2 + 1\cdot 2^1 + 1\cdot 2^0 = 23$$

Für gegebenes $(a_{r-1}, ..., a_0)$ ist es also leicht das Bild unter $\phi_{B,r}$ zu berechnen. Zum Beweis des Satzes verwenden wir einen Algorithmus, der das umgekehrte Problem löst und auf Division mit Rest basiert:

Beweis. Da

$$n = \sum_{i=0}^{r-1} a_i B^i \leq \sum_{i=0}^{r-1} (B-1) \cdot B^i = \sum_{i=0}^{r-1} (B^{i+1} - B^i) = B^r - 1$$

ist die Abbildung wohldefiniert.

Sukzessive Division mit Rest nach B beginnend mit $n_0 = n$ gibt

$$n_i = B \cdot n_{i+1} + a_i$$

mit $a_i \in \{0, ..., B-1\}$. Da $n_{i+1} \leq n_i$ für alle i, terminiert die Division nach $s < \infty$ Schritten mit $n_s = 0$, und sukzessives Einsetzen liefert

$$n = \sum_{i=0}^{s-1} a_i B^i.$$

Jede Zahl n hat also eine B-adische Entwicklung. Für die Surjektivität von $\phi_{B,r}$ müssen noch zeigen, dass wir mit $s \leq r$ Stellen auskommen: Angenommen $s > r$. Nach Definition von s haben wir $a_{s-1} = n_{s-1} \neq 0$. Somit ist $n \geq a_{s-1} B^{s-1} \geq B^{s-1} \geq B^r$, ein Widerspruch zu $n \leq B^r - 1$. Da sowohl Quelle und Ziel B^r Elemente haben, ist $\phi_{B,r}$ biijektiv. ■

Beispiel 2.1.12 *Für $n = 23$ ist*

$$\begin{aligned} 23 &= 10 \cdot 2 + 3 \\ 2 &= 0 \cdot 10 + 2 \end{aligned}$$

also

$$\phi_{10,3}^{-1}(n) = (0, 2, 3)$$

und

$$\begin{aligned} 23 &= 2 \cdot 11 + 1 \\ 11 &= 2 \cdot 5 + 1 \\ 5 &= 2 \cdot 2 + 1 \\ 2 &= 2 \cdot 1 + 0 \\ 1 &= 2 \cdot 0 + 1 \end{aligned}$$

und somit

$$\phi_{2,8}^{-1}(n) = (0, 0, 0, 1, 0, 1, 1, 1).$$

Ausführlich geschrieben haben wir

$$\begin{aligned} 23 &= 2 \cdot 11 + 1 \\ &= 2 \cdot (2 \cdot 5 + 1) + 1 \\ &= 2 \cdot (2 \cdot (2 \cdot 2 + 1) + 1) + 1 \\ &= 2 \cdot (2 \cdot (2 \cdot (2 \cdot 1 + 0) + 1) + 1) + 1 \\ &= 2 \cdot (2 \cdot (2 \cdot (2 \cdot (2 \cdot 0 + 1) + 0) + 1) + 1) + 1 \\ &= 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 \end{aligned}$$

Man beachte noch: Für $r' > r$ erhalten wir bis auf führende Nullen dieselbe Entwicklung von n. Heutige Hardware verwendet üblicherweise je nach Anwendungsfall $B = 2$ und $r = 16, 32, 64$. Addition und Multiplikation von B-adischen Darstellungen kann man in genau derselben Weise, wie in der Schule für $B = 10$ gelernt, durchführen. Erhalten wir durch Addition oder Multiplikation zweier Zahlen in $\mathrm{Bild}(\phi_{2,r})$ eine Zahl, die sich nicht mehr mit r Bits darstellen lässt (d.h. nicht im Bild von $\phi_{2,r}$ liegt), so spricht man von einem **arithmetischen Überlauf**.

In Software kann man natürlich beliebiges B und r emulieren. Für **arbitrary precision integers** verwendet man typischerweise $B = 2^{64}$ und beliebiges r, also Listen mit r Einträgen aus 64-bit Zahlen (wobei man die Länge r und eventuell das Vorzeichen in einem weiteren Eintrag speichert).

Bemerkung 2.1.13 *Die einfachste Möglichkeit, um negative Zahlen darzustellen ist ein zusätzliches Vorzeichenbit. Dieses entscheidet, ob $a \in \mathbb{N}_0$ für $[(a, 0)]$ oder $[(0, a)]$ steht. Allerdings hat dann $0 = [(0, 0)]$ zwei Darstellungen als 0 und -0. Um dies zu vermeiden, verwendet man in der Informatik typischerweise das sogenannte* ***Zweierkomplement****. Hier wird zu einer r-Bit-Zahl $(a_{r-1}, ..., a_0)$ ein weiteres Bit hinzugefügt, dem man den Wert -2^r zuordnet. Mit diesem Verfahren hat jede Zahl in $\{-2^r, ..., 0, ..., 2^r - 1\}$ eine eindeutige Darstellung, denn jede Zahl $0 \le n \le 2^r - 1$ hat eine eindeutige Binärdarstellung mit r Bits (siehe Satz 2.1.10), und jede Zahl $-2^r \le n \le -1$ lässt sich wiederum eindeutig schreiben als*

$$n = -2^r + m$$

mit einer Zahl $0 \le m \le 2^r - 1$.

Beispiel 2.1.14 *In der Zweierkomplementdarstellung mit $r = 7$ schreibt sich die größtmögliche positive Zahl 127 als*

$$(0, 1, 1, 1, 1, 1, 1, 1),$$

die 0 hat die Darstellung

$$(0, 0, 0, 0, 0, 0, 0, 0),$$

-1 erhalten wir als

$$(1, 1, 1, 1, 1, 1, 1, 1),$$

denn $-2^8 + 2^7 + ... + 2^1 + 1 = -1$, und die kleinstmögliche negative Zahl -128 hat die Darstellung

$$(1, 0, 0, 0, 0, 0, 0, 0).$$

Bemerkung 2.1.15 *Explizite Formeln zur Bestimmung der Zweierkomplementdarstellung: Nichtnegative Zahlen* $0 \leq n < 2^r$ *schreibt man*

$$n = -v \cdot 2^r + a_{r-1}2^{r-1} + \ldots + 2^1 a_1 + a_0$$

mit dem Vorzeichenbit $v = 0$ *und* $(a_{r-1}, \ldots, a_0) = \phi_{2,r}^{-1}(n)$.

Für negative Zahlen $-2^r \leq n < 0$ *gilt*

$$n = -v \cdot 2^r + \bar{a}_{r-1}2^{r-1} + \ldots + 2^1 \bar{a}_1 + \bar{a}_0$$

mit dem Vorzeichenbit $v = 1$ *und* $(a_{r-1}, \ldots, a_0) = \phi_{2,r}^{-1}(-n-1)$.

Dabei bezeichnet

$$\bar{a} = \begin{cases} 0 & \text{falls } a = 1 \\ 1 & \text{falls } a = 0 \end{cases}$$

das ***Bit-Komplement****.*

Beweis. Für negatives n gilt

$$\underbrace{(\overline{a_{r-1}}, \ldots, \overline{a_0})}_{2^r+n} + \overbrace{(a_{r-1}, \ldots, a_0)}^{-n-1} = \overbrace{(1, \ldots, 1)}^{2^r-1}$$

und

$$-2^r + (2^r + n) = n.$$

■

Bemerkung 2.1.16 *Als Äquivalenzklasse wird* $0 \leq n < 2^r$ *im Zweierkomplement also repräsentiert als*

$$n = [(n, 0)]$$

und $-2^r \leq n < 0$ *als*

$$n = [(2^r + n,\ 2^r)].$$

Man beachte, dass $2^r + n < 2^r$.

2.2 Euklidischer Algorithmus

In der Praxis führen iterierte Additionen und Multiplikationen mit rationalen Zahlen zu zunehmend unnötig anwachsenden Zählern und Nennern. Dieses Problem lässt sich beheben, indem wir den größten gemeinsamen Teiler aus dem Bruch kürzen.

Definition 2.2.1 *Sind* $a_1, \ldots, a_r \in \mathbb{Z}$, *dann heißt* $d \in \mathbb{N}$ ***größter gemeinsamer Teiler*** *von* $a_1, \ldots, a_r$, *geschrieben* $d = \mathrm{ggT}(a_1, \ldots, a_r)$, *wenn gilt*

1) $d \mid a_j \ \forall j = 1, ..., r$, *d.h.* d *ist ein Teiler von allen* a_j, *und*

2) ist $\tilde{d} \in R$ *ein Teiler aller* a_j, *d.h.* $\tilde{d} \mid a_j \ \forall j = 1, ..., r$, *dann gilt* $\tilde{d} \mid d$.

Gegeben $\frac{r}{s} \in \mathbb{Q}$ schreibe

$$r = r' \cdot \mathrm{ggT}(r, s) \qquad s = s' \cdot \mathrm{ggT}(r, s),$$

dann gilt

$$\frac{r}{s} = \frac{r'}{s'}.$$

Der größte gemeinsame Teiler kann mit dem Euklidischen Algorithmus berechnet werden:

Algorithmus 2.1 Euklidischer Algorithmus

Seien $a_1, a_2 \in \mathbb{Z} \setminus \{0\}$. *Dann terminiert sukzessive Division mit Rest*

$$\begin{aligned} a_1 &= q_1 a_2 + a_3 \\ &\vdots \\ a_j &= q_j a_{j+1} + a_{j+2} \\ &\vdots \\ a_{n-1} &= q_{n-1} a_n + 0 \end{aligned}$$

und

$$\mathrm{ggT}\,(a_1, a_2) = a_n$$

Rückwärtseinsetzen dieser Gleichungen

$$\begin{aligned} a_n &= a_{n-2} - q_{n-2} a_{n-1} \\ &\vdots \\ a_3 &= a_1 - q_1 a_2 \end{aligned}$$

liefert eine Darstellung

$$\mathrm{ggT}\,(a_1, a_2) = x \cdot a_1 + y \cdot a_2$$

mit $x, y \in \mathbb{Z}$.

Beweis. Es ist $|a_{i+1}| < |a_i|$ für $i \geq 2$ und somit muss nach endlich vielen Schritten $a_i = 0$ sein. Weiter ist a_n ein Teiler von a_{n-1}, also auch von $a_{n-2} = q_{n-2} a_{n-1} + a_n$ und induktiv von $a_{n-2}, ..., a_1$. Ist t ein Teiler von a_1 und a_2, dann auch von $a_3, ..., a_n$. ■

Beispiel 2.2.2 *Wir bestimmen den ggT von* 66 *und* 18 *mit Hilfe des Euklidischen Algorithmus, d.h. durch sukzessive Division mit Rest:*

$$\begin{aligned} 66 &= 3 \cdot 18 + 12 \\ 18 &= 1 \cdot 12 + 6 \\ 12 &= 2 \cdot 6 + 0 \end{aligned}$$

Somit ist $\mathrm{ggT}(66,18) = 6$*, denn von unten gelesen gilt*

$$6 \mid 12 \text{ also } 6 \mid 18 \text{ also } 6 \mid 66$$

und von oben gelesen, ist t *ein Teiler von* 66 *und* 18*, dann*

$$t \mid 12 \text{ also } t \mid 6.$$

Weiter erhalten wir eine Darstellung von $\mathrm{ggT}(36,15)$ *als* $\mathbb{Z}$*-Linearkombination von* 66 *und* 18

$$6 = 18 - 1 \cdot 12 = 18 - 1 \cdot (66 - 3 \cdot 18) = 4 \cdot 18 + (-1) \cdot 66.$$

In Julia *können wir den erweiterten Euklidischen Algorithmus durchführen mit:*

```
using Nemo
(g,x,y)=gcdx(66,18)
(6,-1,4)
x*66+y*18
6
```

In der gleichen Weise wie in $\mathbb{Z}$ können wir im Polynomring $K[x]$ in einer Variablen x, den Euklidischen Algorithmus zur Bestimmung des ggT durchführen. Dazu verwenden wir statt Division mit Rest die Polynomdivision. Allgemein bezeichnet man Ringe, in denen man eine Division mit Rest, und damit den Euklidischen Algorithmus durchführen kann, als Euklidische Ringe. Wesentlich ist hier, dass der Rest bezüglich einer geeigneten Kenngröße echt kleiner wird. In $\mathbb{Z}$ ist diese Kenngröße der Absolutbetrag, in $K[x]$ der Grad des Polynoms.

Definition 2.2.3 *Ein* ***Euklidischer Ring*** *ist ein Paar* (R,d) *aus einem Integritätsring* R *und einer Abbildung*

$$d : R\setminus\{0\} \longrightarrow \mathbb{N}_0$$

sodass für je zwei Elemente $a,b \in R\setminus\{0\}$ *Elemente* $g,r \in R$ *existieren mit*

1) $a = g \cdot b + r$ *und*

2) $r = 0$ *oder* $d(r) < d(b)$.

Wir bezeichnen dies als ***Division*** *von* a *durch* b ***mit Rest*** r*. Die Abbildung* d *heißt* ***Euklidische Norm****.*

Beispiel 2.2.4 *1) Der Ring der ganzen Zahlen* $\mathbb{Z}$ *ist euklidisch mit der Betragsabbildung*

$$d(n) = |n|$$

und der üblichen Division mit Rest zur Bestimmung von g *und* r*.*

2) Sei K ein Körper. Der Polynomring $R = K[x]$ in einer Variablen ein euklidischer Ring mit der Gradabbildung

$$d(f) = \deg(f)$$

und der üblichen Polynomdivision zur Berechnung von g und r.

Konkrete Beispiele:

Teilen wir $a = x^2 + \frac{1}{2}x + \frac{1}{2}$ durch $b = 2x - 1$ in $\mathbb{Q}[x]$ erhalten wir

$$\begin{aligned} x^2 + \tfrac{1}{2}x + \tfrac{1}{2} &= (\tfrac{1}{2}x) \cdot (2x-1) + \left(x + \tfrac{1}{2}\right) \\ &= \underbrace{(\tfrac{1}{2}x + \tfrac{1}{2})}_{g} \cdot (2x-1) + \underbrace{1}_{r} \end{aligned}$$

also $d(r) = 0 < 1 = d(b)$.

Teilen wir $a = x^n - 1$, $n \geq 1$ durch $b = x - 1$ erhalten wir

$$\begin{aligned} x^n - 1 &= x^{n-1} \cdot (x-1) + (x^{n-1} - 1) \\ &= (x^{n-1} + x^{n-2}) \cdot (x-1) + (x^{n-2} - 1) \\ &\vdots \qquad\qquad \vdots \\ &= \underbrace{(x^{n-1} + x^{n-2} + \ldots + x + 1)}_{g} \cdot (x-1) + \underbrace{0}_{r} \end{aligned}$$

In JULIA *können wir die Division mit Rest in $\mathbb{Q}[x]$ durchführen mit:*

```
using Nemo
R, x = PolynomialRing(QQ, "x");
(q,r)=divrem(x^2+1/2*x+1/2, 2*x-1)
(1/2*x+1/2, 1)
q*(1/2*x+1/2)+r
x^2+1/2*x+1/2
```

3) $\mathbb{Z}[x]$ und $K[x,y]$ mit K ein Körper sind keine euklidischen Ringe. Darauf werden wir noch in den Übungen zurückkommen.

4) Der Ring der Gaußschen Zahlen

$$R = \mathbb{Z}[i] = \{a_1 + i \cdot a_2 \mid a_1, a_2 \in \mathbb{Z}\} \subset \mathbb{C}$$

ist euklidisch mit

$$\begin{aligned} d: \quad R \setminus \{0\} &\to \mathbb{N}_0 \\ a + b \cdot i &\mapsto a^2 + b^2 \end{aligned}$$

In Übungsaufgabe 2.21 werden wir diskutieren, wie die Division mit Rest durchzuführen ist.

2.3 Primfaktorisierung

Definition 2.3.1 *Ein Element $p \in \mathbb{N}$, $p \geq 2$ heißt **Primzahl**, wenn aus $p = a \cdot b$, $a, b \in \mathbb{N}$ folgt $a = 1$ oder $b = 1$.*

Beispiel 2.3.2 $2, 3, 5, 7, 11, 13, 17, 19, 23...$ *Die Bestimmung aller Primzahlen bis zu einer gegebenen Schranke werden wir im nächsten Abschnitt behandeln.*

Satz 2.3.3 (Fundamentalsatz der Arithmetik) *Jede ganze Zahl $n \in \mathbb{Z} \setminus \{0, 1, -1\}$ hat eine eindeutige Darstellung*

$$n = \pm 1 \cdot p_1^{r_1} \cdot ... \cdot p_r^{r_r}$$

*mit Primzahlen $p_1 < ... < p_r$ und $r_i \in \mathbb{N}$. Die p_i heißen **Primfaktoren** von n.*

Beweis. Existenz der Primfaktorzerlegung mit Induktion nach n:

$n = 2$ ist eine Primzahl. Ist $n > 2$ und keine Primzahl, dann ist $n = a \cdot b$ mit $a, b \neq 1$. Da $a, b < n$, haben a und b nach Induktionsvoraussetzung Zerlegungen, und durch sortieren der Primfaktoren erhalten wir eine Primfaktorzerlegung von $n = a \cdot b$.

Eindeutigkeit mit Induktion nach n:

$n = 2$ ist klar. Sei $n > 2$ und

$$n = p_1 \cdot \ldots \cdot p_s = q_1 \cdot \ldots \cdot q_t$$

mit $p_1 \leq \ldots \leq p_s$ und $q_1 \leq \ldots \leq q_t$. Ist $s = 1$ oder $t = 1$, dann ist n prim, und die Behauptung ist klar. Seien also $s, t \geq 2$.

Ist $p_1 = q_1$ dann hat

$$p_2 \cdot \ldots \cdot p_s = q_2 \cdot \ldots \cdot q_t < n$$

nach Induktionsvoraussetzung eine eindeutige Primfaktorzerlegung und die Behauptung folgt.

Angenommen es wäre $p_1 < q_1$. Dann gilt

$$n > \underbrace{p_1 \cdot (p_2 \cdot \ldots \cdot p_s - q_2 \cdot \ldots \cdot q_t)}_{=:N_1} = \underbrace{(q_1 - p_1) \cdot q_2 \cdot \ldots \cdot q_t}_{=:N_2} \geq 2,$$

also hat $N_1 = N_2$ nach Induktionsvoraussetzung eine eindeutige Primfaktorzerlegung. Wegen $p_1 < q_1 \leq \ldots \leq q_t$ ist $p_1 \neq q_i$, und p_1 ist kein Teiler von $q_1 - p_1$, denn sonst würde p_1 auch q_1 teilen. Somit ist p_1 ein Primfaktor von N_1, jedoch keiner von N_2, ein Widerspruch. ■

Insbesondere ist eine gekürzte Darstellung einer rationalen Zahl $\frac{r}{s}$ eindeutig, wenn wir $s > 0$ fordern. Aus der Eindeutigkeit der Primfaktorzerlegung folgt weiter:

Corollar 2.3.4 (Euklids erster Satz) *Ist $p \in \mathbb{Z}$ prim und $a, b \in \mathbb{Z}$ mit $p \mid ab$, dann $p \mid a$ oder $p \mid b$.*

Corollar 2.3.5 (Euklids zweiter Satz) *Es gibt unendlich viele Primzahlen.*

Beweis. Sei $M = \{p_1, ..., p_r\}$ eine endliche Menge von Primzahlen. Wir zeigen, dass es eine Primzahl gibt, die nicht in M enthalten ist. Die Zahl $N = p_1 \cdot ... \cdot p_r + 1$ ist durch keine der Primzahlen p_i teilbar, denn sonst wäre auch 1 durch p_i teilbar. Ein Primfaktor p von N ist also eine Primzahl, die nicht in M liegt. ■

Wir können sogar den Anteil der Primzahlen unter allen ganzen Zahlen beschreiben. Dies ist wichtig, da z.B. in der Public-Key-Kryptographie große Primzahlen benötigt werden, diese also nicht zu selten und schwer zu finden sein sollten.

Satz 2.3.6 (Primzahlsatz) *Sei für $x \in \mathbb{R}_{>0}$*

$$\pi(x) = |\{p \leq x \mid p \in \mathbb{N} \text{ prim}\}|$$

dann gilt

$$\lim_{x \to \infty} \frac{\pi(x)}{\frac{x}{\ln(x)}} = 1$$

Ohne Beweis.

Beispiel 2.3.7 *Das folgende Programm (in der Syntax von* JULIA*) berechnet $\pi(x)$:*

```
using Nemo
function pi(x)
    p=ZZ(2)
    N=ZZ(0)
    while p<=x
        if isprime(p)
            N=N+1
        end
        p=p+1;
    end
    return N
end

pi(100000);
```

9592

Beispiel 2.3.8 *Mit Hilfe von* JULIA *können wir leicht überprüfen, dass die n-te Fermatzahl $F_n = 2^{2^n} + 1$ prim ist für $0 \leq n \leq 4$ und zusammengesetzt für $5 \leq n \leq 8$. Zum Beispiel:*

```
using Nemo
factor(ZZ(2)^(2^6)+1)
1 * 274177 * 67280421310721
```

Mit parallelem Rechnen kann man diese Tests noch etwas weiterführen. Dennoch ist es unbekannt, ob es mehr Fermat-Primzahlen gibt als die 5 bekannten. Siehe Übung 2.6.

2.4 Probedivision

Zunächst behandeln wir folgendes offensichtliche Primfaktorisierungsverfahren:

Algorithmus 2.2 Probedivision

Sei $n \in \mathbb{Z}$ zusammengesetzt. Für den kleinsten Primteiler p von n gilt

$$p \le m := \lfloor \sqrt{n} \rfloor$$

Kennen wir alle Primzahlen $p \le m$, dann testen wir $p \mid n$ mit Division mit Rest. Damit können wir eine gegebene Zahl n faktorisieren (und beliebig große Primzahlen finden).

Beweis. Ist $n = p \cdot c$ mit p prim und $p \le c$, dann $p^2 \le p \cdot c = n$. ∎

Beispiel 2.4.1 *Zum Faktorisieren von 234 mittels Probedivision testen wir zunächst, ob n durch eine Primzahl $p \le \lfloor \sqrt{234} \rfloor = 15$ teilbar ist. Wir finden*

$$234 = 2 \cdot 117.$$

Ist 117 nicht prim, so muss ein Primteiler $p \le \lfloor \sqrt{117} \rfloor = 10$ vorkommen, wir finden

$$117 = 3 \cdot 39.$$

Ist 39 nicht prim, so muss ein Primteiler $p \le \lfloor \sqrt{39} \rfloor = 6$ vorkommen, und wir finden

$$39 = 3 \cdot 13.$$

Schließlich ist 13 prim, denn 13 ist durch keine Primzahl $p \le \lfloor \sqrt{13} \rfloor = 3$ teilbar.

Die Probedivision erlaubt uns auch, alle Primzahlen $\le n$ induktiv aufzuzählen, denn kennen wir schon alle Primzahlen $p \le \lfloor \sqrt{n} \rfloor < n$, so können wir durch Faktorisieren entscheiden, ob n prim ist.

Beispiel 2.4.2 *Wir bestimmen alle Primzahlen* ≤ 11. *Für den kleinsten Primteiler von* n *gilt* $p \leq m$, *wir erhalten also:*

n	m		
2	1		$\Rightarrow 2$ *prim*
3	1		$\Rightarrow 3$ *prim*
4	2	$4 = 2 \cdot 2$	$\Rightarrow 4$ *nicht prim*
5	2	$2 \nmid 5$	$\Rightarrow 5$ *prim*
6	2	$6 = 2 \cdot 3$	$\Rightarrow 6$ *nicht prim*
7	2	$2 \nmid 7$	$\Rightarrow 7$ *prim*
8	2	$8 = 2 \cdot 4$	$\Rightarrow 8$ *nicht prim*
9	3	$9 = 3 \cdot 3$	$\Rightarrow 9$ *nicht prim*
10	3	$10 = 2 \cdot 5$	$\Rightarrow 10$ *nicht prim*
11	3	$2 \nmid 11$ *und* $3 \nmid 11$	$\Rightarrow 11$ *prim*

Praktisch geht man aber umgekehrt vor, und streicht Vielfache von schon bekannten Primzahlen:

Algorithmus 2.3 Sieb des Eratosthenes

Wir erhalten eine Liste aller Primzahlen kleiner gleich $N \in \mathbb{N}$ *wie folgt: Notiere alle Zahlen von* 2 *bis* n. *Beginnend mit* $p = 2$, *streiche alle* $a \cdot p$ *für* $a > 1$, *und fahre mit dem nächstgrößeren* p *in der Liste fort, das noch nicht gestrichen worden ist. Beachten Sie, dass* p *prim ist, da es kein Vielfaches einer kleineren Primzahl ist. Stoppe wenn* $p > \sqrt{n}$.

In Übung 2.11 behandeln wir das analoge Verfahren für univariate Polynome.

Beispiel 2.4.3 *Wir bestimmen alle Primzahlen* ≤ 15 *und geben in jedem Durchlauf die Liste aller* j *mit* $L_j = true$ *an*

2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	3		5		7		9		11		13		15
2	3		5		7				11		13		

Im ersten Schritt streichen wir alle Vielfachen von 2, *im zweiten Schritt alle Vielfachen von* 3. *Alle verbliebenen Zahlen sind prim, denn* $5 > \sqrt{15}$.

Für große Zahlen gibt es wesentlich effizientere Methoden als Probedivision, um einen Primteiler zu finden. Darauf werden wir später zurückkommen.

Primfaktorisierung ist ein sehr allgemeines Konzept. Neben den ganzen Zahlen haben zum Beispiel auch Polynomringe eine Primfaktorisierung. Ringe mit einer sinnvollen Primfaktorisierung bezeichnet man als **faktorielle Ringe**. Wir werden auf diese Klasse von Ringen später zurückkommen.

2.5 Der Chinesische Restsatz

2.5.1 Kongruenzen

Definition 2.5.1 *Sei $n \in \mathbb{N}$ und $a, b \in \mathbb{Z}$. Dann heißt a **kongruent** zu b modulo n*

$$a \equiv b \bmod n$$

wenn $n \mid (a-b)$.

Kongruent sein ist eine Äquivalenzrelation (Übung). Modulo zu n rechnen bedeutet nichts anderes als mit Restklassen

$$\overline{a} = a + n\mathbb{Z} = \{a, a+n, a-n, a+2n, a-2n, ...\} \subset \mathbb{Z}$$

in dem Ring $\mathbb{Z}/n$ zu rechnen, wobei für $\overline{a}, \overline{b} \in \mathbb{Z}/n$

$$\overline{a} + \overline{b} = \overline{a+b} \qquad \overline{a} \cdot \overline{b} = \overline{a \cdot b} \quad .$$

Die Multiplikation zum Beispiel ist wohldefiniert, denn

$$(a + k_1 \cdot n) \cdot (b + k_2 \cdot n) = a \cdot b + n \cdot (k_1 \cdot b + k_2 \cdot a + k_1 \cdot k_2 \cdot n) .$$

Multiplikation in Ringen der Form $\mathbb{Z}/n$ ist die Basis des RSA Public-Key Kryptosystems.

Beispiel 2.5.2 *Leonhard Euler bewies 1732 durch Rechnen mit Kongruenzen, dass die Fermatzahl $F_5 = 2^{32} + 1$ nicht prim ist. Dazu zeigte er, dass $F_5 \equiv 0 \bmod 641$. Das sieht man wie folgt: Da*

$$641 = 640 + 1 = 5 \cdot 2^7 + 1$$

gilt

$$5 \cdot 2^7 \equiv -1 \bmod 641$$

also für die 4-te Potenz

$$5^4 \cdot 2^{28} \equiv 1 \bmod 641$$

Weiter folgt aus

$$641 = 625 + 16 = 5^4 + 2^4$$

dass

$$5^4 \equiv -2^4 \bmod 641$$

Kombinieren wir beides, erhalten wir

$$2^{32} \equiv -1 \bmod 641.$$

Das wichtigste Werkzeug zum Rechnen mit Kongruenzen ist der Chinesische Restsatz.

Satz 2.5.3 (Chinesischer Restsatz in $\mathbb{Z}$) *Sind $n_1, ..., n_r \in \mathbb{Z}_{>0}$ paarweise teilerfremd und $a_1, ..., a_r \in \mathbb{Z}$, dann ist die* ***simultane Kongruenz***

$$\begin{aligned} x &\equiv a_1 \bmod n_1 \\ &\vdots \\ x &\equiv a_r \bmod n_r \end{aligned}$$

lösbar. Die Lösung ist eindeutig modulo $n = n_1 \cdot ... \cdot n_r$.

Beweis. Sei

$$\hat{n}_i = \frac{n}{n_i}$$

und finde mit dem erweiterten Euklidischen Algorithmus $x_i, y_i \in \mathbb{Z}$ mit

$$1 = \mathrm{ggT}\,(n_i, \hat{n}_i) = x_i n_i + y_i \hat{n}_i$$

Dann ist

$$\begin{aligned} y_i \hat{n}_i &\equiv 0 \bmod n_j \ \ \forall j \neq i \\ y_i \hat{n}_i &\equiv 1 \bmod n_i \end{aligned}$$

Somit erfüllt

$$x = \sum_{i=1}^{r} a_i y_i \hat{n}_i$$

die Kongruenzen und ebenso $x + k \cdot n$ für alle k. Sind x und x' Lösungen, dann $n_i \mid (x - x') \ \forall i$, also

$$n \mid (x - x')$$

■

Diesen Satz kann man wesentlich allgemeiner formulieren. Für ein Beispiel im Kontext von univariaten Polynomringen, siehe Übung 2.20.

Beispiel 2.5.4 *Wir lösen die simultane Kongruenz*

$$\begin{aligned} x &\equiv 2 \bmod 30 \\ x &\equiv 5 \bmod 7 \end{aligned}$$

Es ist $\mathrm{ggT}\,(30, 7) = 1$, also ist die Kongruenz lösbar. Mit dem erweiterten Euklidischen Algorithmus finden wir $x_1 = y_2$ und $y_1 = x_2$ mit

$$x_1 30 + y_1 7 = 1$$

z.B. $x_1 = -3$, $y_1 = 13$. Somit

$$x \equiv 2 \cdot (13 \cdot 7) + 5 \cdot (-3 \cdot 30) \equiv -268 \equiv 152 \bmod 210$$

d.h. die Lösungsmenge ist

$$152 + 210 \cdot \mathbb{Z} = \{152 + k \cdot 210 \mid k \in \mathbb{Z}\}\,.$$

In Termen von Kongruenzen heißt das

$$\left.\begin{array}{l} x \equiv 2 \bmod 30 \\ x \equiv 5 \bmod 7 \end{array}\right\} \Leftrightarrow x \equiv 152 \bmod 210$$

Aus dem Chinesischen Restsatz folgt sofort:

Corollar 2.5.5 *Sind $n, m \in \mathbb{Z}_{>0}$ teilerfremd, dann gilt*

$$\mathbb{Z}/nm \cong (\mathbb{Z}/n) \times (\mathbb{Z}/m)$$

Dabei wird die Klasse von a modulo nm auf das Tupel der Klassen von a modulo n und a modulo m abgebildet.

Beispiel 2.5.6 *Beispiel 2.5.4 können wir wie folgt umschreiben: Nach Corollar 2.5.5 gilt*

$$\mathbb{Z}/210 \cong (\mathbb{Z}/30) \times (\mathbb{Z}/7)$$

und unter diesem Isomorphismus

$$\overline{152} \mapsto (\overline{2}, \overline{5}).$$

Satz 2.5.7 *Seien $a_1, a_2 \in \mathbb{Z}$ und $n_1, n_2 \in \mathbb{Z}_{>0}$. Dann sind die simultanen Kongruenzen*

$$\begin{aligned} x &\equiv a_1 \bmod n_1 \\ x &\equiv a_2 \bmod n_2 \end{aligned}$$

genau dann lösbar, wenn

$$a_1 - a_2 \equiv 0 \bmod \mathrm{ggT}\,(n_1, n_2)\,.$$

Die Lösung ist eindeutig modulo dem $\mathrm{kgV}\,(n_1, n_2)$.

Dies zeigen wir in Übungsaufgabe 2.18.

2.5.2 Anwendung: Modulares Rechnen

Mit Hilfe des Chinesischen Restsatzes lassen sich viele Algorithmen mit ganzzahligem oder rationalem Input und Output beschleunigen. Wir illustrieren im Folgenden das Grundprinzip anhand der Multiplikation von ganzen Zahlen.

Bemerkung 2.5.8 *Der Rechenaufwand der Multiplikation in $\mathbb{Z}$ steigt stärker als linear mit der Bitlänge der Zahlen. Deshalb zerlegt man mit dem Chinesischen Restsatz das Problem in kleinere: Zum Rechnen mit Zahlen $z \in \mathbb{Z}$ mit $|z| < C$ wählt man ein $n = n_1 \cdot \ldots \cdot n_r > 2C$ mit n_i paarweise teilerfremd und alle n_i etwa gleich groß und rechnet in*

$$\mathbb{Z}/n_1 \times \ldots \times \mathbb{Z}/n_r \cong \mathbb{Z}/n$$

ersetzt also eine Operation der Bitlänge N durch r Operationen der Bitlänge $\frac{N}{r}$.

Das Verfahren erlaubt auch die Parallelisierung des Problems, denn die einzelnen Rechnungen in $\mathbb{Z}/n_i$ sind voneinander völlig unabhängig.

Paralleles Rechnen gewinnt an Bedeutung, da Leistungssteigerungen bei Prozessoren zunehmend durch eine größere Zahl von Kernen erreicht werden.

In der Praxis werden für die Multiplikation andere Verfahren verwendet, die jedoch auch auf dem Chinesischen Restsatz beruhen.

Beispiel 2.5.9 *Zur Berechnung von $32 \cdot 45$ betrachten wir*

$$\begin{array}{ccccccccccccccc}
\mathbb{Z} & \to & \mathbb{Z}/2310 & \cong & \mathbb{Z}/2 & \times & \mathbb{Z}/3 & \times & \mathbb{Z}/5 & \times & \mathbb{Z}/7 & \times & \mathbb{Z}/11 \\
32 & \mapsto & \overline{32} & \mapsto & (\,\overline{0} & , & \overline{2} & , & \overline{2} & , & \overline{4} & , & \overline{10}\,) \\
 & & & & & & & & \cdot & & & & \\
45 & \mapsto & \overline{45} & \mapsto & (\,\overline{1} & , & \overline{0} & , & \overline{0} & , & \overline{3} & , & \overline{1}\,) \\
 & & & & & & & & \| & & & & \\
 & & \overline{1440} & \mapsto & (\,\overline{0} & , & \overline{0} & , & \overline{0} & , & \overline{5} & , & \overline{10}\,)
\end{array}$$

Dabei ist $\overline{1440} = 1440 + 2310\mathbb{Z}$ die Lösungsmenge der simultanen Kongruenzen

$$\begin{aligned}
x &\equiv 0 \bmod 2 \\
x &\equiv 0 \bmod 3 \\
x &\equiv 0 \bmod 5 \\
x &\equiv 5 \bmod 7 \\
x &\equiv 10 \bmod 11
\end{aligned}$$

Somit erhalten wir

$$32 \cdot 45 = 1440,$$

wobei das Ergebnis nur korrekt bis auf Addition von Vielfachen von 2310 ist. Man muss also n groß genug wählen, um das korrekte ganzzahlige Ergebnis zu erhalten.

Die Kongruenz lässt sich in SINGULAR *lösen mit:*

```
chinrem(list(0,0,0,5,10), list(2,3,5,7,11));
1440
```

In JULIA *müssen wir iterativ den Chinesischen Restsatz für zwei Kongruenzen anwenden:*

```
using Nemo
crt(ZZ(0),ZZ(30),ZZ(5),ZZ(7))
180
crt(ZZ(180),ZZ(210),ZZ(10),ZZ(11))
1440
```

2.6 Die Einheitengruppe von $\mathbb{Z}/n$

2.6.1 Einheiten und Nullteiler

Wir erinnern kurz an das Konzept von Einheiten und Nullteilern. Betrachten wir etwa den Ring $\mathbb{Z}/6$. Die Verknüpfungen sind gegeben durch

$+$	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{5}$
$\overline{0}$	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{5}$
$\overline{1}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{5}$	$\overline{0}$
$\overline{2}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{5}$	$\overline{0}$	$\overline{1}$
$\overline{3}$	$\overline{3}$	$\overline{4}$	$\overline{5}$	$\overline{0}$	$\overline{1}$	$\overline{2}$
$\overline{4}$	$\overline{4}$	$\overline{5}$	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$
$\overline{5}$	$\overline{5}$	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$

$\cdot$	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{5}$
$\overline{0}$	$\overline{0}$	$\overline{0}$	$\overline{0}$	$\overline{0}$	$\overline{0}$	$\overline{0}$
$\overline{1}$	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{5}$
$\overline{2}$	$\overline{0}$	$\overline{2}$	$\overline{4}$	$\overline{0}$	$\overline{2}$	$\overline{4}$
$\overline{3}$	$\overline{0}$	$\overline{3}$	$\overline{0}$	$\overline{3}$	$\overline{0}$	$\overline{3}$
$\overline{4}$	$\overline{0}$	$\overline{4}$	$\overline{2}$	$\overline{0}$	$\overline{4}$	$\overline{2}$
$\overline{5}$	$\overline{0}$	$\overline{5}$	$\overline{4}$	$\overline{3}$	$\overline{2}$	$\overline{1}$

Während die Additionstabelle keine sonderlich komplizierte Struktur aufweist, ist die Multiplikationstabelle interessanter: Wir sehen, dass $\overline{5}$ bezüglich $\cdot$ ein Inverses besitzt, denn

$$\overline{5} \cdot \overline{5} = \overline{1}$$

Allgemein bezeichnet man ein Element a in einem kommutativen Ring R mit 1 als **Einheit**, wenn ein $b \in R$ existiert mit

$$a \cdot b = 1.$$

Die Menge der Einheiten $R^\times$ bildet bezüglich $\cdot$ eine Gruppe, die **Einheitengruppe**, zum Beispiel hat $(\mathbb{Z}/6)^\times$ die Gruppentafel

$\cdot$	$\overline{1}$	$\overline{5}$
$\overline{1}$	$\overline{1}$	$\overline{5}$
$\overline{5}$	$\overline{5}$	$\overline{1}$

Dagegen ist $\overline{2}$ keine Einheit, es gilt sogar

$$\overline{2} \cdot \overline{3} = \overline{0}.$$

Damit haben wir offenbar auch

$$\overline{4} \cdot \overline{3} = \overline{0}.$$

Die Restklassen $\overline{2}$, $\overline{3}$ und $\overline{4}$ sind also Nullteiler von $\mathbb{Z}/6$. Ein Element kann nicht sowohl Einheit als auch Nullteiler sein, denn ist a eine Einheit und $a \cdot b = 0$, dann ist $b = a^{-1}ab = 0$. In Übung 2.14 werden wir zeigen, dass in einem endlichen Ring R jedes Element entweder Einheit oder Nullteiler ist. Im Allgemeinen gilt dies nicht: In $\mathbb{Z}$ gibt es (außer 0) keine Nullteiler, und die Einheiten sind nur 1 und -1. Wie in Bemerkung 2.1.4 diskutiert, kann man in einem Integritätsring durch Einführen von Brüchen jedes Element außer 0 zu einer Einheit machen und erhält dann den Quotientenkörper $Q(R)$. In einem Körper ist jedes Element $\neq 0$ eine Einheit ist, das heißt $K^\times = K\backslash\{0\}$.

2.6.2 Die Eulersche φ-Funktion

Ein Element $\overline{a} \in \mathbb{Z}/n$ ist invertierbar genau dann, wenn es ein $b \in \mathbb{Z}$ gibt mit $\overline{a} \cdot \overline{b} = \overline{1}$, das heißt, wenn es $b, k \in \mathbb{Z}$ gibt mit

$$a \cdot b + k \cdot n = 1$$

Solche b und k erhalten wir mit dem erweiterten Euklidischen Algorithmus, falls

$$\mathrm{ggT}\,(a, n) = 1$$

Haben wir umgekehrt eine solche Darstellung der 1, dann müssen natürlich a und n teilerfremd sein (denn jeder gemeinsame Teiler teilt auch 1). Somit können wir die Elemente der Einheitengruppe beschreiben:

Satz 2.6.1 *Für $n \in \mathbb{N}$ ist*

$$(\mathbb{Z}/n)^\times = \{\overline{a} \in \mathbb{Z}/n\mathbb{Z} \mid \mathrm{ggT}\,(a, n) = 1\}\,.$$

*Die Elemente heißen **prime Restklassen**. Die Gruppe $(\mathbb{Z}/n)^\times$ bezeichnen wir auch als die **prime Restklassengruppe**.*

Als direkte Folgerung sieht man:

Corollar 2.6.2 *Der Ring $\mathbb{Z}/n$ ist ein Körper genau dann, wenn n eine Primzahl ist.*

Beispiel 2.6.3 *Die Restklasse $\overline{8} \in \mathbb{Z}/15$ hat ein Inverses, d.h. $\overline{8} \in (\mathbb{Z}/15)^\times$, denn*

$$\mathrm{ggT}\,(8, 3 \cdot 5) = 1$$

Mit dem erweiterten Euklidischen Algorithmus erhalten wir eine Darstellung des größten gemeinsamen Teilers

$$1 = (2) \cdot 8 + (-1) \cdot 15$$

also ist

$$\overline{8}^{-1} = \overline{2}$$

Definition 2.6.4 *Die* ***Eulersche φ-Funktion*** $\varphi : \mathbb{N} \to \mathbb{Z}$ *ist definiert durch*

$$\varphi(n) := \left|(\mathbb{Z}/n)^\times\right| = \left|\{r \in \mathbb{Z} \mid 1 \le r \le n,\ \mathrm{ggT}(r,n) = 1\}\right|$$

gibt also für n die Ordnung der Einheitengruppe $(\mathbb{Z}/n)^\times$ an.

Beispiel 2.6.5 *Mit Satz 2.6.1 ist*

$$(\mathbb{Z}/15)^\times = \{\overline{1}, \overline{2}, \overline{4}, \overline{7}, \overline{8}, \overline{11}, \overline{13}, \overline{14}\}$$

also

$$\left|(\mathbb{Z}/15)^\times\right| = 8.$$

Satz 2.6.6 (Satz von Fermat-Euler) *Für alle $a, n \in \mathbb{Z}$, $n \ge 1$ mit* $\mathrm{ggT}(a,n) = 1$ *gilt*

$$a^{\varphi(n)} \equiv 1 \bmod n.$$

Beweis. Da die Ordnung jedes Elements $g \in G$ die Gruppenordnung teilt, ist

$$g^{|G|} = e.$$

Angewendet auf $\overline{a} \in (\mathbb{Z}/n)^\times$ erhalten wir

$$\overline{a}^{\varphi(n)} = \overline{1}.$$

■

Für Primzahlen p ist offenbar

$$\varphi(p) = p - 1,$$

also

$$a^{p-1} \equiv 1 \bmod p \quad \text{falls } p \nmid a$$

und somit (denn für $p \mid a$ ist $a^p \equiv 0 \equiv a \bmod p$):

Corollar 2.6.7 (Kleiner Satz von Fermat) *Ist p eine Primzahl und $a \in \mathbb{Z}$, dann gilt*

$$a^p \equiv a \bmod p.$$

Zur Berechnung der Eulerschen φ-Funktion verwendet man, dass sie multiplikativ über teilerfremde Produkte ist:

Lemma 2.6.8 *Sind $m_1, m_2 \in \mathbb{N}$ teilerfremd, dann gilt*

$$\varphi(m_1 m_2) = \varphi(m_1)\,\varphi(m_2).$$

Beweis. Mit dem Chinesischen Restsatz ist

$$\mathbb{Z}/m_1m_2 \cong \mathbb{Z}/m_1 \times \mathbb{Z}/m_2.$$

Also ist $a + m_1m_2\mathbb{Z} \in (\mathbb{Z}/m_1m_2)^\times$ genau dann, wenn

$$(a + m_1\mathbb{Z}, a + m_2\mathbb{Z}) \in (\mathbb{Z}/m_1 \times \mathbb{Z}/m_2)^\times,$$

äquivalent, wenn $a + m_i\mathbb{Z} \in (\mathbb{Z}/m_i)^\times \ \forall i$, da die Multiplikation komponentenweise definiert ist. Somit

$$(\mathbb{Z}/m_1m_2)^\times \cong (\mathbb{Z}/m_1)^\times \times (\mathbb{Z}/m_2)^\times$$

■

Insbesondere erhalten wir:

Bemerkung 2.6.9 *Ist $n = p \cdot q$ das Produkt von zwei Primzahlen, so gilt*

$$\varphi(n) = (p-1)(q-1).$$

Beispiel 2.6.10 $\varphi(15) = 2 \cdot 4 = 8$*, wie oben schon mittels Satz 2.6.1 berechnet.*

2.7 RSA Public-Key Kryptographie

Beim RSA-Kryptosystem verwendet der Sender den öffentlichen Schlüssel des Empfängers zum Chiffrieren einer Nachricht und dieser seinen privaten Schlüssel zum Dechiffrieren, d.h. es ist ein sogenanntes Public-Key Kryptosystem. Das Verfahren wurde von James Ellis, Clifford Cocks und Malcolm Williamson im britischen Nachrichtendienst entwickelt (und geheim gehalten) und ist nach Ronald Rivest, Adi Shamir und Leonard Adleman benannt, die es später erneut entdeckt haben. Es basiert auf einer Trapdoor (Geheimtür) - Einwegfunktion

$$\{\text{Klartextnachrichten}\} \to \{\text{verschlüsselte Nachrichten}\}$$

die man leicht berechnen kann, das Urbild aber nur unter hohem Rechenaufwand, sofern man nicht die Geheimtür-Information (d.h. den privaten Schlüssel) besitzt.

Im Fall von RSA beruht dies darauf, dass die Primfaktorzerlegung (und damit die Geheimtür) mit dem heutigen Wissen nur schwer zu berechnen ist. Allerdings ist nicht klar, ob nicht in Zukunft schnellere Verfahren zur Verfügung stehen. Auch muss man bei der Verwendung von RSA abschätzen, wie lange die Verschlüsselung unter dem erfahrungsgemäßen exponentiellen Anstieg der Rechenleistung von Computern (Moores Gesetz) sicher ist.

Typischerweise wird aus Gründen der Geschwindigkeit RSA nur zum Austausch eines Schlüssels für ein konventionelles symmetrisches

Kryptosystem (z.B. 3DES, AES, Twofish, Serpent) verwendet (den dann der Sender zum Verschlüsseln und der Empfänger zum Entschlüsseln benützt).

Das RSA Kyptosystem beruht auf der Multiplikation in $\mathbb{Z}/n$ wobei $n = p \cdot q$ und p, q prim. Allerdings können wir nicht alle Elemente verwenden, um eine Gruppe bezüglich $\cdot$ zu erhalten:

2.7.1 Setup für RSA

Man wählt eine große Zahl $N \in \mathbb{N}$ und codiert Nachrichteneinheiten in eine Zahl $0 \leq m < N$ (zum Beispiel $N = 26^k$ und repräsentiert Buchstaben durch Ziffern). In der Praxis verwendet man ein N mit etwa 200 bis 600 Dezimalziffern.

Jeder Benutzer führt nun die folgenden Schritte aus:

1) Wähle 2 Primzahlen p, q mit $p \cdot q > N$.

2) Berechne
$$n := p \cdot q$$
und den Wert der Eulerfunktion
$$\varphi(n) = (p-1)(q-1)$$
Die Zahlen p und q können nun gelöscht werden.

3) Wähle eine Zahl $e \in \mathbb{N}$ mit
$$\mathrm{ggT}(e, \varphi(n)) = 1$$

4) Berechne das Inverse $0 < d < \varphi(n)$ von e modulo $\varphi(n)$, also mit
$$ed \equiv 1 \bmod \varphi(n)$$
Nun kann $\varphi(n)$ gelöscht werden.

Der öffentliche Schlüssel ist das Tupel (n, e) und der private Schlüssel d.

2.7.2 Nachrichtenübertragung

Betrachten wir nun zwei Personen Alice und Bob mit Schlüsseln

	privat	öffentlich
Alice	d_A	(n_A, e_A)
Bob	d_B	(n_B, e_B)

Will Bob an Alice eine Nachricht m senden (wir nehmen an, dass $\mathrm{ggT}(m, n_A) = 1$), verwendet er den öffentlichen Schlüssel von Alice und berechnet
$$c := m^{e_A} \bmod n_A$$

und überträgt c an Alice.

Diese berechnet nun zum Entschlüsseln

$$\tilde{m} := c^{d_A} \bmod n_A$$

Dann gilt modulo n_A, dass

$$\tilde{m} \equiv c^{d_A} \equiv \left(m^{e_A}\right)^{d_A} = m^{e_A d_A} = m^{1+k\cdot\varphi(n_A)} = m \cdot \left(m^{\varphi(n_A)}\right)^k \equiv m \bmod n_A$$

mit dem Satz von Fermat-Euler 2.6.6.

Beispiel 2.7.1 *Alice wählt*

$$n_A = 7 \cdot 11 = 77$$

also

$$\varphi\left(n_A\right) = 6 \cdot 10 = 60$$

und

$$e_A = 13$$

Der öffentliche Schlüssel von Alice ist dann

$$\left(n_A, e_A\right) = (77, 13)$$

Mit dem erweiterten Euklidischen Algorithmus erhalten wir

$$1 = \mathrm{ggT}\left(e_A, \varphi\left(n_A\right)\right) = (-23) \cdot 13 + (5) \cdot 60$$

und somit das Inverse d_A von e_A modulo $\varphi\left(n_A\right)$

$$d_A = 37$$

den privaten Schlüssel von Alice.

Bob möchte die Nachricht $m = 31$ verschlüsselt an Alice senden, berechnet also

$$m^{e_A} \bmod n_A = 31^{13} \bmod 77 = 3 \bmod 77$$

und überträgt

$$c = 3$$

Zum Entschlüsseln berechnet Alice

$$c^{d_A} \bmod n_A = 3^{37} = 31 \bmod 77$$

Siehe auch Übungsaufgabe 2.22.

2.8 Primfaktorisierung mit dem Verfahren von Pollard

Gelingt es uns die Faktorisierung $n = p \cdot q$ zu bestimmen, so erhalten wir $\varphi(n) = (p-1)(q-1)$. Damit können wir aus dem öffentlichen Schlüssel e den privaten Schlüssel d bestimmen und somit jede mit (n, e) verschlüsselte Nachricht mitlesen. Für große Zahlen n ist Probedivision nicht praktikabel. Es gibt wesentlich effizientere Methoden, um einen Primteiler zu finden. Als Beispiel behandeln wir ein Verfahren von John Pollard, das unter folgender Voraussetzung gut funktioniert (die wir bei der RSA-Schlüsselerzeugung also besser vermeiden sollten):

Algorithmus 2.4 Pollard Faktorisierung

Angenommen ein Primfaktor p von n hat die Eigenschaft, dass $p-1$ nur kleine Primpotenzfaktoren $\leq B$ besitzt. Dann lässt sich ein Vielfaches k von $p-1 = \varphi(p)$ bestimmen, ohne p zu kennen:

$$k := \prod_{\substack{q \text{ Primzahl} \\ l \text{ maximal mit } q^l \leq B}} q^l$$

Sei nun $1 < a < n$ beliebig gewählt. Teste zunächst, ob $\mathrm{ggT}(a, n) = 1$ (wenn nicht, haben wir einen echten Teiler gefunden). Sind a und n teilerfremd, erhalten wir einen Faktor von n als

$$\mathrm{ggT}\left(a^k - 1, n\right) > 1$$

denn k ist nach Voraussetzung ein Vielfaches von $\varphi(p)$, also $k = k' \cdot \varphi(p)$. Damit gibt der kleine Fermatsche Satz

$$a^k = \left(a^{\varphi(p)}\right)^{k'} \equiv 1 \bmod p$$

also $p \mid \mathrm{ggT}(a^k - 1, n)$. Falls wir aufgrund der Wahl von a und B keinen echten Teiler finden, ändern wir unsere Wahl.

Wir bemerken, dass

$$\mathrm{ggT}\left(a^k - 1 \bmod n, n\right) = \mathrm{ggT}\left(a^k - 1, n\right),$$

wir können also $a^k - 1$ modulo n reduzieren. Dies ist auch essentiell, da k und damit $a^k - 1$ sehr großsein kann.

Beispiel 2.8.1 *Sei $n = 21733$ und $B = 10$, also*

$$k = 2^3 \cdot 3^2 \cdot 5 \cdot 7.$$

Sei weiter $a = 2$. Dann ist

$$\mathrm{ggT}\left(2^k - 1, n\right) = 211$$

Sowohl 211 *also auch* $\frac{n}{211} = 103$ *sind prim, was man z.B. mit Probedivision sieht. Damit haben wir* n *vollständig faktorisiert. Man beachte, dass die gefundenen Teiler im Allgemeinen nicht prim sein müssen.*

Siehe dazu auch Übungsaufgabe 2.35.

Bemerkung 2.8.2 *Offenbar ist es für RSA und das Pollardverfahren wichtig, schnell Potenzieren zu können. Dies leistet Algorithmus 2.5. Wenn wir in* $R = \mathbb{Z}/n$ *rechnen, dann reduzieren wir im Algorithmus nach jeder arithmetischen Operation modulo* n*, um große ganzzahlige Zwischenergebnisse zu vermeiden.*

Algorithmus 2.5 Potenzieren

Input: x in einem kommutativen Ring R mit 1 und $n \in \mathbb{N}$
Output: x^n

```
1: if n = 0 then
2:     return 1
3: Potenzieren y = x^⌊n/2⌋
4: if n gerade then
5:     return y^2
6: else
7:     return x · y^2
```

2.9 Primzahltests

Wie schon diskutiert, sind für die Probedivision Primzahltests ein wesentliches Hilfsmittel. Die Grundlage für den einfachsten Primzahltest ist wieder der kleine Satz von Fermat:

2.9.1 Fermat Primzahltest

Algorithmus 2.6 Fermat Primzahltest

Wir wollen testen, ob $n \in \mathbb{N}$ *eine Primzahl ist.*

1) Zunächst wählen wir ein $a \in \mathbb{Z}$, $1 < a < n$ *und bestimmen* $\mathrm{ggT}(a, n)$ *mit dem Euklidischen Algorithmus. Falls* $\mathrm{ggT}(a, n) \neq 1$*, war* n *nicht prim.*

2) Ist $\mathrm{ggT}(a, n) = 1$ *(und damit* $\overline{a} \in (\mathbb{Z}/n)^\times$ *nach Satz 2.6.1), dann testen wir, ob*

$$a^{n-1} \equiv 1 \bmod n$$

Gilt dies nicht, dann kann n *nach dem kleinen Satz von Fermat 2.6.6 auch nicht prim gewesen sein. Man bezeichnet* a *dann als* ***Fermat-Zeugen*** *für die Zerlegbarkeit von* n*. Anderenfalls können wir keine Aussage machen und gehen zurück zu* (1).

Falls n prim ist, bricht dieses Verfahren nicht ab, man kann also nur durch mehrfaches Durchlaufen der Schleife (mit verschiedenen a) die Wahrscheinlichkeit erhöhen, dass wir n korrekterweise für prim halten. Es gibt auch Zahlen, bei denen der Test in (2) für kein a mit $\mathrm{ggT}(a,n) = 1$ erkennt, dass sie nicht prim sind, die sogenannten **Carmichael-Zahlen**. Diese erkennt aber Schritt (1) für geeignetes a (was aber natürlich sehr ineffizient ist). Man kann zeigen, dass es unendlich viele Carmichael-Zahlen gibt, der Beweis ist aber nicht einfach und wurde erst 1994 geführt:

Satz 2.9.1 *Sei $C(x)$ die Anzahl der Carmichael-Zahlen $\leq x$. Dann gibt es ein x_0 mit*

$$C(x) > x^{\frac{2}{7}}$$

für alle $x \geq x_0$.

Siehe dazu auch Übung 2.23.

Definition 2.9.2 *Eine Zahl n heißt **Fermatsche Pseudoprimzahl** zur Basis a, wenn n nicht prim ist, aber dennoch $a^{n-1} \equiv 1 \bmod n$ gilt.*

Beispiel 2.9.3 *Die Rechnung*

$$2^8 \equiv 4 \bmod 9$$

beweist, dass 9 nicht prim ist.

Dagegen gilt

$$2^{340} \equiv 1 \bmod 341$$

aber unglücklicherweise ist

$$341 = 11 \cdot 31$$

nicht prim, also 341 eine Fermatsche Pseudoprimzahl zur Basis $a = 2$. Testen wir nochmals zur Basis $a = 3$ erhalten wir

$$3^{340} \equiv 56 \bmod 341$$

und haben damit gezeigt, dass 341 keine Primzahl ist.
Man beachte:
Dies konnten wir erkennen, ohne einen Teiler zu finden.

Wir bemerken noch: Erfüllt n den Fermatschen Primzahltest, kann man mit der folgenden Aussage sicher feststellen, ob n wirklich prim ist:

Proposition 2.9.4 *Sei $n \in \mathbb{Z}_{\geq 2}$. Für ein $a \in \mathbb{Z}$ gelte $a^{n-1} \equiv 1 \bmod n$ und $a^{\frac{n-1}{p}} \not\equiv 1 \bmod n$ für jeden Primteiler p von $n-1$. Dann ist n prim.*

Der leichte Beweis ist Übung 2.24. Zur Anwendung von Proposition 2.9.4 muss man aber $n-1$ faktorisieren können, was praktisch oft nicht möglich ist.

2.9.2 Miller-Rabin Primzahltest

Wie kann man also den Fermatschen Primzahltest so verbessern, dass er mit beliebig hoher Wahrscheinlichkeit ein korrektes Ergebnis liefert (ohne Rückgriff auf das zufällige Finden von Teilern im Test $\mathrm{ggT}(a,n) = 1$)? Der **Miller-Rabin Primzahltest** (auch **Selfridge Primzahltest** nach dem eigentlichen Entdecker) leistet dies. Wir bemerken zunächst, dass sich mit der Einheitengruppe

$$(\mathbb{Z}/n)^\times = \{\overline{a} \in \mathbb{Z}/n \mid 1 \le a \le n,\ \mathrm{ggT}(a,n) = 1\}$$

von $\mathbb{Z}/n$ der kleine Fermatsche Satz (also die Grundlage des Fermatschen Primzahltests) auch formulieren lässt als

$$n \text{ prim}, \overline{a} \in (\mathbb{Z}/n)^\times \implies \overline{a}^{n-1} = 1$$

(wobei wir den trivialen Fall $n \mid a$ ausschließen). Diese Idee lässt sich folgendermaßen erweitern:

Proposition 2.9.5 *Sei n eine Primzahl und $\overline{a} \in (\mathbb{Z}/n)^\times$. Schreibe*

$$n - 1 = 2^t \cdot u$$

mit u ungerade. Dann gilt entweder

$$\overline{a}^u = 1$$

oder es gibt ein $s \in \{1, ..., t\}$ mit

$$\overline{a}^{2^{s-1}\cdot u} = -1$$
$$\overline{a}^{2^s \cdot u} = 1$$

*Gilt dies nicht, dann nennt man a einen **Miller-Rabin-Zeugen** für die Zerlegbarkeit von n. Gilt dies, obwohl n nicht prim ist, dann heißt n **strenge Pseudoprimzahl** zur Basis a.*

Beweis. Sei

$$s = \min\left\{w \in \mathbb{N}_0 \mid \overline{a}^{2^w \cdot u} = 1\right\}$$

Ist n prim, dann gilt mit Satz 2.6.6, dass

$$\overline{a}^{n-1} = 1,$$

also $0 \le s \le t$. Für $s = 0$ ist $\overline{a}^u = 1$, für $s \ge 1$ haben wir

$$(\overline{a}^{2^{s-1}\cdot u})^2 = \overline{a}^{2^s \cdot u} = 1.$$

Somit ist $\overline{a}^{2^{s-1}\cdot u}$ eine Nullstelle von $x^2 - 1 = (x-1)\cdot(x+1)$ in dem Körper $\mathbb{Z}/n$, und damit 1 oder -1. (Dies gilt in jedem Körper, da dieser keine Nullteiler hat. Speziell für $\mathbb{Z}/n$ sehen wir das auch elementar, da aus

Algorithmus 2.7 Miller-Rabin

Input: $n \in \mathbb{Z}_{\geq 3}$ ungerade und $0 < a < n$.

Output: Wenn `false`, dann ist n nicht prim, wenn `true`, dann ist n wahrscheinlich prim (d.h. prim oder eine starke Pseudoprimzahl zum Basis a).

1: Schreibe $n - 1 = 2^t \cdot u$ mit u ungerade.
2: **if** $\mathrm{ggT}(a, n) > 1$ **then return** `false`
3: $b = a^u \bmod n$
4: **if** $b \equiv 1 \bmod n$ **then return** `true`
5: **for** $s = 1, ..., t$ **do**
6: **if** $b \equiv -1 \bmod n$ **then return** `true`
7: $b = b^2 \bmod n$
8: **return** `false`

$n \mid (x^2 - 1)$ und n prim folgt $n \mid (x - 1)$ oder $n \mid (x + 1)$). Nach Wahl von s haben wir also

$$\overline{a}^{2^{s-1} \cdot u} = -1.$$

■

Aus Proposition 2.9.5 erhalten wir den Primzahltest beschrieben in Algorithmus 2.7.

Beispiel 2.9.6 *Wir betrachten die Folge*

$$A := (\overline{a}^u, \overline{a}^{2 \cdot u}, ..., \overline{a}^{2^t \cdot u}) \in (\mathbb{Z}/n)^{t+1}.$$

Für die Primzahl $n = 5$ *und* $a = 2$ *erhalten wir*

$$A = (2, -1, 1),$$

also $s = t = 2$ *in in Proposition 2.9.5.*

Für $n = 341 = 11 \cdot 31$ *ist*

$$2^{340} \equiv 1 \bmod 341,$$

der Fermatsche Primzahltest mit $a = 2$ *erkennt also nicht, dass* n *nicht prim ist. Dagegen ist* $a = 2$ *ein Miller-Rabin-Zeuge für die Zerlegbarkeit von* n*, denn*

$$340 = 2^2 \cdot 85$$

und

$$A = (32, 1, 1),$$

nach Proposition 2.9.5 kann 341 *also nicht prim sein.*

Im Gegensatz dazu ist $n = 2047$ *eine strenge Pseudoprimzahl zur Basis* $a = 2$*, denn*

$$2046 = 2 \cdot 1023$$

und

$$2^{1023} \equiv 1 \bmod 2047$$

also

$$A = (1, 1),$$

aber

$$2047 = 23 \cdot 89.$$

Wiederholen wir den Test mit der Basis $a = 3$*, erhalten wir dagegen*

$$A = (1565, 1013)$$

und erkennen mit Proposition 2.9.5, dass 2047 *nicht prim ist.*

Beispiel 2.9.7 *Für die zusammengesetzte Zahl* $n = 25 = 2^3 \cdot 3 + 1$ *berechnen wir mit dem* JULIA*-Code*

```
using Nemo
n = ZZ(25)
a = ZZ(1)
while a<=24
  a3 = powmod(a,3,n)
  a6 = powmod(a,6,n)
  a12 = powmod(a,12,n)
  if gcd(a,n)==1 && (a3==1 || a3==24 || a6==24 || a12==24)
    println(a,"",a3,"",a6,"",a12)
  end
  a=a+1;
end
```

eine Tabelle mit den positiven Miller-Rabin-Tests zur Basis $\overline{a}$*:*

$\overline{a}$	$\overline{a}^3$	$\overline{a}^6$	$\overline{a}^{12}$
1	1	1	1
7	18	24	1
18	7	24	1
24	24	1	1

Keine Miller-Rabin-Zeugen für die Zerlegbarkeit von 25 *sind also nur* $1, 7, 18, 24$*. Der Anteil der Zeugen unter allen Elementen von* $(\mathbb{Z}/25)^\times$ *ist mit*

$$1 - \frac{4}{20} = \frac{4}{5}$$

also ziemlich groß.

Man kann allgemein zeigen:

Satz 2.9.8 *Ist* $n \in \mathbb{N}$ *nicht prim,* $n > 9$*, dann hat* n *mindestens*

$$\frac{3}{4}\varphi(n)$$

Miller-Rabin-Zeugen für seine Zerlegbarkeit.

Die Wahrscheinlichkeit, dass der Miller-Rabin-Test `true` liefert, obwohl n nicht prim ist, ist also $\leq \frac{1}{4}$, denn $\varphi(n) = \left|(\mathbb{Z}/n)^\times\right| \leq n$. Die Fehlerwahrscheinlichkeit nach m positiven Miller-Rabin-Tests mit zufällig gewählten a ist also

$$\leq \frac{1}{4^m}.$$

Für die in der Praxis übliche Wahl $m = 40$ ist sie also $\leq 10^{-24}$ und damit wesentlich kleiner als die Wahrscheinlichkeit eines Bitfehlers im Computer.

Der Beweis von Satz 2.9.8 ist relativ aufwendig. Etwas schwächer, aber einfacher zu zeigen ist:

Satz 2.9.9 *Sei $n \in \mathbb{N}$ nicht prim. Dann sind mindestens die Hälfte aller $\overline{a} \in (\mathbb{Z}/n)^\times$ Miller-Rabin-Zeugen für die Zerlegbarkeit von n.*

Zunächst betrachten wir den Fall, dass n keine Carmichael-Zahl ist (beachte, dass jeder Fermat-Zeuge auch ein Miller-Rabin-Zeuge ist):

Lemma 2.9.10 *Sei $n \in \mathbb{N}$ nicht prim und keine Carmichael-Zahl, dann sind mindestens die Hälfte aller $\overline{a} \in (\mathbb{Z}/n)^\times$ Fermat-Zeugen für die Zerlegbarkeit von n.*

Beweis. Die Menge

$$\begin{aligned} A &= \{\overline{a} \in (\mathbb{Z}/n)^\times \mid n \text{ ist Fermatsche Pseudoprimzahl zur Basis } \overline{a}\} \\ &= \{\overline{a} \in (\mathbb{Z}/n)^\times \mid \overline{a}^{n-1} \equiv 1 \bmod n\} \end{aligned}$$

ist offenbar eine Untergruppe von $(\mathbb{Z}/n)^\times$. Für n keine Carmichael-Zahl ist A eine echte Untergruppe und hat somit Index

$$\frac{\varphi(n)}{|A|} \geq 2.$$

■

Insbesondere sehen wir, dass der Fermat.Primzahltest nicht schlecht ist, solange wir Carmichael-Zahlen abfangen (z.B. durch eine Tabelle dieser Zahlen bis zu einer vorgegebenen Schranke).

Wir betrachten nun den Fall, dass n eine Carmichael-Zahl ist. In Übung 2.27 beweisen wir:

Satz 2.9.11 *Sei $n \in \mathbb{N}$ zusammengesetzt. Dann ist n eine Carmichael-Zahl genau dann, wenn für alle Primteiler p von n gilt, dass*

$$p^2 \nmid n$$

und

$$(p-1) \mid (n-1).$$

Beispiel 2.9.12 *Die kleinste Carmichael-Zahl ist*

$$561 = 3 \cdot 11 \cdot 17.$$

Corollar 2.9.13 *Eine Carmichael-Zahl ist das Produkt von wenigstens 3 verschiedenen Primzahlen.*

Beweis. Ist $n = p \cdot q$ eine Carmichael-Zahl mit $p \neq q$ prim. Dann gilt $(p-1) \mid (n-1)$ und $(q-1) \mid (n-1)$. Weiter ist

$$n - 1 = p(q-1) + (p-1)$$

und somit $(p-1) \mid (q-1)$. Ebenso sehen wir $(q-1) \mid (p-1)$, also $p = q$. ■

Corollar 2.9.14 *Jede Carmichael-Zahl ist ungerade.*

Beweis. Ist $n = 2 \cdot p \cdot q$ eine Carmichael-Zahl, dann teilt die gerade Zahl $p-1$ die ungerade Zahl $n-1$. ■

Satz 2.9.15 *Die Einheitengruppe $\mathbb{F}_q^\times$ eines endlichen Körpers mit q Elementen ist zyklisch, insbesondere also $(\mathbb{Z}/p)^\times$ für p prim.*

Den Beweis werden wir später im Zuge der Klassifikation von endlich erzeugten abelschen Gruppen führen.

Wir beweisen nun Satz 2.9.9:

Beweis. Nach Lemma 2.9.10 können wir annehmen, dass n eine Carmichael-Zahl ist. Nach Corollar 2.9.13 und Satz 2.9.11 ist

$$n = p_1 \cdot \ldots \cdot p_k$$

mit paarweise verschiedenen ungeraden Primzahlen p_i und $k \geq 3$.

Schreibe

$$n - 1 = 2^t \cdot u$$

mit $t, u \in \mathbb{N}_0$ und u ungerade.

$$S = \left\{ 0 \leq w \leq t \mid \overline{a}^{2^w \cdot u} = 1 \text{ für alle } \overline{a} \in (\mathbb{Z}/n)^\times \right\}.$$

Für die Carmichael-Zahl n ist also $t \in S$. Nach Definition von S ist klar: Falls $w \in S$ mit $0 \leq w < t$, dann auch $w + 1 \in S$.

Sei $p = p_1$ und $\overline{g}$ ein Erzeuger der nach Satz 2.9.15 zyklischen Gruppe $(\mathbb{Z}/p)^\times$. Dann ist $\operatorname{ord}(\overline{g}) = p - 1$ und somit gerade. Wegen u ungerade kann $p-1$ kein Teiler von u sein, also ist $g^u \not\equiv 1 \bmod p$. Dann ist auch $g^u \not\equiv 1 \bmod n$. Also ist $0 \notin S$. Somit gibt es ein $0 \leq w < t$ mit $w \notin S$ und $w + 1 \in S$.

Die Menge

$$G = \{ \overline{a} \in (\mathbb{Z}/n)^\times \mid \overline{a}^{2^w \cdot u} = \pm 1 \}$$

ist offenbar eine Untergruppe von $(\mathbb{Z}/n)^\times$. Alle Basen $\overline{a}$, für die n eine starke Pseudoprimzahl ist, liegen sicher in G: Ist $\overline{a} \notin G$, dann

$$\overline{a}^{2^w \cdot u} \not\equiv \pm 1$$
$$\overline{a}^{2^{w+1} \cdot u} \equiv 1.$$

Es bleibt zu zeigen, dass G eine echte Untergruppe ist: Wegen $w \notin S$ existiert ein $\overline{a} \in (\mathbb{Z}/n)^\times$ mit

$$a^{2^w \cdot u} \not\equiv 1 \bmod n,$$

es muss also auch für ein p_i gelten

$$a^{2^w \cdot u} \not\equiv 1 \bmod p_i.$$

Der Chinesische Restsatz liefert eine Lösung b der simultanen Kongruenzen

$$b \equiv a \bmod p_i$$
$$b \equiv 1 \bmod p_j \text{ für } j \neq i$$

Dann ist

$$b^{2^w \cdot u} \not\equiv 1 \bmod p_i$$
$$b^{2^w \cdot u} = 1 \not\equiv -1 \bmod p_j \text{ für } j \neq i$$

also

$$b^{2^w \cdot u} \not\equiv \pm 1 \bmod n$$

und somit $\overline{b} \notin G$. ■

Bemerkung 2.9.16 *Man kann zeigen, dass für*

$$n < 2^{48}$$

(genauer $n < 341\ 550\ 071\ 728\ 321$) ein für alle

$$a = 2, 3, 5, 7, 11, 13, 17$$

positiver Miller-Rabin-Test beweist, dass n prim ist.

2.10 Das quadratische Sieb

2.10.1 Grundidee

Wir haben schon mit der Pollard-Faktorisierung (Algorithmus 2.4) ein sehr schnelles Verfahren zur Faktorisierung einer Zahl $n \in \mathbb{N}$ kennengelernt. Dieses funktioniert jedoch nur gut, wenn n einen Primfaktor besitzt, sodass $p - 1$ nur kleine Primpotenzfaktoren hat. Eines der

schnellsten Faktorisierungsverfahren, dessen Laufzeit nicht von speziellen Eigenschaften von n, sondern nur von der Grösse von n abhängt, ist das quadratische Sieb. Dieses wurde um 1980 von Carl Pomerance entwickelt. Die Grundidee ist folgende: Wir suchen Zahlen $x, y \in \mathbb{Z}$ sodass

$$x^2 \equiv y^2 \bmod n$$

aber

$$x \not\equiv \pm y \bmod n$$

Dann ist n Teiler von $x^2 - y^2 = (x-y)(x+y)$, aber $n \nmid (x-y)$ und $n \nmid (x+y)$. Somit ist $\mathrm{ggT}(x-y, n)$ ein echter Teiler von n.

Beispiel 2.10.1 *Für $n = 7429$ ist $x^2 \equiv y^2 \bmod n$ für $x = 227$ und $y = 210$ erfüllt und*

$$\mathrm{ggT}(x-y, n) = \mathrm{ggT}(17, 7429) = 17.$$

Lemma 2.10.2 *Ist p ungerade prim, $r \in \mathbb{N}$ und $z \in \mathbb{Z}$ dann existieren in $\mathbb{Z}/p^r$ genau zwei verschiedene Quadratwurzeln von z.*

Beweis. Siehe Übungen. ■

Bemerkung 2.10.3 *Ist n ungerade mit $k \geq 2$ verschiedenen Primteilern, dann gibt es zu jedem x mit $\mathrm{ggT}(x, n) = 1$ (andernfalls haben wir schon einen Teiler von n gefunden) genau 2^k verschiedene y mit $x^2 \equiv y^2 \bmod n$. Für genau zwei von diesen gilt $x \not\equiv \pm y \bmod n$. Die Wahrscheinlichkeit einen nichttrivialen Teiler von n zu finden ist also gleich*

$$1 - \frac{2}{2^k} = 1 - \frac{1}{2^{k-1}} \geq \frac{1}{2}.$$

Beweis. Wir zerlegen n mit dem Chinesischen Restsatz und wenden Lemma 2.10.2 an. ■

Wie findet man x und y? Ein Ansatz ist, x zu suchen, sodass

$$x^2 - n$$

ein möglichst kleines Quadrat y^2 ist. Dazu muss x nahe an der Quadratwurzel von n liegen. Wir definieren deshalb $m = \lfloor \sqrt{n} \rfloor$ und die Funktion

$$f(s) = (s+m)^2 - n,$$

die für kleine Werte von s kleine Werte $f(s)$ liefert. Diese sind im Allgemeinen jedoch kein Quadrat. Die Idee ist nun, mehrere $f(s)$ so als Produkt zu kombinieren, dass ein Quadrat entsteht:

Beispiel 2.10.4 *Für $n = 7429$ ist $m = 86$. Für kleine $|s|$ berechnen wir einige Funktionswerte*

$$\begin{aligned} f(-3) &= 83^2 - n = -540 = -2^2 \cdot 3^3 \cdot 5 \\ f(1) &= 87^2 - n = 140 = 2^2 \cdot 5 \cdot 7 \\ f(2) &= 88^2 - n = 315 = 3^2 \cdot 5 \cdot 7 \end{aligned}$$

Dies zeigt, dass

$$\begin{aligned} 87^2 &\equiv 2^2 \cdot 5 \cdot 7 \bmod n \\ 88^2 &\equiv 3^2 \cdot 5 \cdot 7 \bmod n \end{aligned}$$

Somit gilt

$$87^2 \cdot 88^2 \equiv 2^2 \cdot 3^2 \cdot 5^2 \cdot 7^2 \bmod n$$

also $x^2 \equiv y^2 \bmod n$ für

$$\begin{aligned} x &= 87 \cdot 88 \equiv 227 \bmod n \\ y &= 2 \cdot 3 \cdot 5 \cdot 7 \equiv 210 \bmod n \end{aligned}$$

Weiter gilt für

$$x - y \equiv 17 \bmod n$$

dass

$$\mathrm{ggT}(x - y, n) = \mathrm{ggT}(17, 7429) = 17.$$

2.10.2 Kombination von Kongruenzen

Der Ansatz ist also für kleine s_i

$$f(s_i) = (s_i + m)^2 - n = \prod_j p_j^{a_{i,j}}$$

zu berechnen, und ein Produkt der rechten Seiten der Kongruenzen

$$(s_i + m)^2 \equiv \prod_j p_j^{a_{i,j}} \bmod n$$

zu wählen, das ein Quadrat ist. Ein Produkt der linken Seiten ist sowieso immer ein Quadrat. Das heißt wir suchen $\lambda_i \in \{0, 1\}$, sodass

$$\prod_j p_j^{\sum_i \lambda_i a_{i,j}}$$

ein Quadrat ist, d.h. $\lambda_i \in \mathbb{F}_2 = \mathbb{Z}/2$ mit

$$\sum_i \lambda_i a_{i,j} \equiv 0 \bmod 2 \tag{2.1}$$

Zur Lösung des Problems müssen wir also viele $f(s)$ faktorisieren und ein großes homogenes Gleichungssystem über $\mathbb{F}_2$ lösen.

Beispiel 2.10.5 *In Beispiel 2.10.4 suchen wir λ_i sodass*

$$\begin{aligned}&\left(-1\cdot 2^2\cdot 3^2\cdot 5\right)^{\lambda_1}\left(2^2\cdot 5\cdot 7\right)^{\lambda_2}\left(3^2\cdot 5\cdot 7\right)^{\lambda_3}\\ &= (-1)^{\lambda_1}\cdot 2^{2\lambda_1+2\lambda_2}\cdot 3^{2\lambda_1+2\lambda_3}\cdot 5^{\lambda_1+\lambda_2+\lambda_3}\cdot 7^{\lambda_2+\lambda_3}\end{aligned}$$

ein Quadrat ist, also λ_i mit

$$\begin{aligned}\lambda_1 &\equiv 0 \bmod 2\\ 2\lambda_1+2\lambda_2 &\equiv 0 \bmod 2\\ 2\lambda_1+2\lambda_3 &\equiv 0 \bmod 2\\ \lambda_1+\lambda_2+\lambda_3 &\equiv 0 \bmod 2\\ \lambda_2+\lambda_3 &\equiv 0 \bmod 2\end{aligned}$$

Dies ist ein lineares Gleichungssystem in $\mathbb{F}_2 = \mathbb{Z}/2$. Anwenden des Gauß-Algorithmus gibt linear unabhängige Gleichungen

$$\begin{aligned}\lambda_1 &\equiv 0 \bmod 2\\ \lambda_2+\lambda_3 &\equiv 0 \bmod 2\end{aligned}$$

und damit die Lösungsmenge

$$\left\{\begin{pmatrix}0\\ -\lambda_3\\ \lambda_3\end{pmatrix} \mid \lambda_3\in\mathbb{F}_2\right\} = \left\{\begin{pmatrix}0\\0\\0\end{pmatrix},\begin{pmatrix}0\\1\\1\end{pmatrix}\right\}$$

Wir erhalten also $\lambda_1 = 0$ und $\lambda_2 = \lambda_3 = 1$, d.h.

$$x = 83^0\cdot 87^1\cdot 88^1.$$

2.10.3 Auswahl der Kongruenzen

Zur Durchführung des Algorithmus müssen wir geeignete s_i und kleine Primzahlen p_j wählen, sodass in den $f(s_i)$ nur Primfaktoren p_j vorkommen und das resultierende lineare Gleichungssystem eine nichttriviale Lösung hat. Die p_j spezifizieren wir über eine Konstante β durch die **Faktorbasis**

$$P_\beta = \{p \text{ prim} \mid p \le \beta\} \cup \{-1\},$$

wobei wir, um die Notation zu vereinfachen, auch die Einheit -1 dazunehmen.

Definition 2.10.6 *Eine Zahl m heißt β-glatt, wenn es $a_p \in \mathbb{N}_0$ gibt mit*

$$m = \prod_{p\in P_\beta} p^{a_p}$$

Diese Eigenschaft lässt sich durch Faktorisieren prüfen, dies ist jedoch teuer (wie wir schon empirisch festgestellt haben, und im nachfolgenden Abschnitt quantifizieren werden). Besser finden wir durch Sieben nach den $p \in P_\beta$ (d.h. Probedivision nach p) in einem durch eine Konstante C gegebenen **Siebintervall**

$$S = \{-C, -C+1, ..., 0, 1, ..., C-1, C\} \subset \mathbb{Z}$$

alle $s \in S$, sodass $f(s)$ β-glatt ist. Tatsächlich können wir diesen Ansatz mit der folgenden Bemerkung noch verbessern:

Bemerkung 2.10.7 *Sei p prim, und*

$$V_{\mathbb{F}_p}(f) = \{\overline{s} \in \mathbb{F}_p \mid f(\overline{s}) = 0\}$$

die Menge der Nullstellen von $f(x) = (x+m)^2 - n$ in $\mathbb{F}_p = \mathbb{Z}/p$. Da f Grad 2 hat ist $\left|V_{\mathbb{F}_p}(f)\right| \leq 2$.

Es gilt dann

$$\{s \in \mathbb{Z} \mid f(s) \equiv 0 \bmod p\} = \bigcup_{\overline{s} \in V_{\mathbb{F}_p}(f)} \overline{s}$$

wobei wir $\overline{s} = s + p\mathbb{Z}$ als Teilmenge von $\mathbb{Z}$ auffassen.

Beweis. Es wird $f(s)$ von p geteilt genau dann, wenn $f(s + k \cdot p)$ durch p geteilt wird, denn

$$\begin{aligned} f(s + k \cdot p) - f(s) &= (s + k \cdot p + m)^2 - (s+m)^2 \\ &= 2 \cdot s \cdot k \cdot p + k^2 \cdot p^2 + 2 \cdot k \cdot p \cdot m. \end{aligned}$$

■

Beispiel 2.10.8 *In Beispiel 2.10.4 wähle $\beta = 7$ also $P_\beta = \{-1, 2, 3, 5, 7\}$ und wähle $S = \{-3, ..., 3\}$. Dann liefert Sieben:*

s	-3	-2	-1	0	1	2	3
$f(s) = (s+m)^2 - n$	-540	-373	-204	-33	140	315	492
Sieb nach 2	-135	–	-51	–	35	–	123
Sieb nach 3	-5	–	-17	-11	–	35	41
Sieb nach 5	-1	–	–	–	7	7	–
Sieb nach 7	–	–	–	–	1	1	–

Somit sind $f(-3), f(1), f(2)$ die β-glatten Zahlen in S.

Mit Bemerkung 2.10.7 können wir schon a priori feststellen welche $f(s)$ durch welche Elemente p der Faktorbasis teilbar sind:

p	$f \bmod p$	$s \in \mathbb{F}_p : f(s) = 0$	$s \in S : f(s) \equiv 0 \bmod p$
2	$x^2 + 1$	$\overline{1}$	$.3, -1, 1, 3$
3	$x^2 + x$	$\overline{0}, \overline{2}$	$-3, -1, 0, 2, 3$
5	$x^2 + 2x + 2$	$\overline{1}, \overline{2}$	$-3, 1, 2$
7	$x^2 + 4x + 2$	$\overline{1}, \overline{2}$	$1, 2$

Wir teilen also für die so gefundenen $s \in S$ die $f(s)$ (eventuell mehrfach) durch die in der Tabelle korrespondierenden p. Nur die $f(s)$, für die wir so 1 oder -1 erhalten (und damit eine Faktorisierung in der Faktorbasis), verwenden wir zum Aufstellen des linearen Gleichungssystem. Dadurch sparen wir uns die Berechnung von $f(s)$ und Divisionen mit Rest (deren Aufwand wir im nachfolgenden Abschnitt diskutieren werden).

Bemerkung 2.10.9 *Ist die Anzahl der β-glatten $f(s)$, $s \in S$ größer als $\pi(\beta)+1$, dann ist das homogene lineare Gleichungssystem 2.1 nichttrivial lösbar.*

In Beispiel 2.10.5 hatten wir Glück. Obwohl das Gleichungssystem aus 5 Gleichung in 3 Variablen besteht, war der Rang nur 2.

Für die praktische Wahl von β und C und die Laufzeitanalyse muss man untersuchen, wieviele β-glatte Zahlen es in einem Siebintervall $S = \{-C, -C+1, ..., 0, 1, ..., C\}$ gibt.

2.11 Arithmetik und Laufzeitvergleich

Für eine Laufzeitanalyse des quadratischen Siebs angewendet auf eine Zahl n müssen wir uns überlegen, wie man den Aufwand des Algorithmus in Abhängigkeit von n misst. Wir untersuchen zunächst den wesentlich einfacheren umgekehrten Prozess, die Multiplikation. Dazu rufen wir uns nochmals kurz in Gedächtnis, wie Zahlen im Computer repräsentiert werden: Wie in Definition und Satz 2.1.10 diskutiert, verwendet man zur Darstellung von Zahlen die B-adischen Entwicklung

$$\begin{array}{rccc} \phi_{B,r}: & \{0,...,B-1\}^r & \longrightarrow & \{0,...,B^r-1\} \\ & (a_{r-1},...,a_0) & \longmapsto & \sum_{i=0}^{r-1} a_i B^i \end{array}$$

mit $B \geq 2$.

2.11.1 Addition und Multiplikation

Die Addition und Multiplikation von Zahlen in der B-adischen Entwicklung lassen sich dann wie in der Schule für $B = 10$ gelernt durchführen, siehe Algorithmus 2.8 und 2.9.

Algorithmus 2.8 Addition

Input: Zahlen $n, m \in \mathbb{Z}_{\geq 0}$ in B-adischer Entwicklung $\phi_{B,r}^{-1}(n) = (a_{r-1},...,a_0)$ und $\phi_{B,r}^{-1}(n) = (b_{r-1},...,b_0)$

Output: $\phi_{B,r+1}^{-1}(n+m)$

1: $u = 0$
2: **for** $i = 0,...,r-1$ **do**
3: schreibe $(a_i + b_i + u) =: u \cdot B + c_i$ mit $u \in \{0,1\}$ und $0 \leq c_i < B$.
4: $c_r = u$
5: **return** $(c_r,...,c_0)$

Algorithmus 2.9 Multiplikation

Input: Zahlen $n, m \in \mathbb{Z}_{\geq 0}$ in B-adischer Entwicklung $\phi_{B,r}^{-1}(n) = (a_{r-1}, ..., a_0)$ und $\phi_{B,s}^{-1}(m) = (b_{s-1}, ..., b_0)$

Output: $\phi_{B,r+s}^{-1}(n \cdot m)$

1: $c_0 = ... = c_{r+s} = 0$
2: **for** $i = 0, ..., r-1$ **do**
3: $\quad u = 0$
4: $\quad$ **for** $j = 0, ..., s-1$ **do**
5: $\quad\quad$ schreibe $(c_{i+j} + a_i \cdot b_j + u) =: u \cdot B + c_{i+j}$ mit $0 \leq u < B$ und $0 \leq c_{i+j} < B$.
6: $\quad c_{i+s} := u$
7: **return** $(c_{r+s-1}, ..., c_0)$

Der Multiplikationsalgorithmus ist korrekt:

Beweis. Wir haben $a_i \cdot b_j + c_{i+j} + u \leq (B-1)^2 + (B-1) + (B-1) = B^2 - 1$, also stets $u \leq \left\lfloor \frac{B^2-1}{B} \right\rfloor < B$. ■

Beispiel 2.11.1 *Für $B = 10$ ist*

$$\begin{aligned} 23 + 98 &= (2 \cdot 10^1 + 3 \cdot 10^0) + (9 \cdot 10^1 + 8 \cdot 10^0) \\ &= (2 \cdot 10^1 + 9 \cdot 10^1 + 1 \cdot 10^1) + 1 \cdot 10^0 \\ &= 1 \cdot 10^2 + 2 \cdot 10^1 + 1 \cdot 10^0 \\ &= 121 \end{aligned}$$

und

$$\begin{aligned} 23 \cdot 98 &= \left(2 \cdot 10^1 + 3 \cdot 10^0\right) \cdot \left(9 \cdot 10^1 + 8 \cdot 10^0\right) \\ &= 2 \cdot 10^1 \cdot \left(9 \cdot 10^1 + 8 \cdot 10^0\right) + 3 \cdot 10^0 \cdot \left(9 \cdot 10^1 + 8 \cdot 10^0\right) \\ &= \left(1 \cdot 10^3 + 9 \cdot 10^2 + 6 \cdot 10^1\right) + \left(2 \cdot 10^2 + 9 \cdot 10^1 + 4 \cdot 10^0\right) \\ &= 2 \cdot 10^3 + 2 \cdot 10^2 + 5 \cdot 10^1 + 4 \cdot 10^0 \\ &= 2254 \end{aligned}$$

wobei wir im Algorithmus die grünen Terme tatsächlich schon bei ihrer Berechnung auf die roten Terme addieren:

			10^3	10^2	10^1	10^0
$3 \cdot 10^0$	$\cdot$	$8 \cdot 10^0$			2	4
$3 \cdot 10^0$	$\cdot$	$9 \cdot 10^1$		2	9	4
$2 \cdot 10^1$	$\cdot$	$8 \cdot 10^0$		4	5	4
$2 \cdot 10^1$	$\cdot$	$9 \cdot 10^1$	2	2	5	4

Bei der Addition von zwei Zahlen mit r Stellen ergibt sich eine Zahl mit r Stellen oder $r+1$ Stellen (durch den Übertrag auf die nächste Stelle), bei der Multiplikation von zwei Zahlen mit r und s Stellen eine Zahl mit $r+s$ Stellen.

Zum Rechnen mit beliebig großen Zahlen verwendet man die B-adische Darstellung mit B eine Zweiterpotenz, typischerweise $B = 2^{64}$ und beliebiges r, also Listen mit r Einträgen aus 64-bit Zahlen. Als **B-Operationen** bezeichnen wir Addition, Multiplikation und Division mit Rest von Zahlen in $\{0, ..., B-1\}$. Diese Operationen haben dabei konstante Laufzeit, da sie direkt, unabhängig von Wert, in einer bestimmten festen Zahl von Taktzyklen vom Prozessor abgehandelt werden. Als **Laufzeit** bezeichnen wir die Anzahl der B-Operationen. Diese wird meist als Funktion in der Bitgröße des Inputs angegeben, d.h. in Termen der minimalen Zahl r, sodass der Input im Bild von $\phi_{B,r}$ liegt.

Um für eine Funktion $g : \mathbb{R}_{>0} \to \mathbb{R}$ das Wachstum von $g(x)$ für $x \to \infty$ zu beschreiben, führen wir folgende Notation ein:

Definition 2.11.2 *Für $f : \mathbb{R}_{>0} \to \mathbb{R}$ sei*

$$O(f) = \{g : \mathbb{R}_{>0} \to \mathbb{R} \mid \exists c, x_0 \in \mathbb{R} \text{ mit } |g(x)| \leq c \cdot |f(x)| \ \ \forall x \geq x_0\}$$

*Statt $g \in O(f)$ schreibt man in der sogenannten **Landau-Notation** auch $g = O(f)$.*

Beispiel 2.11.3 $2x^4 + 5x^3 - 2 \in O(x^4)$ *denn* $|2x^4 + 5x^3 - 2| \leq 9 \cdot |x^4|$ *für* $x \geq 1$. *Wir haben folgende Inklusionen*

beschränkt	$O(1)$	
	$\cap$	
logarithmisch	$O(\log(n))$	
	$\cap$	
polylogarithmisch	$O(\log(n)^c)$	*mit* $c \geq 1$
	$\cap$	
linear	$O(n)$	
	$\cap$	
polynomial	$O(n^c)$	*mit* $c \geq 1$
	$\cap$	
exponentiell	$O(c^n)$	*mit* $c > 1$

Bemerkung 2.11.4 *Die Laufzeit des Additionsalgorithmus 2.8 für Zahlen der Bitlänge r ist in $O(r)$.*

Beweis. Für jede Stelle haben wir 2 Additionen (Addition der Stellen und des Übertrags) und eine Division mit Rest, also insgesamt $3r$ Operationen. ■

Bemerkung 2.11.5 *Die Laufzeit des Multiplikationsalgorithmus 2.9 für Zahlen der Bitlänge r bzw. s ist in $O(r \cdot s)$.*

Beweis. Für jede der r Stellen der ersten Zahl und jede der s Stellen der zweiten Zahl haben wir 1 Multiplikation, 2 Additionen und 1 Division mit Rest. Insgesamt also $4rs$ Operationen. ■

Man beachte: Die Bestimmung des aktuellen Stellenindex $i+j$ in Zeile 5 von Algorithmus 2.9 kostet praktisch keine Zeit, da hierfür in jedem Schritt ein Register (d.h. Zähler) im Prozessor von $i,\ldots,i+s-1$ hochgezählt wird.

Asymptotisch kann die Multiplikation deutlich schneller durchgeführt werden:

Algorithmus 2.10 Karatsuba-Multiplikation

Wir teilen eine Zahl der Bitgröße $2r$

$$x = \sum_{i=0}^{2r-1} a_i B^i = a + b \cdot B^r$$

in zwei Zahlen $a, b < B^r$ *der halben Bitgröße* r*, ebenso*

$$y = c + d \cdot B^r$$

Dann gilt

$$\begin{aligned} x \cdot y &= b \cdot d \cdot B^{2r} + (a \cdot d + b \cdot c) \cdot B^r + a \cdot c \\ &= b \cdot d \cdot B^{2r} + (a \cdot c + b \cdot d + (b-a) \cdot (c-d)) \cdot B^r + a \cdot c \end{aligned}$$

Die erste Gleichung ersetzt 1 *Multiplikation von Zahlen der Bitlänge* $2r$ *durch* 4 *Multiplikationen*

$$b \cdot d,\ a \cdot d,\ b \cdot c,\ a \cdot c$$

und 1 *Addition der Bitgröße* r*. Dadurch wäre nichts gewonnen. Die zweite Gleichung zeigt aber, dass wir mit* 3 *Multiplikationen*

$$b \cdot d,\ a \cdot c,\ (b-a) \cdot (c-d)$$

und 4 *Additionen der Bitgröße* r *auskommen. Dieses Verfahren wendet man nun rekursiv auf die* 3 *Multiplikationen an.*

Proposition 2.11.6 *Die Laufzeit des Karatsuba-Algorithmus 2.10 ist in* $O(r^{\log_2(3)}) = O(r^{1.58\ldots})$.

Beweis. Für die Laufzeit $f(r)$ gilt die Rekursionsgleichung

$$f(r) = 3 \cdot f\left(\frac{r}{2}\right) + c \cdot r$$

mit einer Konstanten $c > 0$ (wobei der erste Term den 3 Multiplikationen halber Bitgröße entspricht und der zweite Term den Additionen).

Mit $m = \log_2(r)$ ist also induktiv

$$\begin{aligned} f(2^m) &= 3 \cdot f(2^{m-1}) + c \cdot 2^m = 3^2 \cdot f(2^{m-2}) + 3 \cdot c \cdot 2^{m-1} + c \cdot 2^m = \ldots \\ &= 3^m \cdot f(1) + c \cdot 2^m \cdot \sum_{i=0}^{m-1} \left(\frac{3}{2}\right)^i \\ &= 3^m \cdot f(1) + 2c \cdot (3^m - 2^m) \\ &\leq 3^m \cdot (f(1) + 2c) \\ &= r^{\log_2(3)} \cdot (f(1) + 2c) \\ &\in O(r^{\log_2(3)}) \end{aligned}$$

wobei wir die geometrische Summenformel

$$\sum_{i=0}^{m-1} \left(\frac{3}{2}\right)^i = \frac{1 - \left(\frac{3}{2}\right)^m}{1 - \frac{3}{2}} = 2 \cdot \frac{3^m - 2^m}{2^m}$$

und

$$3^m = \exp\left(\frac{\ln(3) \cdot \ln(r)}{\ln(2)}\right) = r^{\log_2(3)}$$

verwenden. ■

Wir sparen also einen Großteil der B-Multiplikationen. Bei kleinem r fallen allerdings die zusätzlichen Additionen ins Gewicht. Deshalb verwendet man den Karatsuba-Algoirthmus nur für große r (praktisch ab etwa 600 Dezimalstellen der zu multiplizierenden Zahlen).

2.11.2 Division mit Rest

Im diesem Abschnitt wollen wir die Schulbuchdivision formalisieren und die Laufzeit analysieren. Sei $B \in \mathbb{Z}$, $B \geq 2$ und seien zwei Zahlen

$$x = \sum_{i=0}^{u-1} a_i B^i \qquad y = \sum_{i=0}^{v-1} b_i B^i$$

in B-adischer Entwicklung zur Basis B mit $a_{u-1} \neq 0$, $b_{v-1} \neq 0$ gegeben. Das wesentliche Problem ist die effiziente Bestimmung von q in der Divisionsgleichung

$$x = q \cdot y + r$$

Bei der Division von Hand in der Schule löst man dieses Problem wenig formal, für eine Laufzeitanalyse müssen wir genauer vorgehen.

Beispiel 2.11.7 *Wollen wir im Dezimalsystem per Schulbuchdivision* $x = 1000$ *durch* $y = 3$ *teilen, dann multiplizieren wir* y *in Gedanken mit der maximalen* 10*-er Potenz* 10^t*, sodass noch* $y \cdot 10^t \leq x$*. Wir teilen also* 1000 *durch* 300*. Durch diesen Trick erreichen wir, dass in der Divisionsgleichung*

$$x = q_1 \cdot y \cdot 10^t + x_1$$

also

$$1000 = 3 \cdot 300 + 100$$

die Zahl

$$q_1 = \left\lfloor \frac{x}{y \cdot 10^t} \right\rfloor < 10$$

also einstellig ist. Damit haben wir die erste Stelle $q_1 = 3$ *der Zahl* $q = 333$ *in der Divisionsgleichung*

$$1000 = 333 \cdot 3 + 1$$

gefunden.

Diesen Prozess können wir iterieren, denn ist t_1 maximal mit

$$y \cdot 10^{t_1} \leq x_1 < y \cdot 10^t$$

dann muss

$$t_1 < t$$

sein. Für einen einzelnen Divisionsschritt können wir also annehmen, dass

$$\frac{x}{y} < B$$

gilt. Im Folgenden werden wir zeigen, dass sich unter dieser Voraussetzung leicht eine gute Approximation für q mit $x = q \cdot y + r$ finden lässt:

Lemma 2.11.8 *Mit Notation und Voraussetzungen wie oben, gilt für die Zahl*

$$\widetilde{q} := \min\left\{B - 1, \left\lfloor \frac{a_v \cdot B + a_{v-1}}{b_{v-1}} \right\rfloor\right\} \in \mathbb{N}_0$$

dass

$$q \leq \widetilde{q}.$$

Wir verwenden dabei die Konvention $a_i = 0$ für $i \geq u$. Wegen $\frac{x}{y} < B$ kann die Zahl x maximal eine Stelle mehr als y haben, also ist $u \leq v+1$. Man beachte:

Bemerkung 2.11.9 *Die Zahl* $\widetilde{q}$ *lässt sich durch* B*-Operationen (Multiplikation, Addition und Division mit Rest) und damit in konstanter Zeit bestimmen.*

Wir können auch noch eine untere Schranke für q finden:

Lemma 2.11.10 *Für*

$$b_{v-1} \geq \left\lfloor \frac{B}{2} \right\rfloor$$

ist $\widetilde{q} - 2 \leq q$*, zusammen mit Lemma 2.11.8 also*

$$\widetilde{q} - 2 \leq q \leq \widetilde{q}.$$

Entscheidend ist hier, dass der Fehler von $\widetilde{q}$ im Vergleich zu q nicht von B abhängt, das Verfahren ist also auch für sehr große B z.B. $B = 2^{64}$ effizient. Lemma 2.11.8 und Lemma 2.11.10 zeigt man durch direktes Nachrechnen, siehe dazu Übung 2.28.

Beispiel 2.11.11 *Im Dezimalsystem teilen wir* $x = 3000$ *durch* $y = 666$ *und erhalten*

$$\widetilde{q} = \frac{30}{6} = 5,$$

die möglichen Werte für q *sind also* $3, 4, 5$ *und durch Ausprobieren finden wir*

$$q = 4$$

mit

$$3000 = 4 \cdot 666 + 336.$$

Für den Fall $b_{v-1} < \left\lfloor \frac{B}{2} \right\rfloor$ berechnen wir $\widetilde{q}$ aus einem um eine geeignete ganze Zahl k erweiterten Bruch. Man beachte, dass sich

$$q = \left\lfloor \frac{x}{y} \right\rfloor = \left\lfloor \frac{kx}{ky} \right\rfloor.$$

durch Erweitern nicht ändert.

Lemma 2.11.12 *Ist*

$$b_{v-1} < \left\lfloor \frac{B}{2} \right\rfloor$$

dann ersetze x *und* y *durch* kx *und* ky *mit*

$$k := \left\lfloor \frac{B}{b_{v-1} + 1} \right\rfloor.$$

Es gilt dann wieder

$$\widetilde{q} - 2 \leq q \leq \widetilde{q}.$$

Beispiel 2.11.13 *Teilen wir* $x = 2000$ *durch* $y = 299$*, dann ist* $k = 3$ *und wir betrachten*

$$kx = 6000 \text{ und } ky = 897.$$

Daraus erhalten wir

$$\widetilde{q} = 7,$$

und wir finden durch Ausprobieren $q = 6$ *mit*

$$2000 = 6 \cdot 299 + 206.$$

Zum Beweis von Lemma 2.11.12:

Beweis. Schreibe $w = b_{v-1}$. Es ist

$$\begin{aligned} k \cdot w &= \left\lfloor \frac{B}{w+1} \right\rfloor \cdot w > \left(\frac{B}{w+1} - 1 \right) \cdot w \\ &\geq \frac{B}{2} - 1 \geq \left\lfloor \frac{B}{2} \right\rfloor - 1, \end{aligned}$$

denn wegen $1 \leq w < \left\lfloor \frac{B}{2} \right\rfloor$ ist

$$\left(\frac{B}{w+1} - 1 \right) \cdot w - \left(\frac{B}{2} - 1 \right) = \left(\frac{B}{2} - w - 1 \right) (w-1) \frac{1}{w+1} \geq 0.$$

Somit muss

$$k \cdot w \geq \left\lfloor \frac{B}{2} \right\rfloor$$

sein. Um Lemma 2.11.10 anwenden.zu können, müssen wir aber noch zeigen, dass bei der Berechnung der $(v-1)$-ten Stelle w' von $k \cdot y$ kein Übertrag auf die v-te Stelle stattfindet. Bei der Berechnung des Produkts von k mit y erhalten wir die $(v-1)$-te Stelle w' von $k \cdot y$ (mit einem eventuellen Übertrag aus niedrigeren Stellen) als

$$w' = k \cdot w + \left\lfloor \frac{k}{B^{v-1}} (y - b_{v-1} B^{v-1}) \right\rfloor.$$

Um dies nach oben abzuschätzen bemerken zunächst, dass

$$y - b_{v-1} B^{v-1} \leq B^{v-1} - 1$$

(denn $y - b_{v-1} B^{v-1}$ hat maximal $v-1$ Stellen) und

$$B - k = B - \left\lfloor \frac{B}{w+1} \right\rfloor \geq \left\lfloor B - \frac{B}{w+1} \right\rfloor = \left\lfloor \frac{B \cdot w}{w+1} \right\rfloor \geq \left\lfloor \frac{B}{w+1} \right\rfloor \cdot w = k \cdot w.$$

Damit folgt

$$\begin{aligned} w' &\leq B - k + \left\lfloor k \frac{B^{v-1} - 1}{B^{v-1}} \right\rfloor \\ &\leq B - k + k \frac{B^{v-1} - 1}{B^{v-1}} < B. \end{aligned}$$

Die höchste Stelle von $k \cdot y$ ist also

$$w' \geq k \cdot w \geq \left\lfloor \frac{B}{2} \right\rfloor.$$

■

Mit Lemma 2.11.10 und Lemma 2.11.12 folgt dann sofort:

Corollar 2.11.14 *Für*

$$\frac{x}{y} < B$$

lässt sich bei der Division mit Rest die Zahl q in höchstens 3 Versuchen bestimmen.

Bemerkung 2.11.15 *Sind $x, y \in \mathbb{N}$ mit Bitlänge $u \geq v$ und hat $q = \left\lfloor \frac{x}{y} \right\rfloor$ Bitlänge $\leq u - v + 1$ (Übung). Was ist die minimal mögliche Bitlänge von q?*

Satz 2.11.16 *Für die Division mit Rest*

$$x = q \cdot y + r$$

von zwei Zahlen $x, y \in \mathbb{N}$ der Bitlänge $u \geq v$ genügen

$$O(t \cdot v) \subset O((u - v + 1) \cdot v) \subset O(u \cdot v)$$

B-Operationen, wobei t die Bitgröße von q bezeichnet.

Beweis. Wie in Beispiel 2.11.7 schon diskutiert gehen wir im allgemeinen Fall iterativ vor. Seien $x, y \in \mathbb{N}$ mit Bitlängen u bzw. v. Wir dividieren x durch y wie folgt: Wähle t maximal mit $yB^t \leq x$. Setze $x_0 := x$ und führe für $k = 0, \ldots, t$ iterativ den Elementarschritt

$$x_k = q_k \cdot yB^{t-k} + x_{k+1}$$

durch. Damit erhalten wir

$$q = \sum_{k=0}^{t} q_k B^{t-k} = \left\lfloor \frac{x}{y} \right\rfloor$$

mit

$$x = q \cdot y + r \text{ und } 0 \leq r < y.$$

Gemäß Corollar 2.11.14 lässt sich die Bestimmung von q_k in jedem Elementarschritt in 3 Versuchen durchführen. Mit Bemerkung 2.11.4 zur Addition und Bemerkung 2.11.5 zur Multiplikation kann man den Rest bei jedem der 3 Versuche im Elementarschritt in $O(v)$ Operationen berechnen: Eine Multiplikation von y mit der 1-stelligen Zahl q_k und eine Addition von zwei Zahlen mit v Stellen (beachte, dass die letzten $t - k$ Stellen von $q_k \cdot yB^{t-k}$ alle Null sind). Die komplette Division mit Rest im Elementarschritt benötigt also $O(v)$ Operationen. Da $t \leq u - v + 1$ Elementarschritte durchgeführt werden, benötigt die Division höchstens

$$O(t \cdot v)$$

Operationen. ■

Beispiel 2.11.17 *Bei der Division von $x = 9000$ durch $y = 44$ ist $t = 2$ und der erste Elementarschritt für $k = 0$ berechnet*

$$9000 = 2 \cdot 4400 + 200.$$

Im $k = 1$ Schritt finden wir

$$200 = 0 \cdot 440 + 200,$$

im $k = 2$ Schritt schließlich

$$200 = 4 \cdot 44 + 24.$$

Insgesamt ist also $q = 204$ und die wir erhalten den Divisionsausdruck

$$9000 = 204 \cdot 44 + 24.$$

Mit dem Satz können wir die Laufzeit des Euklidischen Algorithmus abschätzen:

Corollar 2.11.18 *Für die Berechnung des größten gemeinsamen Teilers von zwei Zahlen der Bitlänge u und v mit dem Euklidischen Algorithmus 2.1 genügen*

$$O(u \cdot v)$$

B-Operationen.

Zum Beweis zeigen wir in Übung 2.29 zunächst eine obere Schranke für die Anzahl der Divisionen mit Rest im Euklidischen Algorithmus und schätzen damit dann in Übung 2.30 die Laufzeit ab.

2.11.3 Quadratisches Sieb

Beim Quadratischen Sieb hängt die Laufzeit von der Wahl des Siebintervalls $\{-C, \ldots, C\}$ und der Faktorbasis $P_\beta = \{p \text{ prim} \mid p \leq \beta\}$ ab. Zur Beschreibung der Laufzeit und der Konstanten C und β betrachten wir eine Schar von Funktionen, die zwischen polynomial und exponentiell interpoliert:

$$n \mapsto L_n[u, v] = e^{v(\log n)^u (\log\log n)^{1-u}}$$

Dabei ist $\log n = \log_2 n$ die Bitgröße von n.

Beispiel 2.11.19 *Insbesondere haben wir*

$$\begin{array}{ll} L_n[0, v] = (\log n)^v & \textit{polynomial} \\ L_n[1, v] = e^{v \log n} & \textit{exponentiell} \\ & \textit{in } \log n \end{array}$$

Also $L_n[u, v]$ mit $u \in]0, 1[$ ist eine Funktion, die schneller als polynomial und langsamer als exponentiell wächst. Eine solche Laufzeit bezeichnet man auch als ***subexponentiell****.*

Man kann zeigen, dass unter plausiblen, überprüfbaren Annahmen die Laufzeit des quadratischen Siebs in

$$O\left(L_n\left[\frac{1}{2}, 1 + \varepsilon(n)\right]\right) = O\left(\left(e^{\sqrt{\log n \log\log n}}\right)^{1+\varepsilon(n)}\right)$$

liegt, wobei $\lim_{n\to\infty} \varepsilon(n) = 0$, falls wir

$$C = L_n\left[\frac{1}{2}, 1\right] = e^{\sqrt{\log n \log\log n}}$$
$$\beta = L_n\left[\frac{1}{2}, \frac{1}{2}\right] = e^{\frac{1}{2}\sqrt{\log n \log\log n}} = \sqrt{C}$$

wählen. Eine vollständige Laufzeitanalyse gibt es jedoch noch nicht.

Das quadratische Sieb hat also subexponentielle Laufzeit. Die Laufzeit der Probedivision ist dagegen exponentiell:

Bemerkung 2.11.20 *Die Laufzeit der Probedivision: Wir testen alle Primzahlen* $p \le m = \lfloor\sqrt{n}\rfloor$.

1) Die Division mit Rest von n *durch* p *hat Zeitaufwand in*

$$O\left((\log p)\left(\log\frac{n}{p}\right)\right).$$

2) Für die Anzahl

$$\pi(m) = \{p \text{ Primzahl} \mid p \le m\}$$

der Primzahlen $\le m$ *gilt nach dem Primzahlsatz 2.3.6*

$$\pi(m) \approx \frac{m}{\ln m}$$

3) Damit ist der Gesamtaufwand der Probedivision in

$$O\left(\left(\frac{\sqrt{n}}{\log\sqrt{n}}\right)(\log\sqrt{n})(\log n)\right) = O\left(\sqrt{n}\log n\right) = O(e^{\frac{1}{2}\log n}\cdot\log n)$$

also exponentiell in $\log n$.

Dennoch ist die Probedivision das effizienteste Verfahren für Zahlen bis etwa zu der Grössenordnung 10^6.

Konkret gibt der Vergleich zwischen Multiplikation und Faktorisierungsverfahren

Multiplikation		quadratisches Sieb		Probedivision
$(\log n)^2$	$\ll$	$e^{\sqrt{\log(n)\cdot\log(\log(n))}}$	$\ll$	$e^{\frac{1}{2}\log n}\cdot\log n$
polynomial		subexponentiell		exponentiell

siehe Abbildung 2.1 für diese Funktionen in Abhängigkeit von $r = \log n$. Für die zweite Abschätzung beachte, dass

$$\sqrt{\log(n)\cdot\log(\log(n))} \ll \sqrt{\frac{1}{4}\log(n)\cdot\log(n)} = \frac{1}{2}\log(n)$$

für großes n.

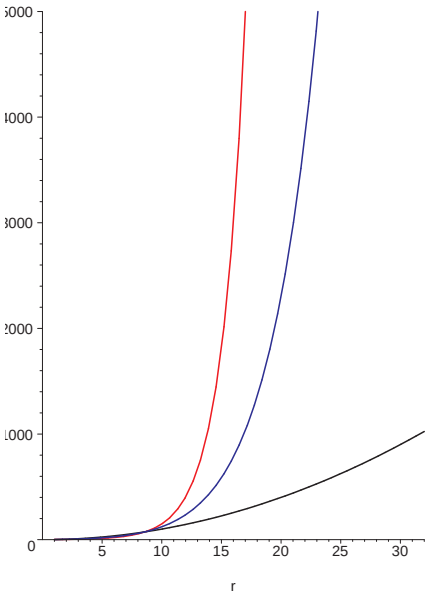


Abbildung 2.1: Laufzeitvergleich zwischen Multiplikation, quadratischem Sieb und Probedivision

2.12 Übungen

Übung 2.1 *Installieren bzw. starten Sie* JULIA, NEMO, SINGULAR, GAP, *und* SURFER *und probieren Sie die Beispiele aus Kapitel 1 aus. Schlagen Sie die verwendeten Kommandos in der Online-Hilfe nach.*

Übung 2.2 *Zeigen Sie:*

1) *Auf* $M = \mathbb{N}_0 \times \mathbb{N}_0$ *ist durch*

$$(a, b) \sim (c, d) \Leftrightarrow a + d = b + c$$

eine Äquivalenzrelation gegeben (das heißt, $\sim$ *ist reflexiv, symmetrisch und transitiv).*

2) *Die Verknüpfungen*

$$[(a, b)] + [(c, d)] = [(a + c, b + d)]$$
$$[(a, b)] \cdot [(c, d)] = [(a \cdot c + b \cdot d, a \cdot d + b \cdot c)]$$

auf

$$\mathbb{Z} = (\mathbb{N}_0 \times \mathbb{N}_0) / \sim$$

sind wohldefiniert (das heißt, das Resultat hängt nicht von der Wahl des Repräsentanten (a, b) *von* $[(a, b)]$ *ab).*

3) *Mit der Identifikation* $a - b = [(a, b)]$ *korrespondieren sie zu den üblichen Verknüpfungen auf* $\mathbb{Z}$.

4) Das neutrale Element bezüglich der Addition ist $0 = [(0,0)]$ *und das Inverse*

$$-[(a,b)] = [(b,a)]$$

5) $(\mathbb{Z}\backslash\{0\},\cdot)$ *ist eine Halbgruppe mit neutralem Element* $1 = [(1,0)]$.

Übung 2.3 *Sei* R *ein Integritätsring und* $S = R\backslash\{0\}$. *Wir konstruieren den Ring von Brüchen*

$$Q(R) = \left\{ \frac{r}{s} \mid r \in R,\ s \in S \right\}$$

als $Q(R) = (R \times S) / \sim$ *mit der Äquivalenzrelation*

$$(r,s) \sim (r',s') \Leftrightarrow rs' - sr' = 0$$

und schreiben $\frac{r}{s} := [(r,s)]$. *Addition und Multiplikation sind gegeben durch*

$$\frac{r_1}{s_1} + \frac{r_2}{s_2} = \frac{r_1 s_2 + r_2 s_1}{s_1 s_2}$$
$$\frac{r_1}{s_1} \cdot \frac{r_2}{s_2} = \frac{r_1 r_2}{s_1 s_2}$$

Zeigen Sie:

1) Addition und Multiplikation sind wohldefiniert, und $Q(R)$ *ist ein Körper.*

2) Die Abbildung $j : R \to Q(R)$, $r \mapsto \frac{r}{1}$ *ist ein Monomorphismus.*

Übung 2.4 *1) Implementieren Sie den Euklidischen Algorithmus zur Berechnung des größten gemeinsamen Teilers* $\mathrm{ggT}(a,b)$ *von* $a, b \in \mathbb{Z}$.

2) Kürzen Sie

$$\frac{90189116021}{18189250063}$$

Übung 2.5 *Seien* $a, b, d \in \mathbb{Z}$. *Zeigen Sie:*

1) Die Gleichung

$$ax + by = d$$

ist genau dann nach $(x,y) \in \mathbb{Z}^2$ *lösbar, wenn*

$$\mathrm{ggT}(a,b) \mid d.$$

2) Ist (x,y) *eine Lösung, dann auch*

$$\left(x + k \cdot \frac{b}{\mathrm{ggT}(a,b)},\ y - k \cdot \frac{a}{\mathrm{ggT}(a,b)} \right)$$

mit $k \in \mathbb{Z}$, *und alle Lösungen sind von dieser Form.*

3) Bestimmen Sie alle Lösungen $(x, y) \in \mathbb{Z}^2$ *von*

$$42 \cdot x + 55 \cdot y = 1.$$

Übung 2.6 *Testen Sie den Primzahlsatz:*

1) Schreiben Sie eine Prozedur zur Berechnung von

$$\pi(x) = |\{p \leq x \mid p \in \mathbb{N} \text{ prim}\}|$$

für $x > 0$.

2) Plotten Sie $\frac{\pi(x)}{x}$ *und* $\frac{1}{\ln(x)}$ *und vergleichen Sie für großes* x.

3) Zeigen Sie, dass die nach dem Primzahlsatz erwartete Zahl von Fermat Primzahlen endlich ist.

Zur Erinnerung: Eine Fermat-Primzahl ist eine Primzahl der Form

$$F_n = 2^{2^n} + 1$$

mit $n \in \mathbb{N}_0$.

Hinweis: Das JULIA/NEMO Kommando `prime`, das MAPLE Kommando `nextPrime` oder das SINGULAR Kommando `prime` können nützlich sein.

Übung 2.7 *Sei* P_N *die Wahrscheinlichkeit, dass zufällig gewählte natürliche Zahlen* $n, m \leq N$ *teilerfremd sind. Zeigen Sie, dass für den Grenzwert gilt*

$$\lim_{N \to \infty} P_N = \frac{6}{\pi^2} \approx 60.7\%$$

Hinweis: Verwenden Sie ohne Beweis die Formel

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{1}{6}\pi^2$$

die man z.B. mit Hilfe von Fourierreihen beweisen kann.

Übung 2.8 *Zeigen Sie:*

1) Ist $r \in \mathbb{N}$ *und* $p = 2^r - 1$ *prim, dann ist* r *prim.*

2) Ist $r \in \mathbb{N}$ *und* $p = 2^r + 1$ *prim, dann ist* $r = 2^k$ *mit* $k \in \mathbb{N}_0$.

Übung 2.9 *Sei* $f = x^3 + 6x^2 + 14x + 9$ *und* $g = x^2 + 5x + 6$. *Führen Sie sowohl in* $\mathbb{Q}[x]$ *als auch in* $\mathbb{F}_3[x]$ *die Division mit Rest von* f *nach* g *durch. Überprüfen Sie Ihr Ergebnis mit* JULIA/NEMO *oder* SINGULAR.

Übung 2.10 *Bestimmen Sie den größten gemeinsamen Teiler*

$$\mathrm{ggT}(x^6 - 1, \quad x^4 + x^3 + 2x^2 + x + 1) \in \mathbb{Q}[x]$$

mit Hilfe von JULIA/NEMO *oder* SINGULAR.

Übung 2.11 *Sei p eine Primzahl und $\mathbb{F}_p = \mathbb{Z}/p = \{\overline{0}, ..., \overline{p-1}\}$ der Körper mit p Elementen.*

1) *Finden Sie, analog zum Sieb von Eratosthenes, alle irreduziblen Polynome in $\mathbb{F}_2[x]$ vom Grad ≤ 3.*

2) *Faktorisieren Sie $x^5 + x^2 + x + 1 \in \mathbb{F}_2[x]$ in ein Produkt von irreduziblen Polynomen.*

Übung 2.12 *Bestimmen Sie die Einheiten und Nullteiler von $\mathbb{Z}/12$ und die Multiplikationstabelle der Einheitengruppe.*

Übung 2.13 *Schreiben Sie eine Prozedur, die für $n \in \mathbb{N}$ den Wert $\varphi(n)$ der Eulerschen φ-Funktion berechnet.*

Die folgenden JULIA/NEMO*-Kommandos können nützlich sein:* `gcd` *zur Berechnung des größten gemeinsamen Teilers,* `while`, `end` *für Schleifen,* `if`, `end` *für bedingte Anweisungen,* `function`, `return`, `end`*, um aus Ihrem Code eine Prozedur zu erzeugen. Schlagen Sie die Syntax der Kommandos in der Online-Hilfe nach.*

Übung 2.14 *Zeigen Sie:*

1) *Jeder Integritätsring mit endlich vielen Elementen ist ein Körper.*

2) *In einem endlichen Ring ist jedes Element entweder eine Einheit oder ein Nullteiler.*

Übung 2.15 *Bestimmen Sie die Menge $L \subset \mathbb{Z}$ aller Lösungen x der simultanen Kongruenzen*

$$x \equiv 1 \bmod 3$$
$$x \equiv 2 \bmod 5$$
$$x \equiv 3 \bmod 7$$

Übung 2.16 *Zeigen Sie: Sind $n, m \in \mathbb{Z}_{>0}$ teilerfremd, dann gilt*

$$\mathbb{Z}/nm \cong (\mathbb{Z}/n) \times (\mathbb{Z}/m).$$

Berechnen Sie das Urbild von $(\overline{1}, \overline{2}, \overline{3})$ unter dem Isomorphismus

$$\mathbb{Z}/105 \cong (\mathbb{Z}/3) \times (\mathbb{Z}/5) \times (\mathbb{Z}/7).$$

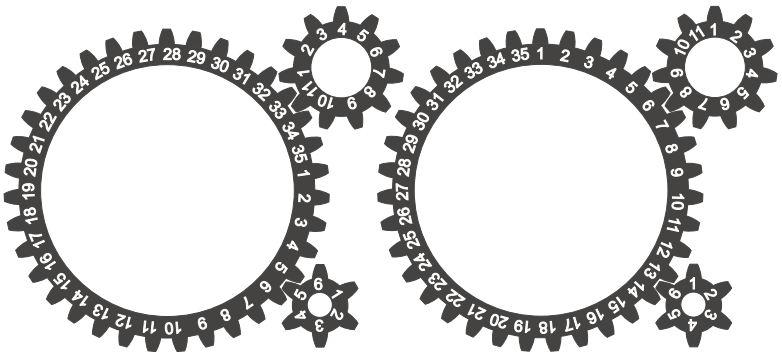


Abbildung 2.2: Zwei Konfigurationen von drei Zahnrädern

Übung 2.17 *Lassen sich die beiden Konfigurationen von Zahnrädern in Abbildung 2.2 durch Drehung ineinander überführen? Falls ja, um wieviele Schritte muss man dafür drehen?*

Übung 2.18 *Zeigen Sie: Die simultanen Kongruenzen*

$$x \equiv a_1 \bmod n_1$$
$$x \equiv a_2 \bmod n_2$$

sind genau dann lösbar, wenn

$$a_1 - a_2 \equiv 0 \bmod \mathrm{ggT}\,(n_1, n_2)$$

Die Lösung ist eindeutig modulo dem $\mathrm{kgV}\,(n_1, n_2)$.

Übung 2.19 *Bestimmen Sie die Menge* $L \subset \mathbb{Z}$ *aller Lösungen* x *der simultanen Kongruenzen*

$$x \equiv 1 \bmod 108$$
$$x \equiv 13 \bmod 40$$
$$x \equiv 28 \bmod 225$$

Übung 2.20 *Für* $f, g, h \in K[x]$ *und* $h \neq 0$ *schreiben wir* $f \equiv g \bmod h$ *falls* $h \mid (f - g)$.

1) *Bestimmen Sie die Menge* $L \subset \mathbb{R}[x]$ *aller Lösungen* f *der simultanen Kongruenzen*

$$f \equiv 2 + 3\,(x - 1) \bmod (x - 1)^2$$
$$f \equiv 1 + 2\,(x + 1) \bmod (x + 1)^2$$

Entwickeln Sie dazu ein Lösungsverfahren analog zu dem Verfahren zur Lösung simultaner Kongruenzen über $\mathbb{Z}$.

2) Zeigen Sie, dass L ein eindeutiges Polynom $f \in K[x]$ vom Grad $\deg(f) \leq 3$ enthält mit

$$\begin{aligned} f(1) &= 2 & f'(1) &= 3 \\ f(-1) &= 1 & f'(-1) &= 2 \end{aligned}$$

Übung 2.21 *Sei $R = \mathbb{Z}[i] = \{a_1 + i \cdot a_2 \mid a_1, a_2 \in \mathbb{Z}\} \subset \mathbb{C}$ mit $i^2 = -1$. Zeigen Sie, dass R zusammen mit*

$$\begin{array}{rccc} d: & R\backslash\{0\} & \to & \mathbb{N}_0 \\ & a + b \cdot i & \mapsto & a^2 + b^2 \end{array}$$

ein euklidischer Ring ist.

1) Geben Sie ein Verfahren an, um die Division mit Rest durchzuführen.

2) Bestimmen Sie den grössten gemeinsamen Teiler

$$\mathrm{ggT}(3 + 4i,\ 1 - 4i) \in \mathbb{Z}[i]\,.$$

Inwiefern ist der größte gemeinsame Teiler eindeutig?

Hinweis: Berechnen Sie zur Division mit Rest von $a + b \cdot i$ durch $c + d \cdot i$ zunächst

$$\frac{a + b \cdot i}{c + d \cdot i} \in \mathbb{Q}[i] = \{a_1 + i \cdot a_2 \mid a_1, a_2 \in \mathbb{Q}\}\,.$$

Übung 2.22 *Der öffentliche RSA-Schlüssel von Alice ist*

$$\begin{aligned} n_A &= 191372480359498044048987808676864667665690167017... \\ &\quad ...150163809808649670406431450799396239185563819 63 \\ e_A &= 2^{16} + 1 \end{aligned}$$

Bob hat eine verschlüsselte Nachricht

$$\begin{aligned} c = 10431252108163715124564523812373627504873232094... \\ ...46483822475432640249389840891211411467552511126 5 \end{aligned}$$

an Alice geschickt. Was war der Inhalt der Nachricht?

Hinweise: Alice hat ungeschickterweise einen Primfaktor p von $n_A = p \cdot q$ gewählt, sodass $\varphi(p)$ nur Primpotenzfaktoren ≤ 200000 hat.

Um für $a, b, n \in \mathbb{N}$ effizient $a^b \bmod n$ zu berechnen, gibt es in JULIA/NEMO *das Kommando*

```
powmod(a,b,n).
```

(das im wesentlichen sukzessive modulo n mit a multipliziert).

Testen Sie, ob auch die JULIA*-Funktion* `factor` *zum Ziel führt.*

Übung 2.23 *Der Fermatsche Primzahltest: n heißt Fermatsche Pseudoprimzahl zur Basis a, wenn n nicht prim ist, aber dennoch $a^{n-1} \equiv 1 \bmod n$ gilt.*

1) *Bestimmen Sie mit Computerhilfe jeweils alle Pseudoprimzahlen $n \leq 2000$ zur Basis a mit $a = 2, 3, 5$ und vergleichen Sie deren Anzahl mit der Anzahl der Primzahlen.*

2) *Eine Zahl n heißt Carmichael-Zahl, wenn sie zu jeder Basis a mit $\mathrm{ggT}(a, n) = 1$ eine Fermatsche Pseudoprimzahl ist. Haben Sie eine Vermutung für eine Carmichael-Zahl?*

Hinweis: JULIA/NEMO-*Funktionen* `prime` *und* `rem`.

Übung 2.24 *Sei $n \in \mathbb{Z}_{\geq 2}$. Für ein $a \in \mathbb{Z}$ gelte $a^{n-1} \equiv 1 \bmod n$ und $a^{\frac{n-1}{p}} \not\equiv 1 \bmod n$ für jeden Primteiler p von $n-1$. Zeigen Sie, dass dann n prim ist.*

Übung 2.25 *Sei x in einem kommutativen Ring R mit 1 und $n \in \mathbb{N}$.*

Übung 2.26 1) *Zeigen Sie:Die Anzahl der Multiplikationen in Algorithmus 2.5 zur Berechnung von x^n ist in $O(\log_2(n))$.*

2) *Wenden Sie das Verfahren an, um 3^{11} in $\mathbb{Z}$ und $\overline{3}^{11}$ in $\mathbb{Z}/7$ zu berechnen.*

Übung 2.27 1) *Sei $n \in \mathbb{N}$ zusammengesetzt. Zeigen Sie, dass n eine Carmichael-Zahl ist genau dann, wenn für alle Primteiler p von n gilt, dass*

$$p^2 \nmid n$$

und

$$(p-1) \mid (n-1).$$

2) *Ist $k \in \mathbb{N}$ mit $6k+1$, $12k+1$ und $18k+1$ prim, dann ist*

$$(6k+1) \cdot (12k+1) \cdot (18k+1)$$

eine Carmichael-Zahl.

3) *Konstruieren Sie eine Carmichael-Zahl $\geq 10^{50}$. Können Sie die Zahl mit* JULIA/NEMO *faktorisieren? Führt das Pollard-Verfahren aus Abschnitt 2.8 zu Ziel?*

Übung 2.28 *Sei $B \in \mathbb{Z}$, $B \geq 2$ und seien zwei Zahlen*

$$x = \sum_{i=0}^{m-1} a_i B^i \qquad y = \sum_{i=0}^{n-1} b_i B^i$$

in B-adischer Entwicklung zur Basis B mit $a_{m-1}, b_{n-1} \neq 0$ und

$$\frac{x}{y} < B$$

gegeben. Sei weiter $x = q \cdot y + r$ mit $0 \leq r < y$ das Resultat der Division mit Rest von x durch y. Wir definieren $\widetilde{q}$ als das Minimum von $B-1$ und

$$\left\lfloor \frac{a_n \cdot B + a_{n-1}}{b_{n-1}} \right\rfloor$$

Wir setzen dabei $a_i = 0$ für $i \geq m$.

1) *Bestimmen Sie q und $\widetilde{q}$ für $B = 10$ und*

$$x = 3142351 \qquad y = 677688.$$

2) *Zeigen Sie allgemein, dass*

$$q \leq \widetilde{q}.$$

3) *Zeigen Sie weiter: Ist*

$$b_{n-1} \geq \left\lfloor \frac{B}{2} \right\rfloor$$

dann gilt

$$\widetilde{q} \leq q + 2.$$

4) *Folgern Sie, dass unter den obigen Voraussetzungen bei der Division mit Rest die Zahl q in höchstens $3 \in O(1)$ Versuchen gefunden werden kann.*

Übung 2.29 *Seien $a_1 \geq a_2 \geq 0$ ganze Zahlen und*

$$\begin{aligned}
a_1 &= q_1 \cdot a_2 + a_3 \\
&\vdots \\
a_j &= q_j \cdot a_{j+1} + a_{j+2} \\
&\vdots \\
a_{n-2} &= q_{n-2} \cdot a_{n-1} + a_n \\
a_{n-1} &= q_{n-1} \cdot a_n + 0
\end{aligned}$$

die sukzessive Division mit Rest im Euklidischen Algorithmus, und sei

$$\phi = \frac{1 + \sqrt{5}}{2}$$

der goldene Schnitt.

1) *Zeigen Sie, dass für $i = n, \ldots, 2$ gilt*

$$a_i \geq \phi^{n-i}.$$

2) Folgern Sie, dass

$$n \leq \frac{\ln(a_2)}{\ln(\phi)} + 2.$$

Übung 2.30 *Zeigen Sie mit Hilfe von Aufgabe 2.29: Für die Berechnung des größten gemeinsamen Teilers von zwei Zahlen der Bitlängen u und v mit dem Euklidischen Algorithmus genügen*

$$O(u \cdot v)$$

B-Operationen.

2.13 Praktische Aufgaben

Übung 2.31 *Implementieren Sie*

1) das Sieb des Eratosthenes und

2) die Faktorisierung von ganzen Zahlen mittels Probedivision.

3) Faktorisieren Sie mit Ihrer Implementierung in $\mathbb{Z}$ die Zahl

$$183720878269532761066013208021559169596728115426694118764 03.$$

Übung 2.32 *Implementieren Sie das Faktorisierungsverfahren von Pollard.*

Testen Sie Ihre Implementierung an Beispielen, und vergleichen Sie die Performance an für das Pollardverfahren geeigneten Beispielen mit der Funktion `factor` *von* Julia/Nemo.

Übung 2.33 *Auf dem Ring der Gaußschen Zahlen*

$$R = \mathbb{Z}[i] = \{a_1 + i \cdot a_2 \mid a_1, a_2 \in \mathbb{Z}\} \subset \mathbb{C}$$

mit $i^2 = -1$ ist durch

$$\begin{array}{rccc} d: & R\backslash\{0\} & \to & \mathbb{N}_0 \\ & a + b \cdot i & \mapsto & a^2 + b^2 \end{array}$$

eine Euklidische Norm gegeben.

1) Implementieren Sie einen Algorithmus zur Durchführung der Division mit Rest in (R, d).

2) Verwenden Sie Ihren Divisionsalgorithmus, um den Eukldischen Algorithmus in R zu implementieren.

3) Berechnen Sie damit

$$\mathrm{ggT}(3 + 4i, \quad -1 + 12i) \in R.$$

Übung 2.34 *Schreiben Sie eine Funktion, die die Lösungsmenge der simultanen Kongruenzen*

$$\begin{aligned} x &\equiv a_1 \bmod n_1 \\ &\vdots \\ x &\equiv a_r \bmod n_r \end{aligned}$$

für $a_1, \ldots, a_r \in \mathbb{Z}$ und paarweise teilerfremde Moduli $n_1, \ldots, n_r \in \mathbb{Z}_{>0}$ bestimmt. Vergleichen Sie mit der JULIA/NEMO*-Funktion* `crt`.

Erweitern Sie die Funktionalität Ihrer Implementierung so, dass sie auch im Fall nicht paarweise teilerfremder n_i korrekt funktioniert.

Übung 2.35 *Implementieren Sie das Faktorisierungsverfahren von Pollard.*

Testen Sie Ihre Implementierung jeweils an Beispielen, siehe insbesondere auch Aufgabe 2.22.

Übung 2.36 *Sei $M = \{0, \ldots, 2^{64} - 1\}$.*

1) *Schreiben Sie eine Funktion* `add64`*, die für $a, b \in M$ Zahlen $c \in M$ und $d \in \{0, 1\}$ bestimmt mit*

$$a + b = c + d \cdot 2^{64}.$$

2) *Schreiben Sie eine Funktion* `mult64`*, die für $a, b \in M$ Zahlen $c, d \in M$ bestimmt mit*

$$a \cdot b = c + d \cdot 2^{64}.$$

3) *Erproben Sie Ihre Funktionen an Beispielen.*

Übung 2.37 *Sei $B = 2^{64}$ und*

$$\begin{array}{rccc} \phi_{B,r} : & \{0, \ldots, B-1\}^r & \longrightarrow & \{0, \ldots, B^r - 1\} \\ & (a_{r-1}, \ldots, a_0) & \longmapsto & \sum_{i=0}^{r-1} a_i B^i \end{array}$$

die B-adische Entwicklung zur Basis B mit r Stellen.

1) *Schreiben Sie eine Funktion, die aus zwei Zahlen in B-adischer Darstellung $a, b \in \{0, \ldots, B-1\}^r$ mittels Schulbuchmultiplikation das Produkt bestimmt, d.h. für minimal mögliches $s \in \mathbb{N}_0$ ein $c \in \{0, \ldots, B-1\}^s$ mit*

$$\phi_{B,s}(c) = \phi_{B,r}(a) \cdot \phi_{B,r}(b).$$

2) *Implementieren Sie für $r = 2^k$ eine Zweierpotenz die rekursive Anwendung der Karatsuba-Multiplikation zur Berechnung des Produkts von a und b.*

3) Erproben Sie Ihre Funktionen an Beispielen und vergleichen Sie die Performance.

Hinweis: Verwenden Sie Ihre Funktionen `add64` *und* `mult64`*.*

Übung 2.38 *Implementieren Sie den Miller-Rabin Primzahltest. Erproben Sie Ihr Programm an Beispielen.*

Übung 2.39 *1) Implementieren Sie die Faktorisierung mit dem quadratischen Sieb.*

2) Vergleichen Sie anhand von Beispielen die Laufzeit mit der Probedivision und dem Pollard-Verfahren. Können Sie jeweils ein Beispiel produzieren, bei dem das quadratische Sieb schneller ist als das Pollard-Verfahren und umgekehrt?

3

Symbolisches Rechnen in endlichen Gruppen

In diesem Abschnitt wollen wir kurz einen Blick auf Computeralgebra in der Gruppentheorie werfen. Das Konzept der Gruppe hat wichtige Anwendungen in fast allen Bereichen der Mathematik und Informatik. Es erlaubt uns Symmetrien in mathematischen Objekten und Problemstellungen zu beschreiben und dadurch das Problem zu vereinfachen. Vom praktischen Standpunkt, kann dies Berechnungen beschleunigen, da diese nur bis auf Symmetrie durchgeführt werden müssen. Für Untergruppen der S_n implementiert das Computeralgebrasystem GAP, siehe [13], im Wesentlichen alle bekannten Algorithmen. Mit Hilfe des Begriffs der Gruppenoperation werden wir sehen, dass man damit schon in allen endlichen Gruppen rechnen kann.

3.1 Gruppenoperationen

Wie können wir alle Symmetrie des Oktaeders in Abbildung 3.1 bestimmen, d.h. alle Drehungen, Spiegelungen und Drehspiegelungen, die den Oktaeder wieder auf sich selbst abbilden? Die Symmetrien bilden eine Gruppe, die Symmetriegruppe des Oktaeders, denn die Komposition von Symmetrien ist wieder eine Symmetrie und jede Symmetrie hat eine inverse Symmetrie. Die identische Abbildung ist das neutrale Element. Formal definiert man die Symmetriegruppe als Untergruppe der Bewegungsgruppe:

Definition 3.1.1 *Eine Euklidische* ***Bewegung*** $f : \mathbb{R}^n \to \mathbb{R}^n$ *ist eine Abbildung, die den Euklidischen Abstand*

$$\|x\| := \sqrt{\sum_{i=1}^{n} x_i^2}$$

erhält, d.h. mit

$$\|x - y\| = \|f(x) - f(y)\|$$

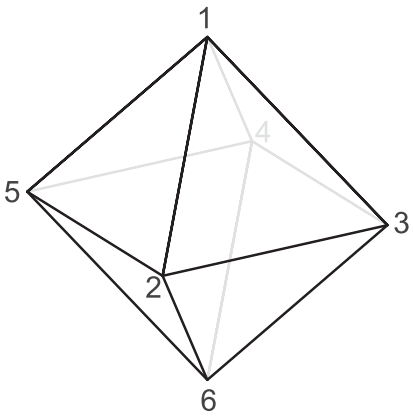


Abbildung 3.1: Oktaeder

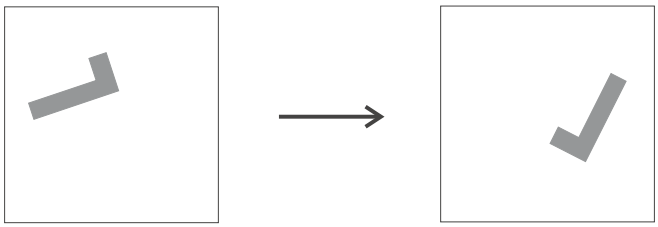


Abbildung 3.2: Beispiel einer Bewegung des $\mathbb{R}^2$.

für alle $x, y \in \mathbb{R}^n$. *Abbildung 3.2 zeigt eine Bewegung, die sich aus einer Translation und einer Drehspiegelung zusammensetzt. Die Menge* $\mathrm{E}(n)$ *der Euklidischen Bewegungen des* $\mathbb{R}^n$ *ist mit der Komposition eine Gruppe, die* ***Bewegungsgruppe****.*

In der linearen Algebra zeigt man, dass sich jede Bewegung schreiben lässt als die Komposition von einer Translation und der Multiplikation mit einer orthogonalen Matrix, d.h.

$$f(x) = v + Q \cdot x$$

mit $v \in \mathbb{R}^n$ *und* $Q \in O(n)$, *also* $Q \cdot Q^t = Q^t \cdot Q = E$. *Wiederholen Sie den Beweis, er ist eine leichte Übungsaufgabe mit Skalarprodukten.*

Sei $M \subset \mathbb{R}^n$ *eine Teilmenge. Die Gruppe*

$$\mathrm{Sym}(M) = \{A \in \mathrm{E}(n) \mid A(M) = M\}$$

heißt ***Symmetriegruppe*** *von* M.

Da Symmetrien des Oktaeders den Mittelpunkt wieder auf sich selbst abbilden müssen, lässt sich also jede Symmetrie als orthogonale Matrix darstellen, ist also ein Vektorraumhomomorphismus. Ein Homomorphismus ist durch die Bilder einer Basis festgelegt. Da die Menge der Ecken des Oktaeders eine Basis von $\mathbb{R}^3$ enthält, sind die

Symmetrien des Oktaeders durch ihre Wirkung auf den Ecken eindeutig bestimmt. Durch Nummerieren der Ecken können wir jede Symmetrie also als ein Element der der symmetrischen Gruppe S_6 auffassen, und damit G als Untergruppe

$$G \hookrightarrow S_6.$$

Zum Beispiel ist die Drehung um 90 Grad gegen den Uhrzeigersinn um die Gerade durch 1 und 6 gegeben durch die Permutation

$$(2,3,4,5) \in S_6$$

(in Zykelschreibweise) und die Spiegelung an der Ebene durch $2,3,4,5$ durch die Transposition

$$(1,6) \in S_6.$$

Ist $E \subset G$ eine Teilmenge, dann natürlich auch das Erzeugnis $\langle E \rangle \subset G$, d.h. die kleinste Untergruppe von G die alle Elemente von E enthält. In GAP können wir die Elemente des Erzeugnisses wie folgt berechnen:

```
gap> G:=Group((2,3,4,5),(1,6));;
gap> Order(G);
8
gap> Elements(G);
[(), (2,3,4,5), (2,4)(3,5), (2,5,4,3), (1,6), (1,6)(2,3,4,5),
(1,6)(2,4)(3,5), (1,6)(2,5,4,3)]
```

Es ist also leicht Elemente von G zu finden, aber wie können wir sicher sein, dass wir alle Erzeuger von G gefunden haben? Dazu verwenden wir den Begriff der Gruppenoperation:

Definition 3.1.2 *Sei $(G,\circ)$ eine Gruppe und M eine Menge. Eine **Operation** von G auf M ist eine Abbildung*

$$\begin{array}{rccc} \cdot : & G \times M & \longrightarrow & M \\ & (g,m) & \longmapsto & g \cdot m \end{array}$$

mit

$$e \cdot m = m$$

für alle $m \in M$ und

$$(a \circ b) \cdot m = a \cdot (b \cdot m)$$

für alle $a,b \in G$ und $m \in M$.

Bemerkung 3.1.3 *Anders formuliert ist eine Operation von G auf M ein Gruppenhomomorphismus*

$$\begin{array}{rccc} \varphi : & G & \longrightarrow & S(M) \\ & g & \mapsto & \varphi(g) := \begin{pmatrix} M & \longrightarrow & M \\ m & \longmapsto & g \cdot m \end{pmatrix} \end{array}$$

von G in die Gruppe der Selbstabbildung

$$S(M) = \{\varphi : M \longrightarrow M \mid \varphi \text{ bijektiv}\}$$

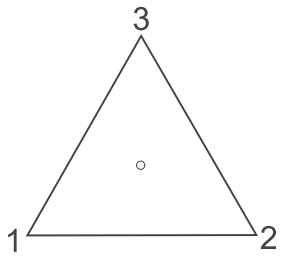
von M.

Analog zu Definition 3.1.2 kann man auch Operationen $M \times G \to M$ von rechts definieren.

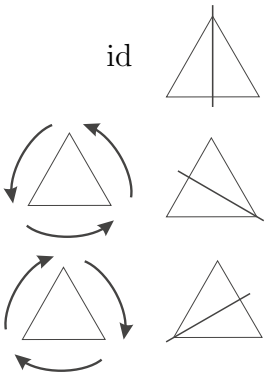
Beispiel 3.1.4 *S_n operiert auf $\{1, ..., n\}$ durch*

$$\begin{array}{ccc} S_n \times \{1, ..., n\} & \longrightarrow & \{1, ..., n\} \\ (\sigma,\ j) & \longmapsto & \sigma(j) \end{array}$$

Beispiel 3.1.5 *Wir beschreiben die Symmetriegruppe $\mathrm{Sym}\,(D)$ des gleichseitigen Dreiecks D.*



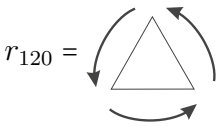
Wir können leicht Symmetrien aufzählen:



Die Operation

$$\mathrm{Sym}\,(D) \times \{1, 2, 3\} \longrightarrow \{1, 2, 3\}$$

von $\mathrm{Sym}\,(D)$ auf der Menge der Ecken von D gibt einen Homomorphismus $\varphi : \mathrm{Sym}\,(D) \to S_3$. Bezeichnet etwa



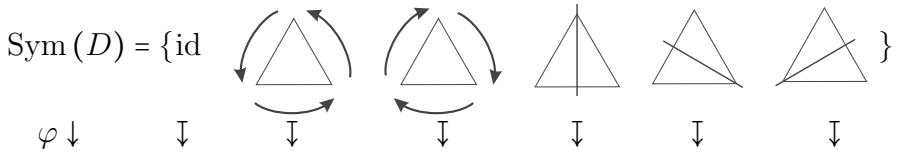
die Drehung um $120°$, dann gibt die Operation die Zuordnung

$$(r_{120}, 1) \mapsto 2,\ \ (r_{120}, 2) \mapsto 3,\ \ (r_{120}, 3) \mapsto 1$$

also

$$\varphi(r_{120}) = (1, 2, 3)$$

in Zykelschreibweise. Da jede Symmetrie durch ihre Wirkung auf den Ecken festgelegt ist, ist φ injektiv, da wir schon 6 Symmetrien gefunden haben also auch surjektiv. Wir haben also einen Gruppenisomorphismus

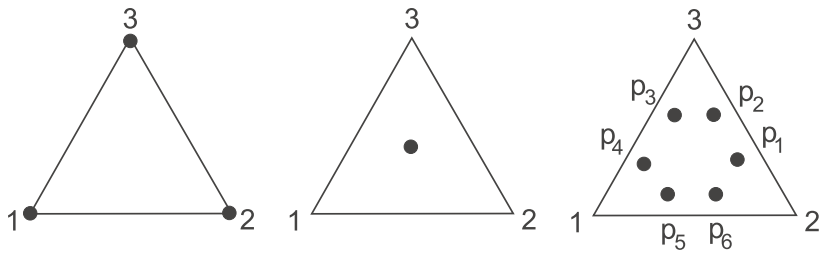


Für die Symmetriegruppe des Tetraeders (der 3-dimensionalen Variante des gleichseitigen Dreiecks) siehe Übungsaufgabe 3.6.

Beispiel 3.1.6 *Gegeben ein Punkt des gleichseitigen Dreiecks D, wollen wir untersuchen, auf welche anderen Punkte dieser unter der Operation*

$$\mathrm{Sym}\,(D) \times D \longrightarrow D$$

abgebildet werden kann. Diese Menge nennt man die Bahn, die Anzahl der Elemente die Länge der Bahn. Beispiele von Bahnen sind



Für einen Punkt $p \in D$ können wir andererseits auch die Menge aller Symmetrien betrachten, die p festhalten, den sogenannten Stabilisator:

p	*Bahn*	*Stabilisator*	
1	$\{1,2,3\}$	$\{\mathrm{id}, (2,3)\}$	$3 \cdot 2 = 6$
m	$\{m\}$	$\mathrm{Sym}\,(D)$	$1 \cdot 6 = 6$
p_1	$\{p_1, ..., p_6\}$	$\{\mathrm{id}\}$	$6 \cdot 1 = 6$

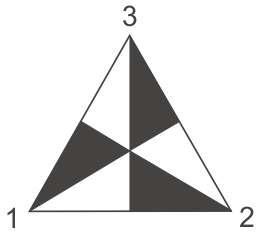
Wir beobachten, dass die Stabilisatoren stets Untergruppen von $\mathrm{Sym}\,(D)$ *sind, und das Produkt der Gruppenordnung mit der Länge der jeweiligen Bahn stets* $|\mathrm{Sym}\,(D)| = 6$ *ergibt.*

Man kann auch eine Bahn der Länge 2 konstruieren: Die Operation auf D induziert eine Operation

$$\mathrm{Sym}\,(D) \times 2^D \longrightarrow 2^D$$

auf der Potenzmenge (d.h. der Menge aller Teilmengen) von D. In der Bahn der schwarzen Teilmenge liegt außerdem noch die weisse

Teilmenge:



Der Stabilisator der schwarzen (und ebenso der weissen) Teilmenge ist die Untergruppe

$$\langle (1,2,3) \rangle$$

der Ordnung 3.

Definition 3.1.7 *Sei* $G \times M \longrightarrow M$ *eine Operation. Für* $m \in M$ *heißt die Menge*

$$G \cdot m = \{gm \mid g \in G\} \subset M$$

die ***Bahn*** *(oder der* ***Orbit****) von* m *und*

$$\mathrm{Stab}\,(m) = \{g \in G \mid gm = m\}$$

der ***Stabilisator*** *von* m.

Die Menge der Bahnen bezeichnen wir mit M/G.

Bemerkung 3.1.8 *In der gleichen Bahn zu sein ist eine Äquivalenzrelation, insbesondere sind Bahnen entweder gleich oder disjunkt.*

Dies zeigen wir in Übung 3.3.

Bemerkung 3.1.9 *Für* $\sigma \in S_n$ *zerlegt die Operation von* $\langle \sigma \rangle$ *die Menge* $\{1, ..., n\}$ *in Bahnen. Wenn wir uns noch merken in welcher Reihenfolge die Bahnen durchlaufen werden erhalten wir die Zykelschreibweise von* σ. *Zum Beispiel für*

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 1 & 2 & 5 & 4 \end{pmatrix}$$

ist die Bahnenzerlegung

$$\{1, \ldots, 5\} = \{1, 2, 3\} \cup \{4, 5\}$$

und die Zykleschreibweise

$$\sigma = (1, 3, 2)(4, 5).$$

In GAP *konvertieren wir* σ *von der Abbildungsschreibweise in die Zykelschreibweise durch:*

```
gap> PermList([3,1,2,5,4]);
(1,3,2)(4,5)
```

Definition und Satz 3.1.10 *Sei $G \times M \to M$ eine Operation. Ein **vollständiges Repräsentantensystem** der Bahnen ist eine Teilmenge $R \subset M$, sodass jede Bahn Gm genau ein Element von R enthält. Dann ist M die disjunkte Vereinigung*

$$M = \dot{\bigcup_{r \in R}} G \cdot r$$

Insbesondere gilt also für eine endliche Menge M, dass

$$|M| = \sum_{r \in R} |G \cdot r| .$$

Für eine Anwendung dieser Formel siehe Aufgabe 3.11.

3.2 Operation durch Translation

Ein wichtiges Beispiel einer Operation ist die einer Gruppe $(G, \circ)$ auf sich selbst

$$\begin{aligned} G \times G &\longrightarrow G \\ (g, h) &\mapsto g \circ h \end{aligned}$$

gegeben durch die Verknüpfung (dies ist eine Operation sowohl von links als auch von rechts). Sie spielt die entscheidende Rolle im Beweis des folgenden Satzes, der eine zentrale Bedeutung für das praktische Rechnen mit Gruppen hat: Er erlaubt es, jede endliche Gruppe als Untergruppe einer S_n aufzufassen. In dieser Darstellung können wir die Gruppe dann im Computer handhaben.

Satz 3.2.1 (Cayley) *Jede Gruppe G ist isomorph zu einer Untergruppe der Gruppe der Selbstabbildungen $S(G)$.*

Inbesondere für $n := |G| < \infty$ können wir G als Untergruppe von $S_n \cong S(G)$ auffassen.

Beweis. Die Abbildung

$$\begin{array}{rccc} \varphi: & G & \to & S(G) \\ & g & \mapsto & \begin{pmatrix} G & \to & G \\ h & \mapsto & g \circ h \end{pmatrix} \end{array}$$

ist ein Gruppenhomomorphismus und

$$\operatorname{Ker} \varphi = \{ g \in G \mid g \circ h = h \ \forall h \in G \} = \{ e \}$$

(mit der Eindeutigkeit des neutralen Elements) also φ injektiv. Somit gilt

$$G \cong \operatorname{Bild}(\varphi) \subset S(G) .$$

■

Für endliche Gruppen kann man die Verknüpfung

$$G \times G \longrightarrow G$$
$$(g, h) \mapsto g \circ h$$

mittels einer Tabelle angeben, der **Verknüpfungstafel**.

Beispiel 3.2.2 *Die Gruppe*

$$G = \mathbb{Z}/4 = \left\{\overline{0}, \overline{1}, \overline{2}, \overline{3}\right\}$$

hat die Verknüpfungstafel

+	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$
$\overline{0}$	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$
$\overline{1}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{0}$
$\overline{2}$	$\overline{2}$	$\overline{3}$	$\overline{0}$	$\overline{1}$
$\overline{3}$	$\overline{3}$	$\overline{0}$	$\overline{1}$	$\overline{2}$

In jeder Zeile und Spalte steht jedes Element genau einmal. Die Zeilen der Verknüpfungstafel spezifizieren $\varphi(g)$, in dem Beispiel ist etwa

$$\varphi(\overline{1}) = \begin{pmatrix} \overline{0} & \overline{1} & \overline{2} & \overline{3} \\ \overline{1} & \overline{2} & \overline{3} & \overline{0} \end{pmatrix} \in S(G) \cong S_4.$$

Eine Gruppe ist abelsch genau dann, wenn ihre Verknüpfungstafel bezüglich der Diagonalen symmetrisch ist. Das Assoziativgesetz lässt sich der Tabelle nicht unmittelbar ansehen.

Analog zur Operation einer Gruppe auf sich selbst kann man auch die Operation einer Untergruppe betrachten:

Definition 3.2.3 *Sei $H \subset G$ eine Untergruppe. Dann definiert die Verknüpfung in G eine Operation von H auf G*

$$H \times G \longrightarrow G,\ (h, g) \longmapsto h \circ g$$

von links, und ebenso eine von rechts

$$G \times H \longrightarrow G,\ (h, g) \longmapsto g \circ h.$$

Für $g \in G$ heißen die Bahnen dieser Operation

$$Hg := H \circ g := \{h \circ g \mid h \in H\}$$

bzw.

$$gH := g \circ H := \{g \circ h \mid h \in H\}$$

rechte bzw. linke ***Nebenklassen*** *von g.*

Satz 3.2.4 *Sei $H \subset G$ eine Untergruppe. Je zwei Nebenklassen von H haben gleich viele Elemente.*

Beweis. Seien $a, b \in G$. Dann stehen aH und bH in Bijektion zueinander durch Multiplikation mit ba^{-1} von links

$$\begin{array}{ccc} g & \longmapsto & b \circ a^{-1} \circ g \\ G & \overset{1:1}{\longrightarrow} & G \\ \cup & & \cup \\ aH & \longrightarrow & bH \\ a \circ h & \longmapsto & b \circ a^{-1} \circ a \circ h = b \circ h \end{array}$$

(was ist die Umkehrabbildung?). Die rechten und linken Nebenklassen Ha und aH stehen in Bijektion vermöge der **Konjugation** mit a

$$\begin{array}{ccc} g & \longmapsto & a \circ g \circ a^{-1} \\ G & \overset{1:1}{\longrightarrow} & G \\ \cup & & \cup \\ Ha & \longrightarrow & aH \\ h \circ a & \longmapsto & a \circ h \circ a \circ a^{-1} = a \circ h \end{array}$$

(was ist die Umkehrabbildung?). ■

Corollar 3.2.5 (Indexformel) *Sei $H \subset G$ eine Untergruppe. Es gilt*

$$|G| = |G/H| \cdot |H|$$

insbesondere in einer endlichen Gruppe teilt $|H|$ die Gruppenordnung $|G|$.

Aus der Indexformel (Satz 3.2.5) erhalten wir mit $H = \langle g \rangle$:

Corollar 3.2.6 *In einer endlichen Gruppe G ist die Ordnung eines Elements $g \in G$ ein Teiler der Gruppenordnung $|G|$, d.h.* $\operatorname{ord}(g) \mid |G|$.

Wir beweisen nun die Indexformel:
Beweis. Wir bemerken zunächst, dass

$$\begin{aligned} H &\to aH \\ h &\mapsto ah \end{aligned}$$

eine Bijektion ist (siehe den Beweis von Satz 3.2.4), also

$$|aH| = |H| .$$

Nach Definition und Satz 3.1.10 ist G die disjunkte Vereinigung aller aH mit a aus einem vollständigen Repräsentantensystem R, also falls $|G| < \infty$ so gilt

$$|G| = \sum_{a \in R} |aH| = |R| \cdot |H|$$

(mit Satz 3.2.4). Ist $|G| = \infty$, dann auch $|G/H| = \infty$ oder $|H| = \infty$. ■

Beispiel 3.2.7 *Die Gruppe*

$$G = \langle (1,2,3,4), (1,3) \rangle \subset S_4$$

ist eine Untergruppe der Ordnung $8 \mid 24$. *Wir zeigen dies mit Hilfe von* GAP*:*

```
gap> G:=Group((1,2,3,4),(1,3));
gap> Order(G);
8
```

Beispiel 3.2.8 *Die Ordnung von* $\sigma = (1,4,3,2)\,(5,9)\,(6,8) \in S_9$ *können wir elementar durch Potenzieren herausfinden:*

$$\begin{aligned}\sigma^2 &= (1,4,3,2)^2(5,9)^2(6,8)^2 = (1,3)(2,4)\\ \sigma^3 &= (1,2,3,4)(5,9)(6,8)\\ \sigma^4 &= \mathrm{id}\,,\end{aligned}$$

oder mit Hilfe von GAP*:*

```
gap> sigma:=(1,4,3,2)(5,9)(6,8);
(1,4,3,2)(5,9)(6,8)
gap> sigma^2;
(1,3)(2,4)
gap> sigma^3;
(1,2,3,4)(5,9)(6,8)
gap> sigma^4;
()
```

Somit gilt $\mathrm{ord}(\sigma) = 4$. *Dies berechnet* GAP *auch durch:*

```
gap> Order(sigma);
4
```

Dabei potenziert GAP *natürlich nicht, sondern verwendet die bekannte* kgV*-Formel für die Ordnung von Permutationen in disjunkter Zykelnotation*

$$\mathrm{ord}(\sigma) = \mathrm{kgV}(4,2,2) = 4.$$

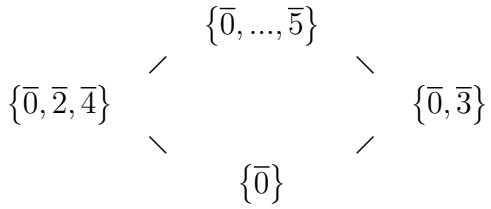
Bei der Arithmetik mit Permutationen in GAP *ist zu beachten, dass man, abweichend von der üblichen Konvention, zur Berechnung von* $\sigma \circ \tau$ *für* $\sigma, \tau \in S_n$ *in* GAP $\tau * \sigma$ *eingeben muss (d.h. Abbildungen nehmen ihr Argument auf der linken Seite). Wir überprüfen in* GAP*, dass mit* $\tau = (2,5)$ *gilt*

$$\begin{aligned}\sigma \circ \tau &= (1,4,3,2)\,(5,9)\,(6,8) \circ (2,5)\\ &= (1,4,3,2,9,5)(6,8).\end{aligned}$$

```
gap> tau:=(2,5);;
gap> tau*sigma;
(1,4,3,2,9,5)(6,8)
```

In Aufgabe 3.10 haben Sie die Chance die Arithmetik in der S_n in der gewohnten Form zu implementieren. Permutationen in Zykelnotation können Sie z.B. als Listen von Listen darstellen. Um eine Eindeutigkeit der Darstellung zu erreichen ist es sinnvoll, den ersten Eintrag in einem Zykel minimal zu wählen und bei Produkten von disjunkten Zykeln diese wiederum nach dem ersten Eintrag zu sortieren.

Beispiel 3.2.9 *Die Gruppe $G = \mathbb{Z}/6$ der Ordnung 6 hat die Untergruppen*



mit den Ordnungen $1, 2, 3$ und 6.

Bemerkung 3.2.10 *Man beachte, dass es in einer Gruppe nicht zu jedem Teiler eine Untergruppe geben muss, z.B. hat die*

$$A_4 = \{\sigma \in S_4 \mid \operatorname{sign}(\sigma) = 1\}$$

keine Untergruppe der Ordnung 6. Der folgende GAP*-Code berechnet alle möglichen Ordnungen von Untergruppen der A_4:*

```
gap> G:=AlternatingGroup(4);;
gap> Order(G);
12
gap> L:=ConjugacyClassesSubgroups(G);;
gap> List(List(L,Representative),Order);
[ 1, 2, 3, 4, 12 ]
```

Tatsächlich werden hier Repräsentaten der Bahnen der Konjugationsoperation

$$\begin{array}{lcl} G \times S & \longrightarrow & S \\ (g, H) & \longmapsto & gHg^{-1} := \{g \circ h \circ g^{-1} \mid u \in H\} \end{array}$$

von G auf der Menge S der Untergruppen von G bestimmt. Man beachte, dass Konjugation die Gruppenordnung nicht ändert (siehe den Beweis von Satz 3.2.4). Für ein weiteres Beispiel zur Konjugation siehe auch die Aufgaben 3.4 und 3.5.

Im Kontext der sogenannten Sylowsätze kann man zeigen, dass es zumindest zu jedem Primpotenzteiler von $|G|$ eine Untergruppe gibt.

Bemerkung 3.2.11 *Hat die Bahn der Untergruppe $H \subset G$ unter der Konjugationsoperation genau ein Element, d.h. gilt*

$$gHg^{-1} = H$$

für alle $g \in G$, so bezeichnet man H als ***Normalteiler*** *von G. Dann ist die Menge der Nebenklassen*

$$G/H$$

mit der von G induzierten Verknüfpung

$$aH \circ bH := (a \circ b)H$$

eine Gruppe (und $\pi : G \to G/H$, $a \mapsto aH$ ein Gruppenepimorphimus). Man rufe sich in Erinnerung, dass die induzierte Verknüpfung für einen Normalteiler H wohldefiniert ist.

Beispiel 3.2.12 *Die* ***Kleinsche Vierergruppe***

$$V_4 = \{(), (1,2)(3,4), (1,3)(2,4), (1,4)(2,3)\}$$

ist ein Normalteiler von S_4 und für die Quotientengruppe gilt

$$S_4/V_4 \cong S_3.$$

Dies zeigen wir in Übungsaufgabe 3.9, wo wir den Isomorphismus geometrisch interpretieren, indem wir die S_4 als Symmetriegruppe des Tetraeders auffassen.

Man kann $S_4/V_4 \cong S_3$ auch mit Hilfe von GAP *beweisen:*

```
S4:=SymmetricGroup(4);;
NormalSubgroups(S4);
[ Group(()),
Group([ (1,4)(2,3), (1,3)(2,4) ]),
Group([ (2,4,3), (1,4)(2,3), (1,3)(2,4) ]),
Sym( [ 1 .. 4 ] ) ]
V4:=Group((1,4)(2,3),(1,3)(2,4));;
Elements(V4);
[ (), (1,2)(3,4), (1,3)(2,4), (1,4)(2,3) ]
Q:=S4/V4;;
Order(Q);
6
IsomorphismGroups(Q,CyclicGroup(6));
fail
IsomorphismGroups(Q,SymmetricGroup(3));
[ f1, f2 ] -> [ (2,3), (1,2,3) ]
```

3.3 Bahnenformel

Wir betrachten nun wieder die Operation einer Gruppe G auf einer Menge M und fragen nach der Beziehung zwischen der Bahn eines Elements $m \in M$ und dem Stabilisator von m.

Satz 3.3.1 *Sei*

$$G \times M \longrightarrow M$$

eine Operation, $m \in M$ *und*

$$H := \mathrm{Stab}\,(m)\,.$$

Dann gibt es eine natürliche Bijektion

$$\begin{array}{ccc} G/H & \longrightarrow & Gm \\ gH & \longmapsto & gm \end{array}$$

Beweis. Die Abbildung ist wohldefiniert: Ist $gH = g'H$, dann $g' \in gH$, also $g' = gh$ mit $h \in H$. Es folgt

$$g'm = ghm = gm,$$

da m von h stabilisiert wird. Die Abbildung ist offenbar surjektiv. Sie ist auch injektiv, denn

$$g_1 m = g_2 m \Rightarrow g_1^{-1} g_2 \in H \Rightarrow$$
$$g_2 = g_1 g_1^{-1} g_2 \in g_1 H \Rightarrow g_1 H = g_2 H.$$

■

Satz 3.3.2 (Bahnenformel) *Sei* $G \times M \longrightarrow M$ *eine Operation. Für jedes* $m \in M$ *gilt*

$$|Gm| \cdot |\mathrm{Stab}\,(m)| = |G|\,.$$

Beweis. Es ist

$$|Gm| = |G/H|$$

mit Satz 3.3.1 und

$$|G/H| \cdot |H| = |G|$$

nach der Indexformel 3.2.5. ■

Beispiel 3.3.3 *Mit Hilfe der Bahnenformel können wir die Ordnung der Symmetriegruppe* $G = \mathrm{Sym}(O)$ *des Oktaeders* O *(Abbildung 3.1) bestimmen: Der Punkt* $p = 1$ *hat sicher die Bahn*

$$G1 = \{1, \ldots, 6\}.$$

Jede Symmetrie des Oktaeders, die 1 *festhält, ist auch eine Symmetrie des Quadrats* Q *in Abbildung 3.3 und umgekehrt. Somit ist*

$$\mathrm{Stab}(1) = \mathrm{Sym}(Q)$$

die Symmetriegruppe des Quadrats. Unter der Operation von $\mathrm{Sym}(Q)$ *wiederum hat* 2 *die Bahn*

$$\mathrm{Sym}(Q)2 = \{2, 3, 4, 5\},$$

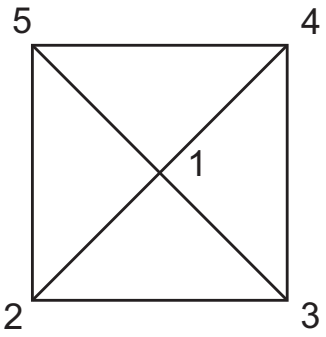


Abbildung 3.3: Quadrat im Oktaeder

während 2 *nur von der Identität* () *und der Spiegelung* $(3,5)$ *stabilisiert wird. Damit ist mit der Bahnenformel*

$$|\mathrm{Sym}(Q)| = 4 \cdot 2 = 8.$$

Siehe auch Aufgabe 3.4, in der wir die Symmetriegruppe des Quadrats explizit bestimmen. Es folgt also wieder mit der Bahnenformel

$$|G| = 8 \cdot 6 = 48.$$

Wir verwenden GAP *um zu zeigen, dass*

$$G = \langle (2,3,4,5), (1,3)(5,6) \rangle$$

Offenbar sind die angegebenen Permutationen Symmetrien des Oktaeders: Wie oben schon gesehen, entspricht $(2,3,4,5)$ *der Drehung um* 90 *Grad um die Gerade durch* 1 *und* 6. *Weiter ist* $(1,3)(5,6)$ *die Spiegelung an der in Abbildung 3.4 gezeigten Ebene. Wir müssen also nur*

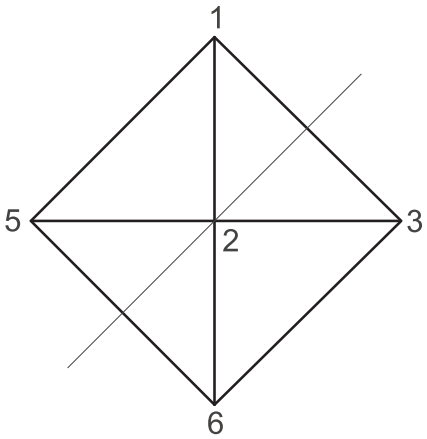


Abbildung 3.4: Spiegelung $(1,3)(5,6)$ als Symmetrie des Oktaeders

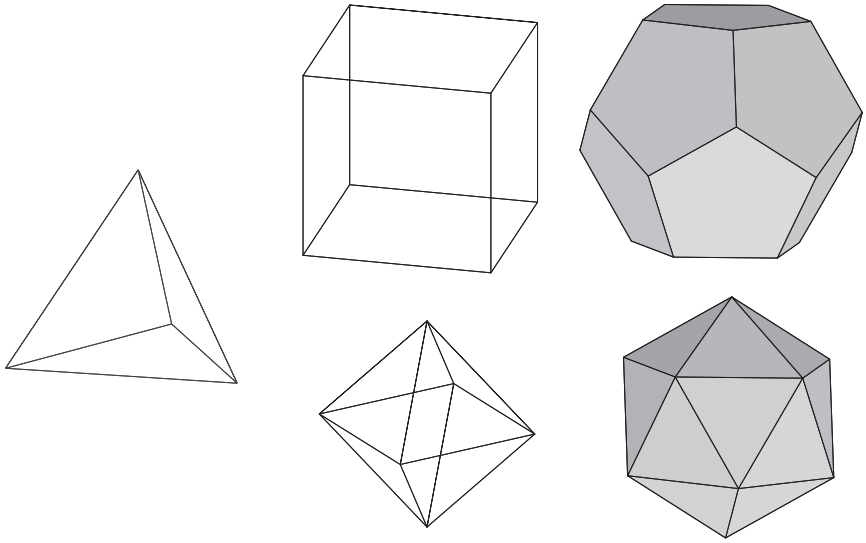
überprüfen, dass die beiden Symmetrien eine Gruppe der Ordnung 48 *erzeugen:*

```
gap> G:=Group((2,3,4,5),(1,3)(5,6));;
gap> Size(G);
48
```

Natürlich können wir statt der Ecken auch die Seiten des Oktaeders nummerieren und erhalten dadurch seine Symmetriegruppe als Untergruppe der S_8. Siehe dazu Aufgabe 3.7. In den Übungen 3.4 und 3.8 bestimmen wir die Symmetriegruppen des Quadrats und des Ikosaders.

3.4 Übungen

Übung 3.1 *Basteln Sie Papiermodelle der Platonischen Körper Tetraeder, Würfel, Oktaeder, Dodekaeder und Ikosaeder. Bitte in die Übung mitbringen.*



Übung 3.2 *Finden Sie für alle Platonischen Körper jeweils eine Drehsymmetrie und eine Spiegelsymmetrie und beschreiben Sie diese als Elemente der symmetrischen Gruppe S_n mit n die Anzahl der Ecken des Platonischen Körpers.*

Übung 3.3 *Sei $G \times M \to M$ eine Operation der Gruppe G auf der Menge M. Zeigen Sie, dass für $a, b \in M$ durch*

$$a \sim b :\Longleftrightarrow Ga = Gb$$

eine Äquivalenzrelation gegeben ist.

Übung 3.4 *Sei G die Symmetriegruppe des Quadrats, wie in Abbildung 3.5.*

1) Berechnen Sie die Gruppenordnung von G.

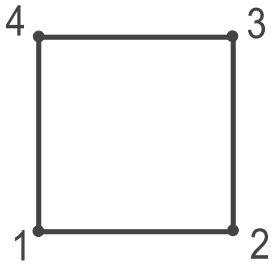


Abbildung 3.5: Quadrat mit Nummerierung

2) Bestimmen Sie Erzeuger von G als Untergruppe von S_4. Beweisen Sie Ihre Behauptung mit Hilfe von GAP.

Übung 3.5 *Sei G eine Gruppe. Zwei Untergruppen $U_1, U_2 \subset G$ heißen konjugiert, wenn es ein $g \in G$ gibt mit*

$$gU_1g^{-1} := \{g \circ u \circ g^{-1} \mid u \in U_1\} = U_2.$$

1) Zeigen Sie, dass konjugiert sein eine Äquivalenzrelation auf der Menge der Untergruppen von G ist.

2) Bestimmen Sie die Konjugationsklassen von Untergruppen der Symmetriegruppe des Quadrats. Welche Untergruppen sind Normalteiler?

Hinweis: Sie können die Klassen direkt bestimmen oder den GAP-*Befehl* `ConjugacyClassesSubgroups` *verwenden.*

Übung 3.6 *Sei G die Symmetriegruppe des Tetraeders (Abbildung 3.6). Zeigen Sie, dass*

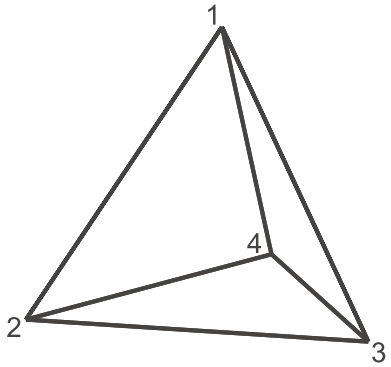


Abbildung 3.6: Tetraeder mit Nummerierung

$$G \cong S_4.$$

Übung 3.7 *Sei $G = \mathrm{Sym}(O)$ die Symmetriegruppe des Oktaeders O.*

1) Durch Nummerieren der Seiten von O (Abbildung 3.7) ist ein Monomorphismus $f_1 : G \to S_8$ gegeben. Finden Sie Erzeuger von $f_1(G)$ und zeigen Sie Ihre Behauptung mit Hilfe von GAP.

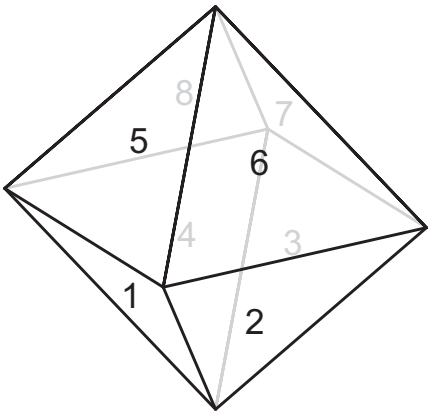


Abbildung 3.7: Oktaeder mit Seitennummerierung

2) Durch Nummerieren der Ecken von O (Abbildung 3.8) ist ein Monomorphismus $f_2 : G \to S_6$ gegeben. Finden Sie Erzeuger von $f_2(G)$ und zeigen Sie Ihre Behauptung mit Hilfe von GAP.

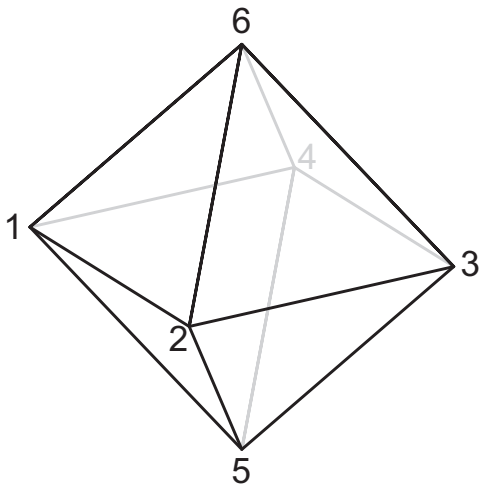


Abbildung 3.8: Oktaeder mit Eckennummerierung

3) Interpretieren Sie die in (a) und (b) gefundenen Erzeuger geometrisch.

4) Bestimmen Sie mit GAP *einen Isomorphismus von $f_2(G) \to f_1(G)$.*

Hinweis:
Verwenden Sie die GAP *Befehle* `Group`, `Size` *und* `IsomorphismGroups`.

Übung 3.8 *Sei G die Symmetriegruppe des Ikosaeders (Abbildung 3.9).*

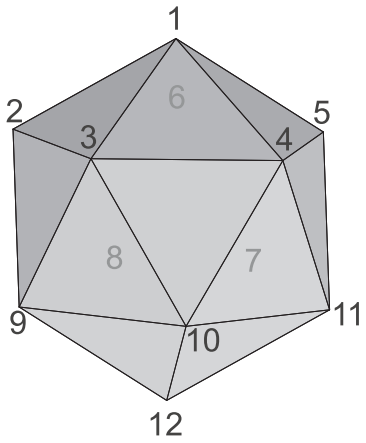


Abbildung 3.9: Ikosaeder mit Nummerierung

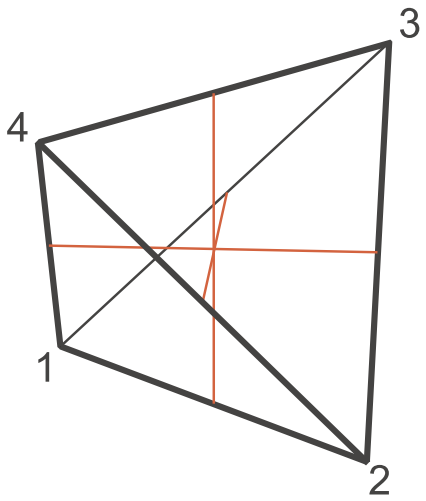
1) *Berechnen Sie die Gruppenordnung von G.*

2) *Bestimmen Sie Erzeuger von G als Untergruppe von S_{12}. Beweisen Sie Ihre Behauptung mit Hilfe von* GAP.

Übung 3.9 *Jede Symmetrie des Tetraeders $T \subset \mathbb{R}^3$ mit den Ecken*

$$e_1 = (1,-1,-1) \quad e_2 = (-1,1,-1) \quad e_3 = (-1,-1,1) \quad e_4 = (1,1,1)$$

permutiert die Koordinatenachsen von $\mathbb{R}^3$.



Dies induziert einen Gruppenhomomorphismus

$$\varphi : S_4 \to S_3.$$

Zeigen Sie mit Hilfe von φ, dass die Kleinsche Vierergruppe

$$V_4 = \{() , (1,2)(3,4) , (1,3)(2,4) , (1,4)(2,3)\}$$

ein Normalteiler in S_4 ist und für die Quotientengruppe gilt

$$S_4/V_4 \cong S_3.$$

3.5 Praktische Aufgaben

Übung 3.10 *Schreiben Sie eine Funktion, die Permutationen von der Abbildungsschreibweise*

$$\sigma = \begin{pmatrix} 1 & \dots & n \\ \sigma(1) & \dots & \sigma(n) \end{pmatrix} \in S_n$$

in die Zykelnotation konvertiert.

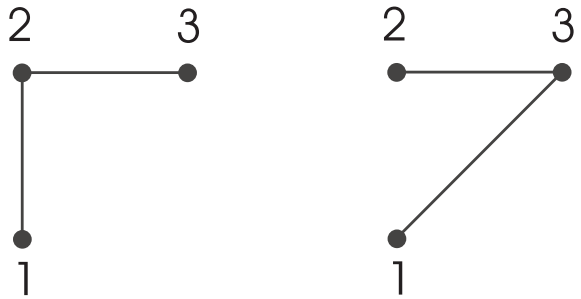
1) *Implementieren Sie die Berechnung von* $\operatorname{sign}(\sigma)$ *und* $\operatorname{ord}(\sigma)$.

2) *Seien $\sigma, \tau \in S_n$ jeweils in (disjunkter) Zykelnotation gegeben. Schreiben Sie eine Funktion die $\sigma \circ \tau$ als Produkt disjunkter Zykel darstellt.*

Übung 3.11 *Ein Graph ist ein Tupel (V, E) aus einer Menge V und einer Teilmenge $E \subset \binom{V}{2}$. Dabei bezeichnet $\binom{V}{2}$ die Menge der zweielementigen Teilmengen von V, und V heißt Menge der Vertices und E Menge der Kanten des Graphen. Zwei Graphen (V_1, E_1) und (V_2, E_2) heißen isomorph, wenn eine bijektive Abbildung $\varphi : V_1 \to V_2$ existiert, sodass*

$$\{v, w\} \in E_1 \iff \{\varphi(v) , \varphi(w)\} \in E_2$$

für alle $v, w \in V_1$. Die folgende Abbildung zeigt zwei isomorphe Graphen:



1) *Wieviele Graphen gibt es auf einer n-elementigen Vertexmenge?*

2) *Zeigen Sie mit Hilfe der Bahnenformel, dass es genau 4 Isomorphieklassen von Graphen mit 3 Vertices gibt.*

3) Schreiben Sie eine Funktion, die für alle Isomorphieklassen von Graphen mit n Vertices einen Repräsentanten berechnet.

Hinweis: Zwei Graphen sind isomorph, wenn sie in derselben Bahn unter der Operation der S_n auf der Menge aller Graphen mit n Vertices liegen.

4

Computeralgebra in Polynomringen

In der algebraischen Geometrie untersucht man Verschwindungsmengen von Polynomen. Rechnungen mit solchen Mengen kann man also auf Rechnungen in multivariaten Polynomringen übersetzen. In diesem Abschnitt diskutieren wir den grundlegenden Algorithmus für solche Rechnungen, den Buchbergeralgorithmus zur Berechnung von Gröbnerbasen. Weiter stellen wir in Anwendungen die Beziehung zur Geometrie her.

4.1 Algebraische Mengen

Definition 4.1.1 *Eine **algebraische Menge** ist die gemeinsame Nullstellenmenge*

$$V(f_1, ..., f_r) = \{p \in K^n \mid f_1(p) = 0, ..., f_r(p) = 0\}$$

von multivariaten Polynomen $f_1, ..., f_r \in K[x_1, ..., x_n]$ über einem Körper K.

Die algebraische Geometrie studiert mit Methoden der kommutativen Algebra solche algebraischen Mengen.

Beispiel 4.1.2 *Einige algebraische Mengen sind auch außerhalb der algebraischen Geometrie allgemein bekannt, zum Beispiel:*

- $V(1) = \varnothing$,
- $V(0) = K^n$,
- *die Menge aller Lösungen $x \in K^m$ eines linearen Gleichungssystems*

  $$A \cdot x - b = 0$$

  mit $A \in K^{n \times m}$ und $b \in K^n$.

- *ein Kreis*
$$V(x_1^2 + x_2^2 - 1) \subset \mathbb{R}^2$$
oder allgemeiner Kegelschnitte (Ellipsen, Parabeln, Hyperbeln).

- *der Graph*
$$\Gamma(g) = V(x_2 \cdot b(x_1) - a(x_1)) \subset K^2$$
einer rationalen Funktion
$$g = \frac{a}{b} \in K(x_1).$$
Zum Beispiel ist der Graph von $g(x_1) = \frac{x_1^3-1}{x_1}$
$$\Gamma(g) = V(x_2x_1 - x_1^3 + 1) \subset K^2,$$
siehe Abbildung 4.1 für $K = \mathbb{R}$. *Den Plot können wir in* MAPLE

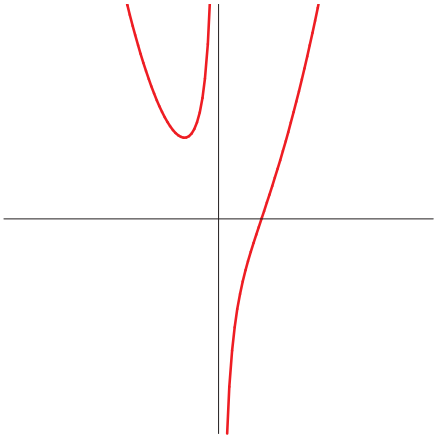


Abbildung 4.1: Graph einer rationalen Funktion

als Funktionsgraphen erstellen durch

```
plot((x1^3-1)/x1, x1=-5..5, view =[-5..5, -5..5]);
```

und als algebraische Menge durch

```
with(plots):
implicitplot(x1*x2-(x1^3-1), x1=-5..5, x2=-5..5,
numpoints=10000);
```

Nicht jede Kurve in K^2 *ist ein Graph (z.B. der Kreis ist kein Graph). In Abschnitt 1 haben wir bereits Beispiele von algebraischen Flächen gesehen, z.B. die Kummerquartik (Abbildung 1.8), die Togliattiquintik (Abbildung 1.9), und die Barthsextik (Abbildung 1.10).*

Eine leichte aber sehr wichtige Beobachtung in Bezug auf die algebraische Menge $V(f_1, ..., f_s)$ mit $f_i \in R = K[x_1, ..., x_n]$ ist die folgende: Ist $f_1(p) = 0, ..., f_s(p) = 0$ für $p \in K^n$, dann verschwindet auch jede R-Linearkombination der f_i auf p, d.h.

$$\left(\sum_{i=1}^{s} r_i \cdot f_i\right)(p) = \sum_{i=1}^{s} r_i(p)\, f_i(p) = 0$$

für alle $r_i \in R$. Dies führt auf einen der wichtigsten Begriffe in der Algebra:

Definition 4.1.3 *Sei R ein kommutativer Ring mit 1. Ein **Ideal** ist eine nichtleere Teilmenge $I \subset R$ mit*

$$a + b \in I$$
$$ra \in I$$

für alle $a, b \in I$ und $r \in R$.

Für $S \subset R$ bezeichnet

$$\langle S \rangle = \{\textstyle\sum_{endlich} r_i \cdot f_i \mid r_i \in R,\ f_i \in S\} \subset R$$

*das von S **erzeugte Ideal** (Übung: dies ist ein Ideal).*

Aufgrund der obigen Beobachtung betrachtet man statt der Nullstellenmenge von festgelegten Gleichungen besser die Nullstellenmenge eines Ideals:

Definition 4.1.4 *Ist $I \subset K[x_1, ..., x_n]$ ein Ideal, dann heißt die Nullstellenmenge*

$$V(I) = \{p \in K^n \mid f(p) = 0\ \forall f \in I\}$$

*die **Verschwindungsmenge** von I.*

Wie gerade gesehen gilt dann

$$V(f_1, ..., f_s) = V(\langle f_1, ..., f_s \rangle).$$

Andererseits ist $V(I)$ immer eine algebraische Menge, denn jedes Ideal $I \subset K[x_1, ..., x_n]$ ist endlich erzeugt, also von der Form $I = \langle f_1, ..., f_s \rangle$ mit $f_i \in R$. Ringe mit dieser Eigenschaft bezeichnet man als Noethersch.

Definition 4.1.5 *Ein kommutativer Ring R mit 1 heißt **Noethersch**, wenn jedes Ideal $I \subset R$ endlich erzeugt ist.*

Diese Ringe heißen Noethersch nach Emmy Noether (1882-1935), die die allgemeine Strukturtheorie dieser Ringe formuliert hat. Der Hilbertsche Basissatz zeigt, dass $R = K[x_1, \ldots, x_n]$ Noethersch ist.

4.2 Der Basissatz

Zum Beweis des Basissatzes verwenden wir die folgende Charakterisierung von Noetherschen Ringen, die auch im algorithmischen Rechnen in multivariaten Polynomringen eine zentrale Rolle spielen wird:

Satz 4.2.1 *Sei R ein kommutativer Ring mit 1. Die folgenden Bedingungen sind äquivalent:*

1) *R ist Noethersch, d.h. jedes Ideal $I \subset R$ ist endlich erzeugt.*

2) *Jede aufsteigende Kette*

$$I_1 \subset I_2 \subset I_3 \subset \ldots \subset I_n \subset \ldots$$

von Idealen wird stationär, d.h. es gibt ein m, sodass

$$I_m = I_{m+1} = I_{m+2} = \ldots$$

gilt.

3) *Jede nicht-leere Menge von Idealen besitzt bezüglich Inklusion ein maximales Element.*

Beweis. Wir beweisen die Implikation (1) $\Rightarrow$ (2), die im Folgenden die entscheidende Rolle spielen wird. Die Implikation (2) $\Rightarrow$ (3) und (3) $\Rightarrow$ (1) sind ebenfalls nicht schwer (siehe dazu Übung 4.2).

Sei $I_1 \subset I_2 \subset \ldots$ eine Kette von Idealen. Dann ist

$$I = \bigcup_{j=1}^{\infty} I_j$$

ebenfalls ein Ideal: Sind $a, b \in I$, so existieren $j_1, j_2 \in \mathbb{N}$ mit $a \in I_{j_1}$, $b \in I_{j_2}$, und somit ist

$$a + b \in I_{\max(j_1, j_2)} \subset I.$$

Nach (1) ist I endlich erzeugt, also gibt es $a_1, \ldots, a_n \in I$ mit $I = \langle a_1, \ldots, a_n \rangle$. Für jedes a_k existiert ein j_k mit $a_k \in I_{j_k}$. Für

$$m := \max \{ j_k \mid k = 1, \ldots, n \}$$

gilt dann $a_1, \ldots, a_n \in I_m$, also

$$I = \langle a_1, \ldots, a_n \rangle \subset I_m \subset I_{m+1} \subset \ldots \subset I$$

und somit

$$I_m = I_{m+1} = \ldots$$

■

Beispiel 4.2.2 *1) Der Ring der ganzen Zahlen $\mathbb{Z}$ ist Noethersch, denn die Ideale von $\mathbb{Z}$ sind alle von der Form*

$$\langle n \rangle = n\mathbb{Z} = \{nk \mid k \in \mathbb{Z}\},$$

also endlich erzeugt (von einem einzigen Element).

2) Ein Körper K besitzt nur die Ideale (0) und $K = (1)$, siehe auch Übungsaufgabe 4.1. Insbesondere ist K Noethersch.

Hilbert hat 1890 gezeigt, dass der Polynomring $K[x_1, \dots x_n]$ Noethersch ist.

Satz 4.2.3 (Hilbertscher Basissatz) *Sei R ein Noetherscher Ring, dann ist $R[x]$ ebenfalls Noethersch.*

Daraus erhalten wir mit Induktion nach der Anzahl der Variablen

$$R[x_1, \dots x_n] = R[x_1, \dots, x_{n-1}][x_n],$$

und da Körper Noethersch sind:

Corollar 4.2.4 *Sei K ein Körper und $n \in \mathbb{N}$. Dann ist der Polynomring $K[x_1, \dots x_n]$ in n Variablen Noethersch.*

Der Beweis des Hilbertschen Basissatzes betrachtet die Leitkoeffizienten in R von Polynomen in $R[x]$:

Definition 4.2.5 *Sei R ein kommutativer Ring mit 1. Ein Element*

$$x^\alpha := x_1^{\alpha_1} \cdot \ldots \cdot x_n^{\alpha_n} \in R[x_1, \ldots, x_n]$$

*heißt **Monom**, ein Element $c \cdot x^\alpha$ mit $c \in K$ heißt **Term**. Jedes Polynom $f \in K[x_1, \ldots, x_n]$ ist eine endliche Summe $f = \sum_\alpha c_\alpha x^\alpha$ von Termen. Der **Grad** $\deg(f)$ von f ist das maximale $|\alpha| := \alpha_1 + \ldots + \alpha_n \in \mathbb{Z}$ mit $c_\alpha \neq 0$.*

Ist $>$ eine Totalordnung auf der Menge der Monome von $R[x_1, \ldots, x_n]$ und $f \in R[x_1, \ldots, x_n]$, dann heißt der größte Term

$$\mathrm{LT}(f) = c \cdot x^\alpha$$

*der **Leitterm** von f,*

$$\mathrm{LC}(f) = c$$

*der **Leitkoeffizient** von f, und*

$$L(f) = x^\alpha$$

*das **Leitmonom** von f.*

*Ist der Leitkoeffizient $c = 1$, so bezeichnet man f als **normiert**.*

Beispiel 4.2.6 *Für univariate Polynome in $R[x]$ gibt die Ordnung nach dem Grad eine Totalordnung $>$ auf der Menge aller Monome.*

Nun zum Beweis von Satz 4.2.3:

Beweis. Angenommen $R[x]$ ist nicht Noethersch. Dann gibt es ein nicht endlich erzeugtes Ideal $I \subset R[x]$. Sei $f_1 \in I$ mit $\deg(f_1)$ minimal, $f_2 \in I \setminus \langle f_1 \rangle$ mit $\deg(f_2)$ minimal, und induktiv

$$f_{k+1} \in I \setminus \langle f_1, \ldots, f_k \rangle$$

mit $\deg(f_{k+1})$ minimal. Man beachte, dass $\langle f_1, \ldots, f_k \rangle \subsetneq I$ für alle $k \in \mathbb{N}$ (sonst wäre I von $f_1, \ldots, f_k$ erzeugt). Damit gilt

$$\deg(f_1) \leq \deg(f_2) \leq \ldots \leq \deg(f_k) \leq \ldots$$

und wir erhalten eine aufsteigende Kette von Idealen in R

$$\langle \mathrm{LC}(f_1) \rangle \subset \langle \mathrm{LC}(f_1), \mathrm{LC}(f_2) \rangle \subset \ldots \subset \langle \mathrm{LC}(f_1), \ldots, \mathrm{LC}(f_k) \rangle \subset \ldots$$

Wir zeigen, dass diese strikt aufsteigend ist (und somit R nach Satz 4.2.1 nicht Noethersch): Angenommen

$$\langle \mathrm{LC}(f_1), \ldots, \mathrm{LC}(f_k) \rangle = \langle \mathrm{LC}(f_1), \ldots, \mathrm{LC}(f_{k+1}) \rangle .$$

Dann können wir schreiben

$$\mathrm{LC}(f_{k+1}) = \sum_{j=1}^{k} b_j \, \mathrm{LC}(f_j)$$

mit $b_j \in R$. Aus dieser Gleichung können wir ein Element $g \in \langle f_1, \ldots, f_k \rangle$ konstruieren, das denselben Leitkoeffizienten wie f_{k+1} hat. Dazu betrachten wir die Linearkombination der f_j mit den Koeffizienten b_j, wobei wir alle Leitterme auf den Grad des Leitterms von f_{k+1} hochmultiplizieren, also

$$g := \sum_{j=1}^{k} b_j \cdot x^{\deg(f_{k+1}) - \deg(f_j)} \cdot f_j \in \langle f_1, \ldots, f_k \rangle .$$

In der Differenz von $g - f_{k+1}$ kürzt sich dann der Leitterm $L(g) = L(f_{k+1})$, also ist

$$\deg(g - f_{k+1}) < \deg(f_{k+1}),$$

ein Widerspruch, da $g - f_{k+1} \in I \setminus \langle f_1, \ldots, f_k \rangle$, und f_{k+1} in dieser Menge mit minimalem Grad gewählt war. ∎

4.3 Univariate Systeme

Es gibt zwei Spezialfälle von algebraischen Gleichungssystemen, die wesentlich einfacher zu handhaben sind, univariate Systeme ($n = 1$) und lineare Gleichungssysteme ($\deg(f_1) = \ldots \deg(f_r) = 1$). Zunächst zu den univariaten Systemen: Für K ein Körper ist, wie in Abschnitt 2.2 diskutiert, $K[X]$ ein Euklidischer Ring. Wir können also den Euklidischen Algorithmus zur Bestimmung des ggT durchführen (wobei die Division mit Rest durch die Polynomdivision realisiert ist). Ideale in Euklidischen Ringen haben eine besonders einfache Struktur, denn jedes Ideal wird von einem einzigen Element erzeugt.

Definition 4.3.1 *Einen kommutativen Ring mit* 1*, in dem jedes Ideal von einem einzigen Element erzeugt ist, bezeichnet man als* ***Hauptidealring****.*

Bemerkung 4.3.2 *1) Jeder Hauptidealring ist also Noethersch.*

2) Der Polynomring in (mindestens) zwei Variablen $K[x, y]$ über einem Körper K ist kein Hauptidealring, siehe Übung 4.3.

Satz 4.3.3 *Euklidische Ringe sind Hauptidealringe.*

Beweis. Sei (R, d) ein euklidischer Ring und $I \subset R$ ein Ideal. Das Ideal $I = \langle 0 \rangle$ ist ein Hauptideal. Sonst betrachten wir $b \in I \setminus \{0\}$ mit $d(b)$ minimal.

Sei $a \in I$ beliebig und $a = g \cdot b + r$ mit $r = 0$ oder $d(r) < d(b)$. Da mit a und b auch $r \in I$ ist, muss $r = 0$ sein, denn sonst hätten wir ein Element kleinerer Norm gefunden. Also ist $a \in \langle b \rangle$.

Damit folgt $I \subset \langle b \rangle \subset I$. ■

Bemerkung 4.3.4 *Der Ring $\mathbb{Z}\left[\frac{1+\sqrt{-19}}{2}\right]$ ist ein Hauptidealring, aber kein euklidischer Ring (ohne Beweis).*

In euklidischen Ringen kann man analog zu $\mathbb{Z}$ die Division mit Rest und den euklidischen Algorithmus 2.1 zur Bestimmung des ggT durchführen, da bei jedem Divisionsschritt die Euklidische Norm kleiner wird. Somit hat man eine Methode, den ggT effizient zu berechnen.

Beispiel 4.3.5 *Wir berechnen den ggT von*

$$f_1 = x^4 + x^3 \quad \text{und} \quad f_2 = x^4 - 1$$

in $\mathbb{Q}[x]$ mit dem Euklidischen Algorithmus

$$\begin{aligned} x^4 + x^3 &= 1 \cdot \left(x^4 - 1\right) + \left(x^3 + 1\right) \\ x^4 - 1 &= x \cdot \left(x^3 + 1\right) + (-x - 1) \\ x^3 + 1 &= \left(-x^2 + x - 1\right) \cdot (-x - 1) + 0 \end{aligned}$$

also

$$\mathrm{ggT}\,(f_1, f_2) = x + 1.$$

Mit dem Euklidischen Algorithmus können wir für ein durch Erzeuger gegebenes Ideal sofort einen einzigen Erzeuger angeben, denn es gilt allgemein für Hauptidealringe:

Satz 4.3.6 *Sei R ein Hauptidealring und $a_1, ..., a_r \in R$. Dann gilt:*

$$\langle a_1, ..., a_r \rangle = \langle \mathrm{ggT}\,(a_1, ..., a_r) \rangle$$

Beweis. Da R ein Hauptidealring ist, ist

$$\langle f_1, ..., f_s \rangle = \langle d \rangle$$

mit einem $d \in R$, und somit $d \mid f_i$ für alle i. Andererseits gibt es $x_i \in R$ mit

$$d = x_1 f_1 + ... + x_s f_s.$$

Somit teilt jeder Teiler von allen f_i auch d. Damit ist

$$d = \mathrm{ggT}\,(f_1, ..., f_s)\,,$$

(wobei diese Gleichung nur bis auf Einheiten Sinn macht). ■

Beispiel 4.3.7 *In $\mathbb{Z}$ gilt*

$$\langle 120, 84 \rangle = \langle \mathrm{ggT}\,(120, 84) \rangle = \langle 12 \rangle\,.$$

In Beispiel 4.3.5 gilt

$$\left\langle x^4 + x^3,\ x^4 - 1 \right\rangle = \langle x + 1 \rangle$$

Dies zeigt, dass

$$V(x^4 + x^3,\ x^4 - 1) = V(x + 1) = \{-1\}\,.$$

Allgemein wird $f = \mathrm{ggT}(f_1, ..., f_r) \in K[x]$ ein Polynom höheren Grades sein. Um die Lösungen von $f(x) = 0$ zu bestimmen, müssen wir die Primfaktorisierung von $\mathbb{Z}$ auf Polynome verallgemeinern. Auf Polynomfaktorisierung werden wir später zurückkommen. Ein alternativer Ansatz ist, die Lösungen mit numerischen Methoden approximativ zu bestimmen.

Bemerkung 4.3.8 *Ist $K = \overline{K}$ algebraisch abgeschlossen, dann korrespondieren nach dem Fundamentalsatz der Algebra die Linearfaktoren von f zu den Lösungen von $f = 0$, das heißt zu den Elementen von $V(f)$.*

Ist K nicht algebraisch abgeschlossen, dann ist dies nicht notwendigerweise so:

Bemerkung 4.3.9 *Über $K = \mathbb{Q}$ haben wir $V(x^2 + 1) = \varnothing$, während $V(x^2 + 1) = \{-i, i\}$ über $K = \mathbb{C}$.*

4.4 Lineare Gleichungssysteme

Der zweite Spezialfall von algebraischen Gleichungssystemen, für die wir bereits einen Algorithmus zur Bestimmung der Lösungsmenge kennen, ist der eines linearen Gleichungssystems. Wir können den Gaußalgorithmus zur Bestimmung einer Zeilenstufenform wie folgt formulieren. Diese Formulierung erlaubt uns später zu sehen, wie sich der Gaußalgorithmus zum Buchbergeralgorithmus verallgemeinert.

Algorithmus 4.1 Gauß

Seien $f_1, ..., f_n \in K[x_1, ..., x_n]$ lineare Polynome. Wir ordnen die Monome der f_i durch

$$x_1 > x_2 > ... > x_m > 1.$$

*Solange es f_i und f_j gibt mit $\mathrm{L}(f_i) = \mathrm{L}(f_j)$ ersetze f_j durch das **S-Polynom** (oder Syzygienpolynom)*

$$\mathrm{spoly}(f_i, f_j) = \mathrm{LC}(f_i)f_j - \mathrm{LC}(f_j)f_i.$$

Ist $f_j = 0$ dann lösche f_j.

Sortieren wir die f_j noch nach der Größe von $L(f_j)$, dann terminiert dieser Algorithmus mit einer **Zeilenstufenform**.

Bemerkung 4.4.1 *Subtrahieren wir von allen f_j geeignete Vielfache aller f_i mit $L(f_i) < L(f_j)$, so können wir erreichen, dass kein Term von*

$$\mathrm{tail}(f_j) := f_j - L(f_j)$$

*durch ein $L(f_i)$ teilbar ist, weiter durch Multipikation mit dem Inversen von $\mathrm{LC}(f_j)$, dass $\mathrm{LC}(f_j) = 1$ für alle j. Damit erhalten wir die eindeutige **reduzierte Zeilenstufenform**.*

Beispiel 4.4.2 *Wir lösen das System*

$$\begin{array}{lllllllllllll} f_1 & = & x_1 & + & x_2 & & & & & + & 1 & = 0 \\ f_2 & = & x_1 & + & x_2 & + & 2x_3 & + & 2x_4 & + & 1 & = 0 \\ f_3 & = & x_1 & + & x_2 & + & x_3 & + & x_4 & + & 1 & = 0 \end{array}$$

Der Gaußalgorithmus liefert das äquivalente System

$$\begin{array}{lllllllllll} f_1 & = & x_1 & + & x_2 & + & & & & + 1 & = 0 \\ \mathrm{spoly}(f_1, f_2) & = & & & & & 2x_3 & + & 2x_4 & & = 0 \\ \mathrm{spoly}(f_1, f_3) & = & & & & & x_3 & + & x_4 & & = 0 \end{array}$$

und schließlich, da das S-Polynom der letzten beiden Polynome verschwindet, das äquivalente System

$$\begin{array}{lllllllllll} f_1 & = & x_1 & + & x_2 & + & & & & + 1 & = 0 \\ \mathrm{spoly}(f_1, f_2) & = & & & & & 2x_3 & + & 2x_4 & & = 0 \end{array}$$

Mit dem (allgemeiner auf jedes algebraische Gleichungssystem anwendbaren) Buchbergeralgorithmus erhalten wir in SINGULAR *die reduzierte Zeilenstufenform durch:*

```
ring R = 0,(x(1..4)),lp;
ideal I = x(1) + x(2) + 1,
x(1) + x(2) + 2*x(3) + 2*x(4) + 1,
x(1) + x(2) + x(3) + x(4) + 1;
option(redSB);
std(I);
_[1] = x(3) + x(4)
_[2] = x(1) + x(2) + 1
```

Für ein Beispiel zur Lösung des Interpolationsproblems mittels linearen Gleichungssystemen siehe Übung 4.5. Dieses Problem kann man auch mit Hilfe des chinesischen Restsatzes für univariate Polynomringe (das heißt mittels des Euklidischen Algorithmus) lösen, siehe Übung 2.20.

4.5 Algebraische Gleichungssysteme und der Nullstellensatz

Bemerkung 4.5.1 *Sei K ein Körper Ein multivariates algebraisches Gleichungssystem in Variablen $x_1, ..., x_n$ über einem Körper K ist durch ein Ideal $I \subset K[x_1, ..., x_n]$ gegeben. Nehmen wir für den Moment der Einfachheit halber an, dass $V(I)$ eine endliche Menge ist. Wir können $V(I)$ bestimmen, indem wir das Problem auf den univariaten Fall zurückführen. Bezeichne mit*

$$\pi_i : K^n \to K,\ (a_1, ..., a_n) \mapsto a_i$$

die Projektion auf die i-te Koordinatenachse. Offenbar gilt

$$\pi_i(V(I)) \subset V(I \cap K[x_i]) \subset K,$$

denn jedes Polynom in $I \cap K[x_i]$ verschwindet, aufgefasst als univariates Polynom, auf $\pi_i(V(I))$.

Beispiel 4.5.2 *Damit erhalten wir den folgenden Algorithmus zur Bestimmung von $V(I)$:*

1) *Berechne $I \cap K[x_i] = \langle f_i \rangle$ mit $f_i \in K[x_i]$. Man beachte, dass $K[x_i]$ ein Hauptidealring ist.*

2) *Bestimme $V(f_i) \subset K$ wie im univariaten Fall.*

3) *Überprüfe, welche Punkte von $V(f_1) \times ... \times V(f_n)$ in $V(I)$ enthalten sind.*

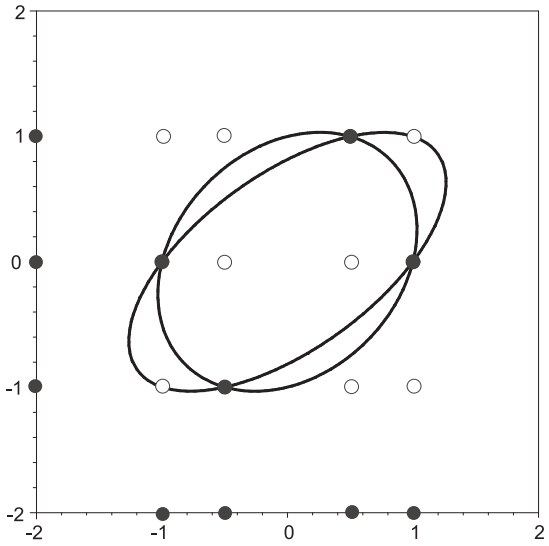


Abbildung 4.2: Projektionen des Durchschnitts von zwei Ellipsen

Siehe dazu Abbildung 4.2 für den Durchschnitt von zwei Ellipsen aus Abschnitt 1. Siehe auch Übung 4.6.

Damit dieses Verfahren algorithmisch funktionieren kann, müssen natürlich die $V(f_i)$ endlich sein. Wir wissen, dass $\pi_i(V(I)) \subset V(f_i)$ endlich ist, aber die beiden Mengen sind i.A. nicht gleich:

Beispiel 4.5.3 *Sei* $I = \langle x_1^2 + x_2^2 \rangle \subset \mathbb{R}[x_1, x_2]$. *Offenbar ist* $V(I) = \{(0,0)\} \subset \mathbb{R}^2$, *jedoch* $I \cap \mathbb{R}[x_i] = \{0\}$. *Wir müssten also für alle (unendlich vielen)* $x_1 \in V(f_1) = \mathbb{R}$ *und* $x_2 \in V(f_2) = \mathbb{R}$ *prüfen, ob* $x_1^2 + x_2^2 = 0$ *erfüllt ist.*

Wir beobachten aber, dass über $K = \mathbb{C}$ *die Menge* $V(x_1^2 + x_2^2)$ *nicht endlich ist, denn für jedes* $x_1 \in \mathbb{C}$ *erhalten wir Lösungen* $(x_1, \pm i \cdot x_1) \in V(x_1^2 + x_2^2)$.

Tatsächlich tritt dieses Problem nicht auf, wenn K algebraisch abgeschlossen ist. Um dies zu sehen, müssen wir die Korrepondenz von algebraischen Menge und Idealen etwas genauer beschreiben. Dazu verwenden wir die folgende Charakterisierung von leeren algebraischen Mengen, die direkt den Fundamentalsatz der Algebra verallgemeinert:

Satz 4.5.4 (Schwacher Nullstellensatz) *Sei* K *ein algebraisch abgeschlossener Körper und* $I \subset K[x_1, \ldots, x_n]$ *ein Ideal. Dann ist*

$$V(I) = \varnothing \iff I = K[x_1, \ldots, x_n]$$

Definition 4.5.5 *Sei* $S \subset K^n$ *eine Teilmenge. Dann ist*

$$I(S) = \{f \in K[x_1, \ldots, x_n] \mid f(p) = 0 \ \forall p \in S\}$$

ein Ideal (wie wir uns oben schon überlegt hatten), das sogenannte ***Verschwindungsideal*** *von* S.

Satz 4.5.6 (Nullstellensatz) *Sei K ein algebraisch abgeschlossener Körper und $J \subset K[x_1, ..., x_n]$ ein Ideal. Dann gilt*

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

wobei

$$\sqrt{J} = \{f \in K[x_1, ..., x_n] \mid \exists a \in \mathbb{N} \text{ mit } f^a \in J\}$$

*das **Radikal** von J bezeichnet. Ist $J = \sqrt{J}$, so heißt J ein **Radikalideal**.*

Beispiel 4.5.7 *Das Radikal vergisst die Information über mehrfache Nullstellen, beispielsweise ist*

$$I(V(\langle x^2 \rangle)) = \sqrt{\langle x^2 \rangle} = \langle x \rangle$$

Genauso haben z.B. die Ideale $\langle x^2, xy, y^2 \rangle$, $\langle x^2, y^2 \rangle$ und $\langle x^2, y \rangle$ alle das selbe Radikal $\langle x, y \rangle$.

Beweis. Nach dem Basissatz können wir J schreiben als $J = \langle f_1, \ldots, f_s \rangle$. Für $f \in I(V(J))$ sei

$$L := \langle J,\ y \cdot f - 1 \rangle \subset K[x_1, \ldots, x_n, y].$$

Da f auf allen gemeinsamen Nullstellen von $f_1, \ldots, f_s$ verschwindet und damit $y \cdot f - 1$ nicht verschwindet, ist $V(L) = \varnothing$. Nach Satz 4.5.4 ist also $L = K[x_1, \ldots, x_n, y]$, d.h. es gibt $c_i, d \in K[x_1, \ldots, x_n, y]$ mit

$$1 = c_1 \cdot f_1 + ... + c_s \cdot f_s + d \cdot (y \cdot f - 1).$$

Setzen wir $y = \frac{1}{f}$ sind die Koeffizienten von der Form $c_i(x_1, \ldots, x_n, \frac{1}{f})$. Multiplizieren wir die Gleichung mit einer genügend hohen Potzenz a von f, dann kürzen sich die Nenner und $f^a \in I$.

Die andere Inklusion ist eine leichte Übung. ■

Mit Hilfe des Nullstellensatzes können wir die Geometrie algebraischer Mengen in die Algebra übersetzen:

Bemerkung 4.5.8 *Für K algebraisch abgeschlossen ist durch*

$$\{\text{algebraische Menge } X \subset K^n\} \underset{V}{\overset{I}{\rightleftarrows}} \{\text{Radikalideale in } K[x_1, \ldots, x_n]\}$$

eine inklusionsumkehrende Bijektion gegeben, also

$$V(I(X)) = X$$
$$I(V(J)) = J$$

für alle algebraischen Mengen X und Radikalideale J.

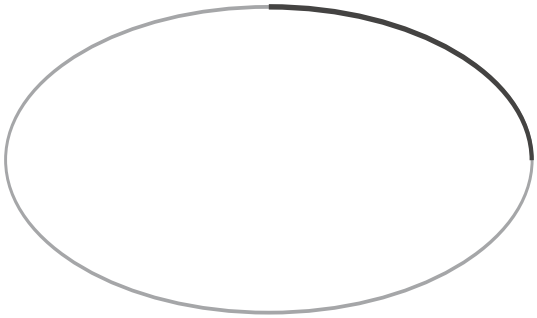


Abbildung 4.3: Ellipsenabschnitt

Beweis. Dass I und V inklusionsumkehrend sind, ist nach Definition klar. Satz 4.5.6 zeigt $I(V(J)) = J$ für J radikal. Ist weiter X eine algebraische Menge, dann verschwindet jedes $f \in I(X)$ auf X, d.h. $X \subset V(I(X))$. Schreiben wir umgekehrt $X = V(f_1, \ldots, f_r)$, dann sind $f_1, \ldots, f_r \in I(X)$, also $\langle f_1, \ldots, f_r \rangle \subset I(X)$ und somit

$$V(I(X)) \subset V(\langle f_1, \ldots, f_r \rangle) = X.$$

■

Bemerkung 4.5.9 *Die Gleichung $V(I(X)) = X$ ist nur für algebraische Mengen korrekt: Sei*

$$S = \left\{(x_1, x_2) \in \mathbb{R}^2 \mid x_1^2 + 2x_2^2 = 1 \text{ and } x_1, x_2 \geq 0\right\}$$

der in Abbildung 4.3 in schwarz eingezeichnete Ellipsenabschnitt. Wir haben

$$I(S) = \left(x_1^2 + 2x_2^2 - 1\right)$$

*also ist $V(I(S))$ die komplette Ellipse, die kleinste algebraische Menge, die S enthält. Dies ist der Abschluss von S in der sogenannten **Zariskitopologie** (in der die algebraischen Mengen genau die abgeschlossenen Menge sind).*

Bemerkung 4.5.10 *Der Beweis von Bemerkung 4.5.8 zeigt, dass die Abbildung $X \mapsto I(X)$ injektiv ist, da $J \mapsto V(J)$ linksinvers dazu ist.*

Nun zurück zu unserem System aus Beispiel 4.5.1:

Beispiel 4.5.11 *Um in unserem Setup zu zeigen, dass $I \cap K[x_i] \neq \langle 0 \rangle$, können wir z.B. verwenden, dass offenbar*

$$g_i := \prod_{a \in V(I)} (x_i - a_i) \in I(V(I)),$$

siehe Abbildung 4.4. Da für K algebraisch abgeschlossen mit dem Nullstellensatz $I(V(I)) = \sqrt{I}$ gilt, ist eine Potenz $g_i^a \in I$ und nach Konstruktion gilt natürlich auch $g_i^a \in K[x_i]$.

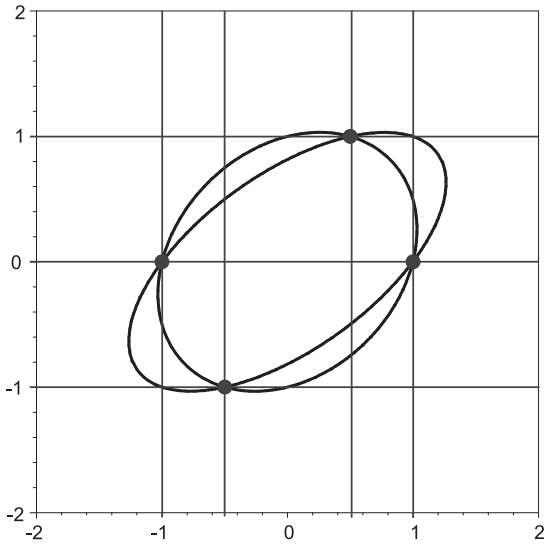


Abbildung 4.4: Elimination für den Durchschnitt von zwei Ellipsen

Das zentrale algorithmische Problem ist nun die Bestimmung der Ideale $I \cap K[x_i] = \langle f_i \rangle$. Analog zum Gaußalgorithmus eliminiert man dazu Variablen, allerdings in nichtlinearen Gleichungen. Ein erst mal leichteres Problem als die Bestimmung von f_i ist die Frage, ob ein gegebenes Polynom (etwa f_i) in einem gegebenen Ideal (etwa $I \cap K[x_i]$) enthalten ist. Wir werden sehen, dass beide Fragestellungen tatsächlich auf dasselbe algorithmische Problem führen.

4.6 Monomordnungen

Zur Elimination von Variablen werden wir den Buchbergeralgorithmus zur Berechnung einer Gröbnerbasis verwenden, der wiederum den Euklidischen Algorithmus im univariaten Fall und den Gaußalgorithmus im linearen Fall verallgemeinert. Dazu benötigen wir eine Verallgemeinerung der Polynomdivision auf mehrere Variablen.

Um in der üblichen Weise durch iteratives Abziehen des Leitterms eine Division mit Rest durchzuführen, müssen wir natürlich zunächst festlegen, welcher Term eines Polynoms der Leitterm ist. Für Monome verwenden wir wie üblich die Multiindexschreibweise $x^\alpha = x_1^{\alpha_1} \cdot \ldots \cdot x_n^{\alpha_n}$ mit dem Exponentenvektor $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}_0^n$.

Definition 4.6.1 *Eine **Monomordnung** (oder **Semigruppenordnung**) auf der multiplikativen Halbgruppe der Monome in den Variablen $x_1, \ldots, x_n$ ist*

1) eine Totalordnung $>$, sodass

2) $>$ *die Multiplikation respektiert, das heißt*

$$x^\alpha > x^\beta \Rightarrow x^\alpha x^\gamma > x^\beta x^\gamma$$

für alle α, β, γ.

Bemerkung 4.6.2 *Die Definition schließt nicht aus, dass* $1 > x$. *Teilen wir zum Beispiel* $\mathbf{1}$ *durch* $\mathbf{1} - x$ *unter dieser Festlegung, so gibt Division mit Rest*

$$\begin{aligned} \mathbf{1} &= 1 \cdot (\mathbf{1} - x) + \boldsymbol{x} \\ &= (1 + x) \cdot (\mathbf{1} - x) + \boldsymbol{x}^2 \\ &\vdots \\ &= \left(\textstyle\sum_{i=0}^{\infty} x^i\right) \cdot (\mathbf{1} - x) + 0 \end{aligned}$$

Wir erhalten somit die Entwicklung der geometrischen Reihe

$$\frac{1}{1-x} = \textstyle\sum_{i=0}^{\infty} x^i.$$

Division mit Rest funktioniert also wie erwartet, gibt aber keine Antwort in endlich vielen Schritten.

Dieses Problem kann man durch Verwendung einer sogenannten globalen Ordnung vermeiden:

Definition und Satz 4.6.3 *Eine* ***globale Ordnung*** *ist eine Monomordnung* $>$ *mit den folgenden äquivalenten Eigenschaften:*

1) $>$ *ist eine Wohlordnung*

(d.h. jede nichtleere Menge von Monomen hat ein kleinstes Element).

2) $x_i > 1 \ \forall i$.

3) $x^\alpha > 1$ *für alle* $0 \neq \alpha \in \mathbb{N}_0^n$.

4) Falls $x^\beta \mid x^\alpha$ *und* $x^\alpha \neq x^\beta$ *dann* $x^\alpha > x^\beta$

(d.h., $>$ *verfeinert die Halbordnung nach Teilbarkeit).*

Ist $x_i < 1 \ \forall i$, *dann heißt* $>$ *eine* ***Lokalordnung****.*

Beweis. Die Implikationen (1) $\Rightarrow$ (2) $\Rightarrow$ (3) $\Rightarrow$ (4) sind leicht zu sehen, siehe Übung 4.10. Für (4) $\Rightarrow$ (1) zeigen wir im folgenden Lemma, dass jede nichtleere Menge von Monomen nur endlich viele minimale Elemente bezüglich Teilbarkeit hat. Dann müssen wir nach Annahme (4) nur diese minimalen Elemente betrachten, und, da $>$ eine Totalordnung ist, gibt es unter diesen endlich vielen ein minimales Element. ■

Lemma 4.6.4 (Dickson, Gordan) *Jede nichtleere Menge von Monomen hat nur endlich viele minimale Elemente bezüglich Teilbarkeit.*

Beweis. Sei $M \neq \varnothing$ eine Menge von Monomen in den Variablen $x_1, \ldots, x_n$, und sei $\langle M \rangle \subset K[x_1, \ldots, x_n]$ das Ideal erzeugt von den Elementen von M. Nach dem Hilbertschen Basissatz 4.2.3 ist $\langle M \rangle = \langle f_1, \ldots, f_s \rangle$ mit Polynomen $f_i = \sum_{j=1}^{u} r_{i,j} m_j$ wobei $r_{i,j} \in K[x_1, \ldots, x_n]$ und $m_1, \ldots, m_u \in M$. Damit ist

$$\langle M \rangle \subset \langle m_1, \ldots, m_u \rangle \subset \langle M \rangle .$$

Unter den $m_1, \ldots, m_u$ wählen wir die minimalen Elemente in Bezug auf Teilbarkeit. ■

Das im Beweis betrachtete Ideal ist ein Beispiel eines monomialen Ideals:

Definition 4.6.5 *Ein Ideal $I \subset K[x_1, \ldots, x_n]$ heißt **monomiales Ideal**, wenn es von Monomen erzeugt wird.*

Corollar 4.6.6 *Jedes monomiale Ideal hat ein eindeutiges **minimales Erzeugendensystem** aus endlich vielen Monomen.*

Beweis. Siehe den Beweis von Lemma 4.6.4 (oder wende das Lemma auf die Menge der Monome in dem Ideal an). ■

In dem Beweis haben wir auch die folgende triviale, aber sehr wichtige Beobachtung gemacht:

Lemma 4.6.7 *Sei $I = \langle M \rangle$ ein monomiales Ideal erzeugt von den Monomen in M. Ist $f \in I$, dann ist jeder Term von f in I.*

Insbesondere falls $f \in I$ ein Monom ist, dann gibt es ein $m \in M$ mit $m \mid f$.

Beweis. Ist $f = \sum_{j=1}^{u} r_j m_j \in I$ mit $r_j \in K[x_1, \ldots, x_n]$ und $m_j \in M$, dann ist jeder Term von f ein Vielfaches von einem Term von einem (eventuell mehreren) $r_j m_j$ und damit ein Vielfaches von m_j. ■

Wir diskutieren nun einige Beispiele von Monomordnungen. Zunächst bemerken wir:

Beispiel 4.6.8 *In einer Variablen x ist durch $x > 1$ eine eindeutige Monomordnung festgelegt, denn mit Definition 4.6.1(1) folgt $x^{i+1} > x^i$ für alle i. Genauso ist durch $x < 1$ eine eindeutige Monomordnung bestimmt. Alle globalen Monomordnungen sind also äquivalent zu $x > 1$, alle lokalen Ordnungen zu $x < 1$.*

Beispiel 4.6.9 *Die folgenden Ordnungen sind globale Monomordnungen*

1) *Die* ***lexikographische*** *Ordnung:*

$x^\alpha > x^\beta \iff$ *der erste Eintrag* $\neq 0$ *von links in* $\alpha - \beta$ *ist positiv*

In SINGULAR *wird diese Ordnung abgekürzt als* ***lp****.*

2) *Die* ***Grad-reverse-lexikographische Ordnung****:*

$x^\alpha > x^\beta \Leftrightarrow \deg x^\alpha > \deg x^\beta$ *oder* ($\deg x^\alpha = \deg x^\beta$ *und* $\exists 1 \leq i \leq n$: $\alpha_n = \beta_n, ..., \alpha_{i+1} = \beta_{i+1}, \alpha_i < \beta_i$).

In SINGULAR *wird diese Ordnung mit* ***dp*** *abgekürzt.*

Ein Beispiel einer lokalen Ordnung ist die ***negative lexikographische Ordnung****:*

$x^\alpha > x^\beta \iff$ *der erste Eintrag* $\neq 0$ *von links in* $\alpha - \beta$ *ist negativ*

In SINGULAR *wird diese Ordnung mit* ***ls*** *abgekürzt.*

Beispiel 4.6.10 *Für lp auf den Monomen in* x, y, z *haben wir (wobei wir Monome und Exponentenvektoren identifizieren)*

$$\begin{aligned} x = (1,0,0) > y = (0,1,0) > z = (0,0,1) \\ xy^2 = (1,2,0) > (0,3,4) = y^3z^4 \\ x^3y^2z^4 = (3,2,4) > (3,1,5) = x^3y^1z^5 \end{aligned}$$

Andererseits erhalten wir für dp

$$\begin{aligned} x = (1,0,0) > y = (0,1,0) > z = (0,0,1) \\ xy^2 = (1,2,0) < (0,3,4) = y^3z^4 \\ x^3y^2z^4 = (3,2,4) > (3,1,5) = x^3yz^5 \end{aligned}$$

und für ls

$$\begin{aligned} x = (1,0,0) < y = (0,1,0) < z = (0,0,1) \\ xy^2 = (1,2,0) < (0,3,4) = y^3z^4 \\ x^3y^2z^4 = (3,2,4) < (3,1,5) = x^3yz^5 \end{aligned}$$

In SINGULAR *können wir Monome folgendermaßen vergleichen:*

```
ring R=0,(x,y,z),lp;
x>y;
1
y>z;
1
x*y^2>y^3*z^4;
1
x^3*y^2*z^4>x^3*y*z^5
1
```

```
ring R=0,(x,y,z),dp;
x>y;
1
y>z;
1
x*y^2>y^3*z^4;
0
x^3*y^2*z^4>x^3*y*z^5
1
ring R=0,(x,y,z),ls;
1>z;
1
z>y;
1
y>x;
1
x*y^2>y^3*z^4;
0
x^3*y^2*z^4>x^3*y*z^5
0
```

Beispiel 4.6.11 *Für lp haben wir*

$$L(5x^2y + yx^2) = x^2y.$$

In SINGULAR *bestimmen wir den Leitterm, das Leitmonom und den Leitkoeffizienten wie folgt:*

```
ring R=0,(x,y,z),lp;
poly f = 5*x^2*y+x*y^2;
lead(f);
5x2y
leadcoef(f);
5
leadmonom(f);
x2y
```

4.7 Division mit Rest und Gröbnerbasen

Gegeben eine globale Monomordnung > auf den Monomen von

$$R = K[x_1, ..., x_n],$$

lässt sich wie in Algorithmus 4.2 beschrieben die Division mit Rest durchführen.

Algorithmus 4.2 Division mit Rest

Input: $f \in R$, $g_1, ..., g_s \in R$, $>$ eine globale Ordnung auf den Monomen von R.

Output: Einen Ausdruck

$$f = q + r = \sum_{i=1}^{s} a_i g_i + r$$

sodass $L(r)$ durch kein $L(g_i)$ teilbar ist.

1: $q = 0$
2: $r = f$
3: **while** $r \neq 0$ **and** $L(g_i) \mid L(r)$ für ein i **do**
4: *Kürze den Leitterm von r:*
5: $a = \frac{LT(r)}{LT(g_i)}$
6: $q = q + a \cdot g_i$
7: $r = r - a \cdot g_i$

Beweis. In jeder Iteration wird der Leitterm von r kleiner bezüglich $>$, somit terminiert der Algorithmus, da $>$ eine Wohlordnung ist. ■

Wie wir in Bemerkung 4.6.2 gesehen haben, können wir für nicht-global Ordnungen im Allgemeinen nicht erwarten, dass die Division in endlich vielen Schritten terminiert.

Beispiel 4.7.1 *Bezüglich der lexikographischen Ordnung auf $K[x, y]$ mit $x > y$ teilen wir $f = x^2y + x$ durch $G = \{y - 1, x^2 - 1\}$*

$$\begin{array}{rl} \boldsymbol{x^2y} + x = & x^2\,(\boldsymbol{y} - 1) + 1 \cdot (\boldsymbol{x^2} - 1) + x + 1 \\ \underline{\quad x^2y - x^2} & \\ \quad \boldsymbol{x^2} + x & \\ \underline{\quad x^2 - 1} & \\ \quad \boldsymbol{x} + 1 & \end{array}$$

und erhalten den Rest $x + 1$. Leitterme notieren wir in rot, Reste in grün.

In SINGULAR *können wir diese Rechnung durchführen mit:*

```
ring R = 0,(x,y),lp;
poly f = x^2*y+x;
ideal I = y-1, x^2-1;
reduce(f,I);
x+1
```

Für ein weiteres Beispiel siehe Übung 4.7.

Wir zeigen nun, dass Algorithm 4.2 das Ideal-Membership-Problem löst, unter der Voraussetzung, dass wir nach einem geeigneten Erzeugendensystem G teilen. Dazu müssen wir in der Lage sein, jeden möglichen Leitterm in dem Ideal $I = \langle G \rangle$ mit einem Leitterm eines Elements von G zu kürzen. Wir formulieren diese Bedingung wie folgt:

Definition 4.7.2 *Gegeben eine Monomordnung $>$ und eine Teilmenge $G \subset R$, definieren wir das* ***Leitideal*** *von G als*

$$L_>(G) = \langle L(f) \mid f \in G \backslash \{0\} \rangle \subset R,$$

das von den Leitmonomen erzeugte monomiale Ideal. Wenn die Wahl von $>$ klar ist, schreiben wir einfach $L(G) = L_>(G)$.

Definition 4.7.3 (Gröbnerbasen) *Sei I ein Ideal und $>$ eine globale Monomordnung. Eine endliche Menge*

$$G \subset I$$

mit $0 \notin G$ heißt ***Gröbnerbasis*** *von I bezüglich $>$, falls*

$$L(G) = L(I).$$

Man beachte, dass die Inklusion $\subset$ für jede Teilmenge G von I erfüllt ist. Die Existenz einer Gröbnerbasis sieht man leicht:

Satz 4.7.4 *Jedes Ideal $I \subset R$ hat eine Gröbnerbasis.*

Beweis. Da $L(I)$ endlich erzeugt ist, ist $L(I) = \langle m_1, \ldots, m_s \rangle$ mit Monomen m_i. Weiter ist nach Lemma 4.6.7 jedes m_i teilbar durch $L(g_i)$ für ein Element $g_i \in I$. Damit ist

$$L(I) = \langle m_1, \ldots, m_s \rangle \subset \langle L(g_1), \ldots, L(g_s) \rangle \subset L(I),$$

also bilden $g_1, \ldots, g_s$ eine Gröbnerbasis von I. ■

Von der Definition ist es erst mal nicht klar, ob G tatsächlich ein Erzeugendensystem von I ist. Indem wir das Ideal-Membership-Problem lösen, werden wir auch diese Frage beantworten.

Um transparent die Division mit Rest verwenden zu können, formlisieren zunächst die abstrakten Eigenschaften von Algorithmus 4.2 in der Definition einer Normalform.

Definition 4.7.5 *Sei $>$ eine globale Monomordnung auf R. Für eine Menge $G = \{g_1, ..., g_s\}$ von Polynomen $g_i \in R$, ist eine* ***Normalform*** *bezüglich $>$ eine Abbildung $\mathrm{NF}(-, G) : R \to R$ mit*

1) $\mathrm{NF}(0, G) = 0$.

2) Ist $\mathrm{NF}(f, G) \neq 0$ *dann ist* $L(\mathrm{NF}(f, G)) \notin L(G)$.

3) Für alle $0 \neq f \in R$ *gibt es* $a_i \in R$ *mit*

$$f - \mathrm{NF}(f, G) = \textstyle\sum_{i=1}^{s} a_i g_i \tag{4.1}$$

und $L(f) \geq L(a_i g_i)$ *für alle i mit* $a_i g_i \neq 0$.

Einen Ausdruck wie in Gleichung 4.1 bezeichnen wir auch als ***Standardausdruck****. Wir sagen auch, dass* NF *eine Normalform ist, falls* $\mathrm{NF}(-, G)$ *eine Normalform für alle* G *ist.*

Lemma 4.7.6 *Gegeben* $G = \{g_1, \ldots, g_s\}$ *und jede gewählte Ordnung der* g_i *in der Division, liefert Algorithm 4.2 eine Normalform* $\mathrm{NF}(-, G)$*. Diese nennen wir die* ***Buchberger Normalform****.*

Beweis. Wir bilden f auf $\mathrm{NF}(f, G) := r$ ab. Falls der Algorithmus $r \neq 0$ zurückgibt, dann ist $L(r)$ durch kein $L(g_i)$ teilbar, also $L(r) \notin L(G)$ nach Lemma 4.6.7. Bedingung (3) gilt, da in jeder Iteration des Algorithmus $L(a \cdot g_i) \leq L(f)$. ■

Mit Hilfe von Gröbnerbasen und einer Normalform können wir nun das Ideal-Membership-Problem lösen:

Satz 4.7.7 (Ideal-Membership) *Sei* $I \subset R$ *ein Ideal und* $f \in R$*. Ist* $G = \{g_1, \ldots, g_s\}$ *eine Gröbnerbasis von* I *und* NF *eine Normalform, dann gilt*

$$f \in I \iff \mathrm{NF}(f, G) = 0.$$

Beweis. Betrachte einen Standardausdruck $f = \sum_i a_i g_i + r$ with $r = \mathrm{NF}(f, G)$, $a_i \in R$. Falls $r = 0$ dann ist $f = \sum_i a_i g_i \in \langle G \rangle \subset I$ und damit $f \in I$. Ist andereseits $r \neq 0$ dann ist nach Definition 4.7.5 (2.)

$$L(r) \notin L(G) = L(I).$$

Nach der Definition des Leitideals haben wir also

$$r \notin I,$$

und damit $f = \sum_i a_i g_i + r \notin I$. ■

Satz 4.7.7 zeigt auch, dass eine Gröbnerbasis ein Erzeugendensystem ist, denn für jedes $f \in I$ haben wir einen Standardausdruck

$$f - \textstyle\sum_{i=1}^{s} a_i g_i = \mathrm{NF}(f, G) = 0$$

mit $a_i \in R$. Diese Beobachtung kann man auch elegant wie folgt formulieren:

Lemma 4.7.8 *Sind* $J \subset I \subset R$ *Ideale mit* $L(J) = L(I)$*, dann ist* $I = J$*.*

Beweis. Sei $G = \{g_1, \ldots, g_s\}$ eine Gröbnerbasis von J, NF eine Normalform, $f \in I$ and $f = \sum_i a_i g_i + r$ ein Standardausdruck für $r = \mathrm{NF}(f, G)$. Also ist $r \in I$. Falls $r \neq 0$, dann muss nach Definition 4.7.5 (2.)

$$L(r) \notin L(G) = L(J) = L(I)$$

gelten. Nach der Definition des Leitideals ist dann aber $r \notin I$, ein Widerspruch. ■

Corollar 4.7.9 *Ist G eine Gröbnerbasis von I, dann gilt*

$$I = \langle G \rangle .$$

Beweis. Wir haben $L(I) = L(G) \subset L(\langle G \rangle) \subset L(I)$, also ist G eine Gröberbasis von $\langle G \rangle \subset I$ und $L(\langle G \rangle) = L(I)$. Gleichheit folgt aus Lemma 4.7.8. ■

Beispiel 4.7.10 *Die Erzeuger des Ideals $I = \langle x^2 - 1, y - 1 \rangle$ bilden schon eine Gröbnerbasis bezüglich lp: Weil $x \notin L(I)$ (Übung) haben wir*

$$L(I) = \langle x^2, y \rangle .$$

Mit Hilfe einer Gröbnerbasis können wir insbesondere überprüfen, ob ein gegebenes Ideal monomial ist:

Beispiel 4.7.11 *Das Ideal*

$$I = \langle xy^3 + x^2y^2,\ xy^3 - x^2y^2,\ x^5y,\ x^2y^4 \rangle$$

ist monomial. Um dies zu sehen, bestimmen wir eine Gröbnerbasis G von I und testen mit Division mit Rest ob jeder Term jedes $g \in G$ in I liegt:

```
ring R = 0,(x,y),lp;
ideal I = x*y^3+x^2*y^2, x*y^3-x^2*y^2,x^5*y,x^2*y^4;
ideal G = std(I);
G;
_[1]=xy3
_[2]=x2y2-xy3
_[3]=x5y
reduce(x2y2,G);
0
reduce(xy3,G);
0
```

Somit ist

$$I = \langle xy^3,\ x^2y^2,\ x^5y \rangle .$$

Abbildung 4.5 zeigt die Monome in I.

Wir bemerken noch, dass die Lösung des Ideal-Membership-Problems vermöge Division mit Rest sogar fäquivalent zur Gröbnerbaseneigenschaft ist:

Proposition 4.7.12 *Sei $I \subset R$ ein Ideal, $>$ eine globale Monomordnung, und $0 \notin G = \{g_1, \ldots, g_s\} \subset I$ und $\mathrm{NF}(-, G)$ eine Normalform. Dann sind äquivalent:*

1) $L(G) = L(I)$, d.h. G ist eine Gröbnerbasis von I,

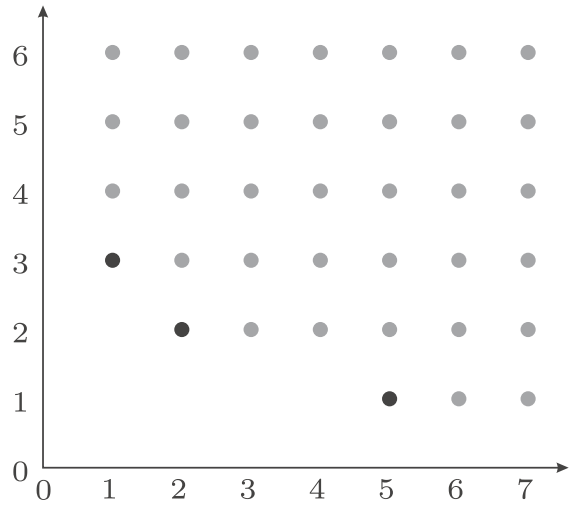


Abbildung 4.5: Monome in $\langle xy^3, x^2y^2, x^5y \rangle$

2) $\mathrm{NF}(f, G) = 0 \Leftrightarrow f \in I$.

Beweis. (1) $\Rightarrow$ (2) haben wir in Satz 4.7.7 gezeigt.

(2) $\Rightarrow$ (1): Für $f \in I$ ist $\mathrm{NF}(f, G) = 0$ und nach Definition 4.7.5(3) haben wir einen Ausdruck

$$f = \sum_{i=1}^{s} a_i g_i$$

mit $L(f) \geq L(a_i g_i)$ für alle i. Somit muss es ein i geben mit $L(f) = L(a_i g_i)$, was wiederum impliziert, dass $L(g_i) \mid L(f)$. Somit ist $L(f) \in L(G)$. ■

4.8 Elimination

Mit Hilfe der lexikograpahischen Ordnung und dem Begriff der Gröbnerbasis können wir das Eliminationsproblem lösen.

Satz 4.8.1 *Sei $I \subset K[x_1, ..., x_n]$ ein Ideal und G eine Gröbnerbasis von I bezüglich der lexikographischen Ordnung. Dann ist für jedes m*

$$H_m = \{g \in G \mid L(g) \in K[x_{m+1}, ..., x_n]\}$$

eine Gröbnerbasis von $J_m = I \cap K[x_{m+1}, ..., x_n]$.

Wir bezeichnen J_m auch als das m-te ***Eliminationsideal****.*

Beweis. Man beachte, dass $L(g) \in K[x_{m+1}, ..., x_n]$ impliziert, dass $g \in K[x_{m+1}, ..., x_n]$.

Insbesondere ist also $H_m \subset J_m$. Sei nun $f \in J_m$. Da G eine Gröbnerbasis von I ist und $J_m \subset I$, gilt $\mathrm{NF}(f, G) = 0$. Da $f \in K[x_{m+1}, ..., x_n]$, können wir zum Kürzen des Leitterms nur Elemente $g \in G$ verwenden

mit $L(g) \in K[x_{m+1}, ..., x_n]$, also mit $g \in H_m$. Das Resultat liegt dann offenbar wieder in $K[x_{m+1}, ..., x_n]$. Induktiv verwendet die Division also nur Elemente aus H_m und terminiert nach Annahme mit 0. Nach Proposition 4.7.12 ist damit H_m eine Gröbnerbasis von J_m. ■

Beispiel 4.8.2 *Betrachte das Ideal*

$$I = \left\langle 2x^2 - xy + 2y^2 - 2,\ 2x^2 - 3xy + 3y^2 - 2 \right\rangle$$

aus Abschnitt 1. Dort haben wir schon eine Gröbnerbasis von I bezüglich der lexikographischen Ordnung mit $y > x$ berechnet:

```
ring R=0,(y,x),lp;
ideal I = 2*x^2-x*y+2*y^2-2, 2*x^2-3*x*y+3*y^2-2;
groebner(I);
_[1]=4x4-5x2+1
_[2]=3y+8x3-8x
```

Dies zeigt, dass

$$I \cap K[x] = \left\langle 4x^4 - 5x^2 + 1 \right\rangle = \langle (x+1)(x-1)(2x+1)(2x-1) \rangle .$$

Um x zu eliminieren, verwenden wir analog die lexikographische Ordnung mit $x > y$:

```
ring R=0,(x,y),lp;
ideal I = 2*x^2-x*y+2*y^2-2, 2*x^2-3*x*y+3*y^2-2;
groebner(I);
_[1] = y3-y
_[2]= 2xy-y2
_[3]= 2x2-3xy+3y2-2
```

Somit ist

$$I \cap K[y] = \left\langle y^3 - y \right\rangle = \langle y(y+1)(y-1) \rangle ,$$

insgesamt also

$$V(I) \subset \{-1, 1, -\frac{1}{2}, \frac{1}{2}\} \times \{0, -1, 1\} .$$

Indem wir testen welche dieser 12 Punkte

$$2x^2 - xy + 2y^2 - 2 = 0 \text{ und}$$
$$2x^2 - 3xy + 3y^2 - 2 = 0$$

erfüllen, erhalten wir

$$V(I) = \left\{ (1,0), (-1,0), \left(\frac{1}{2}, 1\right), \left(-\frac{1}{2}, -1\right) \right\} ,$$

siehe Abbildung 4.2.

Für ein anderes Beispiel siehe Übung 4.8. Eine weitere wichtige Anwendung der Elimination ist die Bestimmung von impliziten Gleichungen von durch Parametrisierungen gegebenen Mengen. Siehe dazu Übung 4.9.

4.9 Buchbergeralgorithmus

Die wesentliche Eigenschaft einer Gröbnerbasis eines Ideals ist, dass die Division nach der Gröbnerbasis das Ideal-Membership-Problem löst. Ist also G eine Gröbnerbasis von $I \subset R = K[x_1, ..., x_n]$ bezüglich $>$ und $\mathrm{NF}(-, G)$ eine Normalform, dann ist

$$f \in I \Leftrightarrow \mathrm{NF}(f, G) = 0.$$

Dies führt direkt auf einen Algorithmus zur Berechnung einer Gröbnerbasis.

Beispiel 4.9.1 *Betrachten wir zum Beispiel die Fragestellung, ob*

$$f = x^2 - y^2 \in I = \left\langle x^2 + y,\ xy + x \right\rangle.$$

Division mit Rest von f bezüglich lp nach dem Erzeugendensystem $G = \{x^2 + y,\ xy + x\}$ von I gibt

$$\begin{array}{ll} \boldsymbol{x}^2 - y^2 = & 1 \cdot (\boldsymbol{x}^2 + y) + (-y^2 - y) \\ \underline{x^2 + y} & \\ \boldsymbol{-y}^2 - y & \end{array}$$

Also erhalten wir $\mathrm{NF}(f, G) = -y^2 - y \neq 0$, jedoch

$$\boldsymbol{x}^2 - y^2 = -y\left(\boldsymbol{x}^2 + y\right) + x\left(\boldsymbol{xy} + x\right) \in I,$$

insbesondere ist G keine Gröbnerbasis von I.

Das Problem wird dadurch verursacht, dass sich in dieser Darstellung von $f \in I$ die Leitterme kürzen, denn eine solche Darstellung würde nie von der Division mit Rest erzeugt. Wie löst man dieses Problem? In dem obigen Beispiel können wir einfach $y^2 + y$ zu G hinzufügen und dann die Division mit dem Rest 0 beenden.

Die Idee ist also systematisch zu G Elemente von I hinzuzunehmen, bis aus der Inklusion

$$L(G) \subset L(I)$$

eine Gleichheit wird, d.h. der Leitterm jedes Elements von I ein Vielfaches des Leitterms eines Elements von G ist (siehe Lemma 4.6.7). In dem Beispiel haben wir das Polynom $x^2 - y^2 \in I$ durch Kürzen der Leitterme der zwei Erzeuger $x^2 + y$ und $xy + x$ von I erhalten. Wie wir sehen werden, ist es auch allgemein ausreichend, die Leitterme von je zwei Elementen von G zu kürzen. Das Kürzen realisiert man durch die Bildung des S-Polynoms:

Definition 4.9.2 *Das **S-Polynom** (oder **Syzygienpolynom**) von $f, g \in K[x_1, ..., x_n]$ ist definiert als*

$$\mathrm{spoly}(f, g) = \frac{\mathrm{kgV}(L(f), L(g))}{\mathrm{LT}(f)} f - \frac{\mathrm{kgV}(L(f), L(g))}{\mathrm{LT}(g)} g.$$

Für $I = \langle G \rangle$ ist mit $f, g \in G$ auch $\mathrm{spoly}(f, g) \in I$ und sollte deshalb zu 0 reduzieren, falls G schon eine Gröbnerbasis war. Wir bilden also $\mathrm{spoly}(f, g)$ für $f, g \in G$, prüfen ob Division mit Rest nach G Null ergibt, falls nicht fügen wir

$$\mathrm{NF}(\mathrm{spoly}(f, g), G) \in I$$

zu G hinzu und iterieren, siehe Algorithmus 4.3. Dieses Verfahren bezeichnet man als den **Buchbergeralgorithmus**.

Algorithmus 4.3 Buchberger

Input: $I = \langle g_1, ..., g_s \rangle \subset R$ ein Ideal, und die globale Ordnung $>$.
Output: Eine Gröbnerbasis G von I bezüglich $>$.

```
1: G = {g_1, ..., g_s}
2: repeat
3:    H = G
4:    for all f, g ∈ H do
5:       r = NF(spoly(f, g), H)
6:       if r ≠ 0 then
7:          G = G ∪ {r}
8: until G = H
```

Beweis. Sind $f, g \in H$ und $r = \mathrm{NF}(\mathrm{spoly}(f, g), H) \neq 0$ dann $L(r) \notin L(H)$ nach Definition 4.7.5(2.), also

$$L(H) \subsetneq L(H \cup \{r\}).$$

Da $K[x_1, ..., x_n]$ nach Satz 4.2.3 Noethersch ist, terminiert der Algorithmus nach Satz 4.2.1.

Es bleibt noch zu zeigen, dass

$$\mathrm{NF}(\mathrm{spoly}(f, g), G) = 0$$

für alle $f, g \in G$ impliziert, dass G eine Gröbnerbasis ist. Darauf werden werden wir in Abschnitt 4.12 über das Buchbergerkriterium zurückkommen. ■

Zunächst erproben wir den Algorithmus an einem Beispiel:

Beispiel 4.9.3 *In obigem Beispiel 4.9.1 bilden wir*

$$\mathrm{spoly}(\boldsymbol{x}^2 + y,\ \boldsymbol{xy} + x) = -\boldsymbol{x}^2 + y^2,$$

reduzieren dies mit Division mit Rest zu

$$\mathrm{NF}(-\boldsymbol{x}^2 + y^2, G) = y^2 + y$$

und erhalten

$$G = \{\boldsymbol{x}^2 + y,\ \boldsymbol{xy} + x,\ \boldsymbol{y}^2 + y\}.$$

In der nächsten Iteration müssen wir nun noch

$$\operatorname{spoly}(\boldsymbol{xy} + x,\ \boldsymbol{y}^2 + y) = 0$$

und

$$\operatorname{spoly}(\boldsymbol{x}^2 + y,\ \boldsymbol{y}^2 + y) = -\boldsymbol{x}^2\boldsymbol{y} + y^3$$

bilden. Division mit Rest liefert

$$\begin{array}{rl} -\boldsymbol{x}^2\boldsymbol{y} + y^3 = & (-y)\cdot(\boldsymbol{x}^2 + y) + y\cdot(y^2 + y) + 0 \\ \underline{\quad -x^2y - y^2} & \\ \boldsymbol{y}^3 + y^2 & \\ \underline{\quad y^3 + y^2} & \\ 0 & \end{array}$$

Somit erhalten wir keine neuen Elemente mehr, und der Buchbergeralgorithmus terminiert.

Beispiel 4.9.4 *In* SINGULAR *können wir diese Rechnung folgendermaßen durchführen:*

```
ring R = 0,(x,y),lp;
ideal I = x^2+y, x*y+x;
groebner(I);
_[1]=y2+y
_[2]=xy+x
_[3]=x2+y
```

Bemerkung 4.9.5 *Für den Buchbergeralgorithmus 4.3 angewendet in n Variablen auf ein Erzeugendensystem mit Elementen vom Grad maximal d ist der Grad der Polynome in der Gröbnerbasis beschränkt durch*

$$2\left(\frac{1}{2}d^2 + d\right)^{2^{n-1}}$$

ist also polynomial in d, und doppelt exponentiell in n d.h. in

$$O(\exp(\exp(n))).$$

In vielen praktischen Beispielen verhält sich der Buchbergeralgorithmus wesentlich besser als dieser worst-case.

Ausgehend von der Gradabschätzung kann man zeigen, dass auch die Laufzeit doppelt exponentiell ist.

Beispiel 4.9.6 *Die Ordnung dp bietet üblicherweise die beste Performance, besonders bei der Berechnung von Gröbnerbasen von homogenen Idealen. Ein* ***homogenes Ideal*** *ist ein Ideal erzeugt von homogenen Polynomen. Ein* ***homogenes Polynom*** *ist eine Summe von Termen, die alle denselben Grad haben. Homogene Ideale treten auf,*

wenn man algebraische Mengen im projektiven Raum (z.B. in der Ebene inklusive dem unendlich fernen Horizont) beschreiben will, siehe Bemerkung 4.9.7 im Anschluss.

Wir vergleichen den Buchbergeralgorithmus für die Ordnungen dp und lp anhand eines zufällig erzeugten homogenen Ideals mit 4 Erzeugern vom Grad 12 und Koeffizienten vom Betrag ≤ 2:

```
LIB "random.lib";
ring R=0,(x,y,z),dp;
ideal I = randomid(maxideal(12),4,50);
int t = timer;
ideal J = groebner(I);
timer - t;
10
size(J);
109
ring S=0,(x,y,z),lp;
ideal I = imap(R,I);
t = timer;
ideal J = groebner(I);
timer - t;
15
size(J);
146
```

Die Rechnung liefert also mit dp schneller eine Gröbnerbasis, und diese hat auch deutlich weniger Elemente. Trotzdem hat die Ordnung lp eine wichtige Anwendung in der Elimination von Variablen.

Ein Beispiel für die explizite Durchführung des Buchbergeralgorithmus unter Verwendung der Ordnung *dp* gibt Übung 4.11.

Bemerkung 4.9.7 *Der n-dimensionale* ***projektive Raum*** *über K ist*

$$\mathbb{P}^n(K) = \left\{1\text{-dimensionale Untervektorräume von } K^{n+1}\right\}$$

Ist $\langle(p_0,\ldots,p_n)\rangle \in \mathbb{P}^n(K)$*, also* $(p_0,\ldots,p_n) \neq 0$*, dann schreibt man*

$$(p_0 : \ldots : p_n) := \langle(p_0,\ldots,p_n)\rangle\,.$$

Die reelle 2-dimensionale projektive Ebene $\mathbb{P}^2(\mathbb{R})$ *können wir uns als eine Halbkubel vorstellen, bei der gegenüberliegende Punkte identifiziert werden (da sie ja den selben 1-dimensionalen Untervektorraum erzeugen), siehe Abbildung 4.6. Um algebraische Teilmengen von* $\mathbb{P}^n(K)$ *zu spezifizieren, verwendet man homogene Polynome: Ist* $f \in R$ *homogenen vom Grad d, dann gilt*

$$f(\lambda\cdot(p_0,\ldots,p_n)) = \lambda^d \cdot f(p_0,\ldots,p_n)$$

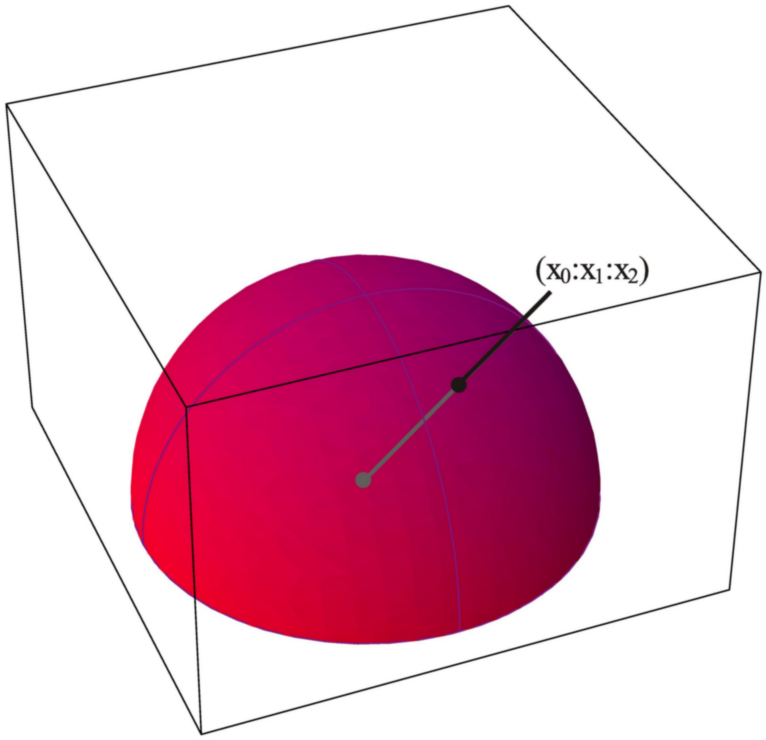


Abbildung 4.6: Projektiver Raum $\mathbb{P}^2(\mathbb{R})$.

für $\lambda \neq 0$, *die Bedingung* $f(p_0 : \ldots : p_n) = 0$ *ist also für* $(p_0 : \ldots : p_n) \in \mathbb{P}^n(K)$ *wohldefiniert (während der Wert von* f *an einem Punkt es nicht ist). Für ein homogenes Ideal* $I \subset R$ *sind (analog zu der Situation von monomialen Idealen) mit einem Polynom* $f \in I$ *auch alle homogenen Summanden von* f *in* I *(Übung). Deshalb macht es Sinn, zu definieren*

$$V(I) = \{p \in \mathbb{P}^n(K) \mid f(p) = 0 \ \forall \ \text{homogenen } f \in I\}\,.$$

Durch die bijektive Abbildung

$$\begin{array}{rccc} \varphi: & K^n & \to & U \\ & (p_1, \ldots, p_n) & \mapsto & (1 : p_1 : \ldots : p_n) \\ & \left(\frac{p_1}{p_0}, \ldots, \frac{p_n}{p_0}\right) & \leftarrow\!\shortmid & (p_0 : p_1 : \ldots : p_n) \end{array}$$

von K^n *auf*

$$U = \{p \in \mathbb{P}^n(K) \mid p_0 \neq 0\} \subset \mathbb{P}^n(K)$$

kann man K^n *als eine Teilmenge von* $\mathbb{P}^n(K)$ *auffassen. Das Komplement ist die* ***unendlich ferne Hyperebene***

$$H = \{p \in \mathbb{P}^n(K) \mid p_0 = 0\}\,.$$

Vermöge der Abbildung φ *korrespondieren die Punkte der Parabel* $V(x_1^2 - x_2) \subset \mathbb{R}^2$ *zu Punkten der projektive Parabel* $V(x_1^2 - x_2 x_0) \subset \mathbb{P}^2(\mathbb{R})$. *Diese hat einen weiteren Punkt im Unendlichen, den Punkt*

$$(0 : 1 : 0) \in H\,.$$

Siehe dazu die Abbildung 4.7 der Parabel in $\mathbb{R}^2$ und Abbildung 4.7 der projektiven Parabel (wobei wir einen 1-dimensionalen Untervektorraum von $\mathbb{R}^3$ mit seinem Erzeuger auf der oberen Einheitshalbkugel identifizieren. Projizieren wir diese in die (x_1, x_2)-Ebene, können wir ein Bild der projektiven Parabel als Teilmenge des Einheitskreisscheibe zeichnen, siehe Abbildung 4.9. Hier sehen wir schon einen Vorteil der projektiven Geometrie: Alle Kegelschnitte (Parabeln, Hyperbeln und Ellipsen) sehen gleich aus. Ein anderer Vorteil (der viel allgemeiner eine Rolle spielt) ist, dass sich je zwei (verschiedene) Geraden in genau einem Punkt schneiden, auch parallele Geraden (Übung: Zeichnen Sie das Bild der Geraden $V(x_1)$ und $V(x_1 - 1)$ in $\mathbb{R}^2$ auf der Einheitskreisscheibe. Was ist der Schnittpunkt?).

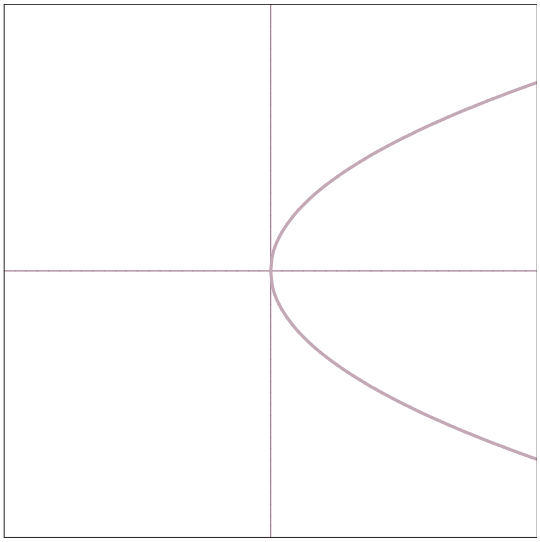


Abbildung 4.7: Parabel $x_1 - x_2^2 = 0$ in $\mathbb{R}^2$.

4.10 Zur Eindeutigkeit des Rests

Für $I = \langle G \rangle$ besagt die Eigenschaft

$$f \in I \Leftrightarrow \mathrm{NF}(f, G) = 0$$

einer Gröbnerbasis G bezüglich $>$, dass falls f in I liegt, jede Normalform $\mathrm{NF}(-, G)$ bezüglich $>$ den eindeutigen Rest 0 liefert.Für $f \notin I$ ist der Rest jedoch nicht eindeutig:

Beispiel 4.10.1 *Teilen wir $f = x^2y + x$ durch $G = \{y - 1, x^2 - 1\}$ mit Algorithmus 4.2 unter Verwendung von lp und bevorzugen $y - 1$ wann*

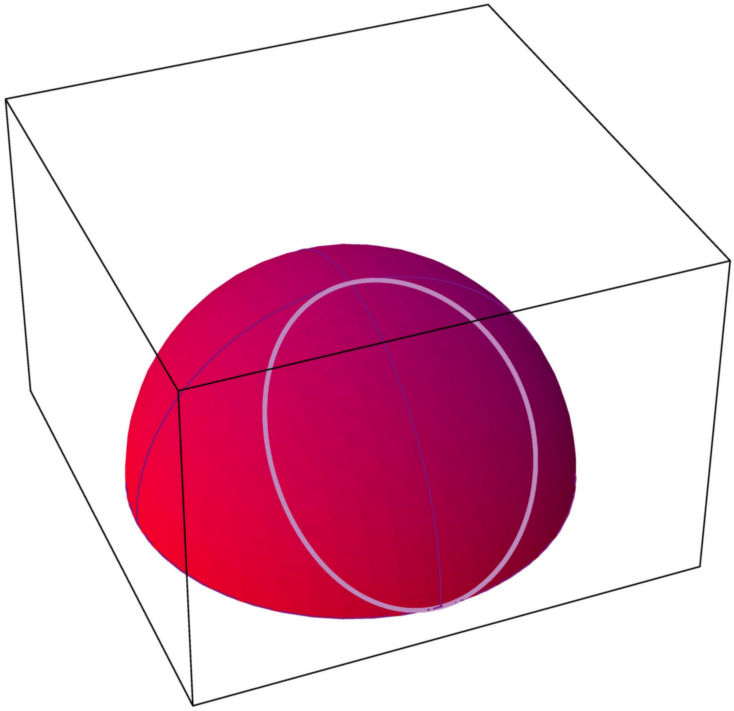


Abbildung 4.8: Projektive Parabel

möglich, so erhalten wir

$$\begin{array}{rl} \boldsymbol{x^2y} + x = & x^2\,(\boldsymbol{y} - 1) + 1 \cdot (\boldsymbol{x^2} - 1) + x + 1 \\ \underline{x^2y - x^2} & \\ \boldsymbol{x^2} + x & \\ \underline{x^2 - 1} & \\ \boldsymbol{x} + 1 & \end{array}$$

(siehe Beispiel 4.7.1). Bevorzugen wir dagegen $x^2 - 1$, liefert der Algorithmus

$$\begin{array}{rl} \boldsymbol{x^2y} + x = & y \cdot (\boldsymbol{x^2} - 1) + x + y \\ \underline{x^2y - y} & \\ \boldsymbol{x} + y & \end{array}$$

also einen anderen Rest.

Die ist ein Problem, wenn wir Reste verwenden wollen, um im Quotientenring R/I zu rechnen. Dazu gehen wir analog zum Fall

$$\mathbb{Z}/n\mathbb{Z} = \{\overline{a} = a + n\mathbb{Z} \mid a \in \mathbb{Z}\}$$

vor, wo wir etwa die Reste $0, \ldots, n-1$ bei ganzzahliger Division nach n als Repräsentanten der Klassen wählen können (siehe Lemma 2.1.5). Eine solche Arithmetik erfordert aber die Eindeutigkeit des Rests, da wir sonst nicht Gleichheit entscheiden können. Zu der algorithmischen

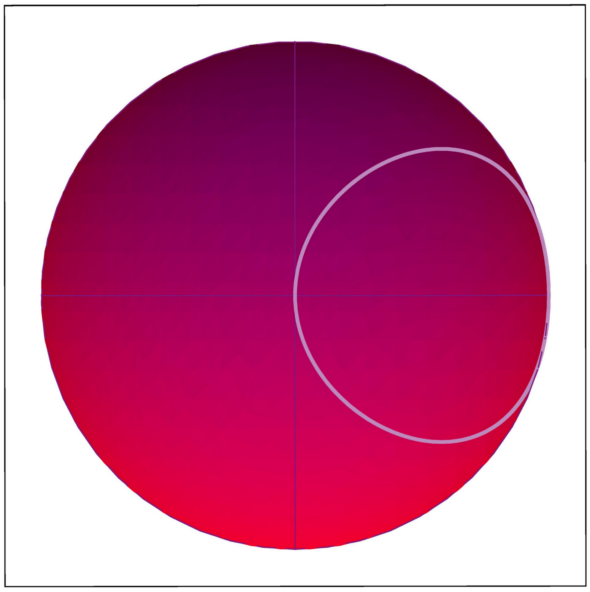


Abbildung 4.9: Projektive Parabel auf der Einheitsscheibe.

Beschreibung von R/I mit Hilfe von Gröbnerbasen und Division mit Rest siehe auch die Übungen 4.14 und 4.15.

Um einen eindeutigen Rest bei der multivariaten Polynomdivision zu erhalten, geht man wie folgt vor:

Definition 4.10.2 *Ein Polynom $f \in R$ heißt* ***reduziert*** *bezüglich einer Teilmenge $G \subset R$, wenn kein Term von f in $L(G)$ liegt.*

Eine Normalform $\mathrm{NF}(-, G)$ heißt ***reduzierte Normalform****, wenn $\mathrm{NF}(f, G)$ bezüglich G reduziert ist für alle $f \in R$.*

Algorithmus 4.4 liefert eine reduzierte Normalform, die **reduzierte Buchberger Normalform**.

Beispiel 4.10.3 *Verschieben wir also auch Terme in Zwischenschritten in den Rest, dann können wir die Division mit Bevorzugung von $x^2 - 1$ in Beispiel 4.10.1 fortführen*

$$
\begin{array}{ll}
\boldsymbol{x^2y} + x = & y \cdot (\boldsymbol{x^2} - 1) + \boldsymbol{x} + 1 \cdot (\boldsymbol{y} - 1) + 1 \\
\hline
x^2y - y & \\
\hline
\boldsymbol{x} + y & \\
\hline
\boldsymbol{y} & \\
y - 1 & \\
\hline
1 &
\end{array}
$$

was wieder zu dem Rest $x + 1$ führt. Tatsächlich ist der Rest jetzt eindeutig, falls wir nach einer Gröbnerbasis teilen:

Algorithmus 4.4 Reduzierte Division mit Rest

Input: $f \in R$, $g_1, ..., g_s \in R$, $>$ eine globale Ordnung auf den Monomen von R.

Output: Einen Ausdruck

$$f = q + r = \sum_{i=1}^{s} a_i g_i + r$$

sodass kein Term von r durch ein $L(g_i)$ teilbar ist.

1: $q = 0$
2: $r = 0$
3: $h = f$
4: **while** $h \neq 0$ **do**
5: **if** $L(g_i) \mid L(h)$ für ein i **then**
6: *Kürze den Leitterm von h:*
7: $a = \frac{LT(h)}{LT(g_i)}$
8: $q = q + a \cdot g_i$
9: $h = h - a \cdot g_i$
10: **else**
11: *Verschiebe den Leitterm in den Rest:*
12: $r = r + LT(h)$
13: $h = h - LT(h)$

Satz 4.10.4 *Sei $>$ eine globale Ordnung, $I \subset R$ ein Ideal, und G eine Gröbnerbasis von I. Ist $\mathrm{NF}(-, G)$ eine reduzierte Normalform, dann ist die Abbildung $\mathrm{NF}(-, G) : R \to R$ eindeutig bestimmt durch $>$ und I. Schreibe für diese $\mathrm{NF}(-, I)$.*

Beweis. Sei $G = \{g_1, ..., g_s\}$ und

$$\begin{aligned} f &= \sum_{i=1}^{s} a_i g_i + r \\ &= \sum_{i=1}^{s} a'_i g_i + r' \end{aligned}$$

Dann ist

$$r - r' = \sum_{i=1}^{s} (a_i - a'_i) g_i \in \langle G \rangle = I$$

(nach Corollar 4.7.9). Falls $r - r' \neq 0$, dann $L(r - r') \in L(I) = L(G)$. Da $L(r - r')$ ein Monom von r oder r' ist, wäre also r oder r' nicht reduziert bezüglich G. ■

Bemerkung 4.10.5 *Tatsächlich hängt $\mathrm{NF}(f, I)$ nur von der Klasse $\overline{f} \in R/I$ und von $>$ ab, wie eine leichte Modifikation des obigen Beweises zeigt, siehe Übung 4.15.*

Bemerkung 4.10.6 *Man beachte: Eine beliebige Normalform $\mathrm{NF}(-, G)$ liefert auch für jedes $f \in R$ ein eindeutiges Ergebnis $\mathrm{NF}(f, G)$, denn $\mathrm{NF}(-, G) : R \to R$ ist ja eine Abbildung. Allerdings kann für $\overline{f} = \overline{g} \in$*

R/I dann $\mathrm{NF}(f,G) \neq \mathrm{NF}(g,G)$ sein: Wie in Beispiel 4.10.1 berechnet liefert die Division von $f = x^2y + x$ durch $G = \{y-1, x^2-1\}$ mit Präferenz für $y-1$ den Rest $x+1$. Dagegen berechnet sie für $g = x+y$ den Rest $x+y$. Man beachte, dass G ein Gröbnerbasis von $I = \langle G\rangle$ ist (siehe Beispiel 4.7.10). Wegen

$$x^2y + x = y \cdot \left(x^2 - 1\right) + x + y$$

sind $\overline{f} = \overline{g} \in R/I$.

Beispiel 4.10.7 *In* SINGULAR *können wir die reduzierte Buchberger Normalform in Beispiel 4.10.3 bestimmen durch:*

```
ring R=0,(x,y),lp;
ideal I = x^2-1,y-1;
```

Wir überprüfen zunächst nochmals, dass die Erzeuger von I eine Gröbnerbasis bilden:

```
I=groebner(I);
I;
I[1]=x-1
I[2]=x2-1
reduce(x^2*y+x,I);
x+1
```

Bemerkung 4.10.8 *Wenn wir eine reduzierte Normalform verwenden und nach einer Gröbnerbasis teilen, ist in der Darstellung $f = \sum_{i=1}^{s} a_i g_i + r$ zwar der Rest r eindeutig, die a_i im Allgemeinen jedoch nicht. In Beispiel 4.10.1 und 4.10.3 haben wir*

$$\boldsymbol{x^2y} + x = y \cdot \left(\boldsymbol{x^2} - 1\right) + 1 \cdot (\boldsymbol{y} - 1) + x + 1$$

bzw.

$$\boldsymbol{x^2y} + x = x^2 \left(\boldsymbol{y} - 1\right) + 1 \cdot \left(\boldsymbol{x^2} - 1\right) + x + 1$$

erhalten.

Teilen wir f durch $G = \{g_1, \ldots, g_s\}$, können wir eine eindeutige Darstellung

$$f = \textstyle\sum_{i=1}^{s} a_i g_i + r$$

erreichen, indem wir fordern, dass kein Term von $a_i L(g_i)$ durch kein $L(g_j)$ mit $j < i$ teilbar ist.

In dem Beispiel würden wir die erste Darstellung dann für $G = \{x^2-1, y-1\}$ erhalten und die zweite für $G = \{x^2-1, y-1\}$.

4.11 Zur Eindeutigkeit von Gröbnerbasen

In Analogie zur reduzierten Zeilenstufenform wollen wir die Eindeutigkeit von Gröbnerbasen untersuchen.

Beispiel 4.11.1 *Wir wenden den Buchbergeralgorithmus auf die Erzeuger von*

$$I = \left\langle \boldsymbol{t^2} - x, \boldsymbol{t^3} - y, \boldsymbol{t^4} - z \right\rangle \subset K\,[t, z, y, x]$$

für die lexikographische Ordnung $t > z > y > x$ an. In jedem Schritt steht die erste Spalte für das S-Polynom und die zweiten Spalte für die Division mit Rest.

	↱	$\boldsymbol{-tx} + y$	↱	$\boldsymbol{-t^2x} + z$	↱	$\boldsymbol{-ty} + z$	↱	$\boldsymbol{t^3y} - zx$
$\boldsymbol{t^2} - x$	t		t^2	x				$-ty$
$\boldsymbol{t^3} - y$	-1				t			
$\boldsymbol{t^4} - z$			-1		-1		x	
$\boldsymbol{tx} - y$		1					$-t^3$	$-y$
$\boldsymbol{z} - x^2$				-1		-1		x
$\boldsymbol{ty} - x^2$						1		
$\boldsymbol{y^2} - x^3$								-1

Führen Sie als Übung die Divisionen mit Rest durch und prüfen Sie dass die restlichen S-Polynome zu 0 reduzieren. Eine Gröbnerbasis ist damit gegeben durch

$$G = \{\boldsymbol{t^2} - x, \boldsymbol{t^3} - y, \boldsymbol{t^4} - z, \boldsymbol{tx} - y, \boldsymbol{ty} - x^2, \boldsymbol{z} - x^2, \boldsymbol{y^2} - x^3\}.$$

Bemerkung 4.11.2 *Nach Satz 4.8.1 gilt in dem Beispiel somit*

$$\left\langle t^2 - x, t^3 - y, t^4 - z \right\rangle \cap K[x, y, z] = \left\langle z - x^2,\ y^2 - x^3 \right\rangle.$$

Wir haben also für die durch die ***Parametrisierung***

$$\varphi : K \to K^3,\ t \mapsto (t^2, t^3, t^4)$$

gegebene Kurve $C = \mathrm{Bild}(\varphi)$ ***implizite Gleichungen*** *gefunden. Abbildung 4.10 zeigt die beiden Flächen $V(z-x^2)$ und $V(y^2-x^3)$ und die Kurve C. Mit Methoden der algebraischen Geometrie kann man genau beschreiben, inwiefern jeder Punkt von $V(z-x^2,\ y^2-x^3)$ im Bild $\mathrm{Bild}(\varphi)$ der Parametrisierung liegt, und auch das wesentlich schwerere umgekehrte Problem lösen: Gegeben eine Kurve in impliziter Form $C = V(J)$, kann man entscheiden, ob sich C durch eine Parametrisierung beschreiben lässt, und falls ja diese bestimmen.*

Eine typische Anwendung ist die Bestimmung des Durchschnitts einer Fläche $S = V(f) \subset K^3$, $f \in K[x, y, z]$ mit einer parametrisch gegebenen Kurve $C = \mathrm{Bild}(\varphi)$. Dazu bestimmt man die Nullstellen von $f \circ \varphi \in K[t]$ und setzt diese in φ ein.

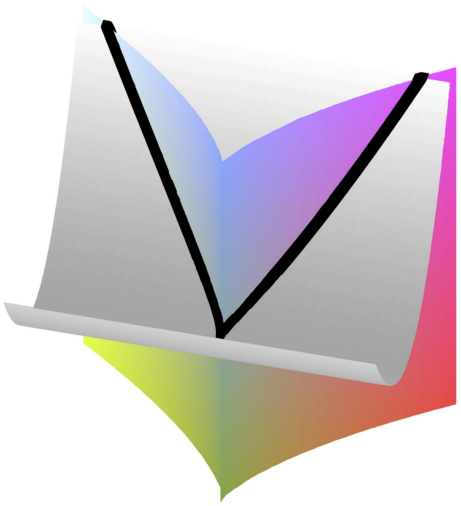


Abbildung 4.10: Kurve gegeben durch eine Parametrisierung bzw. implizite Gleichungen

Benötigen wir alle Polynome in der Gröbnerbasis G?

Definition 4.11.3 *Eine Gröbnerbasis $G = \{g_1, ..., g_s\}$ heißt **minimal**, wenn $L(g_i) \nmid L(g_j)$ für alle $i \neq j$.*

*Wenn außerdem $\mathrm{LC}(g_i) = 1$ und $\mathrm{tail}(g_i) = g_i - \mathrm{LT}(g_i)$ reduziert ist bezüglich G für alle i, dann heißt G **reduziert**.*

Bemerkung 4.11.4 *Aus einer Gröbnerbasis können wir durch Streichen von Elementen eine minimale Gröbnerbasis erhalten.*

Beweis. Sei $G = \{g_1, ..., g_s\}$ eine Gröbnerbasis von I, also $L(I) = \langle L(g_1), ..., L(g_s)\rangle$. Das monomiale Ideal $L(I)$ hat ein minimales Erzeugendensystem (siehe Corollar 4.6.6), das aus den minimalen Elementen von $\{L(g_1), ..., L(g_s)\}$ bezüglich Teilbarkeit besteht. ■

Satz 4.11.5 *Sei $>$ eine globale Ordnung. Jedes Ideal hat bezüglich $>$ eine (bis auf Permutation der Elemente) eindeutige reduzierte Gröbnerbasis.*

Beweis. Seien G und H reduzierte Gröbnerbasen von I. Wegen

$$L(G) = L(I) = L(H)$$

und da G und H minimal sind, gibt es für jedes $g \in G$ ein $h \in H$ mit $L(g) = L(h)$. Dann ist

$$s = g - h = \mathrm{tail}(g) - \mathrm{tail}(h)$$

und, da kein Term von $\mathrm{tail}(g)$ oder $\mathrm{tail}(h)$ durch einen Leitterm teilbar ist, haben wir

$$s = \mathrm{NF}(s, G).$$

Da $s \in I$ ist $\mathrm{NF}(s, G) = 0$. ■

Bemerkung 4.11.6 *Ist* $>$ *global,* NF *eine reduzierte Normalform und* $G = \{g_1, ..., g_s\}$ *eine minimale Gröbnerbasis, dann ist* $H = \{h_1, ..., h_s\}$ *mit*

$$h_i = \mathrm{NF}(g_i, \{g_1, ..., g_{i-1}, g_{i+1}, ..., g_s\})$$

die reduzierte Gröbnerbasis von I*.*

Beweis. Ist G minimal, dann ist $L(g_i)$ durch kein $L(g_j)$ mit $i \neq j$ teilbar, also $L(g_i) = L(h_i)$. Nach Konstruktion ist $\mathrm{tail}(h_i)$ reduziert bezüglich h_j für $j \neq i$. Außerdem ist kein Term von $\mathrm{tail}(h_i)$ durch $L(h_i)$ teilbar, da nach Definition und Satz 4.6.3 die globale Ordnung $>$ Teilbarkeit verfeinert. ■

Beispiel 4.11.7 *In Beispiel 4.11.1 ist eine minimale Gröbnerbasis*

$$G = \{\boldsymbol{t}^2 - x, \boldsymbol{tx} - y, \boldsymbol{ty} - x^2, \boldsymbol{z} - x^2, \boldsymbol{y}^2 - x^3\}.$$

SINGULAR *liefert stets eine minimale Gröbnerbasis. Für Beispiel 4.11.1:*

```
ring R=0,(t,z,y,x),lp;
ideal I = t^2-x,t^3-y,t^4-z;
groebner(I);
_[1]=y2-x3
_[2]=z-x2
_[3]=tx-y
_[4]=ty-x2
_[5]=t2-x
```

Hier ist das Ergebnis von `groebner` *schon reduziert, im Allgemeinen jedoch nicht.*

Beispiel 4.11.8 *Um in* SINGULAR *die reduzierte Gröbnerbasis zu berechnen, müssen wir die Option* `redSB` *setzen. Für ein lineares Gleichungssystem erhalten wir dann die reduzierte Zeilenstufenform:*

```
ring R = 0,(x(1..4)),lp;
LIB random.lib";
ideal I = randomid(maxideal(1),3,101);
groebner(I);
option(redSB);
std(I);
```

4.12 Buchbergerkriterium

4.12.1 Idee

Um zu zeigen, dass der Buchbergeralgorithmus eine Gröbnerbasis berechnet, müssen wir zeigen, dass für $I = \langle G \rangle$ mit $G = \{g_1, \ldots, g_t\} \subset$

$R = K[x_1, \ldots, x_n]$ der Buchbergertest $\mathrm{NF}(\mathrm{spoly}(g,h), G) = 0$ für alle $g, h \in G$ impliziert, dass G eine Gröbnerbasis ist.

Wir müssen zeigen, dass für $f \in I$ das Leitmonom $L(f) \in L(G)$ ist. Nach Voraussetzung gibt es $a_i \in R$ mit

$$f = \sum_{i=1}^{t} a_i g_i.$$

Offenbar ist $L(f) \leq \max_i L(a_i g_i)$. Falls $L(f) < \max_i L(a_i g_i)$, dann müssen Leitterme von Summanden $\sum_{i=1}^{t} a_i g_i$ sich kürzen, etwa $L(a_{i_1} g_{i_1})$ und $L(a_{i_2} g_{i_2})$. Nach Annahme haben wir einen Standardausdruck

$$0 = \mathrm{spoly}(g_{i_1}, g_{i_2}) - \sum_{i=1}^{t} c_i g_i$$

Eine solche Relation zwischen den g_i bezeichnet man als **Syzygie**. Subtrahieren wir ein geeignetes Vielfaches dieser Gleichung von der Gleichung $f = \sum_{i=1}^{t} a_i g_i$ erhalten wir eine neue Darstellung von f mit kleinerem $\max_i L(a_i g_i)$. Wir werden sehen, dass nach endlich vielen Schritten $L(f) = \max_i L(a_i g_i)$, also $L(g_i) \mid L(f)$ für ein i, also $L(f) \in L(G)$.

Mit diesem Argument erhalten wir dann die folgende Charakterisierung der Gröbnerbaseneigenschaft:

Satz 4.12.1 (Buchbergerkriterium) *Sei $>$ eine globale Monomordnung auf R,* NF *eine Normalform, $I \subset R$ ein Ideal und $G \subset R$ endlich mit $0 \notin G$. Dann sind äquvialent:*

Lemma 4.12.2 *1) $L(G) = L(I)$, d.h. G ist eine Gröbnerbasis von I.*

2) $\mathrm{NF}(f, G) = 0$ für alle $f \in I$.

3) $I = \langle G \rangle$ und $\mathrm{NF}(\mathrm{spoly}(g,h), G) = 0$ für alle $g, h \in G$.

Die Äquivalenz (1) $\Leftrightarrow$ (2) haben wir bereits in Proposition 4.7.12 gesehen. Die Implikation (2) $\Rightarrow$ (3) ist klar, da $\mathrm{spoly}(g,h) \in I$ und eine Gröbnerbasis von I ein Erzeugendensystem ist (Corollar 4.7.9).

Um die Darstellung von f mittels Syzygien systematisch auf einen Standardausdruck zu reduzieren, verwenden wir wieder das Konzept von Gröbnerbasen, allerdings betrachten wir Vektoren (a_i) mit polynomialen Einträgen statt nur Polynome. Wir müssen also den Gröbnerbasenbegriff von Idealen in R auf Untermoduln von R^t verallgemeinern.

4.12.2 Buchbergeralgorithmus für Untermoduln

Definition 4.12.3 *Sei R ein kommutativer Ring mit 1. Ein R-**Modul** $(M, +, \cdot)$ ist eine Menge M mit Abbildungen*

$$+ : M \times M \longrightarrow M$$
$$\cdot : R \times M \longrightarrow M$$

sodass

1) $(M,+)$ *eine abelsche Gruppe ist,*

2) Die Skalarmultiplikation $\cdot$ *distributiv über die Addition* $+$ *ist, d.h.*

$$r \cdot (m_1 + m_2) = r \cdot m_1 + r \cdot m_2$$
$$(r_1 \overset{R}{+} r_2) \cdot m = r_1 \cdot m + r_2 \cdot m$$

für alle $r, r_1, r_2 \in R$ *und* $m, m_1, m_2 \in M$*,*

3) für alle $r, s \in R$ *und* $m \in M$

$$(r \overset{R}{\cdot} s) \cdot m = r \cdot (s \cdot m)$$

gilt, und

4) $1 \cdot m = m$*.*

Beispiel 4.12.4 *1) Sei* $R = K$ *ein Körper. Dann ist ein* K*-Modul nichts anderes als ein* K*-Vektorraum.*

2) Ein $\mathbb{Z}$*-Modul* G *ist nichts anderes als eine abelsche Gruppe* $(G,+)$*. Die Skalarmultiplikation ist*

$$\mathbb{Z} \times G \longrightarrow G$$
$$(n,g) \longmapsto n \cdot g := \underbrace{g + \ldots + g}_{n\text{-mal}}$$

mit $(-1) \cdot g := -g$*.*

3) Sei $(R,+,\cdot)$ *ein kommutativer Ring. Dann ist* $I \subset R$ *ein Ideal genau dann, wenn* $(I,+,\cdot)$ *ein* R*-Modul ist.*

4) Sind M_1 *und* M_2 *Moduln über* R*, dann ist auch das direkte Produkt* $M_1 \times M_2$ *ein* R*-Modul mit* $r \cdot (m_1, m_2) = (r \cdot m_1, r \cdot m_2)$*, insbesondere:*

5) Ist R *ein Ring, dann ist*

$$R^n = \underbrace{R \times \ldots \times R}_{n}$$

ein R*-Modul, der* ***freie Modul*** *von Rang* n*.*

6) Sei $(M,+,\cdot)$ *ein* R*-Modul. Ein* ***Untermodul*** $U \subset M$ *ist eine Untergruppe von* $(M,+)$*, auf die sich die Skalarmultiplikation einschränkt, d.h. mit*

$$r \cdot m \in U$$

für alle $m \in U$ *und* $r \in R$*. Ein Untermodul ist wieder ein* R*-Modul.*

Eine Teilmenge $U \subset M$ ist ein Untermodul genau dann, wenn $U \neq \emptyset$ und

$$m_1 + m_2 \in U$$
$$r \cdot m \in U$$

für alle $m_i, m \in U$ und $r \in R$.

7) *Sei M ein R-Modul und $U \subset M$ ein Untermodul. Dann ist die Quotientengruppe M/U wieder ein R-Modul (der **Quotientenmodul**) mit der Skalarmultiplikation*

$$r \cdot (m + U) = r \cdot m + U$$

für $r \in R$ und $m \in M$.

8) *Für eine Matrix $M \in R^{t \times s}$ sind **Kern**, **Bild** und **Cokern***

$$\ker(M) = \{v \in R^s \mid M \cdot v = 0\} \subset R^s$$
$$\mathrm{Bild}(M) = \{M \cdot v \mid v \in R^s\} \subset R^t$$
$$\mathrm{coker}(M) = R^t / \mathrm{Bild}(M)$$

R-Moduln.

9) *Für eine Teilmenge $G \subset R^t$ ist*

$$\langle G \rangle = \{\textstyle\sum_{endlich} r_i v_i \mid r_i \in R,\ v_i \in G\}$$

*ein R-Modul, der von G **erzeugte** Untermodul von R^t.*

Definition 4.12.5 *Ein R-**Modulhomomorphismus** ist ein R-linearer Gruppenhomomorphismus $f : M \to N$ zwischen R-Moduln, das heißt*

1) *$f(m_1 + m_2) = f(m_1) + f(m_2)$ für alle $m_1, m_2 \in M$*

2) *$f(r \cdot m) = r \cdot f(m)$ für alle $m \in M$ und $r \in R$.*

Es gilt der Homomorphiesatz für Moduln

$$M / \ker(f) \cong \mathrm{Bild}(f)$$

Sei im Folgenden sei wieder $R = K[x_1, ..., x_n]$. Um das Gröbnerbasenkonzept auf Untermoduln von R^t zu übertragen, müssen wir zunächst festlegen, was ein Monom ist. Schreibe dazu e_i für den i-ten Standardbasisvektor von R^t. Dann kann jedes Element von R^t geschrieben werden als eine Summe von **Termen** $c_\alpha x^\alpha e_i$, die wiederum K-Vielfache von **Monomen** $x^\alpha e_i$ sind. Zum Beispiel in $K[x, y]^2$

$$\begin{pmatrix} 2x + y^2 \\ x \end{pmatrix} = (2x + y^2)e_1 + xe_2$$
$$= 2xe_1 + y^2 e_1 + xe_2.$$

Definition 4.12.6 *Die Monome in R^t haben eine natürliche Ordnung durch* ***Teilbarkeit***

$$x^\alpha e_i \mid x^\beta e_j \iff i = j \text{ und } x^\alpha \mid x^\beta$$

Für zwei Terme $c_1 x^\alpha e_i$ und $c_2 x^\beta e_j$ mit $x^\alpha e_i \mid x^\beta e_j$ definieren wir

$$\frac{c_2 x^\beta e_j}{c_1 x^\alpha e_i} = \frac{c_2 x^\beta}{c_1 x^\alpha} = \frac{c_2}{c_1} x^{\beta-\alpha} \in R.$$

Definition 4.12.7 *Eine Monomordnung auf R^t ist eine*

1) *Totalordnung $>$ auf den Monomen von R^t, die*
2) *die Multiplikation respektiert, d.h.*

$$x^\alpha e_i > x^\beta e_j \Rightarrow x^\alpha x^\gamma e_i > x^\beta x^\gamma e_j$$

für alle $\alpha, \beta, \gamma, i, j$, und

3) *$x^\alpha e_i > x^\beta e_i \Leftrightarrow x^\alpha e_j > x^\beta e_j$ für alle α, β, i, j erfüllt.*

Bemerkung 4.12.8 *Wegen (3) ist durch $x^\alpha > x^\beta \Leftrightarrow x^\alpha e_i > x^\beta e_i$ eine eindeutige Monomordnung auf R gegeben (die wir wieder mit $>$ bezeichnen).*

Beispiel 4.12.9 *Gegeben eine Monomordnung $>$ auf R, kann man auf die folgenden zwei kanonischen Weisen eine Monomordnung auf R^t erhalten:*

1) *Priorität für die Monome $(>, c)$:*

   $x^\alpha e_i > x^\beta e_j :\iff x^\alpha > x^\beta$ *oder* $(x^\alpha = x^\beta$ *und* $i < j)$

2) *Priorität für die Komponenten $(c, >)$:*

   $x^\alpha e_i > x^\beta e_j :\iff i < j$ *oder* $(i = j$ *und* $x^\alpha > x^\beta)$

Beispiel 4.12.10 *In* SINGULAR *können wir den Monomen Priorität geben durch:*

```
ring R = 0,(x,y),(lp,C);
[1,0]>[0,1];
1
[x,0]>[y,0];
1
[0,x]>[y,0];
1
```

und den Komponenten Priorität geben durch

```
ring R = 0,(x,y),(C,lp);
[y,0]>[0,x];
1
```

Ersetzt man `C` *durch* `c`*, dann wird die Anordnung der Standardbasisvektoren umgekehrt.*

Bemerkung 4.12.11 *Es überträgt sich direkt von R auf R^t,*

1) *die Definition einer* ***globalen Ordnung*** *als Wohlordnung,*
2) *die Definition des* ***Leitterms***, ***Leitmonoms***, *und* ***Leitkoeffizienten***,
3) *der Begriff* ***Noethersch*** *in dem Sinne, dass jeder Untermodul von R^t endlich erzeugt ist (darauf kommen wir noch genauer zurück),*
4) *die Definition von* ***reduziert***, *und*
5) *die Division mit Rest und die reduzierte Division mit Rest.*

Teilen wir zum Beispiel in $K[x_1, x_2, x_3, x_4]^3$

$$f = \begin{pmatrix} 0 \\ -x_2x_4 \\ x_1x_3 \end{pmatrix} \text{ durch } g_1 = \begin{pmatrix} -x_2 \\ 0 \\ x_1 \end{pmatrix} \text{ und } g_2 = \begin{pmatrix} x_3 \\ -x_4 \\ 0 \end{pmatrix}$$

bezüglich (lp, c), so erhalten wir

$$\begin{aligned} f &= x_3 \cdot g_1 + \begin{pmatrix} x_2x_3 \\ -x_2x_4 \\ 0 \end{pmatrix} \\ &= x_3 \cdot g_1 + x_2 \cdot g_2 + 0. \end{aligned}$$

Definition 4.12.12 *Für eine Monomordnung $>$ und eine Teilmenge $G \subset R^t$, definieren wir den* ***Leitmodul*** *von G als den Untermodul*

$$L(G) = L_>(G) = \langle L(f) \mid f \in G \backslash \{0\} \rangle \subset R^t$$

erzeugt von den Leitmonomen. Der Begriff der ***Normalform*** *überträgt sich dann auch direkt von R auf R^t.*

Definition und Satz 4.12.13 *Sei $U \subset R^t$ ein Untermodul und $>$ eine globale Monomordnung auf R^t und* NF *eine Normalform. Eine endliche Teilmenge $0 \notin G \subset U$ heißt* ***Gröbnerbasis*** *von U bezüglich $>$, wenn die folgenden äquivalenten Bedingungen erfüllt sind*

1) $L(G) = L(U)$.
2) $\mathrm{NF}(f, G) = 0 \Leftrightarrow f \in I$.

Beweis. Identisch zum Beweis von Proposition 4.7.12. ∎

Bemerkung 4.12.14 *Der Buchbergeralgorithmus überträgt sich direkt von Idealen $I \subset R$ auf Untermoduln $U \subset R^t$. Wir werden im Folgenden zeigen, dass er für eine globale Ordnung $>$ nach endlich vielen Schritten mit einer Gröbnerbasis terminiert. Dies schließt dann insbesondere dann den Fall $t = 1$ von Idealen ein.*

4.12.3 Beweis des Buchbergerkritierums und Syzygien

Definition 4.12.15 *Für $g_1, ..., g_s \in R^t$ definiere*

$$\begin{array}{rccc} M_G: & R^s & \to & R^t \\ & e_i & \mapsto & g_i \end{array}$$

Wir können diese Abbildung als die Matrix $M_G = (g_1 \mid ... \mid g_s) \in R^{t\times s}$ darstellen.

*Der **Syzygienmodul** von G ist der Kern*

$$\mathrm{Syz}(G) = \ker(M_G) \subset R^s.$$

Bemerkung 4.12.16 *Jede **Syzygie***

$$v = \sum_{i=1}^{s} v_i e_i \in \mathrm{Syz}(G) \subset R^s$$

korrespondiert zu einer Relation

$$\sum_{i=1}^{s} v_i g_i = 0.$$

Mit Hilfe des Syzygienmoduls lässt sich z.B. der Durchschnitt von Idealen bestimmen:

Algorithmus 4.5 Durchschnitt von Idealen

Seien $I = \langle f_1, ..., f_a\rangle$ und $J = \langle g_1, ..., g_b\rangle$ Ideale in $R = K[x_1, ..., x_n]$. Ist

$$M_G = \begin{pmatrix} f_1 & ... & f_a & 0 & ... & 0 & 1 \\ 0 & ... & 0 & g_1 & ... & g_b & 1 \end{pmatrix} \in R^{2\times(a+b+1)}$$

und

$$\ker(M_G) = \mathrm{Bild}(S)$$

mit einer Matrix $S \in R^{(a+b+1)\times g}$, dann wird $I \cap J$ von den Einträgen der letzten Zeile von S erzeugt.

Beweis. Sei $f \in R$. Es gibt v_i mit

$$\begin{pmatrix} v_1 \\ \vdots \\ v_{a+b} \\ f \end{pmatrix} \in \ker(M_G)$$

genau dann, wenn es v_i gibt mit

$$-f = v_1 f_1 + ... + v_a f_a = v_{a+1} g_1 + ... + v_{a+b} g_b$$

genau dann wenn $f \in I \cap J$. ■

Bemerkung 4.12.17 *Für beliebige Ideale $I, J \subset K[x_1, ..., x_n]$ gilt*

$$V(I \cap J) = V(I) \cup V(J)$$
$$V(I + J) = V(I) \cap V(J)$$

(Übung), insbesondere sind Vereinigungen und Durchschnitte von algebraischen Mengen wieder algebraische Mengen. Im Gegensatz zu $I \cap J$ ist die Berechnung von $I + J$ trivial, denn

$$\langle f_1, ..., f_a \rangle + \langle g_1, ..., g_b \rangle = \langle f_1, ..., f_a, g_1, ..., g_b \rangle .$$

Beispiel 4.12.18 *In* SINGULAR *können wir den Syzygienmodul mit dem Kommando* `syz` *berechnen:*

```
ring R=0,(x,y,z),lp;
matrix M[2][5]= x,y-z^2,0,0,1,0,0,y,z,1;
print(syz(M));
0,  0,  -z,-y,
0,  -1,  0,  0,
-z,-1,  0,  -x,
y,  z,  -x,0,
0,  y-z^2,x*z,x*y
```

Das Beispiel zeigt, dass

$$\left\langle x, y - z^2 \right\rangle \cap \langle y, z \rangle = \left\langle y - z^2, xz, xy \right\rangle ,$$

geometrisch ist also

$$V(y - z^2, xz, xy) = V(x, y - z^2) \cup V(y, z)$$

die Vereinigung einer Parabel in der $z = 0$ Ebene und einer Geraden, siehe Abbildung 4.11.
Das Ergebnis dieser Rechnung erhält man direkt durch:

```
intersect(ideal(x,y-z^2), ideal(y,z));
_[1]=-y+z^2
_[2]=x*z
_[3]=x*y
```

Wie berechnet man den Syzygienmodul? Der Beweis des Buchbergertests liefert zugleich auch einen Algorithmus dazu:

Bemerkung 4.12.19 *Gegeben eine globale Ordnung $>$ und $G = \{g_1, ..., g_s\} \subset R$, gibt jeder erfolgreiche Buchbergertest*

$$\mathrm{NF}(\mathrm{spoly}(g_i, g_j), G) = 0$$

eine Relation

$$\underbrace{\frac{\mathrm{lcm}(L(g_i), L(g_j))}{\mathrm{LT}(g_i)}}_{m} \cdot g_i \quad - \quad \underbrace{\frac{\mathrm{lcm}(L(g_i), L(g_j))}{\mathrm{LT}(g_j)}}_{w} \cdot g_j \quad - \quad \sum_{k=1}^{s} a_k g_k \quad = \quad 0$$

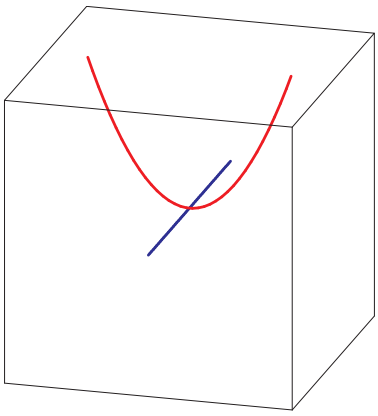


Abbildung 4.11: Vereinigung von algebraischen Mengen

mit Monomen $m, w \in R$ und damit eine Syzygie

$$\begin{array}{rcl} \mathrm{s}(g_i, g_j) & := & m \cdot e_i - w \cdot e_j - \sum_{k=1}^{s} a_k e_k \\ & \in & \mathrm{Syz}(G) \subset R^s \end{array}$$

Beispiel 4.12.20 *Wir führen den Buchbergertest durch für*

$$G = \{\boldsymbol{x_2^2} - x_1x_3,\ \boldsymbol{x_1x_2} - x_0x_3,\ \boldsymbol{x_1^2} - x_0x_2\}$$

und die Ordnung dp:

	↱	$\boldsymbol{-x_1^2x_3} + x_0x_2x_3$	↱	$\boldsymbol{x_0x_2^2} - x_0x_1x_3$	↱	$\boldsymbol{x_0x_2^3} - x_1^3x_3$
$\boldsymbol{x_2^2} - x_1x_3$	x_1			$-x_0$	x_1^2	$-x_0x_2$
$\boldsymbol{x_1x_2} - x_0x_3$	$-x_2$		x_1			
$\boldsymbol{x_1^2} - x_0x_2$		x_3	$-x_2$		$-x_2^2$	$+x_1x_3$

Somit erhalten wir Syzygien

$$\begin{pmatrix} x_1 \\ -x_2 \\ x_3 \end{pmatrix}, \begin{pmatrix} -x_0 \\ x_1 \\ -x_2 \end{pmatrix}, \begin{pmatrix} x_1^2 - x_0x_2 \\ 0 \\ -x_2^2 + x_1x_3 \end{pmatrix} \in \mathrm{Syz}(G)$$

Wir zeigen nun, dass die Syzygien $\mathrm{s}(g_i, g_j)$ schon $\mathrm{Syz}(G)$ erzeugen, tatsächlich bilden sie sogar eine Gröbnerbasis bezüglich einer passend gewählten Ordnung:

Definition 4.12.21 *Sei $G = \{g_1, ..., g_s\}$ eine Menge von Vektoren $g_i \in R^t$ und $>$ eine Monomordnung auf R^t. Dann ist durch*

$$\begin{array}{rcl} x^\alpha e_i >_G x^\beta e_j & :\Longleftrightarrow & L(x^\alpha g_i) > L(x^\beta g_j) \text{ oder} \\ & & (L(x^\alpha g_i) = L(x^\beta g_j) \text{ und } i < j) \end{array}$$

*eine Monomordnung $>_G$ auf R^s definiert, die sogenannte **Schreyer-Ordnung** induziert von $>$ und G.*

Bezüglich dieser Ordnung lässt sich der Leitterm einer Syzygie $\mathrm{s}(g_i, g_j)$ leicht bestimmen:

Lemma 4.12.22 *Wir verwenden die Notation von Bemerkung 4.12.19. Ist $i < j$ und $\mathrm{s}(g_i, g_j) \neq 0$, dann*

$$L(\mathrm{s}(g_i, g_j)) = \frac{\mathrm{lcm}(L(g_i), L(g_j))}{L(g_i)} e_i$$

wobei die linke Seite bezüglich $>_G$ und die rechte Seite bezüglich $>$ bestimmt wird.

Beweis. Da sich im S-Polynom $\mathrm{spoly}(g_i, g_j)$ die Leitterme kürzen gilt

$$L(m \cdot g_i) = L(w \cdot g_j),$$

also in der Schreyer-Ordnung

$$m \cdot e_i >_G w \cdot e_j$$

wegen $i < j$. Da NF eine Normalform ist, haben wir für alle k

$$L(a_k g_k) \leq L(\mathrm{spoly}(g_i, g_j)) < L(m \cdot g_i),$$

also in der Schreyer-Ordnung

$$m \cdot e_i >_G L(a_k e_k).$$

Insgesamt erhalten wir also

$$L(\mathrm{s}(g_i, g_j)) = m \cdot e_i.$$

■

Satz 4.12.23 (Buchbergerkriterium) *Sei $R = K[x_1, ..., x_n]$, $U = \langle g_1, ..., g_s \rangle \subset R^t$ ein Untermodul, $>$ eine globale Ordnung,* NF *eine Normalform.*

Ist

$$\mathrm{NF}(\mathrm{spoly}(g_i, g_j), G) = 0$$

für alle $i < j$, dann

1) *ist $G = \{g_1, ..., g_s\}$ eine Gröbnerbasis von U bezüglich $>$, und*
2) *die $\mathrm{s}(g_i, g_j) \neq 0$ für $i < j$ bilden eine Gröbnerbasis G' von $\mathrm{Syz}(G)$ bezüglich $>_G$.*

Beispiel 4.12.24 *Mit dem Buchbergerkriterium können wir z.B. leicht verifizieren, dass* $G = \{\boldsymbol{x}^2 - 1, \boldsymbol{y} - 1\}$ *eine Gröbnerbasis von* $I = \langle G\rangle \subset K[x,y]$ *bezüglich lp ist: Für das S-Polynom*

$$s = y(\boldsymbol{x}^2 - 1) - x^2(\boldsymbol{y} - 1) = x^2 - y$$

gibt Division mit Rest $\mathrm{NF}(s, G) = 0$*:*

$$\begin{array}{ll} \boldsymbol{x}^2 - y = & 1\cdot(\boldsymbol{x}^2 - 1) - 1\cdot(\boldsymbol{y} - 1) + 0 \\ x^2 - 1 & \\ \hline \boldsymbol{-y} + 1 & \\ -y + 1 & \\ \hline 0 & \end{array}$$

Beispiel 4.12.25 *Nach Satz 4.12.23, zeigt die Rechnung in Beispiel 4.12.20, dass*

$$G = \{\boldsymbol{x}_2^2 - x_1x_3,\ \boldsymbol{x}_1\boldsymbol{x}_2 - x_0x_3,\ \boldsymbol{x}_1^2 - x_0x_2\}$$

eine Gröbnerbasis von $I = \langle G\rangle$ *bezüglich dp ist. Außerdem sehen wir, dass*

$$G' = \left(\begin{pmatrix} \boldsymbol{x}_1 \\ -x_2 \\ x_3 \end{pmatrix}, \begin{pmatrix} -x_0 \\ \boldsymbol{x}_1 \\ -x_2 \end{pmatrix}, \begin{pmatrix} \boldsymbol{x}_1^2 - x_0x_2 \\ 0 \\ -x_2^2 + x_1x_3 \end{pmatrix}\right)$$

eine Gröbnerbasis von $\mathrm{Syz}(G)$ *bezüglich* $>_G$ *ist. Zur Bestimmung der Leitterme für* $>_G$ *beachte man, dass bezüglich dp*

$$\begin{aligned} x_1\cdot x_2^2 &= x_2\cdot x_1x_2 > x_3\cdot x_1^2 \\ x_0\cdot x_2^2 &< x_1\cdot x_1x_2 = x_1^2\cdot x_2 \\ x_1^2\cdot x_2^2 &= x_2^2\cdot x_1^2. \end{aligned}$$

Der letzte Vektor kann gestrichen werden, da sein Leitmonom $x_1^2e_1$ *durch das Leitmonom* x_1e_1 *des ersten Vektors teilbar ist. Dann ist*

$$G' = \left(\begin{pmatrix} \boldsymbol{x}_1 \\ -x_2 \\ x_3 \end{pmatrix}, \begin{pmatrix} -x_0 \\ \boldsymbol{x}_1 \\ -x_2 \end{pmatrix}\right)$$

eine minimale (und auch reduzierte) Gröbnerbasis von $\mathrm{Syz}(G)$*.*

Beweis. Sei

$$f = \textstyle\sum_{i=1}^s a_ig_i \in U$$

ein beliebiges Element. Dann gilt für

$$g := \textstyle\sum_{i=1}^s a_ie_i \in R^s$$

dass

$$M_G(g) = f.$$

Mit einer reduzierten Normalform NF' auf R^s erhalten wir unter Verwendung von $>_G$ eine Darstellung

$$g = \sum_{i,j} g_{i,j} \cdot \mathrm{s}(g_i, g_j) + \underbrace{\mathrm{NF}'(g, G')}_{\sum_{k=1}^{s} r_k e_k}$$

mit $g_{i,j} \in R$. Dann gilt für alle k mit $r_k \neq 0$, dass

$$L(r_k e_k) \notin \langle L(\mathrm{s}(g_i, g_j)) \mid i, j \text{ mit } \mathrm{s}(g_i, g_j) \neq 0 \rangle .$$

Da nach Lemma 4.12.22

$$L(\mathrm{s}(g_k, g_j)) = \frac{\mathrm{lcm}(L(g_k), L(g_j))}{L(g_k)} e_k,$$

für $k < j$, folgt

$$\frac{\mathrm{lcm}(L(g_k), L(g_j))}{L(g_k)} \text{ teilt nicht } L(r_k) \tag{4.2}$$

für alle $k < j$ mit $\mathrm{s}(g_k, g_j) \neq 0$.

Wegen

$$g - \mathrm{NF}'(g, G') \in \langle G' \rangle \subset \mathrm{Syz}(G) = \ker(M_G)$$

erhalten wir

$$f = M_G(\mathrm{NF}'(g, G')) = \sum_{k=1}^{s} r_k g_k.$$

Wir zeigen, dass sich in dieser Summe keine Leitterme kürzen können: Gilt für $k < j$ mit $r_k, r_j \neq 0$ dass

$$L(r_k g_k) = L(r_j g_j),$$

dann ist $L(r_k g_k) = L(r_k) L(g_k) \in R^t$ teilbar durch $L(g_k)$ und $L(g_j)$, also auch durch $\mathrm{lcm}(L(g_k), L(g_j))$. Also

$$\frac{\mathrm{lcm}(L(g_k), L(g_j))}{L(g_k)} \text{ teilt } L(r_k),$$

ein Widerspruch zu Gleichung 4.2 (da $\mathrm{s}(g_k, g_j) \neq 0$ nach Annahme $L(r_k g_k) = L(r_j g_j)$).

1) Für $f \neq 0$ ist also $L(f) = L(r_k g_k)$ für ein k, und damit $L(f) \in L(G)$, also $L(U) = L(G)$.

2) Für $f = 0$ ist g eine beliebige Syzygie. Das obige Argument zeigt dann, dass $r_k = 0$ für alle k. Somit ist $\mathrm{NF}'(g, G') = 0$ für alle $g \in \mathrm{Syz}(G)$.

■

Beispiel 4.12.26 *Da eine Gröbnerbasis insbesondere ein Erzeugendensystem ist, gilt in Beispiel 4.12.25*

$$\ker\left(\boldsymbol{x}_2^2 - x_1x_3 \quad \boldsymbol{x}_1\boldsymbol{x}_2 - x_0x_3 \quad \boldsymbol{x}_1^2 - x_0x_2\right) = \mathrm{Bild}\left(\begin{array}{c|c} \boldsymbol{x}_1 & -x_0 \\ -x_2 & \boldsymbol{x}_1 \\ x_3 & -x_2 \end{array}\right)$$

Diesen Prozess kann man iterieren:

Bemerkung 4.12.27 *Gegeben ein Ideal $I = \langle G_0\rangle \subset R = K[x_1,...,x_n]$ (oder allgemeiner Untermodul $U \subset R^s$) mit einer Gröbnerbasis G_0, erhält man durch iteriertes Bestimmen von Gröbnerbasen G_i mit*

$$\langle G_{i+1}\rangle = \mathrm{Syz}(G_i)$$

*eine sogenannte **freie Auflösung***

$$0 \to R^{s_l} \overset{M_{G_{l-1}}}{\to} ... \overset{M_{G_1}}{\to} R^{s_1} \overset{M_{G_0}}{\to} I \to 0$$

mit $\mathrm{Bild}(M_{G_{i+1}}) = \ker(M_{G_i})$. Hilberts Syzygiensatz besagt, dass diese Iteration nach endlich vielen Schritten stoppt. Aus der freien Auflösung läßt sich zum Beispiel die Dimension $\dim(V(I))$ bestimmen, oder für $\dim(V(I)) = 0$ auch die Anzahl der Lösungen eines algebraischen Gleichungssystems.

4.12.4 Terminierung des Buchbergeralgorithmus für Moduln

Es bleibt noch zu zeigen, dass der Buchbergeralgorithmus auch für Untermoduln von R^s terminiert. Dazu übertragen wir den Begriff Noethersch und die Kettenbedingung auf Moduln.

Definition 4.12.28 *Sei M ein R-Modul.*

1) *M heißt **endlich erzeugt**, wenn es einen surjektiven R-Modulhomomorphismus*

$$\varphi : R^r \to M$$

gibt.

Mit $m_i = \varphi(e_i)$ ist φ surjektiv genau dann, wenn sich jedes $m \in M$ als

$$m = a_1m_1 + ... + a_rm_r$$

mit $a_i \in R$ schreiben lässt, d.h. $M = \langle m_1,...,m_r\rangle$.

2) *M heißt **frei** vom Rang r, wenn es einen Isomorphismus $\varphi : R^r \to M$ gibt, d.h.*

$$M \cong R^r$$

*Obige Darstellung $m = a_1m_1 + ... + a_rm_r$ ist dann eindeutig, und wir bezeichnen $m_1,...,m_r$ als eine **Basis** von M.*

*3) M heißt **endlich präsentiert**, wenn M endlich erzeugt ist und $\ker\varphi$ ebenfalls endlich erzeugt ist.*

Wie kann man einen Modul allgemein im Computer darstellen? Endlich präsentierte Moduln werden einfach durch eine Matrix gegeben. Dazu führen wir folgende (auch allgemein sehr nützliche) Kurzschreibweise ein:

Definition 4.12.29 *Eine Sequenz von R-Modulhomomorphismen*

$$\ldots \to M_i \overset{\varphi_i}{\to} M_{i+1} \overset{\varphi_{i+1}}{\to} M_{i+2} \to \ldots$$

*heißt **exakt**, wenn*

$$\operatorname{Bild}(\varphi_i) = \ker(\varphi_{i+1}) \quad \forall i$$

Bemerkung 4.12.30 *Ein Homomorphismus $\pi : N \to M$ ist also surjektiv, wenn die Sequenz*

$$N \overset{\pi}{\to} M \to 0$$

exakt ist, bzw. injektiv, wenn

$$0 \to N \overset{\pi}{\to} M$$

exakt ist.

Bemerkung 4.12.31 *Ein endlich erzeugter Modul M ist also eine exakte Sequenz*

$$R^n \overset{\pi}{\to} M \to 0$$

Mit der Inklusion des Kerns erhalten wir auch eine exakte Sequenz

$$0 \to \ker(\pi) \to R^n \overset{\pi}{\to} M \to 0$$

Somit ist ein endlich präsentierter Modul M eine exakte Sequenz

$$R^m \overset{A}{\to} R^n \overset{\pi}{\to} M \to 0$$

*mit $A \in R^{n\times m}$. Diese Matrix A heißt **Präsentationsmatrix** von M. Mit dem Homomorphiesatz gilt*

$$M \cong R^n / \ker(\pi) = R^n / \operatorname{Bild}(A).$$

Durch iteratives Berechnen des Kerns von A erhalten wir (analog zu Beispiel 4.12.27 für Untermoduln) eine freie Auflösung von M.

Tatsächlich lässt sich sogar schon jeder endlich erzeugte Modul über einem Noetherschen Ring (insbesondere also auch über einem Euklidischen Ring oder Hauptidealring) durch eine Präsentationsmatrix darstellen: Analog zum Resultat für Ideale kann man folgende Äquivalenz zeigen (Übung 4.25):

Definition und Satz 4.12.32 *Sei R ein kommutativer Ring mit 1. Ein R-Modul M heißt **Noethersch**, wenn er folgende äquivalente Bedingungen erfüllt:*

1) *Jede aufsteigende Kette von Untermoduln wird stationär.*
2) *Jeder Untermodul von M ist endlich erzeugt.*
3) *Jede Teilmenge von Untermoduln enthält ein maximales Element.*

Beispiel 4.12.33 *Vorsicht, nicht jeder Untermodul eines endlich erzeugten Moduls ist endlich erzeugt: Der Ring $R = K[x_1, x_2, ...]$ der Polynome in abzählbar vielen Variablen ist als R-Modul endlich erzeugt (von 1), der Untermodul*

$$\begin{aligned} M &= \{f \in R \mid f(0) = 0\} \\ &= \langle x_1, x_2, ... \rangle \end{aligned}$$

jedoch nicht, da jede endliche Menge von Polynomen nur endlich viele Variablen involviert.

Mit der Kettenbedingung zeigt man (Übung 4.26):

Lemma 4.12.34 *Sei*

$$0 \to U \to F \to M \to 0$$

eine exakte Sequenz von R-Moduln. Dann ist F Noethersch genau dann, wenn U und M Noethersch sind.

Daraus folgert man leicht mit Induktion (Übung 4.27):

Lemma 4.12.35 *Ist R ein Noetherscher Ring, dann ist R^n ein Noetherscher Modul.*

Da jeder endlich erzeugte Modul M von der Form $M \cong R^n/U$ mit einem Untermodul $U \subset R^n$ ist, folgt sofort:

Satz 4.12.36 *Endlich erzeugte Moduln über Noetherschen Ringen sind schon endlich präsentiert.*

Insbesondere können wir endlich erzeugte $\mathbb{Z}$-Moduln mittels einer Präsentationsmatrix beschreiben. Die endlich erzeugten $\mathbb{Z}$-Moduln sind genau die endlich erzeugten abelschen Gruppen, denn jede solche Gruppe G hat durch

$$\begin{array}{ccc} \mathbb{Z} \times G & \longrightarrow & G \\ (n, g) & \longmapsto & n \cdot g := \underbrace{g + \ldots + g}_{n} \end{array}$$

eine kanonische $\mathbb{Z}$-Modulstruktur.

Beispiel 4.12.37 *Für die abelsche Gruppe G mit der Präsentation als $\mathbb{Z}$-Modul*

$$\mathbb{Z}^3 \xrightarrow{D} \mathbb{Z}^4 \to G \to 0$$

gegeben durch die Matrix

$$D = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 4 \\ 0 & 0 & 0 \end{pmatrix}$$

gilt mit dem Homomorphiesatz

$$G \cong \mathbb{Z}/4 \times \mathbb{Z}/4 \times \mathbb{Z},$$

denn

$$G \cong \mathbb{Z}^4/\operatorname{Bild}(D) = \mathbb{Z}^4/\langle e_1, 4e_2, 4e_3\rangle$$

ist erzeugt von den Klassen der Einheitsbasisvektoren $e_1, \ldots, e_4$ mit den Relationen

$$1 \cdot e_1 = 0$$
$$4 \cdot e_2 = 0$$
$$4 \cdot e_3 = 0.$$

Die entscheidende Frage für durch Präsentationen gegebene abelsche Gruppen ist dann: Können wir mittes Basiswechsel eine Präsentation finden, aus der sich wie in Beispiel 4.12.37 der Isomorphietyp der Gruppe sofort ablesen lässt.

4.13 Algebraische Gleichungssysteme revisited

In Abschnitt 4.5 hatten wir ein Verfahren entwickelt zur Bestimmung von $V(I) \subset K^n$ für $I \subset K[x_1, ..., x_n]$ mit $|V(I)| < \infty$ und $K = \overline{K}$. Die wesentliche Idee war durch Elimination die Ideale $I \cap K[x_i] = \langle f_i \rangle$ zu bestimmen und dann zu testen, welche Punkte von $V(f_1) \times ... \times V(f_n)$ in $V(I)$ liegen. Mit unserem erweiterten Wissen über Gröbnerbasen können wir zeigen, dass tatsächlich eine einzige Gröbnerbasenrechnung ausreicht:

Satz 4.13.1 *Sei $K = \overline{K}$, $I \subset K[x_1, ..., x_n]$. Dann sind äquivalent:*

1) $|V(I)| < \infty$
2) *Ist G eine Gröbnerbasis von I, dann gibt es für jedes i ein $g \in G$ mit*

$$L(g) = x_i^{\alpha_i}$$

und $\alpha_i \geq 0$.

3) $\dim_K(K[x_1,...,x_n]/I) < \infty$.

Beweis. Für (1) ⇒ (2) hat man wie gehabt

$$f = \prod_{t\in\pi(V(I))}(x_i - t) \in I(V(I)) = \sqrt{I},$$

mit der Projektion

$$\pi : V(I) \to K,\ (a_1,...,a_n) \mapsto a_i$$

Somit ist $f^w \in I$ für ein $w \geq 1$, also $L(f^w) \in L(I)$.

(2) ⇔ (3) folgt da nach Übung 4.15

$$K[x_1,...,x_n]/I \cong {}_K\langle x^\alpha \mid x^\alpha \notin L(I)\rangle$$

als K-Vektorräume.

(3) ⇒ (1): Da $\dim_K(K[x_1,...,x_n]/I) < \infty$ sind die Klassen $\bar{x}_i^0, \bar{x}_i^1, \bar{x}_i^2, ...$ linear abhängig, also gibt es ein $0 \neq f_i \in K[x_i]$ mit $f_i \in I$. Wie gehabt gilt dann

$$V(I) \subset V(f_1) \times ... \times V(f_n).$$

■

Bemerkung 4.13.2 *Ist $V(I) \neq \varnothing$ endlich, dann enthält also die minimale Gröbnerbasis von I bezüglich lp Gleichungen der Form*

$$\begin{aligned} f_1 &= x_1^{\alpha_1} - g_1(x_1,...,x_n) \\ f_2 &= x_2^{\alpha_2} - g_2(x_2,...,x_n) \\ &\vdots \\ f_{n-1} &= x_{n-1}^{\alpha_{n-1}} - g_{n-1}(x_{n-1},x_n) \\ f_n &= g_n(x_n) \end{aligned}$$

mit $\alpha_i > 0$.

Um $V(I)$ zu berechnen, bestimmen wir dann $V(f_1,...,f_n)$ und testen welche Lösungen Nullstellen der restlichen Gröbnerbasenelemente sind.

Beispiel 4.13.3 *Wir betrachten wieder den Durchschnitt von zwei Ellipsen aus Abbildung 4.2.*

Für die lexikographische Ordnung mit $x > y$ erhalten wir:

```
ring R=0,(x,y),lp;
ideal I = 2*x^2-x*y+2*y^2-2, 2*x^2-3*x*y+3*y^2-2;
groebner(I);
_[1]=y3-y
_[2]=2xy-y2
_[3]=2x2-3xy+3y2-2
```

also

$$y = 0, 1, -1.$$

Lösen der dritten Gleichung liefert

$$V(I) \subset \left\{(1,0),(-1,0),(1,1),(\frac{1}{2},1),(-1,-1),(-\frac{1}{2},-1)\right\}.$$

Um $V(I)$ zu erhalten, streichen wir die Lösungen $(1,1),(-1,-1)$ die nicht $2xy - y^2 = 0$ erfüllen. Statt 12 Punkte wie in Beispiel 4.8.2, müssen wir also nur 6 Punkte testen, siehe Abbildung 4.12.

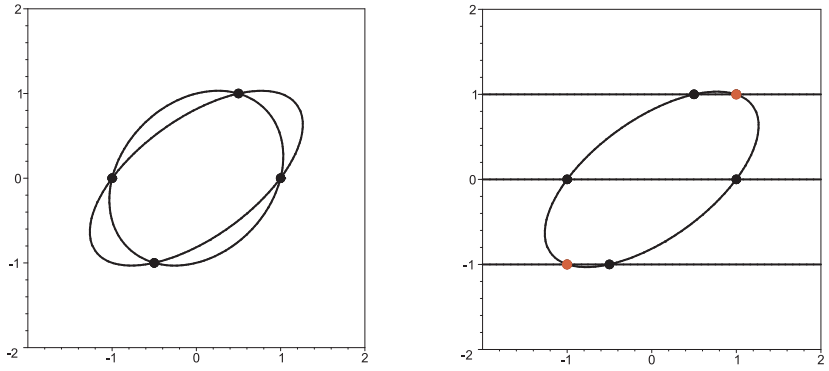


Abbildung 4.12: Buchberger-Algorithmus mit $x > y$ für den Schnitt von zwei Ellipsen

Wenden wir das Verfahren für $y > x$ an, erhalten wir sogar sofort die Lösungsmenge:

```
ring R=0,(y,x),lp;
ideal I = 2x2-xy+2y2-2, 2x2-3xy+3y2-2;
groebner(I);
_[1]=4x4-5x2+1
_[2]=3y+8x3-8x
```

Lösen von $4x^4 - 5x^2 + 1 = 0$ liefert

$$x = 1, -1, \frac{1}{2}, -\frac{1}{2}$$

und mit der zweiten Gleichung

$$V(I) = \left\{(1,0),(-1,0),(\frac{1}{2},1),(-\frac{1}{2},-1)\right\},$$

siehe Abbildung 1.7.

4.14 Gewichtsordnungen, Gröbnerfächer und tropische Varietäten

Definition 4.14.1 *Eine Monomordnung $>$ heißt **Gewichtsordnung**, falls es ein $w \in \mathbb{R}^n$ mit allen Einträgen $\neq 0$ gibt, sodass*

$$w \cdot \alpha > w \cdot \beta \Rightarrow x^\alpha > x^\beta.$$

Beispiel 4.14.2 *Die Grad-reverse-lexikographische Ordnung ist eine Gewichtsordnung mit* $w = (1, \dots, 1)$. *Allgemeiner kann man die* ***gewichtet-reverse-lexikographische Ordnung*** *definieren , indem man* $\deg x^\alpha$ *durch* $w \cdot \alpha$ *ersetzt. In* SINGULAR *erzeugt man diese Ordnung wie folgt (wobei wir den Gewichtsvektor* $w = (2,2,3)$ *verwenden):*

```
ring R=0,(x,y,z),wp(2,2,3);
poly f = x*z+y*z+z;
lead(f);
xz
poly f = x*z+y*z+z^2;
lead(f);
z2
```

Wir bemerken noch, dass sich tatsächlich jede Monomordnung auf einer *endlichen* Menge von Monomen durch eine ganzzahlige Gewichtung der Variablen beschreiben lässt. Für $v, w \in \mathbb{R}^n$ schreiben wir $v \cdot w = \sum_{i=1}^n v_i w_i$.

Proposition 4.14.3 *Gegeben eine Monomordnung* $>$ *und eine endliche Menge von Monomen* M *in* $x_1, ..., x_n$, *gibt es einen Gewichtsvektor* $w \in \mathbb{Z}^n$ *mit*

$$x^\alpha > x^\beta \iff w \cdot \alpha > w \cdot \beta$$

für alle $x^\alpha, x^\beta \in M$. *Man kann* w *so wählen, dass* $w_i > 0$ *falls* $x_i > 1$ *und* $w_i < 0$ *falls* $x_i < 1$.

Die Beweisidee ist folgende: Wir bilden die Menge

$$M_> = \left\{ \alpha - \beta \in \mathbb{Z}^n \mid x^\alpha, x^\beta \in M,\ x^\alpha > x^\beta \right\}.$$

Da 0 nicht in der konvexen Hülle von $M_>$ liegt, kann man zeigen, dass es ein w gibt mit $w \cdot \delta > 0$ für alle $\delta \in M_>$.

Beispiel 4.14.4 *Für die lexikographische Ordnung* $>$ *in den Variablen* x_1, x_2 *zeigt Abbildung 4.13 die Menge* $M_>$ *und ihre konvexe Hülle. Die Abbildung zeigt auch die Gerade* $w \cdot \delta = 0$, *wobei* w *ein Gewichtsvektor ist, der lp auf allen Monomen vom Grad* ≤ 4 *repräsentiert. Man beachte, dass für wachsenden Grad sich diese Gerade der Vertikalen durch* 0 *annähern muss.*

4.15 Übungen

Übung 4.1 *Sei* R *ein kommutativer Ring mit* 1 *und* $I \subset R$ *ein Ideal. Zeigen Sie:* $I = R$ *genau dann, wenn* $I \cap R^\times \neq \varnothing$.

Übung 4.2 *Sei* R *ein kommutativer Ring mit* 1. *Zeigen Sie, dass die folgenden Aussagen äquivalent sind:*

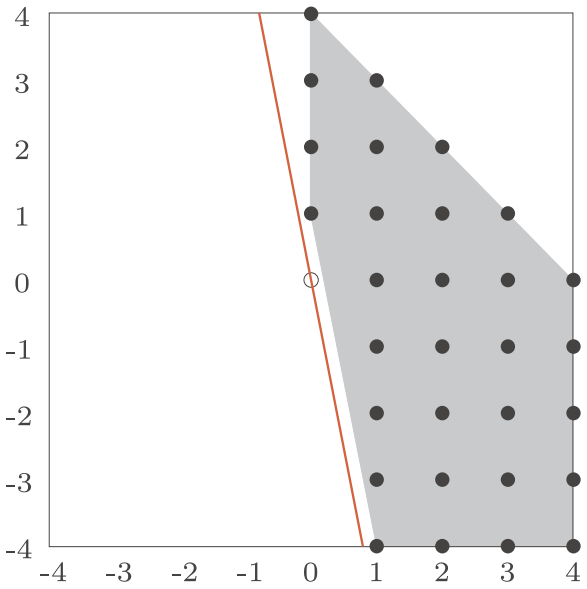


Abbildung 4.13: $M_>$ für die Ordnung *lp* und Monome vom Grad ≤ 4

1) R ist Noethersch, d.h. jedes Ideal $I \subset R$ ist endlich erzeugt.

2) Jede aufsteigende Kette

$$I_1 \subset I_2 \subset I_3 \subset \ldots \subset I_n \subset \ldots$$

von Idealen wird stationär, d.h. es gibt ein m, sodass

$$I_m = I_{m+1} = I_{m+2} = \ldots$$

gilt.

3) Jede nicht-leere Menge von Idealen besitzt bezüglich Inklusion ein maximales Element.

Übung 4.3 *Sei K ein Körper.*

1) Zeigen Sie: Das Ideal $\langle x, y\rangle \subset K[x, y]$ ist kein Hauptideal.

2) Ist $\mathbb{Z}[x]$ ein Hauptidealring?

Übung 4.4

$$\langle 3 + 4i, \quad -1 + 12i\rangle \subset \mathbb{Z}[i]\,.$$

Übung 4.5 *Sei $f \in \mathbb{R}[x]$ vom Grad $\deg(f) \leq 3$ mit*

$$\begin{array}{ll} f(1) = 2 & f'(1) = 3 \\ f(-1) = 1 & f'(-1) = 2 \end{array}$$

1) *Stellen Sie ein lineares Gleichungssystem zur Bestimmung von f auf.*

2) *Lösen Sie das lineare Gleichungssystem, indem Sie den Buchberger Algorithmus anwenden.*

3) *Erstellen Sie einen Plot des Graphen $\Gamma(f) \subset \mathbb{R}^2$.*

4) *Können Sie f auch mit Hilfe des Chinesischen Restsatzes finden?*

Übung 4.6 *Bestimmen Sie über $K = \mathbb{C}$ die Lösungsmenge*

1) *$V(I) \subset K$ für*

$$I = \left\langle x^5 + 2x^4 - x - 2,\ x^4 + x^2 - 2 \right\rangle \subset K[x]$$

2) *$V(J) \subset K^2$ für*

$$J = \left\langle x^5 + 2x^4 - x - 2,\ x^4 + x^2 - 2,\ x^2 + y^2 - 10,\ y^2 - 4y + 3 \right\rangle \subset K[x, y]$$

Übung 4.7 *Berechnen Sie den Rest $\mathrm{NF}(f, G)$ der Division mit Rest von*

$$f = x^5 + y^5 - 1 \in \mathbb{Q}[x, y]$$

nach

$$G = \left\{ x^2 + y^2 - 1,\ xy - x,\ y^2 - 1 \right\}$$

bezüglich der lexikographischen Ordnung mit $x > y$.

Übung 4.8 *Sei $I = \langle f_1, f_2, f_3 \rangle \subset \mathbb{R}[x, y, z]$ das Ideal erzeugt von*

$$f_1 = x^2 + y^2 - 1 \qquad f_2 = x^2 + z^2 - 1 \qquad f_3 = x + y + z.$$

Bestimmen Sie $V(I) \subset \mathbb{R}^3$.

Übung 4.9 *Sei $C = \{(t^2 - 1, t^3 - t) \in \mathbb{R}^2 \mid t \in \mathbb{R}\}$ die Kurve in Abbildung 4.14.*

1) *Berechnen Sie mit Hilfe von* Singular *eine Gröbnerbasis von*

$$I = \left\langle x - (t^2 - 1),\ y - (t^3 - t) \right\rangle \subset \mathbb{R}[t, x, y]$$

bezüglich der lexikographischen Ordnung mit $t > x > y$.

2) *Folgern Sie, dass $C \subset V(x^3 + x^2 - y^2)$.*

Hinweis: Betrachten Sie die Projektion $\pi : \mathbb{R}^3 \to \mathbb{R}^2$, $(t, x, y) \mapsto (x, y)$.

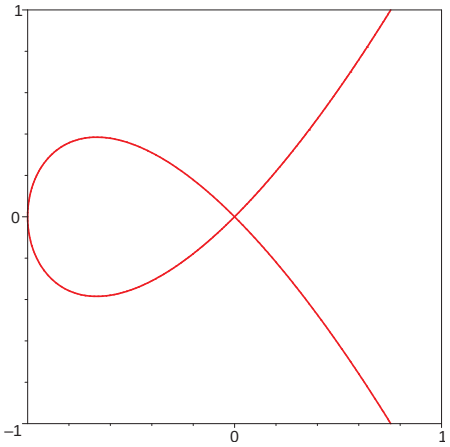


Abbildung 4.14: Nodale Kubik

Übung 4.10 *Sei $>$ eine Monomordnung in den Variablen $x_1, ..., x_n$. Zeigen Sie, dass die folgenden Bedingungen äquivalent sind:*

1) $>$ ist eine Wohlordnung

(d.h. jede nichtleere Menge von Monomen hat ein kleinstes Element).

2) $x_i > 1 \ \forall i$.

3) $x^\alpha > 1$ für alle $0 \neq \alpha \in \mathbb{N}_0^n$.

4) Falls $x^\beta \mid x^\alpha$ und $x^\alpha \neq x^\beta$ dann $x^\alpha > x^\beta$.

Hinweis: (4) $\Rightarrow$ (1) haben wir schon im Beweis von Definition und Satz 4.6.3 mit Hilfe von Dicksons Lemma gezeigt.

Übung 4.11 *1) Sortieren Sie alle Monome vom Grad ≤ 3 in x, y, z bezüglich dp.*

2) Berechnen Sie mit dem Buchbergeralgorithmus eine Gröbnerbasis von

$$I = \left\langle x^2 + yz + z^2,\ xy + y^2 + yz \right\rangle \subset \mathbb{Q}[x, y, z]$$

bezüglich dp. Überprüfen Sie Ihr Ergebnis mit Hilfe von SINGULAR.

Übung 4.12 *Berechnen Sie mit Hilfe des Buchbergeralgorithmus die reduzierte Gröbnerbasis von*

$$I = \left\langle t^2 - x,\ t^3 - y,\ t^4 - z \right\rangle \subset \mathbb{Q}[t, z, y, x]$$

bezüglich der lexikographischen Ordnung $t > z > y > x$.

Übung 4.13 *Sei*

$$I = \langle st - x,\ t - y,\ s^2 - z \rangle \subset \mathbb{Q}[s,t,z,y,x].$$

1) *Bestimmen Sie mit Hilfe des Buchbergeralgorithmus die reduzierte Gröbnerbasis von I bezüglich der lexikographischen Ordnung $t > s > z > y > x$. Überprüfen Sie Ihr Ergebnis mit* SINGULAR.

2) *Berechnen Sie $J = I \cap \mathbb{Q}[z,y,x]$.*

3) *Visualisieren Sie $V(J) \subset \mathbb{R}^3$ mit Hilfe von* SURFER.

Übung 4.14 *Sei*

$$I = \langle y + x^2 + 1,\ xy + y^2 + y \rangle \subset R = \mathbb{Q}[x,y]$$

und betrachten Sie die lexikographische Ordnung lp.

1) *Bestimmen Sie minimale Erzeuger von $L(I)$.*

2) *Zeigen Sie, dass $\overline{1}, \overline{x}, \overline{y}, \overline{y^2}$ eine Vektorraumbasis von R/I bilden*

3) *Bestimmen Sie die Multiplikationstabelle von R/I bezüglich dieser Basis.*

Übung 4.15 *1) Sei R ein kommutativer Ring mit 1 und $I \subset R$ ein Ideal. Zeigen Sie, dass auf $R/I = \{\overline{f} = f + I \mid f \in R\}$ die repräsentantenweise Addition und Multiplikation*

$$\overline{f} + \overline{g} = \overline{f+g} \qquad \overline{f} \cdot \overline{g} = \overline{f \cdot g}$$

wohldefiniert sind, und R/I damit zu einem kommutativen Ring mit 1 wird.

2) *Sei $R = K[x_1, ..., x_n]$, $I \subset R$ ein Ideal und* NF *eine reduzierte Normalform. Zeigen Sie, dass durch*

$$\begin{array}{rccc} \varphi: & R/I & \to & {}_K\langle x^\alpha \mid x^\alpha \notin L(I)\rangle \\ & \overline{f} & \mapsto & \mathrm{NF}(f,I) \end{array}$$

ein Isomorphismus von K-Vektorräumen gegeben ist. Können Sie auf ${}_K\langle x^\alpha \mid x^\alpha \notin L(I)\rangle$ eine Addition und Multiplikation definieren, sodass φ zu einem Isomorphismus von K-Algebren wird?

Übung 4.16 *Ein kommutativer Ring R mit 1 heißt Noethersch, wenn jedes Ideal $I \subset R$ endlich erzeugt ist. Zeigen Sie:*

1) *Ist R ein Noetherscher Ring, so ist auch $R[x]$ Noethersch.*

   Folgern Sie, dass der Polynomring $K[x_1, ..., x_n]$ über einem Körper K und $\mathbb{Z}[x_1, ..., x_n]$ Noethersch sind.

2) *Der Polynomring $K[x_1, x_2, ...]$ in unendlich vielen Variablen über einem Körper K ist nicht Noethersch.*

Hinweis: Verwenden Sie die Kettenbedingung aus Satz 4.2.1.

Übung 4.17 *Sei*

$$R = \{f \in \mathbb{Q}[x] \mid f(0) \in \mathbb{Z}\}$$

Zeigen Sie:

1) *R ist ein Ring.*

2) *Jedes von zwei Elementen erzeugte Ideal von R ist ein Hauptideal (d.h. von einem einzigen Element erzeugt).*

3) *Das Ideal*
$$I = \left\langle \frac{x}{2^n} \mid n \in \mathbb{N} \right\rangle \subset R$$
ist kein Hauptideal.

4) *Jedes endlich erzeugte Ideal von R ist ein Hauptideal.*

5) *R ist nicht Noethersch.*

Übung 4.18 *Sei $R = K[x, y]$ und $M = R^3$ und e_i der i-te Einheitsbasisvektor. Sortieren Sie die Monome*

$$\begin{aligned} &xy \cdot e_1,\ y^2 \cdot e_1,\ xy^2 \cdot e_2,\ e_2,\ x^2y \cdot e_3,\ x^2 \cdot e_3,\\ &xy^2 \cdot e_3,\ xy \cdot e_3,\ y \cdot e_1,\ x \cdot e_3,\ x \cdot e_1,\ e_1,\ y^2 \cdot e_2 \end{aligned}$$

bezüglich der Ordnung, die lp von R auf R^3 erweitert durch

1) *Priorität für die Monome, also (lp, c).*

2) *Priorität für die Komponenten, also (c, lp).*

Überprüfen Sie Ihr Ergebnis mit Hilfe von SINGULAR.

Übung 4.19 *Sei $R = K[x, y]$ und*

$$f = \begin{pmatrix} xy + y^2 \\ xy^2 - 1 \\ x^2y + x^2 + xy^2 + xy \end{pmatrix}, \quad g_1 = \begin{pmatrix} y \\ 0 \\ xy + x \end{pmatrix}, \quad g_2 = \begin{pmatrix} x + 1 \\ y^2 \\ 0 \end{pmatrix} \in R^3$$

Teilen Sie f durch g_1, g_2 bezüglich der Erweiterung der lexigraphischen Ordnung auf R^3 mit

1) *Priorität für die Monome von R,*

2) *Priorität für die Komponenten.*

Überprüfen Sie Ihre Ergebnisse mit Hilfe des SINGULAR *Kommandos* `reduce`.

Übung 4.20 *Sei* $R = K[x_0, x_1, x_2, x_3]$ *und* $>$ *die Ordnung dp.*

1) *Verifizieren Sie mit dem Buchbergerkriterium, dass* $G = \{x_0, x_1, x_2, x_3\}$ *eine Gröbnerbasis von*

$$I = \langle x_0, x_1, x_2, x_3 \rangle \subset R,$$

bezüglich $>$ *ist.*

2) *Bestimmen Sie mit Hilfe Ihrer Rechnung aus (1) eine Gröbnerbasis von* $\mathrm{Syz}(G)$ *bezüglich* $>_G$.

Übung 4.21 *Berechnen Sie in* $K[x, y]$ *jeweils den Durchschnitt* $I_1 \cap I_2$ *der Ideale*

1) $I_1 = \langle x, y \rangle$ *und* $I_2 = \langle x - 1, y - 1 \rangle$.

2) $I_1 = \langle x, y^3 \rangle$ *und* $I_2 = \langle x^2, y \rangle$.

Überprüfen Sie Ihre Ergebnisse mit Hilfe des SINGULAR *Befehls* `intersect`.

Übung 4.22 *Sei* $R = K[x_0, ..., x_4]$ *und* $>$ *die Ordnung lp und*

$$G = \{x_0x_1, x_1x_2, x_2x_3, x_3x_4, x_4x_0\}.$$

Berechnen Sie eine minimale Gröbnerbasis G' *von* $\mathrm{Syz}(G)$ *bezüglich* $>_G$.

Überprüfen Sie Ihr Ergebnis mit dem SINGULAR *Kommando* `syz`.

Übung 4.23 *Bestimmen Sie* $V(f_1, f_2, f_3) \subset \mathbb{R}^3$ *für*

$$\begin{aligned} f_1 &= x^2 + y^2 + z^2 - 9 \\ f_2 &= xy - z \\ f_3 &= x^2 + y^2 - z^2 - 1. \end{aligned}$$

siehe Abbildung 4.15.

Erstellen Sie jeweils einen Plot von $V(f_i)$.

Übung 4.24 *Sei* $K = \overline{K}$, *und* $I \subset K[x_1, ..., x_n]$ *ein Ideal mit* $|V(I)| < \infty$. *Zeigen Sie, dass*

$$|V(I)| \leq \dim_K \left(K[x_1, ..., x_n]/I \right).$$

Hinweis: Interpolation.

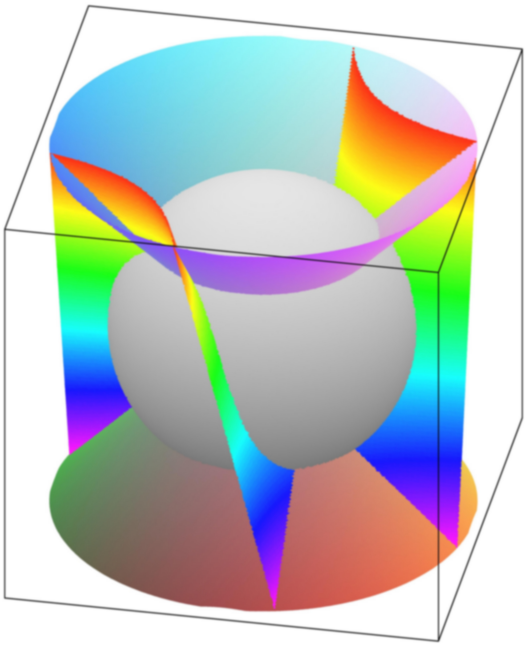


Abbildung 4.15: Durchschnitt von drei Kegelschnitten

Übung 4.25 *Ein R-Modul M heißt Noethersch, wenn er folgende äquivalente Bedingungen erfüllt:*

1) *Jede aufsteigende Kette von Untermoduln wird stationär.*
2) *Jeder Untermodul von M ist endlich erzeugt.*
3) *Jede Teilmenge von Untermoduln enthält ein maximales Element.*

Zeigen Sie die Äquivalenz.

Übung 4.26 *Sei*

$$0 \to M' \xrightarrow{i} M \xrightarrow{p} M'' \to 0$$

eine exakte Sequenz von R-Moduln. Zeigen Sie M ist Noethersch genau dann, wenn M' und M'' Noethersch sind.

Hinweis: Verwenden Sie die Kettenbedingung aus Übung 4.25.

Übung 4.27 *Folgern Sie aus Übung 4.26 mit Induktion nach n: Ist R ein Noetherscher Ring und $n \in \mathbb{N}$, dann ist R^n ein Noetherscher Modul.*

4.16 Praktische Aufgaben

Übung 4.28 *Sei K ein in* Julia/Nemo *verfügbarer Körper und $R = K[x_1, \ldots, x_n]$.*

1) *Schreiben Sie Funktionen die für $f \in R$ das Leitmonom $L(f)$, den Leitterm $LT(f)$ und den Leitkoeffizienten $LC(f)$ bestimmen bezüglich*

   (a) *der lexikographischen Ordnung lp, und*

   (b) *der Grad-reverse-lexikographischen Ordnung dp.*

2) *Implementieren Sie mit Hilfe der Funktionen aus (1) einen Divisionsalgorithmus, der für $f \in R$ und $G = \{g_1, \ldots, g_s\} \subset R$ einen Standardausdruck*

$$f = \sum_{i=1}^{s} a_i g_i + r$$

   bestimmt mit

$$a_i g_i \neq 0 \Rightarrow L(f) \geq L(a_i g_i)$$

   für alle i und

$$r \neq 0 \Rightarrow L(r) \notin L(G).$$

   Geben Sie sowohl die $a_1, \ldots, a_s$ als auch r zurück.

3) *Erproben Sie Ihre Implementierung an der Division von*

$$f = xy^2 + xyz - y^2z - yz^2 \in \mathbb{Q}[x, y, z]$$

   nach

$$G = \{x^2 + yz + z^2,\ xy + y^2 + yz\}$$

   bezüglich dp.

Übung 4.29 *Sei K ein in* Julia/Nemo *verfügbarer Körper, $R = K[x_1, \ldots, x_n]$ und $I = \langle f_1, \ldots, f_r \rangle \subset R$ ein Ideal.*

1) *Implementieren Sie die Berechnung einer Gröbnerbasis $G = \{g_1, \ldots, g_s\}$ von I bezüglich lp und dp mit Hilfe des Buchbergeralgorithmus.*

2) *Minimieren Sie das Ergebnis in dem Sinne, dass kein $L(g_i)$ ein $L(g_j)$ mit $i \neq j$ teilt. Ist das Resultat weiterhin eine Gröbnerbasis von I?*

3) *Erproben Sie Ihre Implementierung an dem Ideal*

$$I = \left\langle st - x,\ t - y,\ s^2 - z \right\rangle \subset \mathbb{Q}[t, s, z, y, x].$$

Übung 4.30 *Sei K ein in* Julia/Nemo *verfügbarer Körper, $R = K[x_1, \ldots, x_n]$ und $I = \langle f_1, \ldots, f_r \rangle \subset R$ ein Ideal und $1 \leq m \leq n - 1$.*

1) *Schreiben Sie eine Funktion, die Erzeuger von*

$$I \cap K[x_{m+1}, \ldots, x_n].$$

berechnet.

2) *Verwenden Sie Ihre Implementierung, um*

$$\langle st - x,\ t - y,\ s^2 - z \rangle \cap \mathbb{Q}[z, y, x]$$

zu bestimmen.

Übung 4.31 *Sei K ein in* Julia/Nemo *verfügbarer Körper und $R = K[x_1, \ldots, x_n]$, $G = \{g_1, \ldots, g_r\} \subset R$ mit $0 \notin G$ und $>$ die lexikographische Ordnung.*

1) *Modifizieren Sie Ihre Implementierung der Division mit Rest aus Aufgabe 4.28 so, dass sie eine reduzierte Normalform $\mathrm{NF}(-, G)$ bezüglich $>$ realisiert.*

2) *Sei G eine Gröbnerbasis von $I = \langle G \rangle$. Implementieren Sie die Ringstruktur des Quotientenrings R/I, d.h. Funktionen für Addition und Multiplikation, wobei Elemente vermöge des Isomorphismus*

$$\begin{array}{ccc} R/I & \to & {}_K\langle x^\alpha \mid x^\alpha \notin L(I) \rangle \\ \overline{f} & \mapsto & \mathrm{NF}(f, G) \end{array}$$

dargestellt werden.

3) *Bestimmen Sie in dieser Darstellung die Multiplikationstabelle von R/I für*

$$I = \langle x^2 + y^2 + z^2 - 9, \quad xy - z, \quad x^2 + y^2 - z^2 - 1 \rangle \subset \mathbb{Q}[x, y, z].$$

Übung 4.32 *1) Sei K ein in* Julia/Nemo *verfügbarer Körper und $R = K[x_1, \ldots, x_n]$ und $>$ die grad-reverse-lexikographische Ordnung. Sei G eine Gröbnerbasis von $I = \langle G \rangle$ bezüglich $>$. Implementieren Sie eine Funktion, die ein Erzeugendensystem des Syzygienmoduls*

$$\mathrm{Syz}(G)$$

bestimmt.

2) *Sei $I \subset \mathbb{Q}[x, y, z, w]$ das Ideal erzeugt von den 2×2-Minoren der Matrix*

$$A = \begin{pmatrix} x & y & z & w \\ y & z & w & x \end{pmatrix}$$

(a) *Verifizieren Sie, dass die Minoren eine Gröbnerbasis G von I bezüglich der grad-reverse-lexikographischen Ordnung bilden.*

(b) *Bestimmen Sie den Syzygienmodul $\mathrm{Syz}(G)$ dieser Gröbnerbasis.*

5

Lineare Algebra über $\mathbb{Z}$

5.1 Übersicht

In diesem Abschnitt wollen wir die Ideen der linearen Algebra von Körpern auf $\mathbb{Z}$ übertragen. Gegeben eine Matrix $A \in \mathbb{Z}^{n\times m}$, sind die entscheidenden Fragen:

1) Wie berechnet man den Kern $\ker(A)$?

2) Wie bestimmt man eine Basis des Spaltenraums $\mathrm{Bild}(A)$?

3) Wie vereinfacht man eine Präsentation

$$\mathbb{Z}^m \xrightarrow{A} \mathbb{Z}^n \to M \to 0$$

eines endlich präsentierten $\mathbb{Z}$-Moduls $M \cong \mathbb{Z}^n/\mathrm{Bild}(A) = \mathrm{coker}(A)$ so, dass man die Struktur von M (bis auf Isomorphie) wie in Beispiel 4.12.37 direkt ablesen kann? Dies erlaubt uns dann z.B. endlich erzeugte abelsche Gruppen zu klassifizieren.

Wir werden sehen, dass es einen Algorithmus gibt, der alle diese Fragen löst. Wir verallgemeinern dazu den Gauß-Algorithmus zur Normalformbestimmung über Körpern auf $\mathbb{Z}$ und allgemeiner Euklidische Ringe. Aus der linearen Algebra wissen wir: Ist K ein Körper und $A \in K^{n\times m}$, dann gibt es Basiswechsel gegeben durch invertierbare Matrizen $T \in \mathrm{GL}(m, K)$ und $S \in \mathrm{GL}(n, K)$, die A in die Normalform

$$S \cdot A \cdot T = D = \left(\begin{array}{ccc|c} 1 & & 0 & \\ & \ddots & & 0 \\ 0 & & 1 & \\ \hline & 0 & & 0 \end{array}\right)$$

bringen (mit $r = \mathrm{rang}(A)$ Einsen auf der Diagonalen). Ersetzen wir K durch einen Ring R können wir dies nicht mehr erwarten:

Bemerkung 5.1.1 *Sei R ein kommutativer Ring mit 1. Eine Matrix $A \in R^{n\times n}$ ist invertierbar (d.h. $A \in \mathrm{GL}(n,R)$) genau dann, wenn ihre Determinante eine Einheit ist, d.h.*

$$\det(A) \in R^\times.$$

Beweis. Ist $A \cdot A^{-1} = E$, dann $\det(A)\cdot\det(A^{-1}) = 1$. Umgekehrt: Falls $\det(A) \in R^\times$, dann ist

$$A^{-1} = \frac{A^{adj}}{\det(A)} \in R^{n\times n}$$

mit der adjungierten Matrix $A^{adj} = \left((-1)^{i+j}\det(A_{ji})\right)_{i,j} \in R^{n\times n}$, wobei A_{ji} durch Streichen der j-ten Zeile und i-ten Spalte von A entsteht. ■

Somit ist zum Beispiel $A = (2) \in \mathbb{Z}^{1\times 1}$ schon in Normalform, denn $\mathbb{Z}^\times = \{\pm 1\}$.

Im Allgemeinen können wir für eine Matrix $A \in R^{n\times m}$ über einem Euklidischen Ring R (sogar allgemeiner über einem Hauptidealring) mit Basiswechseln $T \in \mathrm{GL}(m,R)$ und $S \in \mathrm{GL}(n,R)$ erreichen, dass

$$S \cdot A \cdot T = D = \left(\begin{array}{ccc|c} d_1 & & 0 & \\ & \ddots & & 0 \\ 0 & & d_r & \\ \hline & 0 & & 0 \end{array}\right) \in R^{n\times m}$$

wobei jedes d_i ein Teiler von d_{i+1} ist. Im folgenden Abschnitt werden wir einen Algorithmus kennenlernen, der solche S, T und D bestimmt.

In jedem Fall erlaubt es uns eine derartige Normalform, eine Basis des Bilds

$$\mathrm{Bild}(A) = \{A \cdot x \mid x \in R^m\}$$

d.h. des von den Spalten von A erzeugten Gitters und des Kerns

$$\ker(A) = \{x \in R^m \mid A \cdot x = 0\}$$

d.h. der Lösungsmenge des homogenen linearen Gleichungssystems $A \cdot x = 0$ zu berechnen:

Bemerkung 5.1.2 *Gegeben $A \in R^{n\times m}$ ist $x \in R^m$ eine Lösung von*

$$A \cdot x = 0$$

genau dann, wenn $y = T^{-1} \cdot x$ eine Lösung von $D \cdot y = 0$ ist, denn

$$A \cdot x = 0 \Leftrightarrow S \cdot A \cdot x = 0 \Leftrightarrow S \cdot A \cdot T \cdot T^{-1} \cdot x = 0 \Leftrightarrow D \cdot y = 0.$$

Die Lösungen von $D \cdot y = 0$ können wir aber sofort hinschreiben

$$\ker(D) = \langle e_{r+1}, \ldots, e_m \rangle$$

also

$$\ker(A) = T \cdot \ker(D) = \langle Te_{r+1}, ..., Te_m \rangle .$$

Genauso erhalten wir wegen $A \cdot T = S^{-1} \cdot D$ *das Bild*

$$\text{Bild}(A) = \text{Bild}(A \cdot T) = \text{Bild}(S^{-1} \cdot D) = \left\langle S^{-1}d_1e_1, ..., S^{-1}d_re_r \right\rangle .$$

Diese Erzeuger von Kern und Bild bilden offenbar Basen: Sind $\lambda_i \in R$ *mit*

$$0 = \sum_{i=r+1}^{m} \lambda_i(Te_i) = T \sum_{i=r+1}^{m} \lambda_i e_i$$

also

$$0 = \sum_{i=r+1}^{m} \lambda_i e_i$$

also alle $\lambda_i = 0$. *Genauso für die Erzeuger des Bilds. Insbesondere sind also Kern und Bild einer Matrix über einem Hauptidealring freie Moduln.*

Beispiel 5.1.3 15 *Hühner legen jeden Tag ein Ei, und diese werden im halben Dutzend verkauft. An welchen Tagen bleibt kein Ei übrig? Wir müssen die Menge* L *aller*

$$\begin{pmatrix} x \\ y \end{pmatrix} \in \mathbb{Z}^2$$

mit

$$15x + 6y = 0$$

finden, also den Kern von

$$A = \begin{pmatrix} 15 & 6 \end{pmatrix} \in \mathbb{Z}^{1\times 2}.$$

Dazu transformieren wir A *durch Spaltentransformationen in eine Matrix* D*, für die wir den Kern direkt ablesen können. Wie in der linearen Algebra über Körpern erhalten wir eine Transformationsmatrix* $T \in \mathbb{Z}^{2\times 2}$ *mit*

$$A \cdot T = D$$

indem wir dieselben Tranformationen mit einer 2×2*-Einheitsmatrix durchführen:*

$$\begin{array}{ll} \begin{pmatrix} 15 & 6 \end{pmatrix} & \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\ \begin{pmatrix} 3 & 6 \end{pmatrix} & \begin{pmatrix} 1 & 0 \\ -2 & 1 \end{pmatrix} \\ \underbrace{\begin{pmatrix} 3 & 0 \end{pmatrix}}_{D} & \underbrace{\begin{pmatrix} 1 & -2 \\ -2 & 5 \end{pmatrix}}_{T} \end{array}$$

Der Kern von D ist offenbar der von dem zweiten Einheitsbasisvektor erzeugte $\mathbb{Z}$-Modul

$$\ker D = \left\langle\left(\begin{array}{c} 0 \\ 1 \end{array} \right)\right\rangle$$

und somit

$$\ker A = T \cdot \ker D = \left\langle\left(\begin{array}{c} -2 \\ 5 \end{array} \right)\right\rangle.$$

Nach zwei Tagen oder 5 verkauften Packungen bleibt also kein Ei übrig.

5.2 Der Elementarteiler-Algorithmus

Wir zeigen zunächst im Elementarteilersatz die Existenz dieser Normalform. Der Beweis gibt gleichzeitig einen rekursiven Algorithmus zur Bestimmung von S, T und D.

Satz 5.2.1 (Elementarteilersatz) *Sei R ein Euklidischer Ring und $A \in R^{n\times m}$ eine Matrix. Dann gibt es Basiswechsel $S \in \mathrm{GL}(n, R)$ und $T \in \mathrm{GL}(m, R)$ und ein $r \leq \min(n, m)$ mit*

$$S \cdot A \cdot T = D = \left(\begin{array}{ccc|c} d_1 & & & \\ & \ddots & & 0 \\ & & d_r & \\ \hline & 0 & & 0 \end{array} \right) \in R^{n\times m}$$

und

$$d_1 \mid d_2, \quad d_2 \mid d_3 \quad \ldots \quad d_{r-1} \mid d_r \neq 0$$

*Die d_i sind bis auf Multiplikation mit Einheiten durch A eindeutig bestimmt und heißen **Elementarteiler** von A, und D heißt die **Smith-Normalform** von A.*

Bemerkung 5.2.2 *Betrachten wir zunächst die Berechnung der Normalform im Spezialfall für $R = K$ ein Körper, d.h. den Gauß-Algorithmus (mit Zeilen- und Spaltenoperationen) in der linearen Algebra:*

Ist $A = (a_{i,j}) \in K^{n\times m}$, dann gibt es Basiswechsel $T \in \mathrm{GL}(m, K)$ und $S \in \mathrm{GL}(n, K)$ in Quelle und Ziel, die A in die Normalform

$$S \cdot A \cdot T = \left(\begin{array}{ccc|c} 1 & & & \\ & \ddots & & 0 \\ & & 1 & \\ \hline & 0 & & 0 \end{array} \right)$$

bringen mit $r = \mathrm{rang}(A)$ Einsen auf der Diagonalen.

Dabei erhalten wir S und T als Produkt von Zeilen- bzw. Spaltenoperationen: Durch Permutation von Zeilen und Spalten können wir $a_{11} \neq 0$ annehmen (falls $A \neq 0$). Subtraktion des $\frac{a_{1,j}}{a_{1,1}}$-fachen der ersten

Spalte von der j-ten Spalte (und analog für die Zeilen) bringt A in die Form

$$\left(\begin{array}{c|ccc} a_{1,1} & 0 & \cdots & 0 \\ \hline 0 & & & \\ \vdots & & * & \\ 0 & & & \end{array}\right)$$

Schließlich multiplizieren wir die erste Spalte mit $\frac{1}{a_{1,1}}$. Mit Induktion folgt die Behauptung.

In einem Ring R ist es dagegen im Allgemeinen nicht möglich, $\frac{a_{1,j}}{a_{1,1}}$ zu bilden, da $a_{1,1}$ keine Einheit sein muss. Die Idee für den Beweis von Satz 5.2.1 ist die Division $\frac{a_{1,j}}{a_{1,1}}$ durch Division mit Rest zu ersetzen. Der Beweis gibt einen Algorithmus zur Berechnung der Smith-Normalform, den **Elementarteiler-Algorithmus**.

Beweis. Sei R ein euklidischer Ring mit Norm d und $A \neq 0$.

1) Durch Zeilen- und Spaltenvertauschungen können wir annehmen, dass $a_{1,1} \neq 0$ und

$$d(a_{1,1}) \leq d(a_{i,j}) \text{ oder } a_{i,j} = 0$$

für alle $(i,j) \neq (1,1)$.

2) Ist ein Eintrag $a_{1,j}$ der ersten Zeile (analog für die erste Spalte) nicht durch $a_{1,1}$ teilbar, dann schreibe $a_{1,j}$ mit Division mit Rest

$$a_{1,j} = q \cdot a_{1,1} + r$$

mit $d(r) < d(a_{1,1})$. Der Fall $r = 0$ tritt nicht auf, da nach Voraussetzung $a_{1,1} \nmid a_{1,j}$.

Nach Subtraktion des q-fachen der 1-ten Spalte von der j-ten Spalte erreichen wir also

$$d(a_{1,1}) > d(a_{1,j})$$

Gehe nun zurück zu Schritt (1). Dieser Prozess terminiert, da $d(a_{1,1})$ in jedem Durchlauf echt kleiner wird.

3) Sind alle Einträge der ersten Zeile und Spalte durch $a_{1,1}$ teilbar, dann können wir durch Addition von Vielfachen der ersten Spalte A in die Form

$$\left(\begin{array}{c|ccc} a_{1,1} & 0 & \cdots & 0 \\ \hline * & & * & \end{array}\right)$$

und durch Addition von Vielfachen der ersten Zeile auf die Form

$$\left(\begin{array}{c|ccc} a_{1,1} & 0 & \cdots & 0 \\ \hline 0 & & & \\ \vdots & & A' & \\ 0 & & & \end{array}\right)$$

bringen. Hat A' einen Eintrag $a_{i,j}$, der nicht durch $a_{1,1}$ teilbar ist, dann addieren wir die i-te Zeile zu der ersten Zeile und gehen zurück zu (2). Danach wird $d(a_{1,1})$ wieder echt kleiner.

4) Sind alle Einträge von A durch $a_{1,1}$ teilbar, dann auch die Einträge von A', da sie R-Linearkombinationen von Einträgen von A sind.

   Mit Induktion nach $\min(n, m)$ folgt die Behauptung.

Für den Induktionsanfang ($n = 1$ oder $m = 1$) sind die Schritte (1) – (3) der euklidische Algorithmus auf den Einträgen.
Auf die Eindeutigkeit kommen wir gleich zurück. ■

Beispiel 5.2.3 *Wir bestimmen die Smith-Normalform von*

$$A = \begin{pmatrix} 6 & 9 & 6 \\ 6 & 6 & 7 \end{pmatrix} \in \mathbb{Z}^{2\times 3}$$

und gleichzeitig $S \in \mathbb{Z}^{2\times 2}$ und $T \in \mathbb{Z}^{3\times 3}$ durch simultanes Ausführen der Zeilen- bzw. Spaltenoperation auf der 2×2 bzw. 3×3 Einheitsmatrix. Division mit Rest (Schritt 2) gibt

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 6 & 3 & 6 \\ 6 & 0 & 7 \end{pmatrix} \quad \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Vertauschen (Schritt 1)

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 3 & 6 & 6 \\ 0 & 6 & 7 \end{pmatrix} \quad \begin{pmatrix} -1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Reduktion der ersten Zeile und Spalte (Schritt 3)

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 3 & 0 & 0 \\ 0 & 6 & 7 \end{pmatrix} \quad \begin{pmatrix} -1 & 3 & 2 \\ 1 & -2 & -2 \\ 0 & 0 & 1 \end{pmatrix}$$

Da 7 nicht durch 3 teilbar ist, addieren wir die zweite zur ersten Zeile

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 3 & 6 & 7 \\ 0 & 6 & 7 \end{pmatrix} \quad \begin{pmatrix} -1 & 3 & 2 \\ 1 & -2 & -2 \\ 0 & 0 & 1 \end{pmatrix}$$

Division mit Rest (Schritt 2)

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 3 & 6 & 1 \\ 0 & 6 & 7 \end{pmatrix} \quad \begin{pmatrix} -1 & 3 & 4 \\ 1 & -2 & -4 \\ 0 & 0 & 1 \end{pmatrix}$$

Vertauschen (Schritt 1)

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} 1 & 6 & 3 \\ 7 & 6 & 0 \end{pmatrix} \qquad \begin{pmatrix} 4 & 3 & -1 \\ -4 & -2 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

Reduktion der ersten Zeile und Spalte (Schritt 3 und 4)

$$\begin{pmatrix} 1 & 1 \\ -7 & -6 \end{pmatrix} \qquad \begin{pmatrix} 1 & 0 & 0 \\ 0 & -36 & -21 \end{pmatrix} \qquad \begin{pmatrix} 4 & -21 & -13 \\ -4 & 22 & 13 \\ 1 & -6 & -3 \end{pmatrix}$$

Rekursives Anwenden des Algorithmus auf die durch Streichen (bzw. Ignorieren) der ersten Zeile und Spalte erhaltene Untermatrix, gibt

$$\begin{pmatrix} -36 & -21 \end{pmatrix} \mapsto \begin{pmatrix} -15 & -21 \end{pmatrix} \mapsto \begin{pmatrix} -15 & -6 \end{pmatrix} \mapsto \begin{pmatrix} -3 & -6 \end{pmatrix} \mapsto \begin{pmatrix} -3 & 0 \end{pmatrix}$$

(in diesem Fall ist dies genau der euklidische Algorithmus) und wir erhalten die Smith-Normalform

$$D = S \cdot A \cdot T$$

mit

$$S = \begin{pmatrix} 1 & 1 \\ -7 & -6 \end{pmatrix} \qquad D = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -3 & 0 \end{pmatrix} \qquad T = \begin{pmatrix} 4 & 2 & -9 \\ -4 & 1 & 2 \\ 1 & -3 & 6 \end{pmatrix}$$

Die Elementarteiler von A sind also

$$d_1 = 1 \qquad d_2 = 3$$

bis auf Multiplikation mit Einheiten in $\mathbb{Z}^\times = \{1, -1\}$.

Beispiel 5.2.4 *Den Kern erhalten wir als*

$$\ker(A) = \langle Te_3 \rangle = \left\langle \begin{pmatrix} -9 \\ 2 \\ 6 \end{pmatrix} \right\rangle$$

und mit

$$S^{-1} = \begin{pmatrix} -6 & -1 \\ 7 & 1 \end{pmatrix}$$

das Bild als

$$\mathrm{Bild}(A) = \left\langle S^{-1}e_1, -3S^{-1}e_2 \right\rangle = \left\langle \begin{pmatrix} -6 \\ 7 \end{pmatrix}, \begin{pmatrix} 3 \\ -3 \end{pmatrix} \right\rangle$$

Offenbar gibt es einfachere Basen als die gefundene, denn

$$\mathrm{Bild}(A) = \left\langle \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 3 \\ -3 \end{pmatrix} \right\rangle = \left\langle \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 3 \\ 0 \end{pmatrix} \right\rangle$$

*Mit dem **LLL-Algorithmus** werden wir ein Verfahren kennenlernen, das es uns erlaubt, Basen aus kurzen Vektoren zu finden.*

Bemerkung 5.2.5 *Vorsicht, anders als bei Vektorräumen lässt sich nicht jedes Erzeugendensystem eines Moduls zu einer Basis verkürzen. In Beispiel 5.2.4 erzeugen keine zwei Spalten von A das Bild von A, zum Beispiel ist*

$$\begin{pmatrix} 0 \\ 1 \end{pmatrix} \notin \left\langle \begin{pmatrix} 6 \\ 6 \end{pmatrix}, \begin{pmatrix} 9 \\ 6 \end{pmatrix} \right\rangle$$

Beispiel 5.2.6 *In* JULIA/NEMO *können wir die Smith Normalform wie folgt bestimmen:*

```
using Nemo
S = MatrixSpace(ZZ, 2, 3)
Matrix Space of 2 rows and 3 columns over Integer Ring
A = S([6 9 6; 6 6 7]);
D = snf(A)
```

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 3 & 0 \end{pmatrix}$$

Auf die Berechnung von S and T werden wir in den praktischen Übungsaufgaben zurückkommen. Man beachte, dass S und T nicht eindeutig bestimmt sind.

Dass $r \leq n$ und die Elementarteiler d_i bis auf Einheiten eindeutig bestimmt sind, folgt aus folgendem Satz:

Satz 5.2.7 *Für die Elementarteiler $d_1, ..., d_r$ von $A \in R^{n\times m}$ in Satz 5.2.1 gilt für $i \leq r$ (bis auf Einheiten)*

$$d_1 \cdot ... \cdot d_i = \mathrm{ggT}\left(\det\left(A_{I,J}\right) \mid |I| = |J| = i\right) =: D_i$$

Für $i > r$ sind alle $\det\left(A_{I,J}\right) = 0$.

Hier bezeichnet für $I \subset \{1, ..., n\}$ und $J \subset \{1, ..., m\}$

$$A_{I,J} \in R^{|I|\times|J|}$$

*die Untermatrix von A mit den Zeilen aus I und Spalten aus J. Die $\det\left(A_{I,J}\right)$ mit $|I| = |J| = i$ heißen $i \times i$-**Minoren** von A.*

*Man nennt D_i auch den i-ten **Determinantenteiler** von A.*

Insbesondere ist $d_1 = D_1$ der größte gemeinsame Teiler aller Einträge von A.

Beweis. Ist A in Smith-Normalform, dann ist der Satz klar. Dass ein Basiswechsel die Minoren nur um Einheiten ändert können wir hier nicht zeigen. Es ist aber plausibel, da z.B. für die Determinante einer quadratischen Matrix A gilt

$$\det(S \cdot A \cdot T) = \det(S)\det(A)\det(T)$$

und $\det(S), \det(T) \in R^\times$ für invertierbare S und T. ■

Beispiel 5.2.8 *Wir bestimmen für*

$$A = \begin{pmatrix} 1 & 1 & 1 \\ -3 & 1 & 1 \\ 1 & -3 & 1 \\ 1 & 1 & -3 \end{pmatrix}$$

die Smith-Normalform:

$$A \mapsto \begin{pmatrix} 1 & 0 & 0 \\ 0 & 4 & 4 \\ 0 & -4 & 0 \\ 0 & 0 & -4 \end{pmatrix} \mapsto \begin{pmatrix} 1 & 0 & 0 \\ 0 & 4 & 4 \\ 0 & -4 & 0 \\ 0 & 0 & -4 \end{pmatrix}$$

$$\mapsto \begin{pmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 4 \\ 0 & 0 & -4 \end{pmatrix} \mapsto \begin{pmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 4 \\ 0 & 0 & 0 \end{pmatrix} = D$$

Andererseits ist offenbar

$$D_1 = 1 = d_1$$

die 2×2*-Minoren sind* $0, \pm 4$ *oder* 8*, also*

$$D_2 = 4 = d_1 \cdot d_2$$

und die 3×3 *Minoren sind* ± 16*, also*

$$D_3 = 16 = d_1 \cdot d_2 \cdot d_3$$

Aus dem Elementarteilersatz 5.2.1 erhalten wir sofort folgendes zentrale Resultat der Gruppentheorie, den Hauptsatz über endlich erzeugte abelsche Gruppen:

Satz 5.2.9 (Hauptsatz endlich erzeugte abelsche Gruppen) *Sei G eine endlich erzeugte abelsche Gruppe. Dann gibt es Zahlen $0 \leq r \leq n$ und $d_1, \ldots, d_r \geq 2$ mit $d_i \mid d_{i+1}$ für $i = 1, \ldots, r-1$, sodass*

$$G \cong \mathbb{Z}/d_1 \times \ldots \times \mathbb{Z}/d_r \times \mathbb{Z}^{n-r}$$

Beweis. Da G endlich erzeugt ist, ist G nach Satz 4.12.36 endlich präsentiert (denn nach Definition und Satz 4.2.1 und Satz 4.3.3 sind Euklidische Ringe Noethersch), es gibt also eine Präsentation

$$\mathbb{Z}^m \xrightarrow{A} \mathbb{Z}^n \to G \to 0.$$

Mit dem Satz über die Smith-Normalform gibt es Isomorphismen

$$\begin{array}{ccccccc} \mathbb{Z}^m & \xrightarrow{A} & \mathbb{Z}^n & \to & G & \to & 0 \\ \cong & & \cong & & \cong & & \\ \mathbb{Z}^m & \xrightarrow{D} & \mathbb{Z}^n & \to & G' & \to & 0 \end{array}$$

mit D in Normalform. Der Cokern von D hat also die behauptete Struktur, wobei wir Faktoren mit $d_i = 1$ weglassen können, da $\mathbb{Z}/1 = \{0\}$ die triviale Gruppe ist. ∎

Beispiel 5.2.10 *Für die endlich erzeugte abelsche Gruppe* $G = \operatorname{coker} A$ *mit*

$$A = \begin{pmatrix} 1 & 1 & 1 \\ -3 & 1 & 1 \\ 1 & -3 & 1 \\ 1 & 1 & -3 \end{pmatrix}$$

erhalten wir aus der Smith-Normalform

$$D = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 4 \\ 0 & 0 & 0 \end{pmatrix}$$

(siehe Beispiel 5.2.8), dass

$$G = \operatorname{coker} A \cong \operatorname{coker} D \cong \mathbb{Z}/4 \times \mathbb{Z}/4 \times \mathbb{Z}$$

(siehe dazu Beispiel 4.12.37).

Bemerkung 5.2.11 *Freie Auflösungen über Hauptidealringen* R *haben eine viel einfachere Struktur als über einem multivariaten Polynomring. Ist* M *ein endlich erzeugter* R*-Modul und*

$$R^m \xrightarrow{A} R^n \to M \to 0$$

eine Präsentation, dann können wir vermöge einem Basiswechsel in R^n *und* R^m *annehmen, dass* A *in Smith-Normalform ist. Durch Streichen der Nullspalten in* A *erhalten wir eine Matrix* $\widetilde{A} \in R^{n \times \tilde{m}}$ *mit* $\ker \widetilde{A} = \{0\}$ *und eine Präsentation*

$$0 \to R^{\tilde{m}} \xrightarrow{\widetilde{A}} R^m \to M \to 0,$$

denn durch das Streichen ändert sich das Bild der Matrix nicht.

Beispiel 5.2.12 *Für*

$$A = \begin{pmatrix} 6 & 9 & 6 \\ 6 & 6 & 7 \end{pmatrix}$$

wie in Beispiel 5.2.3 ist

$$M = \operatorname{coker} A \cong \operatorname{coker} \begin{pmatrix} 1 & 0 & 0 \\ 0 & -3 & 0 \end{pmatrix} \cong \operatorname{coker} \begin{pmatrix} 1 & 0 \\ 0 & -3 \end{pmatrix}$$

5.3 Hermite-Normalform

Mit der Smith Normalform können wir (wie in Satz 5.2.9) die Struktur jedes endlich erzeugten Moduls über einem beliebigen Euklidischen

Ring vollständig analysieren, indem wir die Normalform seiner Präsentationsmatrix bestimmen.

In vielen praktischen Anwendungen ist ein Modul jedoch nicht durch eine Präsentation (d.h. durch Relationen zwischen Erzeugern), sondern als Bild oder Kern einer Matrix gegeben. Beispielsweise sind ganzzahlige Gitter als Bild und Lösungsmengen von homogenen linearen Gleichungssystemen als Kern spezifiziert.

Wie schon in Bemerkung 5.1.2 gesehen, löst die Smith Normalform auch dieses Problem, allerdings lassen sich Basen von Kern und Bild einer Matrix auch mit einem einfacheren Normalformenalgorithmus finden. Offenbar ändern Spaltentransformationen z.B. nicht das Bild. Welche Art von Normalform können wir also nur durch Spaltentransformationen erreichen?

Definition 5.3.1 *Sei R ein Euklidischer Ring mit Norm d. Eine Matrix $A = (a_{i,j}) \in R^{n\times m}$ ist in (Spalten-)**Hermite-Normalform**, wenn es ein $r \leq m$ und eine streng monotone Funktion*

$$f : \{1, ..., r\} \to \{1, ..., n\}$$

gibt mit

	$a_{i,j} = 0$ *für alle* $i < f(j)$
$d(a_{f(j),j'}) < d(a_{f(j),j})$ *oder* $a_{f(j),j'} = 0$ *für alle* $j' < j$	$\boldsymbol{a_{f(j),j} \neq 0}$

und

$$a_{i,j} = 0 \text{ für alle } i \text{ und } j > r$$

Beispiel 5.3.2 *Die Matrix*

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ \boxed{2} & 0 & 0 & 0 & 0 \\ * & 0 & 0 & 0 & 0 \\ 1 & \boxed{5} & 0 & 0 & 0 \\ * & * & 0 & 0 & 0 \\ * & * & 0 & 0 & 0 \\ 2 & 0 & \boxed{3} & 0 & 0 \end{pmatrix}$$

ist in Hermite Normalform.

Satz 5.3.3 *Sei R ein Euklidischer Ring und $A \in R^{n\times m}$. Dann gibt es ein $T \in \mathrm{GL}(m, R)$ sodass*

$$D = A \cdot T$$

in Hermite-Normalform ist.

Bevor wir den Satz beweisen, einige Bemerkungen zu Anwendungen:

Bemerkung 5.3.4 *Aus der Hermite-Normalform D erhalten wir direkt eine Basis von*

$$\mathrm{Bild}(A) = \mathrm{Bild}(D)$$

und wie bei der Smith-Normalform eine Basis des Kerns

$$\ker(A) = \langle Te_{r+1}, ..., Te_m \rangle .$$

Bemerkung 5.3.5 *Die Struktur des Cokerns lässt sich dagegen nicht unmittelbar aus der Hermite-Normalform ablesen: Zum Beispiel ist*

$$A = \begin{pmatrix} 6 & 0 \\ 0 & 8 \\ 0 & 4 \end{pmatrix}$$

schon in Hermite-Normalform. Mit dem Smith-Normalform-Algorithmus liefert

$$\begin{aligned} A \mapsto \begin{pmatrix} 6 & 8 \\ 0 & 8 \\ 0 & 4 \end{pmatrix} \mapsto \begin{pmatrix} 6 & 2 \\ 0 & 8 \\ 0 & 4 \end{pmatrix} \mapsto \begin{pmatrix} 2 & 6 \\ 8 & 0 \\ 4 & 0 \end{pmatrix} \\ \mapsto \begin{pmatrix} 2 & 0 \\ 8 & -24 \\ 4 & -12 \end{pmatrix} \mapsto \begin{pmatrix} 2 & 0 \\ 0 & -24 \\ 0 & -12 \end{pmatrix} \mapsto \begin{pmatrix} 2 & 0 \\ 0 & 12 \\ 0 & 0 \end{pmatrix} \end{aligned}$$

für die Gruppe G mit Präsentation

$$\mathbb{Z}^2 \xrightarrow{A} \mathbb{Z}^3 \to G \to 0$$

somit, dass

$$G = \mathrm{coker}(A) \cong \mathbb{Z}/2 \times \mathbb{Z}/12 \times \mathbb{Z}.$$

Also Vorsicht: Die Diagonalelemente von Smith- und Hermite-Normalform stimmen im Allgemeinen nicht überein, nicht einmal das Produkt der Diagonalelemente.

Beispiel 5.3.6 *Ist $A \in R^{n \times n}$ dagegen quadratisch und $D = S \cdot A \cdot T$ die Smith-Normalform, dann*

$$\det(D) = \det(S) \cdot \det(A) \cdot \det(T) = u \cdot \det(A)$$

mit $u \in R^\times$, und analog für die Hermite-Normalform. Das Produkt der Diagonalelemente stimmt also bis auf Einheiten überein. Auch für quadratische Matrizen können die Diagonalelemente selbst verschieden sein: Die Matrix

$$A = \begin{pmatrix} 2 & 0 \\ 1 & 2 \end{pmatrix}$$

ist schon in Hermite-Normalform, und als Smith-Normalform erhalten wir

$$D = \begin{pmatrix} 1 & 0 \\ 0 & 4 \end{pmatrix}$$

Wie bei der Smith-Normalform liefert der Beweis von Satz 5.3.3 einen Algorithmus zur Bestimmung von D und T, den Hermite-Normalformen-Algorithmus:

Beweis. Wir zeigen die Behauptung mit Induktion nach n.

1) Für

$$A = \left(\begin{array}{ccc} 0 & \cdots & 0 \\ \hline & A' & \end{array}\right)$$

wende die Induktionsvoraussetzung auf A' an.

Anderenfalls können wir durch Spaltenvertauschungen erreichen, dass $a_{1,1} \neq 0$ und

$$d(a_{1,1}) \leq d(a_{1,j}) \text{ oder } a_{i,j} = 0$$

für alle $j \geq 1$.

2) Ist ein Eintrag $a_{1,j}$ der ersten Zeile nicht durch $a_{1,1}$ teilbar, dann schreibe $a_{1,j}$ mit Division mit Rest

$$a_{1,j} = q \cdot a_{1,1} + r$$

mit $d(r) < d(a_{1,1})$. Der Fall $r = 0$ tritt nicht auf, da nach Voraussetzung $a_{1,1} \nmid a_{1,j}$.

Nach Subtraktion des q-fachen der 1-ten Spalte von der j-ten Spalte erreichen wir also

$$d(a_{1,1}) > d(a_{1,j})$$

Gehe nun zurück zu Schritt (1). Dieser Prozess terminiert, da $d(a_{1,1})$ in jedem Durchlauf echt kleiner wird.

3) Sind alle Einträge der ersten Zeile durch $a_{1,1}$ teilbar, können wir durch Addition von Vielfachen der ersten Spalte A in die Form

$$\left(\begin{array}{c|ccc} a_{1,1} & 0 & \cdots & 0 \\ \hline * & & A' & \end{array}\right)$$

bringen. Für $n = 1$ ist die Matrix nun in Hermite-Normalform. Anderenfalls wende die Induktionsvoraussetzung auf A' an.

4) Sei r minimal mit $a_{i,j} = 0$ für alle i und $j > r$. Für $j = 1, ..., r$ sei

$$f(j) = \min\{i \mid a_{i,j} \neq 0\}$$

der Zeilenindex der j-ten Stufe. Für $j = 2, ..., r$ und $j' = 1, ..., j-1$ schreibe

$$a_{f(j),j'} = q \cdot a_{f(j),j} + r$$

mit $d(r) < d(a_{f(j),j})$ und subtrahiere das q-fache der j-ten Spalte von der j'-ten Spalte. Danach gilt dann

$$d(a_{f(j),j'}) < d(a_{f(j),j}) \text{ oder } a_{f(j),j'} = 0$$

für alle $1 \leq j' < j \leq r$.

■

Eindeutigkeit der Hermite-Normalform erreichen wir, wenn wir den ggT und die Divisionsreste geeignet normieren. Beispielsweise können wir über $\mathbb{Z}$ festlegen, dass

$$\begin{aligned} &a_{f(j),j} > 0 \\ &a_{f(j),j} > a_{f(j),j'} \geq 0 \quad \text{für alle } j' < j \end{aligned}$$

Dann gilt:

Satz 5.3.7 *Die Hermite-Normalform ist eindeutig.*

Beweis. Die Zahl r ist der Rang des nach Bemerkung 5.1.2 freien Moduls $M = \text{Bild}(A) = \text{Bild}(D)$. Ebenso ist für jedes j

$$f(j) = \max\{i \mid \text{rank}(M \cap \langle e_i, ..., e_n\rangle) = r - j + 1\}$$

durch A festgelegt. Wir sagen, dass eine Basis $b_1, ..., b_r$ in Hermite-Form ist, wenn dies für die Matrix $(b_1 \mid ... \mid b_r)$ gilt.

Wir zeigen mit Induktion nach r, dass M eine eindeutige Basis in Hermite-Form hat: Ein Modul vom Rang $r = 1$ hat einen eindeutigen normierten Basisvektor (über den ganzen Zahlen würden wir etwa den ersten Eintrag $\neq 0$ auf positives Vorzeichen normieren). Für $r > 1$ ist für $i = f(r)$

$$\text{rank}(M \cap \langle e_i, ..., e_n\rangle) = 1$$

es gibt also wieder einen eindeutigen normierten Basisvektor $v \in M \cap \langle e_i, ..., e_n\rangle$.

Die Induktionsvoraussetzung liefert für den freien Modul

$$M/\langle v\rangle \cong M/\langle e_i, ..., e_n\rangle \subset R^n/\langle e_i, ..., e_n\rangle \cong R^{n-i}$$

eindeutige Basisvektoren $\bar{b}_1, ..., \bar{b}_{r-1}$ in Hermite-Form. Jedes $\bar{b}_i$ hat einen Repräsentanten $b_i \in M$ und dieser ist eindeutig bis auf Vielfache von v. Legen wir ein Restesystem für den Divisionsrest fest, so erhalten wir ein eindeutiges b_i. Die Vektoren $b_1, ..., b_s, v$ bilden dann die eindeutige normierte Basis in Hermite-Form. ■

Für die eindeutige Hermite-Normalform von $A \in R^{n\times m}$ schreibe $\text{HNF}(A)$.

Beispiel 5.3.8 *Wir bestimmen die Hermite-Normalform der Matrix aus Beispiel 5.2.3 und zugleich den Basiswechsel T in der Quelle*

$$\begin{pmatrix} 6 & 9 & 6 \\ 6 & 6 & 7 \end{pmatrix} \quad \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Division mit Rest (Schritt 2) gibt

$$\begin{pmatrix} 6 & 3 & 6 \\ 6 & 0 & 7 \end{pmatrix} \quad \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Vertauschen (Schritt 1)

$$\begin{pmatrix} 3 & 6 & 6 \\ 0 & 6 & 7 \end{pmatrix} \quad \begin{pmatrix} -1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Reduktion der ersten Zeile (Schritt 3)

$$\begin{pmatrix} 3 & 0 & 0 \\ 0 & 6 & 7 \end{pmatrix} \quad \begin{pmatrix} -1 & 3 & 2 \\ 1 & -2 & -2 \\ 0 & 0 & 1 \end{pmatrix}$$

Division mit Rest (Schritt 2)

$$\begin{pmatrix} 3 & 0 & 0 \\ 0 & 6 & 1 \end{pmatrix} \quad \begin{pmatrix} -1 & 3 & -1 \\ 1 & -2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Vertauschen (Schritt 1)

$$\begin{pmatrix} 3 & 0 & 0 \\ 0 & 1 & 6 \end{pmatrix} \quad \begin{pmatrix} -1 & -1 & 3 \\ 1 & 0 & -2 \\ 0 & 1 & 0 \end{pmatrix}$$

Reduktion der zweiten Zeile (Schritt 3)

$$\begin{pmatrix} 3 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} -1 & -1 & 9 \\ 1 & 0 & -2 \\ 0 & 1 & -6 \end{pmatrix}$$

Eine Basis des Bilds ist also

$$\mathrm{Bild}(A) = \left\langle \begin{pmatrix} 3 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\rangle$$

und eine Basis des Kerns

$$\ker(A) = \langle Te_3 \rangle = \left\langle \begin{pmatrix} 9 \\ -2 \\ -6 \end{pmatrix} \right\rangle$$

MAPLE *berechnet eine Zeilen-Hermite-Normalform, wir müssen also die Matrix vor und nach der Berechnung transponieren (an der Diagonalen spiegeln):*

```
using Nemo
S = MatrixSpace(ZZ, 2, 3)
Matrix Space of 2 rows and 3 columns over Integer Ring
A = S([6 9 6; 6 6 7]);
D = transpose(hnf(transpose(A)))
```

$$\begin{pmatrix} 3 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

Zur Bestimmung der Spaltentransformationen gehen wir analog vor. Man beachte, dass beim Transponieren gilt

$$A \cdot T = D \Leftrightarrow T^t \cdot A^t = D^t.$$

```
(tD,tT) = hnf_with_transform(transpose(A));
T = transpose(tT)
```

$$\begin{pmatrix} -1 & 8 & -9 \\ 1 & -2 & 2 \\ 0 & -5 & 6 \end{pmatrix}$$

```
A*T
```

$$\begin{pmatrix} 3 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

Beispiel 5.3.9 *Für ein weiteres Beispiel betrachte*

$$A = \begin{pmatrix} 4 & 2 \\ 13 & 5 \end{pmatrix}$$

Der Hermite-Normalform-Algorithmus liefert dann

$$A \mapsto \begin{pmatrix} 2 & 4 \\ 5 & 13 \end{pmatrix} \mapsto \begin{pmatrix} 2 & 0 \\ 5 & 3 \end{pmatrix} \mapsto \begin{pmatrix} 2 & 0 \\ 2 & 3 \end{pmatrix} = D$$

Bemerkung 5.3.10 *Seien $A \in R^{n\times m}$ und $B \in R^{n\times m'}$. Mit der Hermite-Normalform kann man Gleichheit, Inklusion und Modulmitgliedschaft testen.*

$$\begin{aligned} \text{Bild}(A) = \text{Bild}(B) &\Leftrightarrow \text{HNF}(A) = \text{HNF}(B) \\ \text{Bild}(B) \subset \text{Bild}(A) &\Leftrightarrow \text{HNF}(A) = \text{HNF}(A \mid B) \\ v \in \text{Bild}(A) &\Leftrightarrow \text{HNF}(A) = \text{HNF}(A \mid v) \end{aligned}$$

Dabei betrachten wir zwei Hermite-Normalformen als gleich, wenn sie sich nur um Nullspalten unterscheiden.

Beispiel 5.3.11 *Wir zeigen, dass*

$$\begin{pmatrix} 6 \\ 9 \end{pmatrix} \in \text{Bild} \begin{pmatrix} 4 & 2 \\ 13 & 5 \end{pmatrix}$$

Ausgehend von der Hermite-Normalform aus Beispiel 5.3.9 berechnen wir dazu die Hermite-Normalform von

$$\begin{pmatrix} 2 & 0 & 6 \\ 2 & 3 & 9 \end{pmatrix} \mapsto \begin{pmatrix} 2 & 0 & 0 \\ 2 & 3 & 3 \end{pmatrix} \mapsto \begin{pmatrix} 2 & 0 & 0 \\ 2 & 3 & 0 \end{pmatrix}$$

Beispiel 5.3.12 *Der Hermite-Normalformen-Algorithmus führt im Allgemeinen zu in der Anzahl n der Zeilen exponentiellem Koeffizientenwachstum. Schon bei zwei Zeilen:*

$$\begin{aligned} A = \begin{pmatrix} 2 & 1007 \\ 1013 & 3 \end{pmatrix} &\mapsto \begin{pmatrix} 2 & 1 \\ 1013 & -509536 \end{pmatrix} \\ \mapsto \begin{pmatrix} 1 & 2 \\ -509536 & 1013 \end{pmatrix} &\mapsto \begin{pmatrix} 1 & 0 \\ -509536 & 1020085 \end{pmatrix} = D \end{aligned}$$

Die Zwischenergebnisse können sogar noch wesentlich stärker mit der Anzahl der Zeilen von A ansteigen (überprüfen Sie dies mit Ihrer Implementierung aus Übung 5.7 anhand von zufällig gewählten Beispielen). Dies lässt sich mit der folgenden Strategie vermeiden:

Satz 5.3.13 *Sei $A \in R^{n\times m}$ mit Hermite-Normalform*

$$D = \begin{pmatrix} d_1 & & 0 & 0 & \cdots & 0 \\ & \ddots & & \vdots & & \vdots \\ * & & d_n & 0 & \cdots & 0 \end{pmatrix}$$

Mit

$$D_i = \prod_{j\geq i} d_j$$

ist

$$D_i \cdot e_i \in \mathrm{Bild}(A)$$

für alle i.

Beweis. Offenbar ist $d_n \cdot e_n \in \mathrm{Bild}(A)$. Induktiv seien

$$D_n \cdot e_n, ..., D_{i+1} \cdot e_{i+1} \in \mathrm{Bild}(A).$$

Für das $\frac{D_i}{d_i}$-Vielfache der i-ten Spalte von D (mit Einträgen $0, \ldots, 0, d_i, b_{i+1}, \ldots, b_n$) gilt

$$\begin{pmatrix} 0 \\ \vdots \\ 0 \\ D_i \\ \frac{D_i}{d_i} b_{i+1} \\ \vdots \\ \frac{D_i}{d_i} b_n \end{pmatrix} = \frac{D_i}{d_i} \cdot \begin{pmatrix} 0 \\ \vdots \\ 0 \\ d_i \\ b_{i+1} \\ \vdots \\ b_n \end{pmatrix} \in \mathrm{Bild}(D) = \mathrm{Bild}(A).$$

Da $D_s = \prod_{j \geq s} d_j$ die Zahl $\frac{D_i}{d_i} = \prod_{j>i} d_j$ für $s > i$ teilt ist nach Induktionsvoraussetzung auch

$$\begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ \frac{D_i}{d_i} b_{i+1} \\ \vdots \\ \frac{D_i}{d_i} b_n \end{pmatrix} \in \mathrm{Bild}(A)$$

und damit $D_i e_i \in \mathrm{Bild}(A)$. ■

Algorithmus 5.1 Hermite modulo Determinante

Ist z.B. $A \in R^{n \times n}$ quadratisch mit $d = \det(A) \neq 0$ so können wir damit das Koeffizientenwachstum in den Zwischenergebnissen im Hermite-Normalformen-Algorithmus begrenzen: Haben wir schon

$$D = \left(\begin{array}{ccc|c} d_1 & & 0 & \\ & \ddots & & 0 \\ * & & d_{i-1} & \\ \hline & * & & A' \\ & & & \end{array}\right)$$

berechnet, dann ist (bis auf Einheiten)

$$D_i = \det(A') = \frac{d}{d_1 \cdot \ldots \cdot d_{i-1}}$$

und $D_i \cdot e_i, \ldots, D_i \cdot e_n \in \mathrm{Bild}(A')$. Zur Bestimmung der i-ten Zeile der Hermite Normalform fügen wir zu D die Spalte $D_i \cdot e_i$ hinzu. Dies ändert das Ergebnis nicht (bis auf eine zusätzliche Nullspalte). Außerdem können wir dann alle Zeilen $j > i$ modulo D_i reduzieren (da wir im j-ten Schritt $D_j \cdot e_j$ wieder hinzufügen und $D_j \mid D_i$).

5.4 Übungen

Übung 5.1 *Bestimmen Sie für*

$$A = \begin{pmatrix} 4 & 6 & 2 \\ 2 & 3 & 2 \\ 2 & -2 & -2 \end{pmatrix} \in \mathbb{Z}^{3 \times 3}$$

die Smith-Normalform D und $S, T \in \mathrm{GL}(3, \mathbb{Z})$ mit

$$S \cdot A \cdot T = D.$$

Übung 5.2 *Seien*

$$g_1 = \begin{pmatrix} 1 \\ 0 \\ 2 \end{pmatrix},\ g_2 = \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix},\ g_3 = \begin{pmatrix} -3 \\ -3 \\ -3 \end{pmatrix},\ g_4 = \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix} \in \mathbb{Z}^3$$

Bestimmen Sie eine Basis (d.h. ein $\mathbb{Z}$-linear unabhängiges Erzeugendensystem) der von $g_1, ..., g_4$ erzeugten Untergruppe von $\mathbb{Z}^3$.

Übung 5.3 *Sei*

$$A = \begin{pmatrix} 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 2 & 4 & 6 & 2 \\ 3 & 3 & 6 & 3 \end{pmatrix} \in \mathbb{Z}^{4\times 4}$$

Bestimmen Sie

1) *die Smith-Normalform von A.*
2) *die Hermite-Normalform D von A und $T \in \mathrm{GL}(4, \mathbb{Z})$ mit*

$$A \cdot T = D,$$

3) *eine Basis des Bilds von A und*
4) *eine Basis des Kerns von A.*

Übung 5.4 *Sei $R = \mathbb{C}[x]$. Bestimmen Sie jeweils die Hermite-Normalform und die Smith-Normalform von*

1) $A = \begin{pmatrix} 1-x & 1 \\ 0 & 1-x \end{pmatrix} \in R^{2\times 2}$
2) $A = \begin{pmatrix} 1-x & 1 & 1 \\ 1 & 1-x & 1 \\ 1 & 1 & 1-x \end{pmatrix} \in R^{3\times 3}$
3) *Was sagt das Resultat über $A(0)$?*

Übung 5.5 *Sei G eine endliche abelsche Gruppe und*

$$\mathbb{Z}^n \xrightarrow{A} \mathbb{Z}^n \to G \to 0$$

mit $A = (a_{i,j}) \in \mathbb{Z}^{n\times n}$ eine Präsentation. Zeigen Sie, dass für die Gruppenordnung von G gilt

$$|G| = |\det A|$$

5.5 Praktische Aufgaben

Übung 5.6 *1) Implementieren Sie den Algorithmus zur Bestimmung der Smith-Normalform D einer ganzzahligen Matrix $A \in \mathbb{Z}^{n\times m}$:*

2) Modifizieren Sie Ihre Funktion so, dass sie die Zeilen- bzw. Spaltentransformationen simultan auch auf der $n \times n$ bzw. $m \times m$ Einheitsmatrix durchführt, und dadurch $S \in \mathrm{GL}(n,\mathbb{Z})$ und $T \in \mathrm{GL}(m,\mathbb{Z})$ mit $S \cdot A \cdot T = D$ bestimmt.

3) Bestimmen Sie S, T und D für

$$A = \begin{pmatrix} 10 & 41 & 6 & -19 \\ -6 & -19 & -4 & 9 \\ -6 & -41 & -2 & 19 \\ -12 & -62 & -8 & 30 \end{pmatrix}$$

4) Welche Ordnung hat die Gruppe

$$G = \mathrm{coker}(A) = \mathbb{Z}^4 / \mathrm{Bild}(A).$$

Klassifizieren Sie die Gruppe G.

Übung 5.7 *Implementieren Sie über $R = \mathbb{Z}$ den Elementarteileralgorithmus zur Bestimmung der Smith-Normalform und den Algorithmus zur Bestimmung der Hermite-Normalform.*

Übung 5.8 *1) Modifizieren Sie Ihre Implementierung der Smith-Normalform so, dass sie auch für Matrizen $A \in \mathbb{Q}[x]^{n\times m}$ funktioniert.*

2) Sei

$$B = \begin{pmatrix} 3 & 6 & 0 & 0 & 4 & -2 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & -2 & 0 & 0 \\ 1 & 0 & -1 & 0 & 2 & 0 & 1 \\ -1 & -3 & 0 & 0 & -1 & 1 & 0 \\ 0 & 3 & 0 & 0 & 0 & -1 & -1 \\ 0 & -2 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \in \mathrm{End}\left(\mathbb{C}^7\right)$$

Berechnen Sie die Smith-Normalform D von

$$A = xE - B \in \mathbb{C}[x]^{7\times 7}$$

und $S, T \in \mathrm{GL}(7, \mathbb{C}[x])$ mit $D = S \cdot A \cdot T$.

3) Bestimmen Sie die Jordansche Normalform von B. Können Sie eine Beziehung zu dem Ergebnis aus (b) herstellen?

Übung 5.9 *1) Implementieren Sie die Berechnung der Hermite-Normalform D einer ganzzahligen Matrix $A \in \mathbb{Z}^{n\times m}$.*

2) Erweitern Sie Ihre Implementierung so, dass sie auch eine Matrix $T \in \mathrm{GL}(m, \mathbb{Z})$ mit

$$A \cdot T = D$$

liefert.

3) Bestimmen Sie für

$$A = \begin{pmatrix} 9 & -9 & 9 \\ 12 & -17 & 12 \\ 7 & -10 & 7 \end{pmatrix} \in \mathbb{Z}^{3\times 3}$$

Basen von $\mathrm{Bild}(A)$ *und* $\ker(A)$.

6

Algorithmen für Gitter

6.1 Übersicht

Für $A \in \mathbb{Z}^{n\times m}$ finden wir mit dem Hermite-Normalform-Algorithmus eine Basis von $\mathrm{Bild}(A)$. Allerdings hat $\mathrm{Bild}(A)$ im Allgemeinen viele verschiedene Basen. Wie lässt sich die Qualität einer Basis messen? Eine Möglichkeit ist die Betrachtung der Länge der Basisvektoren. Da der Hermite-Normalformen-Algorithmus zu exponentiellem Koeffizientenwachstum führt, sind die resultierenden Basisvektoren typischerweise viel länger als notwendig.

Beispiel 6.1.1 *In Beispiel 5.3.12 haben wir Basen*

$$\begin{aligned}\mathrm{Bild}(A) &= \left\langle \begin{pmatrix} 2 \\ 1013 \end{pmatrix}, \begin{pmatrix} 1009 \\ 1016 \end{pmatrix} \right\rangle \\ &= \left\langle \begin{pmatrix} 1 \\ -509536 \end{pmatrix}, \begin{pmatrix} 0 \\ 1020085 \end{pmatrix} \right\rangle\end{aligned}$$

Die Länge der Vektoren der zweiten Basis ist etwa 10^6*, die Länge der ersten dagegen nur* 10^3*.*

In diesem Kapitel werden wir ein Verfahren kennenlernen, das es uns erlaubt, in sehr effizienter Weise kurze Basen und einen kürzesten Vektor in $\mathrm{Bild}(A)$ zu finden. Bei dieser Fragestellung sind wir nicht auf Untermoduln von $\mathbb{Z}^n$ beschränkt, sondern betrachten allgemeiner Gitter:

Definition 6.1.2 *Ein* ***Gitter*** *ist ein endlich erzeugter, freier* $\mathbb{Z}$*-Modul* $L \subset \mathbb{R}^n$*, der eine Basis aus* $\mathbb{R}$*-linear unabhängigen Vektoren besitzt. Diese bezeichnen wir auch als* ***Gitterbasis****.*

Es gibt also einen $\mathbb{Z}$-Modul-Isomorphismus

$$\varphi : \mathbb{Z}^r \to L$$

sodass $\sum_{i=1}^{r} a_i \varphi(e_i) = 0$ mit $a_i \in \mathbb{R}$ impliziert, dass alle $a_i = 0$. Insbesondere ist $r \leq n$.

Beispiel 6.1.3 *Gitter sind z.B.*

1) $\mathbb{Z}^n \subset \mathbb{R}^n$*, allgemeiner*

2) $\mathrm{Bild}(A) \subset \mathbb{Z}^n \subset \mathbb{R}^n$ *mit* $A \in \mathbb{Z}^{n\times m}$*,*

3) $${}_{\mathbb{Z}}\left\langle \begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}2\\\sqrt{5}\end{pmatrix} \right\rangle = \mathbb{Z}\begin{pmatrix}1\\1\end{pmatrix} + \mathbb{Z}\begin{pmatrix}2\\\sqrt{5}\end{pmatrix}$$

Kein Gitter ist z.B. der $\mathbb{Z}$*-Untermodul*

$$ {}_{\mathbb{Z}}\langle e, \pi\rangle \subset \mathbb{R}^1$$

da die Erzeuger zwar $\mathbb{Z}$*-linear unabhängig, aber nicht* $\mathbb{R}$*-linear unabhängig sind.*

Wie kann man nun aus gegebenen Erzeugern eines Gitters eine möglichst kurze Basis konstruieren? Für $a, b \in \mathbb{Z}$ ist

$$L = \langle a, b\rangle = \langle \mathrm{ggT}(a, b)\rangle \subset \mathbb{Z}$$

Der Euklidische Algorithmus liefert direkt eine kürzeste Linearkombination

$$g = \mathrm{ggT}(a, b) = ua + vb \neq 0$$

von a und b mit $u, v \in \mathbb{Z}$, denn g teilt jedes Element von L. Im letzten Abschnitt haben wir zumindest für Gitter in $\mathbb{Z}^n$ gesehen, wie man aus Erzeugern eine Basis erhält. Die Idee zur Berechnung einer kurzen Basis eines Gitters ist, eine Art von Euklidischem Algorithmus zu entwickeln, der statt mit Zahlen mit Vektoren arbeitet. Der **Gauss-Lagrange-Algorithmus** findet für $n = 2$ einen kürzesten Vektor. Für beliebiges n verwendet man den **LLL-Algorithmus** benannt nach L. Lovász, H. Lenstra und A. Lenstra. Er kann in etwa genauso effizient wie der Euklidische Algorithmus durchgeführt werden, findet dann aber nicht mehr sicher den kürzesten Vektor.

6.2 Anwendung: Rationale Rekonstruktion

Angenommen ein Algorithmus berechnet eine rationale Zahl $\frac{a}{b} \in \mathbb{Q}$. Um die Größe von Zwischenergebnissen zu beschränken rechnet, man statt in $\mathbb{Q}$ in $\mathbb{Z}/N$. Wir nehmen an, dass $\gcd(b, N) = 1$. Ist N ein Produkt von großen Primzahlen, dann ist dies sehr wahrscheinlich. Wie kann man aus $\overline{a} \cdot \overline{b}^{-1} = \overline{r} \in \mathbb{Z}/N$ wieder a und b rekonstruieren?

Beispiel 6.2.1 *Sei* $\frac{a}{b} = \frac{2}{3}$ *und* $N = 101$*. Mit dem Euklidischen Algorithmus können wir* 3 *modulo* 101 *invertieren*

$$\overline{3}^{-1} = \overline{34} \in \mathbb{Z}/101.$$

also

$$\overline{r} = \overline{68} \in \mathbb{Z}/101.$$

Können wir aus dieser Restklasse wieder die rationale Zahl $\frac{2}{3}$ *rekonstruieren?*

Der folgende Satz zeigt, dass alle kürzesten Vektoren in dem Gitter L erzeugt von

$$\begin{pmatrix} N \\ 0 \end{pmatrix}, \begin{pmatrix} r \\ 1 \end{pmatrix}$$

ein Vielfaches von

$$\begin{pmatrix} a \\ b \end{pmatrix}$$

sind, vorausgesetzt, dass a und b nicht zu groß waren im Vergleich zu N. Wir setzen voraus, dass

$$a^2 + b^2 < N.$$

Satz 6.2.2 *Für alle* $(x, y) \in L$ *mit* $x^2 + y^2 < N$ *gilt*

$$\frac{x}{y} = \frac{a}{b},$$

und es gibt einen solchen Vektor $(x, y) \in L$.

Wir zeigen zunächst:

Lemma 6.2.3 *Alle* $(x, y) \in L$ *mit* $x^2 + y^2 < N$ *sind collinear.*

Beweis. Seien

$$\lambda = \begin{pmatrix} x \\ y \end{pmatrix}, \ \mu = \begin{pmatrix} c \\ d \end{pmatrix} \in L$$

mit $x^2 + y^2 < N$ und $c^2 + d^2 < N$. Dann ist auch

$$\begin{pmatrix} yc - xd \\ 0 \end{pmatrix} = y \cdot \mu - d \cdot \lambda \in L$$

also

$$N \mid (yc - xd).$$

Mit

$$\mu' = \begin{pmatrix} d \\ -c \end{pmatrix}$$

und der Cauchy-Schwarz-Ungleichung gilt aber

$$|yc - xd| = |\langle \lambda, \mu' \rangle| \leq \|\lambda\| \cdot \|\mu'\| = \|\lambda\| \cdot \|\mu\| < N$$

und damit

$$\det \begin{pmatrix} c & x \\ d & y \end{pmatrix} = yc - xd = 0.$$

■

Nun zu Satz 6.2.2:

Beweis. Wir haben nach Voraussetzung

$$a \equiv b \cdot r \bmod N$$

d.h. es gibt ein $k \in \mathbb{Z}$ mit $a - b \cdot r = k \cdot N$. Also ist

$$\begin{pmatrix} a \\ b \end{pmatrix} - b \begin{pmatrix} r \\ 1 \end{pmatrix} = \begin{pmatrix} a - br \\ 0 \end{pmatrix} = \begin{pmatrix} k \cdot N \\ 0 \end{pmatrix} \in L$$

und damit

$$\begin{pmatrix} a \\ b \end{pmatrix} \in L.$$

Da $a^2 + b^2 < N$ gibt Lemma 6.2.3, dass

$$\frac{a}{b} = \frac{x}{y}.$$

■

Beispiel 6.2.4 *In* JULIA/NEMO *finden wir für Beispiel 6.2.1 einen kürzesten Vektor in dem Gitter*

$$\begin{pmatrix} 101 \\ 0 \end{pmatrix}, \begin{pmatrix} 68 \\ 1 \end{pmatrix}$$

mit:

```
using Nemo
S = MatrixSpace(ZZ, 2, 2)
Matrix Space of 2 rows and 2 columns over Integer Ring
A = S([101 0; 68 1]);
lll(A)
[2, 3], [-23, 16]
```

Wir erhalten also den kürzesten Vektor $(2, 3) \in L$*, der die rationale Zahl* $\frac{2}{3}$ *repräsentiert.*

Bemerkung 6.2.5 *Praktisch rechnet man modulo verschiedenen Primzahlen* $p_1, ..., p_n$*, liftet dann die Ergebnisse* $\overline{r_1}, ..., \overline{r_n}$ *mit dem Chinesischen Restsatz*

$$\begin{array}{ccc} \mathbb{Z}/p_1 \times ... \times \mathbb{Z}/p_n & \cong & \mathbb{Z}/(p_1 \cdot ... \cdot p_n) \\ (\overline{r_1}, ..., \overline{r_n}) & \mapsto & \overline{r} \end{array}$$

und wendet die rationale Rekonstruktion auf $\overline{r}$ *und* $N = p_1 \cdot ... \cdot p_n$ *an.*

Beispiel 6.2.6 *Verwende* $p_1 = 11$ *und* $p_2 = 13$*. Wir berechnen*

$$\frac{2}{3} \cdot \frac{6}{5} = \ ?$$

Wir berechnen das Produkt modulo p_1 und p_2 und kombinieren die Ergebnisse mit dem Chinesischen Restsatz

$$\begin{array}{ccccc} \mathbb{Q} & \to & \mathbb{Z}/11 \times \mathbb{Z}/13 & \cong & \mathbb{Z}/(11\cdot 13) \\ \frac{2}{3} & \mapsto & (\overline{8}, \overline{5}) & & \\ & & \cdot & & \\ \frac{6}{5} & \mapsto & (\overline{10}, \overline{9}) & & \\ & & \| & & \\ & & (\overline{3}, \overline{6}) & \mapsto & \overline{58} \end{array}$$

In JULIA/NEMO *führen wir diese Rechnung durch mit*

```
using Nemo
mod(ZZ(2)*invmod(ZZ(3),ZZ(11),ZZ(11))
8
```

für die Reduktion von Brüchen modulo Primzahlen und analog für die anderen Komponenten. Die Multiplikation führen wir durch mit

```
mod(ZZ(8)*ZZ(10),ZZ(11))
3
```

und analog für die zweite Komponente des kartesischen Produkts $\mathbb{Z}/11 \times \mathbb{Z}/13$. *Mit dem Chinesischen Restsatz bilden wir ab nach* $\mathbb{Z}/(11\cdot 13)$ *mit*

```
crt(ZZ(3),ZZ(11),ZZ(6),ZZ(13))
58
```

Der Gauß-Lagrange-Algorithmus oder der LLL-Algorithmus ergibt (wobei die Erzeuger des Gitters in den Zeilen der Matrix A stehen)

```
S = MatrixSpace(ZZ, 2, 2);
A = S([11*13 0; 58 1]);
lll(A)
[[4, 5], [1267, -1017]]
```

also nach Satz 6.2.2

$$\frac{2}{3} \cdot \frac{6}{5} = \frac{4}{5}$$

da $4^2 + 5^2 = 41 < N = 11 \cdot 13$.

Siehe auch Übung 6.2. In manchen Anwendungen hat man keine Schranke für die Größe des korrekten Ergebnisses $\frac{a}{b}$ oder kann $\mathrm{ggT}(b, N) = 1$ nicht testen. Dann muss man das modulare Ergebnis verifizieren, was oft trotzdem billiger ist als es über $\mathbb{Q}$ zu berechnen.

6.3 Gram-Schmidt-Verfahren und Determinante

Wie lässt sich die Länge einer Gitterbasis $v_1, ..., v_r$ messen? Mit der **Euklidischen Norm**

$$\|x\| = \sqrt{x_1^2 + ... + x_n^2} = \sqrt{x^t \cdot x}$$

von $x \in \mathbb{R}^n$ ist

$$\|v_1\| \cdot ... \cdot \|v_r\| \in \mathbb{R}$$

ein Maß für die Länge der Basis. Die Länge eines kürzesten Vektors in dem Gitter hängt natürlich von dem Gitter ab. Somit macht dieses Maß nur Sinn relativ zu einer Invariante $d(L)$ des Gitters, die wiederum unabhängig von der Wahl einer Basis ist. Wir werden zeigen, dass $d(L)$ eine untere Schranke für $\|v_1\| \cdot ... \cdot \|v_r\|$ ist.

Definition und Satz 6.3.1 *Sei $L \subset \mathbb{R}^n$ ein Gitter mit Basis $v_1, ..., v_r$ und $X = (v_1 \mid ... \mid v_r) \in \mathbb{R}^{n \times r}$. Die **Determinante** (oder Diskriminante)*

$$d(L) = \sqrt{\det(X^t \cdot X)}.$$

von L ist unabhängig von der Wahl der Basis.

Beweis. Ist $w_1, ..., w_r$ eine weitere Basis von L und

$$Y = (w_1 \mid ... \mid w_r).$$

Wir haben dann ein Diagramm von $\mathbb{Z}$-Modulisomorphismen

$$\begin{array}{lccc} X: & \mathbb{Z}^r & \longrightarrow & L \\ & \downarrow & & \| \\ Y: & \mathbb{Z}^r & \longrightarrow & L \end{array}$$

es gibt also eine Matrix $Q \in \mathrm{GL}(r, \mathbb{Z})$ mit $X = Y \cdot Q$. Somit gilt

$$\det(X^t \cdot X) = \det(Q^t \cdot Y^t \cdot Y \cdot Q) = \det(Q) \cdot \det(Y^t \cdot Y) \cdot \det(Q) = \det(Y^t \cdot Y),$$

denn $\det(Q) = \pm 1$. ■

Bemerkung 6.3.2 *Für ein Gitter L von vollem Rang $r = n$ ist*

$$d(L) = \sqrt{\det(X^t \cdot X)} = |\det(X)|.$$

Um zu zeigen, dass $d(L)$ tatsächlich eine untere Schranke für $\|v_1\| \cdot ... \cdot \|v_r\|$ darstellt, verwenden wir das Gram-Schmidt-Verfahren. Dieses berechnet mittels Projektion von Vektoren aus einer Basis eine Orthogonalbasis.

Definition 6.3.3 *Sei*

$$\begin{array}{rccc} \langle -, - \rangle: & \mathbb{R}^n \times \mathbb{R}^n & \to & \mathbb{R} \\ & (v, w) & \mapsto & v^t \cdot w = \sum_{i=1}^n v_i w_i \end{array}$$

*das **Euklidische Skalarprodukt**. Eine **Orthogonalbasis** von $\mathbb{R}^n$ ist eine Basis $(w_1, ..., w_n)$ mit*

$$\langle w_i, w_j \rangle = 0$$

für alle $i \neq j$.

Bemerkung 6.3.4 *Ist* $v \in \langle w_1, ..., w_n \rangle$ *dann gibt es* λ_s *mit*

$$v = \sum_{s=1}^{n} \lambda_s w_s$$

und sind die w_i *orthogonal, so ist*

$$\langle v, w_s \rangle = \lambda_s \cdot \langle w_s, w_s \rangle$$

also

$$v = \sum_{s=1}^{n} \frac{\langle v, w_s \rangle}{\langle w_s, w_s \rangle} w_s.$$

Algorithmus 6.1 Gram-Schmidt

Gegeben eine Basis $(v_1, ..., v_n)$ *von* $\mathbb{R}^n$*, erhalten wir durch*

$$w_1 := v_1$$
$$w_i := v_i - \sum_{s=1}^{i-1} \frac{\langle w_s, v_i \rangle}{\langle w_s, w_s \rangle} \cdot w_s$$

eine Orthogonalbasis $(w_1, ..., w_n)$ *von* $\mathbb{R}^n$*.*

Beweis. Ist schon

$$\langle w_j, w_s \rangle = 0$$

für $j \neq s$ mit $j, s < i$ gezeigt, dann folgt mit der Bilinearität des Skalarprodukts

$$\begin{aligned} \langle w_j, w_i \rangle &= \langle w_j, v_i \rangle - \sum_{s=1}^{i-1} \frac{\langle w_s, v_i \rangle}{\langle w_s, w_s \rangle} \cdot \langle w_j, w_s \rangle \\ &= \langle w_j, v_i \rangle - \frac{\langle w_j, v_i \rangle}{\langle w_j, w_j \rangle} \cdot \langle w_j, w_j \rangle = 0 \end{aligned}$$

für alle $j < i$. Die Symmetrie des Skalarprodukts gibt die Behauptung. ■

Beispiel 6.3.5 *Für*

$$v_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \qquad v_2 = \begin{pmatrix} 3 \\ 1 \end{pmatrix}$$

erhalten wir mit dem Gram-Schmidt-Verfahren die Orthogonalbasis

$$w_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \qquad w_2 = \begin{pmatrix} 3 \\ 1 \end{pmatrix} - \frac{4}{2} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

siehe Abbildung 6.1. Der Beweis von Satz 6.3.1 zeigt auch:

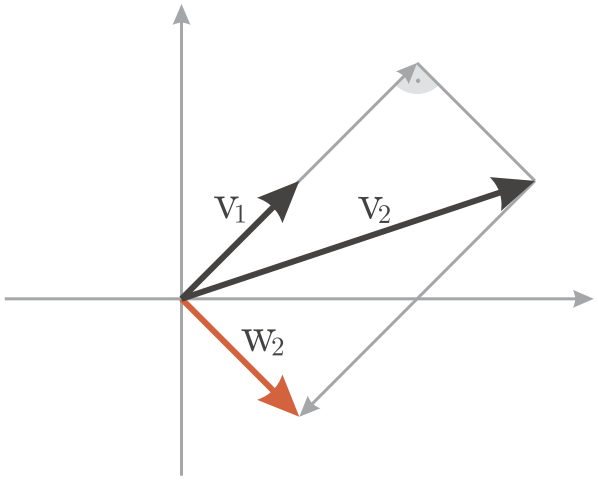


Abbildung 6.1: Gram-Schmidt-Verfahren und Projektion

Lemma 6.3.6 *Ist* $(w_1, ..., w_r)$ *eine Orthogonalbasis von* L*, dann gilt*

$$d(L) = \|w_1\| \cdot ... \cdot \|w_r\| .$$

Beweis. Ist $(w_1, ..., w_r)$ eine Orthogonalbasis von L, dann

$$\begin{aligned} \det(X^t \cdot X) &= \det \begin{pmatrix} \langle w_1, w_1 \rangle & \cdots & \langle w_1, w_r \rangle \\ \vdots & & \vdots \\ \langle w_r, w_1 \rangle & \cdots & \langle w_r, w_r \rangle \end{pmatrix} \\ &= \det \begin{pmatrix} \langle w_1, w_1 \rangle & & 0 \\ & \ddots & \\ 0 & & \langle w_r, w_r \rangle \end{pmatrix} = \|w_1\|^2 \cdot ... \cdot \|w_r\|^2 \geq 0 \end{aligned}$$

■

Satz 6.3.7 (Hadamard Ungleichung) *Sei* $L \subset \mathbb{R}^n$ *ein Gitter mit Basis* $v_1, ..., v_r$*. Dann gilt*

$$d(L) \leq \|v_1\| \cdot ... \cdot \|v_r\| .$$

Beweis. Sei $(w_1, ..., w_r)$ die mit Algorithmus 6.1 aus $(v_1, ..., v_r)$ konstruierte Orthogonalbasis. Dann ist mit Lemma 6.3.6

$$d(L) = \|w_1\| \cdot ... \cdot \|w_r\| .$$

Außerdem werden unter Projektion Vektoren höchstens kürzer, denn

$$\begin{aligned} \|w_i\|^2 &= \left\langle v_i - \sum_{s=1}^{i-1} \frac{\langle w_s, v_i \rangle}{\langle w_s, w_s \rangle} \cdot w_s,\ v_i - \sum_{t=1}^{i-1} \frac{\langle w_t, v_i \rangle}{\langle w_t, w_t \rangle} \cdot w_t \right\rangle \\ &= \langle v_i, v_i \rangle - 2 \sum_{s=1}^{i-1} \frac{\langle w_s, v_i \rangle^2}{\langle w_s, w_s \rangle} + \sum_{s=1}^{i-1} \frac{\langle w_s, v_i \rangle^2}{\langle w_s, w_s \rangle} \\ &= \|v_i\|^2 - \sum_{s=1}^{i-1} \left(\frac{\langle w_s, v_i \rangle}{\|w_s\|} \right)^2 \end{aligned}$$

also

$$\|v_i\|^2 \geq \|w_i\|^2.$$

■

Beispiel 6.3.8 *In Beispiel 6.3.5 gilt*

$$\left|\det\begin{pmatrix} 1 & 3 \\ 1 & 1 \end{pmatrix}\right| = 2 = \left|\det\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\right| = \sqrt{2}\cdot\sqrt{2} \leq \sqrt{2}\cdot\sqrt{10}$$

Für eine Orthogonalbasis $v_1, ..., v_r$ nimmt also $\|v_1\| \cdot ... \cdot \|v_r\|$ den minimal möglichen Wert an. Allerdings löst das unser Problem nicht, denn das Gram-Schmidt-Verfahren liefert im Allgemeinen auch bei ganzzahligem Input keinen ganzzahligen Output.

6.4 Ganzzahlige Gram-Schmidt-Reduktion

Als Basis von Algorithmen zur Bestimmung von kurzen Vektoren in Gittern verwendet man eine gerundete Version des Gram-Schmidt-Verfahrens. Gegeben zwei Vektoren v_1, v_2 in dem Gitter L können wir v_2 um Vielfache von v_1 abändern, um einen möglichst kurzen Vektor zu erhalten. Wir betrachten also

$$v_2' = v_2 + \lambda \cdot v_1$$

mit $\lambda \in \mathbb{R}$ und die Längendifferenz-Parabel

$$f(\lambda) = \|v_2'\|^2 - \|v_2\|^2 = 2 \cdot \langle v_2, v_1 \rangle \cdot \lambda + \lambda^2 \cdot \|v_1\|^2$$

Das Minimum dieser Parabel liegt bei

$$\lambda_0 = -\frac{\langle v_2, v_1 \rangle}{\|v_1\|^2}$$

zwischen den beiden Nullstellen $\lambda = 0$ und $\lambda = -2 \cdot \frac{\langle v_2, v_1 \rangle}{\|v_1\|^2}$. Wir erhalten also den kürzest möglichen Vektor v_2' für den im Gram-Schmidt-Verfahren gewählten Wert, siehe Abbildung 6.1. Im Allgemeinen wird $\lambda_0 \in \mathbb{R}$ aber keine ganze Zahl sein. Die Idee ist nun λ_0 auf die nächste ganze Zahl $\lfloor \lambda_0 \rceil$ zu runden.

Notation 6.4.1 *Für $x \in \mathbb{R}$ sei*

$$\lfloor x \rceil := \begin{cases} \lfloor x + \frac{1}{2} \rfloor & \text{für } x \geq 0 \\ \lfloor x - \frac{1}{2} \rfloor & \text{für } x < 0 \end{cases}$$

die übliche Rundung auf die nächste ganze Zahl.

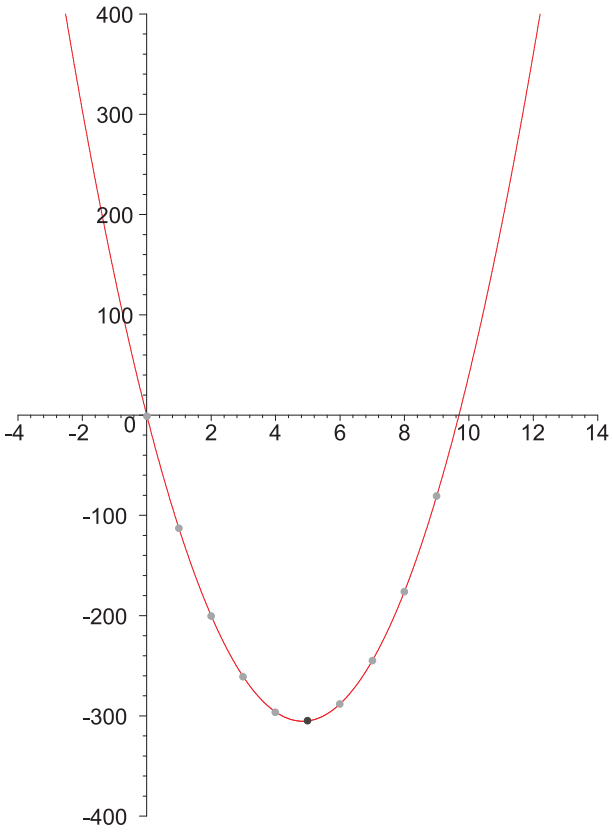


Abbildung 6.2: Längendifferenz vor Reduktion

Beispiel 6.4.2 *Für*

$$v_1 = \begin{pmatrix} 2 \\ 3 \end{pmatrix} \qquad v_2 = \begin{pmatrix} -33 \\ 1 \end{pmatrix}$$

haben wir

$$f(\lambda) = 2 \cdot (-63) \cdot \lambda + 13 \cdot \lambda^2,$$

siehe Abbildung 6.2, Hier ist also

$$\lambda_0 = \frac{63}{13} \approx 4.8$$

und für eine ganzzahlige Reduktion runden wir auf

$$\lfloor \lambda_0 \rceil = 5.$$

Wir wir gleich sehen werden, führt dies zu dem kürzest möglichen Vektor der Form $v_2 + \lambda \cdot v_1$ im Gitter.

Definition 6.4.3 *Sei $(v_1, ..., v_n)$ eine Basis des Gitters $L \subset \mathbb{R}^n$ und*

$$w_1 = v_1$$
$$w_i = v_i - \sum_{s=1}^{i-1} \frac{\langle w_s, v_i \rangle}{\|w_s\|^2} \cdot w_s$$

für $i = 2, ..., n$ die Gram-Schmidt-Basis.

Die Basis $(v_1, ..., v_n)$ heißt ***Gram-Schmidt-reduziert*** *(oder* ***längenreduziert****) wenn*

$$\frac{|\langle w_s, v_i\rangle|}{\|w_s\|^2} \leq \frac{1}{2}$$

für alle $1 \leq s < i \leq n$.

Gram-Schmidt-Reduziertheit können wir durch iteratives Anwenden von Algorithmus 6.2 erreichen.

Algorithmus 6.2 Reduziere v_i nach v_s für $1 \leq s < i \leq n$

Input: Basis $(v_1, ..., v_n)$ des Gitters $L \subset \mathbb{Q}^n$.
Output: Basis $(v_1, ..., v_n)$ von L sodass

$$\left|\frac{\langle w_s, v_i\rangle}{\|w_s\|^2}\right| \leq \frac{1}{2}$$

wobei $(w_1, ..., w_n)$ die aus $(v_1, ..., v_n)$ bestimmte Gram-Schmidt-Basis bezeichnet.

1: $\mu = \frac{\langle w_s, v_i\rangle}{\|w_s\|^2}$
2: **if** $|\mu| > \frac{1}{2}$ **then**
3: $\quad v_i = v_i - \lfloor \mu \rceil \cdot v_s$
4: **return** $(v_1, ..., v_n)$

Wir zeigen die Korrektheit von Algorithmus 6.2.

Beweis. Beachte, dass nach Konstruktion der w_i gilt

$$\langle w_s, v_s\rangle = \left\langle w_s, w_s + \sum_{t=1}^{s-1} \frac{\langle w_t, v_s\rangle}{\|w_t\|^2} \cdot w_t \right\rangle = \langle w_s, w_s\rangle .$$

Somit ist

$$\frac{\langle w_s, v_i - \lfloor \mu \rceil \cdot v_s\rangle}{\|w_s\|^2} = \mu - \lfloor \mu \rceil \cdot \frac{\langle w_s, v_s\rangle}{\|w_s\|^2} = \mu - \lfloor \mu \rceil$$

■

Bemerkung 6.4.4 *Nach der Reduktion von $v_2 \in \mathbb{R}^n$ nach $v_1 \in \mathbb{R}^n$ gilt für jedes $v_2' = v_2 + \lambda \cdot v_1$ mit $\lambda \in \mathbb{Z}$, dass*

$$\|v_2'\| \geq \|v_2\| .$$

Beweis. Nach Algorithmus 6.2 haben wir

$$\frac{|\langle v_2, v_1\rangle|}{\|v_1\|^2} \leq \frac{1}{2}$$

Angenommen

$$\|v_2\|^2 > \|v_2'\|^2 = \|v_2\|^2 + 2 \cdot \langle v_2, v_1 \rangle \cdot \lambda + \lambda^2 \cdot \|v_1\|^2$$

äquivalent

$$\lambda^2 < -2 \cdot \frac{\langle v_2, v_1 \rangle}{\|v_1\|^2} \cdot \lambda.$$

Dann ist

$$|\lambda| < 2 \frac{|\langle v_2, v_1 \rangle|}{\|v_1\|^2} \leq 1$$

also $\lambda = 0$ und somit $\|v_2'\| = \|v_2\|$, ein Widerspruch. ■

Nach der Reduktion von v_2 nach v_1 liegt also zwischen den Nullstellen $\lambda = 0$ und $\lambda = -2 \cdot \frac{\langle v_2, v_1 \rangle}{\|v_1\|^2}$ der Längendifferenz-Parabel

$$f(\lambda) = \|v_2'\|^2 - \|v_2\|^2 = 2 \cdot \langle v_2, v_1 \rangle \cdot \lambda + \lambda^2 \cdot \|v_1\|^2$$

keine ganze Zahl mehr.

Beispiel 6.4.5 *Nach der Reduktion der Vektoren in Beispiel 6.4.2 ist*

$$v_1 = \begin{pmatrix} 2 \\ 3 \end{pmatrix} \qquad v_2 = \begin{pmatrix} -33 \\ 1 \end{pmatrix} - \left\lfloor -\tfrac{63}{13} \right\rceil \cdot \begin{pmatrix} 2 \\ 3 \end{pmatrix} = \begin{pmatrix} -23 \\ 16 \end{pmatrix}$$

also

$$f(\lambda) = 2 \cdot 2 \cdot \lambda + 13 \cdot \lambda^2,$$

siehe Abbildung 6.3. Der Vektor v_2 lässt sich also durch Addition von ganzzahligen Vielfachen von v_1 nicht mehr verkürzen.

6.5 Gauß-Lagrange-Algorithmus

Wir konzentrieren uns zunächst auf den Fall $n = 2$. Iteratives Anwenden der ganzzahligen Gram-Schmidt-Reduktion gibt Gauß-Lagrange Algorithmus 6.3, der einen kürzesten Vektor in einem Gitter bestimmt.

Algorithmus 6.3 Gauß-Lagrange

Input: Basis (v_1, v_2) des Gitters $L \subset \mathbb{Q}^2$.
Output: Basis (v_1, v_2), sodass v_1 ein kürzester Vektor in L ist.

1: **reduziere** v_2 nach v_1.
2: **if** $\|v_2\| < \|v_1\|$ **then**
3: **vertausche** v_1 und v_2
4: **goto** 1
5: **return** (v_1, v_2)

Für den Beweis der Terminierung und Korrektheit von Algorithmus 6.3 siehe Übungsaufgabe 6.1.

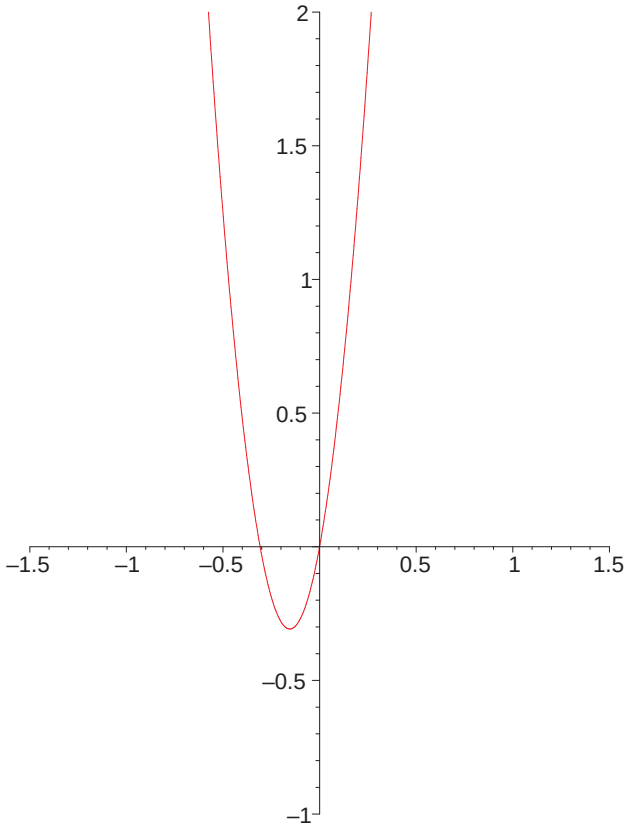


Abbildung 6.3: Längendifferenz nach Reduktion

Beispiel 6.5.1 *Wir wenden den Gauß-Lagrange-Algorithmus auf die Gitterbasis*

$$v_1 = \begin{pmatrix} 101 \\ 0 \end{pmatrix} \qquad v_2 = \begin{pmatrix} 68 \\ 1 \end{pmatrix}$$

aus Beispiel 6.2.1 an:
Reduktion:

$$v_1 = \begin{pmatrix} 101 \\ 0 \end{pmatrix} \qquad v_2 = \begin{pmatrix} 68 \\ 1 \end{pmatrix} - \left\lfloor \tfrac{68}{101} \right\rceil \cdot \begin{pmatrix} 101 \\ 0 \end{pmatrix} = \begin{pmatrix} -33 \\ 1 \end{pmatrix}$$

Es ist

$$\|v_1\|^2 = 101^2 > 1090 = \|v_2\|^2$$

also vertauschen wir:

$$v_1 = \begin{pmatrix} -33 \\ 1 \end{pmatrix} \qquad v_2 = \begin{pmatrix} 101 \\ 0 \end{pmatrix}$$

Reduktion:

$$v_1 = \begin{pmatrix} -33 \\ 1 \end{pmatrix} \qquad v_2 = \begin{pmatrix} 101 \\ 0 \end{pmatrix} + \left\lfloor \tfrac{3333}{1090} \right\rceil \cdot \begin{pmatrix} -33 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$$

Somit

$$\|v_1\|^2 = 1090 > 13 = \|v_2\|^2$$

vertausche also:

$$v_1 = \begin{pmatrix} 2 \\ 3 \end{pmatrix} \qquad v_2 = \begin{pmatrix} -33 \\ 1 \end{pmatrix}$$

Reduktion:

$$v_1 = \begin{pmatrix} 2 \\ 3 \end{pmatrix} \qquad v_2 = \begin{pmatrix} -33 \\ 1 \end{pmatrix} - \left\lfloor -\tfrac{63}{13} \right\rceil \cdot \begin{pmatrix} 2 \\ 3 \end{pmatrix} = \begin{pmatrix} -23 \\ 16 \end{pmatrix}$$

Nun ist

$$\|v_1\|^2 = 13 \leq 785 = \|v_2\|^2,$$

der Algorithmus terminiert also mit

$$v_1 = \begin{pmatrix} 2 \\ 3 \end{pmatrix} \qquad v_2 = \begin{pmatrix} -23 \\ 16 \end{pmatrix}$$

und v_1 ist ein kürzester Vektor im Gitter.

Für weitere Beispiele und die Implementierung des Gauß-Lagrange-Algorithmus siehe die Übungen 6.7 und 6.6.

6.6 Anwendungen von kurzen Vektoren in höher Dimension

Bevor wir nun den Gauß-Algorithmus auf Gitter in höhere Dimension verallgemeinern, diskutieren wir einige Anwendungen.

Beispiel 6.6.1 *Allgemein benötigt man kurze Vektoren in Gittern in der*

- *linearen Algebra über $\mathbb{Z}$, z.B. die Konstruktion einer kurzen Basis von* $\mathrm{Bild}(A)$ *für $A \in \mathbb{Z}^{n \times m}$,*
- *der algebraischen Zahlentheorie,*
- *bei der Suche nach dichtesten Kugelpackungen. Beispielsweise ist die effizienteste Packung von Kreisen in der Ebene die hexagonale Packung, siehe Abbildung 6.4.*
- *in der Kryptanalyse, d.h. beim Mitlesen von verschlüsselten Nachrichten. Mit Hilfe von LLL wurden verschiedene Public-Key-Kryptosysteme gebrochen. Ein prominentes Beispiel ist das Merkle-Hellman Kryptosystem, das auf dem Rucksackproblem basiert (wie kann man einen Rucksack mit Gegenständen verschiedenen Wertes und Gewichts packen, sodass der Inhalt maximalen Wert hat bei festgelegtem Maximalgewicht).*

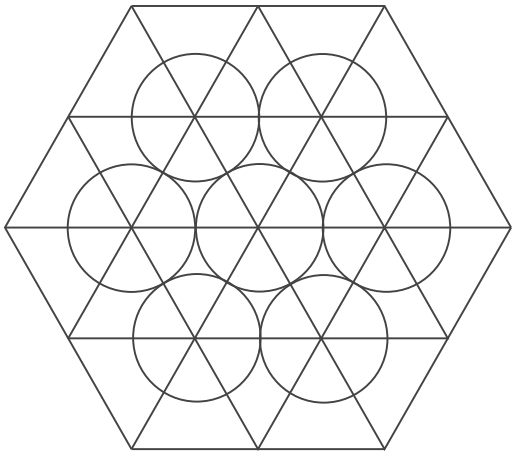


Abbildung 6.4: Hexagonale Kugelpackung

- *bei der Faktorisierung: Zum Beispiel wurde mit Hilfe von LLL ein Produkt von zwei Primzahlen $N = p \cdot q$ mit 232 Stellen aus der RSA-Factoring-Challange zerlegt (d.h. jeder RSA-Schlüssel, der N verwendet gebrochen, siehe Abschnitt 2.7). Faktorisierung (von Polynomen) war auch die ursprüngliche Motivation von Lovász, Lenstra und Lenstra. Wie man LLL dazu verwenden kann, werden wir im nächsten Kapitel über Faktorisierung sehen.*

Wir gehen noch auf zwei konkrete Anwendungen ein:

Beispiel 6.6.2 *Konstruktion von Minimalpolynomen für algebraische Zahlen: Nehmen wir an, wir haben (als Ergebnis eines gegebenen Algorithmus) z.B. die Zahl*

$$a = 2.732$$

numerisch berechnet, und gehen davon aus, dass a eine Näherung für eine Nullstelle eines Polynoms $f \in \mathbb{Z}[x]$ vom Grad d sein sollte. Wie können wir f finden? Wähle ein der Rechengenauigkeit angepasstes $\lambda = 10^n$ und bestimme einen kurzen Vektor

$$\begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_d \\ c_{d+1} \end{pmatrix} \in \left\langle \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ \lambda \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ \lambda \cdot a \end{pmatrix}, \ldots, \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ \lambda \cdot a^d \end{pmatrix} \right\rangle$$

Dann gilt

$$\lambda \left(c_0 + c_1 a + \ldots + c_d a^d \right) = c_{d+1}.$$

Für großes λ und kleines c_{d+1} haben wir also

$$c_0 + c_1 a + \ldots + c_d a^d \approx 0,$$

das heißt wir können vermuten, dass a eine Nullstelle des Polynoms

$$f = \sum_{i=0}^{d} c_i t^i \in \mathbb{Z}[t]$$

ist. Diese Vermutung lässt sich nun eventuell in dem jeweiligen Kontext verifizieren.
In JULIA/NEMO *für unser Beispiel mit* $\lambda = 10^3$*:*

```
using Nemo
S = MatrixSpace(ZZ, 3, 4);
A = S([1 0 0 1000; 0 1 0 2732; 0 0 1 7463]);
lll(A)
[[-2, -2, 1, -1],
[27, -29, 7, 13],
[32, -9, -1, -51]
```

Somit ist

$$10^3 \cdot (a^2 - 2a - 2) = -1$$

und damit sollte a eine Nullstelle des Polynoms

$$f = x^2 - 2x - 2$$

sein, also $a = 1 \pm \sqrt{3}$.

Numerisches Auswerten der Nullstellen von f

```
1+sqrt(3);
2.732050807568877
evalf(1-sqrt(3));
-.7320508075688772
```

führt zu der Vermutung, dass

$$a = 1 + \sqrt{3}.$$

Siehe auch Übung 6.3.

Beispiel 6.6.3 *Konstruktion von Potenzreihenformeln für transzendente Zahlen: Mit Hilfe von LLL wurde z.B. die Darstellung*

$$\pi = \sum_{k=0}^{\infty} \frac{1}{16^k} \left(\frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6} \right)$$

gefunden, die wesentlich schneller gegen π konvergiert als etwa die Leibnizreihe

$$\pi = \sum_{k=0}^{\infty} (-1)^k \frac{4}{2k+1}$$

In MAPLE *können wir die Formel auswerten mit:*

```
s:=1/16^k*(4/(8*k+1)-2/(8*k+4)-1/(8*k+5)-1/(8*k+6);
evalf(sum(s,k=0..5));
3.141592653
evalf(Pi);
3.141592654
evalf(sum(4*(-1)^k/(2*k+1),k=0..5));
2.976046176
```

6.7 LLL-Algorithmus

Der *LLL*-Algorithmus verallgemeinert den Ansatz des Gauß-Lagrange-Algorithmus für $n \geq 3$. Das zentrale Problem, das von Lovász, Lenstra und Lenstra gelöst wurde, ist, eine vernünftige Notation einer kurzen Basis zu finden, die es auch erlaubt eine solche Basis effizient zu berechnen. Diese Eigenschaft wird über das Verhalten des Gram-Schmidt-Orthogonalisierungsverfahrens gemessen. Im Längenvergleich führt man einen Parameter $\frac{1}{4} < \delta \leq 1$ ein, der es zulässt, dass der erste Vektor in der Basis zwar kurz, aber in wenigen Fällen nicht unbedingt ein kürzester Vektor im Gitter ist.

Definition 6.7.1 *Sei $(v_1, ..., v_n)$ eine Basis des Gitters $L \subset \mathbb{R}^n$ und*

$$w_1 = v_1$$
$$w_i = v_i - \sum_{s=1}^{i-1} \frac{\langle w_s, v_i \rangle}{\|w_s\|^2} \cdot w_s$$

für $i = 2, ..., n$ die Gram-Schmidt-Basis.

*Die Basis $(v_1, ..., v_n)$ heißt δ-**LLL-reduziert** für $\frac{1}{4} < \delta \leq 1$ wenn sie*

*1) **Gram-Schmidt-reduziert** ist, das heißt*

$$\frac{|\langle w_s, v_i \rangle|}{\|w_s\|^2} \leq \frac{1}{2}$$

für alle $1 \leq s < i \leq n$ und

*2) die δ-**Lovász-Bedingung** erfüllt, das heißt*

$$\left(\delta - \left(\frac{\langle w_{i-1}, v_i \rangle}{\|w_{i-1}\|^2}\right)^2\right) \cdot \|w_{i-1}\|^2 \leq \|w_i\|^2$$

für alle $2 \leq i \leq n$.

Man beachte, dass falls die v_i Gram-Schmidt-reduziert sind $0 < \delta - \left(\frac{\langle w_{i-1}, v_i \rangle}{\|w_{i-1}\|^2}\right)^2 \leq 1$ gilt.

Bemerkung 6.7.2 *Für zwei Vektoren v_1, v_2 ist die Lovász-Bedingung äquivalent zu*

$$\left(\delta - \left(\frac{\langle v_1, v_2 \rangle}{\|v_1\|^2}\right)^2\right) \cdot \|v_1\|^2 \leq \|w_2\|^2 = \|v_2\|^2 - 2\frac{\langle v_1, v_2 \rangle^2}{\|v_1\|^2} + \frac{\langle v_1, v_2 \rangle^2}{\|v_1\|^4} \|v_1\|^2$$

also zu

$$\delta \cdot \|v_1\|^2 - \frac{\langle v_1, v_2 \rangle^2}{\|v_1\|^2} \leq \|v_2\|^2 - \frac{\langle v_1, v_2 \rangle^2}{\|v_1\|^2}$$

d.h.

$$\delta \cdot \|v_1\|^2 \leq \|v_2\|^2 .$$

Bemerkung 6.7.3 *1) Die Lovász-Bedingung ist trivialerweise erfüllt, wenn*

$$\|w_{i-1}\|^2 \leq \|w_i\|^2$$

denn $\delta - \left(\frac{\langle w_{i-1}, v_i\rangle}{\|w_{i-1}\|^2}\right)^2 < 1$.

2) Die Lovász-Bedingung ist äquivalent zu

$$\delta \cdot \|w_{i-1}\|^2 \leq \left\| w_i + \frac{\langle w_{i-1}, v_i\rangle}{\|w_{i-1}\|^2} w_{i-1} \right\|^2$$

denn w_{i-1} *und* w_i *sind orthogonal.*

3) Damit können wir die Änderung der Länge des $(i-1)$*-ten Gram-Schmidt-Vektors unter Vertauschung von* v_{i-1} *und* v_i *beschreiben: Vor dem Vertauschen gilt für den* i*-ten Gram-Schmidt-Vektor*

$$w_i = v_i - \sum_{s=1}^{i-2} \frac{\langle w_s, v_i\rangle}{\|w_s\|^2} \cdot w_s - \frac{\langle w_{i-1}, v_i\rangle}{\|w_{i-1}\|^2} \cdot w_{i-1}$$

und damit nach dem Vertauschen für den $(i-1)$*-ten Gram-Schmidt-Vektors, dass*

$$\begin{aligned} w'_{i-1} &= v_i - \sum_{s=1}^{i-2} \frac{\langle w_s, v_i\rangle}{\|w_s\|^2} \cdot w_s \\ &= w_i + \frac{\langle w_{i-1}, v_i\rangle}{\|w_{i-1}\|^2} \cdot w_{i-1} \end{aligned}$$

also können wir die Lovász-Bedingung als

$$\frac{\|w'_{i-1}\|^2}{\|w_{i-1}\|^2} \geq \delta$$

umformulieren.

4) Die Idee im LLL-Algorithmus zur Berechnung einer δ*-LLL-reduzierten Basis ist, eine ganzzahlige Gram-Schmidt-Reduktion durchzuführen und dann* v_{i-1} *und* v_i *zu vertauschen, wenn*

$$\frac{\|w'_{i-1}\|^2}{\|w_{i-1}\|^2} < \delta$$

(für ein i*). Dies stellt sicher, dass dieser Quotient um den Faktor* δ *kleiner wird und der Algorithmus terminiert.*

5) In vielen Programmen ist der LLL-Algorithmus für $\delta = \frac{3}{4}$ *implementiert. Er verhält sich aber sehr ähnlich für alle*

$$\frac{1}{4} < \delta < 1$$

(jedoch nicht für $\delta = 1$ *wie wir sehen werden).*

6) Für $\delta = 1$ und $n = 2$ ist der LLL-Algorithmus wegen Bemerkung 6.7.2 genau der Gauß-Lagrange-Algorithmus.

Beispiel 6.7.4 *Für*

$$v_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \qquad v_2 = \begin{pmatrix} 4 \\ 1 \end{pmatrix}$$

gibt das Gram-Schmidt-Verfahren

$$w_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \qquad w_2 = \begin{pmatrix} 4 \\ 1 \end{pmatrix} - \tfrac{5}{2}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{3}{2} \\ -\frac{3}{2} \end{pmatrix}$$

Wegen

$$\left| \frac{\langle w_1, v_2 \rangle}{\|w_1\|^2} \right| = \frac{5}{2} > \frac{1}{2}$$

ist (v_1, v_2) also nicht LLL-reduziert.

Mit der Hermite-Normalform sehen wir, dass auch

$$v_1' = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \qquad v_2' = \begin{pmatrix} 1 \\ -2 \end{pmatrix}$$

eine Basis von $\langle v_1, v_2 \rangle$ bilden, denn

$$\mathrm{HNF}\begin{pmatrix} 1 & 4 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 1 & 3 \end{pmatrix} = \mathrm{HNF}\begin{pmatrix} 1 & 1 \\ 1 & -2 \end{pmatrix}$$

Anwendung des Gram-Schmidt-Verfahrens auf (v_1', v_2') liefert

$$w_1' = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \qquad w_2' = \begin{pmatrix} 1 \\ -2 \end{pmatrix} + \tfrac{1}{2}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{3}{2} \\ -\frac{3}{2} \end{pmatrix}$$

Dafür gilt

$$\left| \frac{\langle w_1', v_2' \rangle}{\|w_1'\|^2} \right| = \frac{1}{2} \leq \frac{1}{2}$$

und

$$\|w_1'\|^2 = 2 < \frac{9}{2} = \|w_2'\|^2$$

Die Basis (v_1', v_2') ist also LLL-reduziert für jedes δ.
In Julia/Nemo *berechnen wir für $\delta = \frac{3}{4}$ aus (v_1, v_2) eine LLL-reduzierte Basis mit:*

```
using Nemo
S = MatrixSpace(ZZ, 2, 2);
A = S([1 1; 4 1]);
lll(A, lll_ctx(0.75, 0.5))
[1,1]
[1,-2]
```

Der zweite Parameter 0.5 bezieht sich hier auf die Schranke in der Definition von Gram-Schmidt-Reduziertheit $\frac{|\langle w_s, v_i \rangle|}{\|w_s\|^2} \leq \frac{1}{2}$. Die Modifikation dieses Parameters ist sinnvoll, wenn man nicht exakt, sondern mit Fließkommazahlen rechnet.

Beispiel 6.7.5 *Die Basis*

$$(v_1'', v_2'') = (v_2', v_1') = \left(\begin{pmatrix} 1 \\ -2 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right)$$

ist dagegen nicht LLL-reduziert für $\delta = \frac{3}{4}$*: Das Gram-Schmidt-Verfahren liefert*

$$w_1'' = \begin{pmatrix} 1 \\ -2 \end{pmatrix} \qquad w_2'' = \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \frac{1}{5} \begin{pmatrix} 1 \\ -2 \end{pmatrix} = \begin{pmatrix} \frac{6}{5} \\ \frac{3}{5} \end{pmatrix}$$

Wir haben damit zwar

$$\frac{|\langle w_1'', v_2'' \rangle|}{\|w_1''\|^2} = \frac{1}{5} \leq \frac{1}{2}$$

d.h. (v_1'', v_2'') *ist Gram-Schmidt-reduziert, aber mit*

$$\|w_1''\|^2 = 5$$
$$\|w_2''\|^2 = 2,$$

ist die δ*-Lovász-Bedingung äquivalent zu*

$$5\delta - \frac{1}{5} = \left(\delta - \frac{1}{25}\right) \cdot \|w_1'\|^2 \leq 2,$$

d.h. sie ist erfüllt für $\delta \leq \frac{11}{25}$ *und nicht erfüllt für* $\delta > \frac{11}{25}$*.*

Bevor wir den Algorithmus zur Bestimmung einer LLL-reduzierten Basis beschreiben, untersuchen wir die grundlegenden Eigenschaften einer solchen Basis. Die wesentliche Beobachtung ist, dass für eine LLL-reduzierte Basis das Maß $\|v_1\| \cdot \ldots \cdot \|v_n\|$ für die Länge der Basis nicht wesentlich größer sein kann als die Invariante $d(L)$:

Satz 6.7.6 *Sei* $(v_1, \ldots, v_n)$ *eine* δ*-LLL-reduzierte Basis des Gitters* $L \subset \mathbb{R}^n$ *und* $(w_1, \ldots, w_n)$ *die Gram-Schmidt-Basis und*

$$\tau = \frac{4}{4\delta - 1}$$

Dann gilt:

1)

$$d(L) \leq \|v_1\| \cdot \ldots \cdot \|v_n\| \leq \tau^{\frac{n(n-1)}{4}} \cdot d(L)$$

2) Für $1 \leq j \leq i \leq n$ *ist*

$$\|v_j\| \leq \tau^{\frac{i-1}{2}} \|w_i\|$$

3)

$$\|v_1\| \leq \tau^{\frac{n-1}{4}} \cdot \sqrt[n]{d(L)}$$

4) Für jedes $0 \neq v \in L$ *ist*

$$\|v_1\| \leq \tau^{\frac{n-1}{2}} \cdot \|v\|$$

5) Für jede linear unabhängige Familie von Vektoren $y_1, ..., y_l \in L$ *ist*

$$\|v_j\| \leq \tau^{\frac{n-1}{2}} \cdot \max\{\|y_1\|, ..., \|y_l\|\}$$

für alle $j = 1, ..., l$.

Beweis.

1) Die erste Ungleichung ist die Hadamard-Ungleichung aus Satz 6.3.7. Da $(v_1, ..., v_n)$ eine LLL-reduzierte Basis ist, gilt mit

$$\mu = \frac{\langle w_{i-1}, v_i \rangle}{\|w_{i-1}\|^2}$$

dass

$$\left(\delta - \frac{1}{4}\right) \cdot \|w_{i-1}\|^2 \leq \left(\delta - \mu^2\right) \cdot \|w_{i-1}\|^2 \leq \|w_i\|^2$$

für alle $2 \leq i \leq n$. Die erste Ungleichung gilt wegen $|\mu| \leq \frac{1}{2}$. Mit Induktion folgt damit

$$\|w_j\|^2 \leq \left(\frac{4}{4\delta - 1}\right)^{i-j} \cdot \|w_i\|^2$$

für alle $j \leq i$. Dies impliziert zusammen mit der Gram-Schmidt-Reduziertheit der v_i, dass

$$\begin{aligned}
\|v_i\|^2 &= \|w_i\|^2 + \sum_{s=1}^{i-1} \frac{\langle w_s, v_i \rangle^2}{\|w_s\|^4} \cdot \|w_s\|^2 \\
&\leq \|w_i\|^2 \cdot \left(1 + \sum_{s=1}^{i-1} \frac{1}{2^2} \cdot \left(\frac{4}{4\delta - 1}\right)^{i-s}\right) \\
&= \|w_i\|^2 \cdot \left(\frac{4 - 4\delta}{5 - 4\delta} + \frac{1}{5 - 4\delta}\left(\frac{4}{4\delta - 1}\right)^{i-1}\right)
\end{aligned}$$

also

$$\begin{aligned}
\|v_1\| \cdot ... \cdot \|v_n\| &\leq \sqrt{\prod_{i=1}^{n} \left(\frac{4}{4\delta - 1}\right)^{i-1}} \cdot \|w_1\| \cdot ... \cdot \|w_n\| \\
&= \left(\frac{4}{4\delta - 1}\right)^{\frac{n(n-1)}{4}} \cdot |\det(v_1 \mid ... \mid v_n)| .
\end{aligned}$$

Hier verwenden wir, dass wegen $\frac{4\delta - 1}{4} \leq 1$

$$(4 - 4\delta)\left(\frac{4\delta - 1}{4}\right)^{i-1} + 1 \leq 5 - 4\delta$$

und somit die obige Ungleichung mit

$$\frac{1}{5-4\delta}\left(4-4\delta+\left(\frac{4}{4\delta-1}\right)^{i-1}\right)\leq\left(\frac{4}{4\delta-1}\right)^{i-1}$$

folgt. Des Weiteren erhalten wir die Gleichheit mit

$$\begin{aligned}\|w_1\|\cdot\ldots\cdot\|w_n\| &= |\det(w_1\mid\ldots\mid w_n)|\\ &= |\det(v_1\mid\ldots\mid v_n)|\end{aligned}$$

(siehe den Beweis von Lemma 6.3.6).

2) Mit den Ungleichungen aus dem Beweis von 1) gilt für $1\leq j\leq i\leq n$ dass

$$\begin{aligned}\|v_j\|^2 &\leq \|w_j\|^2\cdot\left(\frac{4-4\delta}{5-4\delta}+\frac{1}{5-4\delta}\left(\frac{4}{4\delta-1}\right)^{j-1}\right)\\ &\leq \|w_i\|^2\cdot\left(\frac{4-4\delta}{5-4\delta}+\frac{1}{5-4\delta}\left(\frac{4}{4\delta-1}\right)^{j-1}\right)\cdot\left(\frac{4}{4\delta-1}\right)^{i-j}\\ &\leq \|w_i\|^2\cdot\left(\frac{4}{4\delta-1}\right)^{i-1}\end{aligned}$$

3) Mit 2) folgt für alle i

$$\|v_1\|^2\leq\left(\frac{4}{4\delta-1}\right)^{i-1}\cdot\|w_i\|^2$$

also

$$\begin{aligned}\|v_1\|^{2n} &\leq \prod_{i=1}^{n}\left(\left(\frac{4}{4\delta-1}\right)^{i-1}\cdot\|w_i\|^2\right)\\ &= \left(\frac{4}{4\delta-1}\right)^{\frac{n(n-1)}{2}}\cdot\|w_1\|^2\cdot\ldots\cdot\|w_n\|^2\\ &= \left(\frac{4}{4\delta-1}\right)^{\frac{n(n-1)}{2}}\cdot|\det(v_1\mid\ldots\mid v_n)|^2\end{aligned}$$

Durch Ziehen der $2n$-ten Wurzel erhält man

$$\|v_1\|\leq\left(\frac{4}{4\delta-1}\right)^{\frac{n-1}{4}}\cdot|\det(v_1\mid\ldots\mid v_n)|^{\frac{1}{n}}$$

4) Sei i minimal mit

$$v\in\langle v_1,\ldots,v_i\rangle$$

Wegen $\langle v_1,\ldots,v_i\rangle=\langle w_1,\ldots,w_i\rangle$ gibt es $a_j\in\mathbb{Z}$ und $r_j\in\mathbb{R}$ mit

$$v=\sum_{j=1}^{i}a_jv_j=\sum_{j=1}^{i}r_jw_j$$

also

$$\begin{aligned} r_i &= \langle v, w_i \rangle = \textstyle\sum_{j=1}^{i} a_j \langle v_j, w_i \rangle \\ &= \textstyle\sum_{j=1}^{i} a_j \left\langle w_j + \sum_{s=1}^{j-1} \frac{\langle w_s, v_j \rangle}{\|w_s\|^2} \cdot w_s, w_i \right\rangle \\ &= a_i \end{aligned}$$

Nach Wahl von i ist $a_i = r_i \neq 0$. Somit

$$\begin{aligned} \|v\|^2 &= \textstyle\sum_{j=1}^{i} r_j^2 \|w_j\|^2 \geq r_i^2 \|w_i\|^2 = a_i^2 \|w_i\|^2 \\ &\geq \|w_i\|^2 \geq \left(\frac{4}{4\delta - 1}\right)^{1-i} \|v_1\|^2 \geq \left(\frac{4}{4\delta - 1}\right)^{1-n} \|v_1\|^2 \end{aligned}$$

mit 2).

5) Folgt aus 4), siehe Übung 6.5.

■

Bemerkung 6.7.7 *Für den Standardwert $\delta = \frac{3}{4}$ haben wir $\tau = 2$ und für $\delta = 1$ ist $\tau = \frac{4}{3}$. Aufgrund der durchgeführten Abschätzungen zeigt Satz 6.7.6 also nicht die volle Stärke des Gauß-Lagrange-Algorithmus.*

Der *LLL*-Algorithmus in seiner einfachsten Form funktioniert wie in Algorithmus 6.4 beschrieben.

Algorithmus 6.4 LLL

Input: Basis $(v_1, ..., v_n)$ des Gitters $L \subset \mathbb{Q}^n$ und $\frac{1}{4} < \delta \leq 1$.
Output: *LLL*-reduzierte Basis $(v_1, ..., v_n)$.

1: Berechne die Gram-Schmidt-Basis $w_1, ..., w_n$ aus $v_1, ..., v_n$.
2: **for** $i = 2, ..., n$ **do**
3: **for** $s = i - 1, ..., 1$ **do**
4: **reduziere** v_i nach v_s
5: **if** $\exists i$ mit $\left(\delta - \left(\frac{\langle w_{i-1}, v_i \rangle}{\|w_{i-1}\|^2}\right)^2\right) \cdot \|w_{i-1}\|^2 > \|w_i\|^2$ **then**
6: **vertausche** v_{i-1} und v_i
7: **goto** 1
8: **return** $(v_1, ..., v_n)$

Beweis. Wenn der Algorithmus terminiert, dann erfüllt wegen des Vertauschungsschritts das Ergebnis offensichtlich die Lovász-Bedingung.

Wegen des Reduktionsschritts ist das Ergebnis Gram-Schmidt-reduziert. Wir müssen noch zeigen, dass die Reduktion $w_1, ..., w_n$ nicht ändert, diese also nur nach Vertauschungen aktualisiert werden müssen. Sei

$$v_i' = v_i + c \cdot v_j$$

mit $j < i$ und $c \in \mathbb{R}$. Dann ist mit Bemerkung 6.3.4 der aus v_i' bestimmte Gram-Schmidt-Vektor w_i' gleich dem aus v_i bestimmten Vektor w_i:

$$\begin{aligned} w_i' &= v_i' - \sum_{s=1}^{i-1} \frac{\langle w_s, v_i' \rangle}{\|w_s\|^2} \cdot w_s \\ &= v_i + c \cdot v_j - \sum_{s=1}^{i-1} \frac{\langle w_s, v_i \rangle}{\|w_s\|^2} \cdot w_s - c \cdot \underbrace{\sum_{s=1}^{i-1} \frac{\langle w_s, v_j \rangle}{\|w_s\|^2} \cdot w_s}_{v_j} \\ &= w_i \end{aligned}$$

Auf die Terminierung kommen wir noch in Satz 6.7.14 zurück. ■

Bemerkung 6.7.8 *Um den Algorithmus für $v_1, \ldots, v_n \in \mathbb{R}^n$ durchzuführen, rechnet man mit floating-point Approximationen und verwendet Methoden zur Kontrolle der Rundungsfehler.*

Den Reduktions- und Vertauschungsschritt können wir noch wie in Algorithmus 6.5 etwas effizienter anordnen.

Algorithmus 6.5 Verbesserter LLL

Input: Basis $(v_1, \ldots, v_n)$ des Gitters $L \subset \mathbb{Q}^n$ und $\frac{1}{4} < \delta \leq 1$.

Output: *LLL*-reduzierte Basis $(v_1, \ldots, v_n)$. Zu jedem Zeitpunkt sei $w_1, \ldots, w_n$ die aus $v_1, \ldots, v_n$ berechnete Gram-Schmidt-Basis.

1: $i = 2$
2: **reduziere** v_i nach v_{i-1}.
3: **if** $\left(\delta - \left(\frac{\langle w_{i-1}, v_i \rangle}{\|w_{i-1}\|^2}\right)^2\right) \cdot \|w_{i-1}\|^2 > \|w_i\|^2$ **then**
4: **vertausche** v_{i-1} und v_i
5: $i = i - 1$
6: **if** $i = 1$ **then**
7: **goto** 1 **else goto** 2
8: **for** $s = i - 2, \ldots, 1$ **do**
9: **reduziere** v_i nach v_s
10: $i = i + 1$
11: **if** $i > n$ **then**
12: **return** $(v_1, \ldots, v_n)$
13: **else**
14: **goto** 2

Hier bestimmt man natürlich nur die w_i, die im jeweiligen Schritt benötigt werden.

Beispiel 6.7.9 *Wir wenden den LLL-Algorithmus auf die Basis*

$$v_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \qquad v_2 = \begin{pmatrix} 4 \\ 1 \end{pmatrix}$$

aus Beispiel 6.7.4 an: Gram-Schmidt gibt

$$w_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \qquad w_2 = \begin{pmatrix} 4 \\ 1 \end{pmatrix} - \tfrac{5}{2}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{3}{2} \\ -\frac{3}{2} \end{pmatrix}$$

Ganzzahlige Reduktion liefert also

$$v_2 = \begin{pmatrix} 4 \\ 1 \end{pmatrix} - 3\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ -2 \end{pmatrix}$$

und (v_1, v_2) ist, wie schon oben überprüft, LLL-reduziert für beliebiges δ. Insbesondere stimmt das Ergebnis mit dem des Gauß-Lagrange-Algorithmus ($\delta = 1$) überein.

Beispiel 6.7.10 *Wir wenden den LLL-Algorithmus mit $\delta = \frac{3}{4}$ auf das Gitter erzeugt von*

$$v_1 = \begin{pmatrix} 101 \\ 0 \end{pmatrix} \qquad v_2 = \begin{pmatrix} 68 \\ 1 \end{pmatrix}$$

aus Beispiel 6.2.1 an.
Reduktion:

$$v_1 = \begin{pmatrix} 101 \\ 0 \end{pmatrix} \qquad v_2 = \begin{pmatrix} 68 \\ 1 \end{pmatrix} - \left\lfloor \tfrac{68}{101} \right\rceil \cdot \begin{pmatrix} 101 \\ 0 \end{pmatrix} = \begin{pmatrix} -33 \\ 1 \end{pmatrix}$$

Testen der Lovász-Bedingung:

$$\mu = \frac{\langle w_1, v_2 \rangle}{\|w_1\|^2} = -\frac{33}{101}$$

$$w_2 = \begin{pmatrix} -33 \\ 1 \end{pmatrix} - \mu \begin{pmatrix} 101 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$\left(\frac{3}{4} - \mu^2\right) \cdot \|w_1\|^2 = \frac{26247}{40804} \cdot 101^2 > 1 = \|w_2\|^2$$

Also Vertauschen:

$$v_1 = \begin{pmatrix} -33 \\ 1 \end{pmatrix} \qquad v_2 = \begin{pmatrix} 101 \\ 0 \end{pmatrix}$$

Reduktion:

$$v_1 = \begin{pmatrix} -33 \\ 1 \end{pmatrix} \qquad v_2 = \begin{pmatrix} 101 \\ 0 \end{pmatrix} + \left\lfloor \tfrac{3333}{1090} \right\rceil \cdot \begin{pmatrix} -33 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$$

Testen der Lovász-Bedingung:

$$\mu = \frac{\langle w_1, v_2 \rangle}{\|w_1\|^2} = -\frac{63}{1090}$$

$$w_2 = \begin{pmatrix} 2 \\ 3 \end{pmatrix} - \mu \begin{pmatrix} -33 \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{101}{1090} \\ \frac{3333}{1090} \end{pmatrix}$$

$$\left(\frac{3}{4} - \mu^2\right) \cdot \|w_1\|^2 = \frac{443553}{594050} \cdot 1090 > \frac{10201}{1090} = \|w_2\|^2$$

Vertauschen:

$$v_1 = \begin{pmatrix} 2 \\ 3 \end{pmatrix} \qquad v_2 = \begin{pmatrix} -33 \\ 1 \end{pmatrix}$$

Reduktion:

$$v_1 = \begin{pmatrix} 2 \\ 3 \end{pmatrix} \qquad v_2 = \begin{pmatrix} -33 \\ 1 \end{pmatrix} - \left\lfloor -\tfrac{63}{13} \right\rceil \cdot \begin{pmatrix} 2 \\ 3 \end{pmatrix} = \begin{pmatrix} -23 \\ 16 \end{pmatrix}$$

Nun erfüllt (v_1, v_2) *die Lovász-Bedingung*

$$\mu = \frac{\langle w_1, v_2 \rangle}{\|w_1\|^2} = \frac{2}{13}$$

$$w_2 = \begin{pmatrix} -23 \\ 16 \end{pmatrix} - \frac{2}{13} \begin{pmatrix} 2 \\ 3 \end{pmatrix} = \begin{pmatrix} -\frac{303}{13} \\ \frac{202}{13} \end{pmatrix}$$

$$\|w_1\|^2 = 13 < \frac{10201}{13} = \|w_2\|^2$$

und ist wegen

$$|\mu| = \frac{2}{13} < \frac{1}{2}$$

auch Gram-Schmidt-reduziert. Letzteres ist nach Konstruktion von Algorithmus 6.2 sowieso klar.

Der Algorithmus terminiert also mit der LLL-reduzierten Basis

$$v_1 = \begin{pmatrix} 2 \\ 3 \end{pmatrix} \qquad v_2 = \begin{pmatrix} -23 \\ 16 \end{pmatrix}$$

Er verhält sich in diesem Beispiel exakt wie der Gauß-Lagrange-Algorithmus. Dies ist durchaus typisch.

Für ein weiteres Beispiel siehe Übungsaufgabe 6.4.

Bemerkung 6.7.11 *Vorsicht, der erste Vektor in der Basis, die vom LLL-Algorithmus mit* $\frac{1}{4} < \delta < 1$ *berechnet wird, ist nicht notwendig ein kürzester Vektor im Gitter: Die Basis*

$$v_1 = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \qquad v_2 = \begin{pmatrix} 0 \\ 2 \end{pmatrix}$$

ist schon LLL-reduziert für $\delta = \frac{3}{4}$*, denn*

$$\frac{3}{4} \cdot (2^2 + 1^2) = \frac{3}{4} \cdot 5 \leq 4 = 2^2$$

wobei wir Bemerkung 6.7.2 verwenden. Der LLL-Algorithmus würde also die Basis (v_1, v_2) *nicht modifizieren.*

Der Gauß-Lagrange-Algorithmus dagegen vertauscht

$$v_1 = \begin{pmatrix} 0 \\ 2 \end{pmatrix} \qquad v_2 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

reduziert

$$v_1 = \begin{pmatrix} 0 \\ 2 \end{pmatrix} \qquad v_2 = \begin{pmatrix} 2 \\ 1 \end{pmatrix} - \left\lfloor \tfrac{2}{4} \right\rceil \cdot \begin{pmatrix} 0 \\ 2 \end{pmatrix} = \begin{pmatrix} 2 \\ -1 \end{pmatrix}$$

und terminiert dann.

Wollen wir sicher den kürzesten Vektor in dem Gitter erzeugt von v_1 und v_2 finden, müssen wir den LLL-Algorithmus mit $\delta = 1$ verwenden. Für $n = 2$ spezialisiert sich der LLL-Algorithmus dann in den Gauß-Lagrange-Algorithmus 6.3.

Zum Abschluss untersuchen wir noch, das Laufzeitverhalten von Algorithmus 6.5. Dies beweist insbesondere, dass der Algorithmus nach endlich vielen Schritten terminiert. Dazu führen wir eine Kenngröße ein, die den Basisvektor v_i stärker gewichtet als v_j, wenn $i < j$.

Definition 6.7.12 *Das **Potential** der Basis $(v_1, ..., v_i)$ sei*

$$P(v_1, ..., v_n) = \prod_{j=1}^{n} d(\langle v_1, ..., v_j \rangle),$$

wobei $\langle v_1, ..., v_j \rangle$ das Gitter erzeugt von der Basis $(v_1, ..., v_j)$ bezeichnet.

Bemerkung 6.7.13 *Es gilt*

$$\begin{aligned} P(v_1, ..., v_n) &= \prod_{j=1}^{n} \|w_1\| \cdot ... \cdot \|w_j\| \\ &= \|w_1\|^n \cdot \|w_2\|^{n-1} \cdot ... \cdot \|w_n\| \end{aligned}$$

wobei $(w_1, ..., w_n)$ die Gram-Schmidt-Basis zu $(v_1, ..., v_n)$ bezeichnet.

Der Einfachheit halber beweisen wir die Terminierung nur für $L \subset \mathbb{Z}^n$:

Satz 6.7.14 *Der LLL-Algorithmus angewendet auf eine Basis $(v_1, ..., v_n)$ eines Gitters $L \subset \mathbb{Z}^n$ terminiert für jedes $\frac{1}{4} < \delta \leq 1$.*

Beweis. Es ist

$$d(\langle v_1, ..., v_j \rangle)^2 = \det(X^t \cdot X) \in \mathbb{Z}_{\geq 1}$$

mit $X = (v_1 \mid ... \mid v_j)$ für jedes $j \geq 1$, also auch

$$P(v_1, ..., v_n)^2 \in \mathbb{Z}_{\geq 1}.$$

Wie schon im Beweis der Korrektheit gezeigt, ändert der Reduktionsschritt die Gram-Schmidt-Basis $(w_1, ..., w_n)$ und damit $P(v_1, ..., v_n)^2$ nicht.

Beim Vertauschen von v_{i-1} und v_i ändert sich für $j \neq i-1$ der Wert von $d(\langle v_1, ..., v_j \rangle)^2$ nicht. Für $j < i-1$ ist dies offensichtlich, für $j \geq i$ haben wir

$$d(\langle v_1, ..., v_{i-1}, v_i, ..., v_j \rangle)^2 = \det \begin{pmatrix} \langle v_1, v_1 \rangle & \cdots & \langle v_1, v_j \rangle \\ \vdots & & \vdots \\ \langle v_j, v_1 \rangle & \cdots & \langle v_j, v_j \rangle \end{pmatrix}$$

$$= -\det \left(\begin{array}{ccc|c|c|ccc} \langle v_1, v_1 \rangle & \cdots & \langle v_1, v_{i-1} \rangle & \langle v_1, v_i \rangle & \cdots & \langle v_1, v_j \rangle \\ \vdots & & \vdots & & & \vdots \\ \langle v_i, v_1 \rangle & \cdots & \langle v_i, v_{i-1} \rangle & \langle v_i, v_i \rangle & \cdots & \langle v_i, v_j \rangle \\ \langle v_{i-1}, v_1 \rangle & \cdots & \langle v_{i-1}, v_{i-1} \rangle & \langle v_{i-1}, v_i \rangle & \cdots & \langle v_{i-1}, v_j \rangle \\ \vdots & & \vdots & \vdots & & \vdots \\ \langle v_j, v_1 \rangle & \cdots & \langle v_j, v_{i-1} \rangle & \langle v_j, v_i \rangle & \cdots & \langle v_j, v_j \rangle \end{array}\right)$$

$$= \det \left(\begin{array}{ccc|c|c|ccc} \langle v_1, v_1 \rangle & \cdots & \langle v_1, v_i \rangle & \langle v_1, v_{i-1} \rangle & \cdots & \langle v_1, v_j \rangle \\ \vdots & & \vdots & & & \vdots \\ \langle v_i, v_1 \rangle & \cdots & \langle v_i, v_i \rangle & \langle v_i, v_{i-1} \rangle & \cdots & \langle v_i, v_j \rangle \\ \langle v_{i-1}, v_1 \rangle & \cdots & \langle v_{i-1}, v_i \rangle & \langle v_{i-1}, v_{i-1} \rangle & \cdots & \langle v_{i-1}, v_j \rangle \\ \vdots & & \vdots & \vdots & & \vdots \\ \langle v_j, v_1 \rangle & \cdots & \langle v_j, v_i \rangle & \langle v_j, v_{i-1} \rangle & \cdots & \langle v_j, v_j \rangle \end{array}\right)$$

$$= d(\langle v_1, ..., v_{i-2}, v_i, v_{i-1}, v_{i+1}, ..., v_j \rangle)^2$$

da sich die Determinante beim Vertauschen der Spalten $i-1$ und i und Zeilen $i-1$ und i jeweils um -1 ändert.

Sei nun $j = i-1$: Sei w_{i-1} der $(i-1)$-te Gram-Schmidt-Vektor vor der Vertauschung und w'_{i-1} der entsprechende Vektor danach. Wie in Bemerkung 6.7.3.(3) gezeigt, ist die δ-Lovász-Bedingung dann äquivalent zu

$$\frac{\|w'_{i-1}\|^2}{\|w_{i-1}\|^2} \geq \delta.$$

Wir erhalten damit

$$\frac{d(\langle v_1, ..., v_{i-2}, v_i \rangle)^2}{d(\langle v_1, ..., v_{i-2}, v_{i-1} \rangle)^2} = \frac{\|w_1\|^2 \cdot ... \cdot \|w_{i-2}\|^2 \cdot \|w'_{i-1}\|^2}{\|w_1\|^2 \cdot ... \cdot \|w_{i-2}\|^2 \cdot \|w_{i-1}\|^2} = \frac{\|w'_{i-1}\|^2}{\|w_{i-1}\|^2} < \delta,$$

denn der Algorithmus vertauscht genau dann, wenn die Lovász-Bedingung nicht erfüllt ist.

Für das Potential gilt dann also

$$\frac{P(v_1, ..., v_i, v_{i-1}, ..., v_n)^2}{P(v_1, ..., v_{i-1}, v_i, ..., v_n)^2} < \delta$$

und somit wird die positive ganze Zahl $P(v_1, ..., v_n)^2 \in \mathbb{Z}$ bei jeder Vertauschung echt kleiner. ■

Mit der Ungleichung

$$\frac{P(v_1, ..., v_i, v_{i-1}, ..., v_n)^2}{P(v_1, ..., v_{i-1}, v_i, ..., v_n)^2} < \sqrt{\delta}$$

können wir die Anzahl der Vertauschungen abschätzen: Eine untere Schranke für die Größe des Inputs ist

$$M = \max\{n, \log_2(\max\{\|v_1\|, ..., \|v_n\|\})\}$$

da wir für jeden Inputvektor v_j mindestens 1 Bit benötigen, und ein Vektor v der Norm $\|v\|$ mindestens $\log_2 \|v\|$ Bits benötigt. Man beachte: Ist die Anzahl der Vertauschungen des *LLL*-Algorithmus polynomial in der unteren Schranke M, dann auch in der wirklichen Bitgröße.

Satz 6.7.15 *Für $\delta < 1$ ist die Anzahl der Vertauschungen im LLL-Algorithmus polynomial in M.*

Beweis. Mit

$$V = \max\{\|v_1\|, ..., \|v_n\|\}$$

haben wir

$$\begin{aligned} P(v_1, ..., v_n) &= \|w_1\|^n \cdot \|w_2\|^{n-1} \cdot ... \cdot \|w_n\| \\ &\leq \|v_1\|^n \cdot \|v_2\|^{n-1} \cdot ... \cdot \|v_n\| \\ &\leq V^{\sum_{i=1}^n i} = V^{\frac{n(n+1)}{2}} \end{aligned}$$

da $\|w_i\| \leq \|v_i\|$, wie im Beweis der Hadamard-Ungleichung (Satz 6.3.7) gezeigt.

Nach dem im Beweis von Satz 6.7.14 wird $P(v_1, ..., v_n)$ bei jeder Vertauschung wenigstens um den Faktor $\sqrt{\delta}$ kleiner. Die Anzahl der Vertauschungen ist also beschränkt durch

$$\begin{aligned} \log_{\sqrt{\delta}} P(v_1, ..., v_n) &= \frac{1}{\log_2 \sqrt{\delta}} \log_2 P(v_1, ..., v_n) \\ &\leq \frac{1}{\log_2 \sqrt{\delta}} \cdot \frac{n \cdot (n+1)}{2} \log_2 V \\ &\leq \frac{1}{\log_2 \sqrt{\delta}} \cdot \frac{M^2(M+1)}{2} \end{aligned}$$

d.h. durch ein Polynom in M. ■

Bemerkung 6.7.16 *Wie man leicht sieht, benötigt auch die Gram-Schmidt-Reduktion nur polynomial viele Rechenoperationen. Um polynomiale Laufzeit des LLL-Algorithmus in M vollständig zu beweisen, müsste man aber noch zeigen, dass die Zahlen, die bei der Reduktion auftreten, durch in M polynomial viele Bits repräsentiert werden können. Dies ist eine leichte, aber aufwendige Rechnung, die wir hier nicht durchführen wollen. Um zu sehen, dass dies notwendig ist, betrachte z.B. n-maliges Quadrieren einer Zahl a: Obwohl wir nur n Rechenschritte durchführen wächst die Bitgröße (und damit die Laufzeit, wie wir im nächsten Abschnitt sehen werden) exponentiell in der Größe der Zahl a, denn*

$$\log_2(a^{2^n}) = \log_2(a) \cdot 2^n.$$

6.8 Übungen

Übung 6.1 *Sei $L \subset \mathbb{R}^2$ ein Gitter mit Basis (v_1, v_2).*

1) *Zeigen Sie, dass der Gauß-Lagrange-Algorithmus 6.3 terminiert.*

2) *Zeigen Sie, dass in der vom Gauß-Lagrange-Algorithmus berechneten Basis (v_1, v_2) der Vektor v_1 ein kürzester Vektor in L ist, d.h.*

$$\|v\| \geq \|v_1\|$$

für alle $v \in L$.

Übung 6.2 *Seien $p_1 = 101$ und $p_2 = 103$ und*

$$r = \frac{a}{b} \in \mathbb{Q}$$

mit $p_i \nmid b \ \forall i$ und $a^2 + b^2 < N = p_1 \cdot p_2$. Modulo p_1 und p_2 wurde $\overline{r} = \overline{a} \cdot \overline{b}^{-1}$ berechnet als

$$\overline{r} = \overline{79} \in \mathbb{Z}/p_1$$
$$\overline{r} = \overline{27} \in \mathbb{Z}/p_2$$

1) *Bestimmen Sie mit dem Chinesischen Restsatz $x \in \mathbb{Z}$ mit $0 \leq x < N$ und*

$$x \equiv 79 \bmod p_1$$
$$x \equiv 27 \bmod p_2$$

2) *Bestimmen Sie a und b aus r, indem Sie in dem Gitter der $\mathbb{Z}$-Linearkombinationen von*

$$\begin{pmatrix} N \\ 0 \end{pmatrix}, \begin{pmatrix} r \\ 1 \end{pmatrix}$$

einen kürzesten Vektor finden.

3) *Überprüfen Sie, dass Ihr Resultat die obigen Kongruenzen erfüllt.*

Übung 6.3 *Leonardo da Vinci hat vermutet, dass das Längenverhältnis $\frac{b}{a}$ in Abbildung 6.5 eine algebraische Zahl $r \in \overline{\mathbb{Q}}$ ist.*

1) *Messen Sie a und b so genau wie möglich ab, und berechnen Sie numerisch eine Näherung für $\frac{b}{a}$.*

2) *Bestimmen Sie mit Hilfe des LLL-Algorithmus aus der Näherung einen Kandidaten für r.*

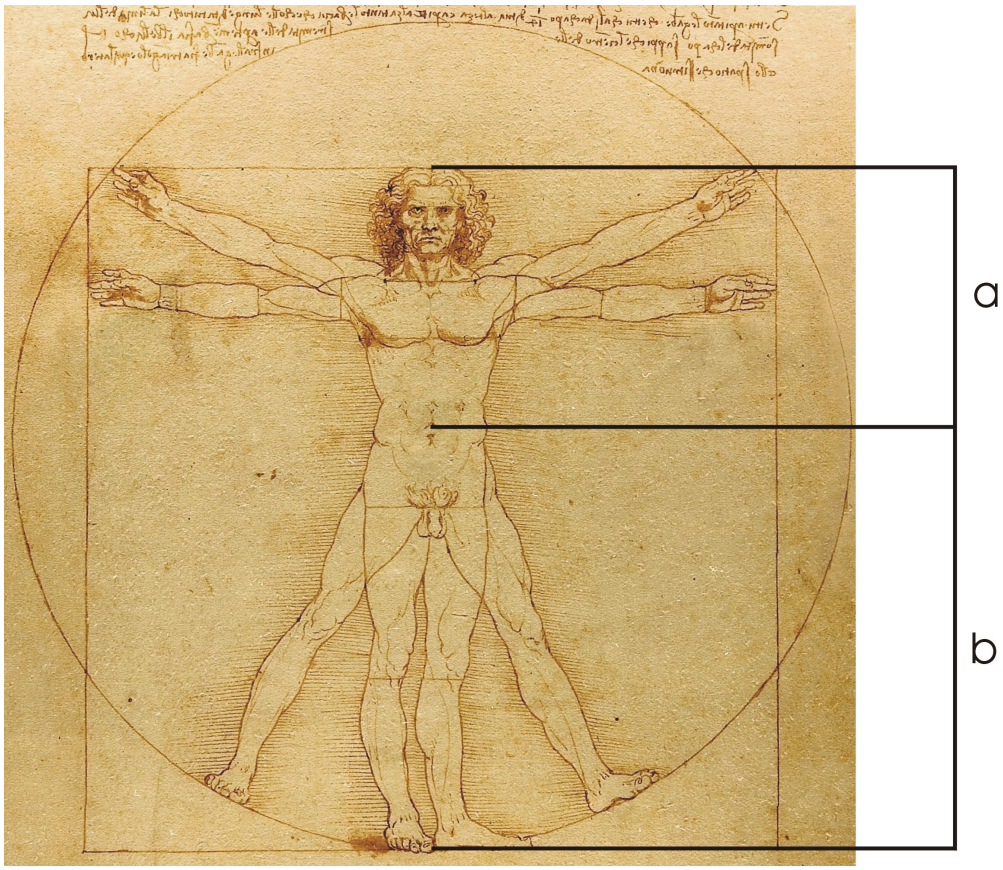


Abbildung 6.5: Proportionen des Menschen von Leonardo da Vinci

3) Erproben Sie das Verfahren auch an Ihren eigenen Maßen.

Übung 6.4 *Berechnen Sie eine LLL-reduzierte Basis des Gitters L erzeugt von den Vektoren*

$$\begin{pmatrix} 4 \\ 4 \\ 6 \end{pmatrix}, \begin{pmatrix} 5 \\ 8 \\ 2 \end{pmatrix}, \begin{pmatrix} 1 \\ 2 \\ 6 \end{pmatrix} \in \mathbb{Z}^3$$

Folgen Sie dabei dem Algorithmus Schritt für Schritt.

Übung 6.5 *Sei $L \subset \mathbb{R}^n$ ein Gitter von Rang n und $(v_1, ..., v_n)$ eine δ–LLL-reduzierte Basis von L. Seien weiter $y_1, ..., y_l$ linear unabhängige Vektoren in L. Zeigen Sie, dass mit*

$$\tau = \frac{4}{4\delta - 1}$$

gilt

$$\|v_j\| \leq \tau^{\frac{n-1}{2}} \cdot \max\{\|y_1\|, ..., \|y_l\|\}$$

für alle $j = 1, ..., l$.

Übung 6.6 *Wenden Sie den Gauß-Lagrange-Algorithmus auf folgende Basen an:*

1)

$$v_1 = \begin{pmatrix} 101 \\ 0 \end{pmatrix} \qquad v_2 = \begin{pmatrix} 68 \\ 1 \end{pmatrix}$$

2)

$$v_1 = \begin{pmatrix} 7 \\ 17 \end{pmatrix} \qquad v_2 = \begin{pmatrix} 3 \\ 7 \end{pmatrix}$$

3)

$$v_1 = \begin{pmatrix} 19 \\ 18 \end{pmatrix} \qquad v_2 = \begin{pmatrix} 15 \\ 12 \end{pmatrix}$$

Vergleichen Sie das Verhalten mit dem des LLL-Algorithmus.

6.9 Praktische Aufgaben

Übung 6.7 *Schreiben Sie jeweils eine Funktion, die*

1) für $v_1, v_2 \in \mathbb{Z}^2$ vom Typ `intvec` *das Skalarprodukt $\langle v_1, v_2 \rangle$ bestimmt.*

2) $x \in \mathbb{Q}$ auf die nächste ganze Zahl $\lfloor x \rceil \in \mathbb{Z}$ rundet.

3) $v_2 \in \mathbb{Z}^2$ nach $v_1 \in \mathbb{Z}^2$ reduziert.

4) Sei $L \subset \mathbb{Z}^2$ ein Gitter von Rang 2. Schreiben Sie eine Funktion, die mit Hilfe des Gauß-Lagrange-Algorithmus eine Basis (v_1, v_2) von L bestimmt, sodass v_1 ein kürzester Vektor in L ist.

5) Erproben Sie Ihre Implementierung an Beispielen, insbesondere an dem Gitter

$$\left\langle \begin{pmatrix} 10403 \\ 0 \end{pmatrix}, \begin{pmatrix} 4162 \\ 1 \end{pmatrix} \right\rangle$$

Übung 6.8 *1) Implementieren Sie den LLL-Algorithmus 6.4 für Gitter $L \subset \mathbb{Z}^n$.*

2) Erproben Sie Ihre Implementierung an dem Gitter aus Aufgabe 6.4.

7

Polynomfaktorisierung

7.1 Übersicht

Ein grundlegendes Prinzip für unser bisheriges Vorgehen war die Übertragung von wichtigen Eigenschaften der ganzen Zahlen auf allgemeinere, interessante Klassen von Ringen. Wie in $\mathbb{Z}$ lässt sich z.B. für jeden Integritätsring der Quotientenkörper konstruieren (siehe Übung 2.3), oder in jedem Euklidischen Ring (siehe Definition 2.2.3) der Euklidische Algorithmus 2.1 zur Bestimmung des ggT durchführen. Ebenso wie in $\mathbb{Z}$ wird in jedem Hauptidealring (siehe 4.3.1) jedes Ideal vom größten gemeinsamen Teiler der Erzeuger erzeugt. Als eine Verallgemeinerung davon haben wir Noethersche Ringe untersucht (siehe Definition und Satz 4.2.1), in denen jedes Ideal von endlich vielen Elementen erzeugt wird. Speziell im multivariaten Polynomring $K[x_1, ..., x_n]$ erlaubt uns die Notation der Gröbnerbasis, Ideale und deren Lösungsmengen algorithmisch zu untersuchen (Buchbergeralgorithmus 4.3). Ein multivariates Gleichungssystem wird gegeben durch ein Ideal $I \subset K[x_1, ..., x_n]$. Ist die Lösungsmenge $V(I)$ in $\overline{K}^n$ endlich, so enthält die lexikographische Gröbnerbasis von I nach Bemerkung 4.13.2 Gleichungen in Dreiecksgestalt

$$\begin{aligned} f_1 &= \boldsymbol{x}_1^{\alpha_1} - g_1(x_1, ..., x_n) \\ f_2 &= \boldsymbol{x}_2^{\alpha_2} - g_2(x_2, ..., x_n) \\ &\vdots \\ f_{n-1} &= \boldsymbol{x}_{n-1}^{\alpha_{n-1}} - g_{n-1}(x_{n-1}, x_n) \\ f_n &= g_n(x_n) \end{aligned}$$

Zur Bestimmung von $V(I)$ lösen wir das System rückwärts auf, d.h. wir lösen die univariate Gleichung

$$f_n(x_n) = 0$$

setzen die Ergebnisse x_n in

$$f_{n-1}(x_{n-1}, x_n) = 0$$

ein und fahren induktiv mit $f_{n-2}, ..., f_1$ fort. Schliesslich testen wir, welche Punkte wirklich in $V(I)$ liegen.

Wenn wir ein exaktes Ergebnis benötigen, können wir nicht numerisch vorgehen (ausser wir verwenden Methoden wie in Bemerkung 6.6.2). Um die Gleichung $f(x) = 0$ mit $f \in K[x]$ zu lösen, gehen wir folgendermaßen vor: Zerfällt f in Linearfaktoren

$$f = c \cdot (x - c_1) \cdot ... \cdot (x - c_d)$$

so sind $c_1, ..., c_d \in K$ die Lösungen. Falls nicht, faktorisieren wir

$$f = f_1 \cdot ... \cdot f_r$$

in Primfaktoren und betrachten f_i einzeln. In dem Körper $K' = K[x]/\langle f_i \rangle$ hat f_i dann die Nullstelle $\overline{x}$. Zerfällt f_i über K' in Linearfaktoren, kennen wir wieder die Nullstellen, falls nicht, wenden wir das Verfahren induktiv auf einen Primfaktor von $f_i \in K'[x]$ an.

Es stellt sich also die allgemeine Frage, in welchen Ringen wir wie in $\mathbb{Z}$ den Satz 2.3.3 über die eindeutige Primfaktorisierung beweisen können. Dies sind die sogenannten faktoriellen Ringe. Wir werden zeigen, dass diese wie Noethersche Ringe eine Verallgemeinerung von Hauptidealringen darstellen. Wir haben also folgende Typen von Ringen:

	Eigenschaften	Beispiel für $R = \mathbb{Z}$
{Integritätsringe}	Quotientenkörper-konstruktion	$\mathbb{Q}$
$\cup$		
{Faktorielle Ringe}	Eindeutige Primfaktorisierung (bis auf Einheiten), Existenz des ggT	$120 = 2^3 \cdot 3 \cdot 5$ $84 = 2^2 \cdot 3 \cdot 7$ $\mathrm{ggT}(120, 84) = 2^2 \cdot 3$
$\cup$		
{Hauptidealringe}	Jedes Ideal wird von einem Element erzeugt	$120\mathbb{Z} + 84\mathbb{Z} = \underbrace{12}_{\mathrm{ggT}(120,84)}\mathbb{Z}$
$\cup$		
{Euklidische Ringe}	Euklidischer Algorithmus zur Bestimmung des ggT	$120 = 1 \cdot 84 + 36$ $84 = 2 \cdot 36 + 12$ $36 = 3 \cdot 12 + 0$

Noethersche Ringe, die auch Hauptidealringe verallgemeinern, sind im Allgemeinen nicht faktoriell (siehe Übung 7.1). Speziell der multivariate Polynomring $K[x_1, ..., x_n]$ über einem Körper ist aber faktoriell.

7.2 Faktorielle Ringe

Definition 7.2.1 *1) Zwei Elemente $a, b \in R$ heißen **assoziiert**, wenn es ein $u \in R^\times$ gibt mit $a = u \cdot b$. Wir schreiben dann $a \sim b$. Dies ist eine Äquivalenzrelation.*

2) Ein Element $q \in R$, $q \neq 0$, $q \notin R^\times$ heißt ***irreduzibel****, wenn gilt*

$$q = a \cdot b \text{ mit } a, b \in R \Longrightarrow a \in R^\times \text{ oder } b \in R^\times$$

3) Ein Element $p \in R$, $p \neq 0$, $p \notin R^\times$ heißt ***Primelement****, wenn gilt*

$$p \mid a \cdot b \text{ mit } a, b \in R \Longrightarrow p \mid a \text{ oder } p \mid b$$

Satz 7.2.2 *Ist R ein Integritätsring und $p \in R$, dann*

$$p \text{ prim} \implies p \text{ irreduzibel}$$

Beweis. Sei p prim und $p = a \cdot b$, dann $p \mid a \cdot b$ also ohne Einschränkung $p \mid a$ und somit $a = p \cdot r$. Dann folgt $p = p \cdot r \cdot b$ also mit der Kürzungsregel in Integritätsringen $1 = r \cdot b$ und somit $b \in R^\times$. ■

Satz 7.2.3 *Ist R Noethersch, dann gilt: Jedes $a \in R$, $a \neq 0$, $a \notin R^\times$ ist ein Produkt*

$$a = q_1 \cdot \ldots \cdot q_r$$

von irreduziblen Elementen.

Beweis. Ist a irreduzibel, ist nichts zu zeigen. Sei a reduzibel, etwa $a = a_1 b_1$ mit $a_1, b_1 \notin R^\times$. Wenn a_1 und b_1 irreduzibel sind, stimmt die Behauptung. Ist ohne Einschränkung a_1 nicht irreduzibel, dann existieren $a_2, b_2 \notin R^\times$ mit $a_1 = a_2 b_2$. Somit erhalten wir eine Folge von Elementen a_i und eine Kette von Hauptidealen

$$\langle a \rangle \subset \langle a_1 \rangle \subset \langle a_1 \rangle \subset \ldots$$

die stationär werden muss. ■

Definition 7.2.4 *Ein Integritätsring heißt* ***faktoriell****, wenn jedes $a \in R$, $a \neq 0$, $a \notin R^\times$ ein Produkt*

$$a = p_1 \cdot \ldots \cdot p_r$$

von Primelementen p_i ist.

Beispiel 7.2.5 *1) $\mathbb{Z}$ ist faktoriell.*

2) Der Integritätsring

$$R = K[x, y, z, w] / (xy - zw)$$

ist nicht faktoriell, denn

$$\bar{x}\bar{y} = \bar{z}\bar{w}$$

also $\bar{z}$ teilt $\bar{x}\bar{y}$ aber nicht $\bar{x}$ oder $\bar{y}$.

Primelemente sind nach Satz 7.2.2 stets irreduzibel, in faktoriellen Ringen gilt auch die Umkehrung:

Satz 7.2.6 *Sei R faktoriell und $q \in R$, dann gilt*

$$q \text{ prim} \iff q \text{ irreduzibel}$$

Beweis. Ist q irreduzibel, dann ist q kein Produkt von mindestens zwei Nichteinheiten, also auch nicht von Primelementen. In der Darstellung $a = p_1 \cdot \ldots \cdot p_r$ muss also $r = 1$ und $q = p_1$ prim sein. ■

Satz 7.2.7 *Ein Integritätsring R ist faktoriell genau dann, wenn jedes $a \in R$, $a \neq 0$, $a \notin R^\times$ ein bis auf Permutation und Einheiten eindeutiges Produkt von irreduziblen Elementen ist.*

Das heißt, a lässt sich schreiben als

$$a = p_1 \cdot \ldots \cdot p_r$$

mit p_i irreduzibel, und sind

$$p_1 \cdot \ldots \cdot p_r = a = q_1 \cdot \ldots \cdot q_s$$

zwei solche Darstellungen, dann ist $r = s$ und es existiert eine Permutation $\sigma \in S_r$ sodass $p_i \sim q_{\sigma(i)}$.

Beweis. Ist R faktoriell, dann gibt es eine Zerlegung in irreduzible (äquivalent prime) Elemente. Zur Eindeutigkeit: Seien

$$p_1 \cdot \ldots \cdot p_r = q_1 \cdot \ldots \cdot q_s$$

zwei solche Zerlegungen. Da p_1 prim ist, gilt $p_1 \mid q_j$ für ein j, ohne Einschränkung $j = 1$, also

$$q_1 = w \cdot p_1$$

und $w \in R^\times$ (wegen q_1 irreduzibel und p_1 prim).

Mit der Kürzungsregel folgt aus

$$p_1 \cdot \ldots \cdot p_r = w \cdot p_1 \cdot q_2 \cdot \ldots \cdot q_s$$

dass

$$p_2 \cdot \ldots \cdot p_r = (w \cdot q_2) \cdot \ldots \cdot q_s$$

Induktion nach r gibt die Behauptung.

Umgekehrt müssen wir nur zeigen, dass jedes irreduzible Element prim ist:

Sei q irreduzibel und $q \mid a \cdot b$. Ist a eine Einheit, dann $q \mid b$, ist $a = 0$ dann $q \mid a$.

Sind $a, b \notin R^\times$ und $a, b \neq 0$, dann

$$a \cdot b = q \cdot w$$

Nach Voraussetzung haben a, b, w Zerlegungen in irreduzible Elemente. Setzen wir diese ein, dann liefert die Eindeutigkeit, dass q bis auf eine Einheit einer der irreduziblen Faktoren von a oder b sein muss, also $q \mid a$ oder $q \mid b$. Somit ist q prim. ■

Als Corollar zu Satz 7.2.3 und Satz 7.2.6 folgt sofort:

Corollar 7.2.8 *Ein Noetherscher Integritätsring ist genau dann faktoriell, wenn jedes irreduzible Element auch prim ist.*

Beispiel 7.2.9 *Der Ring*

$$R = \mathbb{Z}\left[\sqrt{-3}\right] = \left\{a + b\sqrt{-3} \mid a, b \in \mathbb{Z}\right\} \subset \mathbb{C}$$

ist nicht faktoriell, denn

$$4 = 2 \cdot 2 = \left(1 - \sqrt{-3}\right)\left(1 + \sqrt{-3}\right)$$

sind Zerlegungen in irreduzible (nicht prime) Elemente und die Faktoren 2 *und* $1 \pm \sqrt{-3}$ *unterscheiden sich nicht nur um Einheiten (Übung). Beachte, dass* R *Noethersch ist. Siehe auch Übungsaufgabe 7.1.*

Primelemente sind nach Satz 7.2.2 stets irreduzibel. In Hauptidealringen ist auch die Umkehrung richtig:

Satz 7.2.10 *In einem Hauptidealring gilt*

$$p \text{ irreduzibel} \Longrightarrow p \text{ prim}$$

Mit Bemerkung 4.3.2 und Satz 7.2.8 folgt daraus sofort:

Corollar 7.2.11 *Hauptidealringe sind faktoriell.*

Wir zeigen noch Satz 7.2.10:

Beweis. Sei p irreduzibel und $p \mid ab$. Es ist $\langle p \rangle \subset \langle p, a \rangle$ und $\langle p \rangle \subset \langle p, b \rangle$. Es können nicht $\langle p, a \rangle$ und $\langle p, b \rangle$ beide gleich $\langle 1 \rangle$ sein, denn sonst gäbe es r_i, s_i mit

$$r_1 a + s_1 p = 1 = r_2 b + s_2 p$$

und somit

$$1 = (r_1 a + s_1 p) \cdot (r_2 b + s_2 p) = r_1 r_2 ab + r_1 a s_2 p + s_1 r_2 b p + s_1 s_2 p^2 \in \langle p \rangle$$

Sei also ohne Einschränkung $\langle p, a \rangle = \langle d \rangle \subsetneq R$ mit $d \notin R^\times$, also $p = cd$ und $a = ed$. Da p irreduzibel ist, folgt $c \in R^\times$ und damit $a = ec^{-1}p \in \langle p \rangle$, d.h. $p \mid a$. ■

Somit ist mit Satz 4.3.3 jeder Euklidische Ring faktoriell, insbesondere $\mathbb{Z}$ (das haben wir schon in Satz 2.3.3 gezeigt) und der univariate Polynomring $K[x]$ über einem Körper K.

Wir bemerken noch folgenden Satz, den wir hier nicht zeigen können:

Satz 7.2.12 (Satz von Gauß) *Sei* R *ein Integritätsring. Dann gilt*

$$R \text{ faktoriell} \iff R[x] \text{ faktoriell}$$

Induktiv ist also jeder Polynomring $R[x_1, \ldots, x_n]$ faktoriell, wenn R faktoriell ist (insbesondere wenn R ein Körper oder $R = \mathbb{Z}$ ist).

7.3 Quadratfreie Polynomfaktorisierung

Sei R ein faktorieller Ring. Dann ist nach Satz 7.2.12 auch der Polynomring $R[x]$ faktoriell.

Definition 7.3.1 *Die **formale Ableitung** ist die Abbildung*

$$(-)' : R[x] \to R[x],\ \sum_{i=0}^{d} a_i x^i \mapsto \sum_{i=1}^{d} i a_i x^{i-1}$$

Bemerkung 7.3.2 *Es gilt $f' = 0$ für $f \in R$ (Vorsicht: Die Umkehrung ist im Allgemeinen nicht richtig, siehe unten) und*

$$\begin{aligned}(f+g)' &= f' + g' \\ (f \cdot g)' &= f' \cdot g + f \cdot g' \\ (f^n)' &= n \cdot f^{n-1} \cdot f'\end{aligned}$$

für alle $f, g \in R[x]$, $n \in \mathbb{N}$.

Definition 7.3.3 *Ein Polynom $0 \neq f \in R[x]$ heißt **quadratfrei**, wenn es kein $g \in R[x]$ mit $\deg(g) > 0$ gibt mit $g^2 \mid f$.*

*Eine **quadratfreie Faktorisierung** von $f \in R[x]$ ist eine Darstellung*

$$f = \prod_{i=1}^{s} f_i^i$$

mit $f_i \in R[x]$ quadratfrei und paarweise teilerfremd.

Bemerkung 7.3.4 *Mit der Faktorisierung*

$$f = p_1^{a_1} \cdot \ldots \cdot p_r^{a_r}$$

von $f \in R[x]$ gilt

$$f_i = \prod_{j \text{ mit } a_j = i} p_j$$

Insbesondere ist die quadratfreie Faktorisierung eindeutig (bis auf Einheiten).

Bemerkung 7.3.5 *Ist $f = g \cdot h^i \in R[x]$ dann gilt*

$$f' = g' \cdot h^i + a \cdot g \cdot h^{i-1} \cdot h'$$

also $h^{i-1} \mid \mathrm{ggT}(f, f')$. Ist also

$$f = \prod_{i=1}^{s} f_i^i$$

dann gilt

$$\prod_{i=2}^{s} f_i^{i-1} \mid \mathrm{ggT}(f, f').$$

Angenommen wir haben Gleichheit

$$d := \prod_{i=2}^{s} f_i^{i-1} = \mathrm{ggT}(f, f'). \tag{7.1}$$

Dann ist

$$\tilde{f} = \frac{f}{\mathrm{ggT}(f, f')} = f_1 \cdot \ldots \cdot f_s$$

Mit demselben Argument angewendet auf d erhalten wir

$$\tilde{d} = \frac{d}{\mathrm{ggT}(d, d')} = f_2 \cdot \ldots \cdot f_s$$

und als Quotient somit

$$\frac{\tilde{f}}{\tilde{d}} = f_1.$$

Wir wenden dann das Verfahren induktiv auf d an.

Beispiel 7.3.6 *Für*

$$f = x^4 + 5x^3 + 9x^2 + 7x + 2 = (x+1)^3(x+2)$$

ist

$$f' = 4x^3 + 15x^2 + 18x + 7 = 3(x+1)^2(x+2) + (x+1)^3$$

also

$$d = \mathrm{ggT}(f, f') = x^2 + 2x + 2 = (x+1)^2$$

und

$$\tilde{f} = \frac{f}{\mathrm{ggT}(f, f')} = x^2 + 3x + 2 = (x+1)(x+2)$$

Nochmal für d angewendet ist

$$\tilde{d} = \frac{d}{\mathrm{ggT}(d, d')} = x + 1$$

und somit erhalten wir

$$\frac{\tilde{f}}{\tilde{d}} = x + 2 = f_1$$

Das folgende Beispiel zeigt, dass dies nicht immer so funktioniert, da Gleichung 7.1 im Allgemeinen nicht korrekt ist:

Beispiel 7.3.7 *Für $f = x^p - 1 \in \mathbb{F}_p[x]$ gilt*

$$f' = px^{p-1} = 0$$

also

$$\mathrm{ggT}(f, f') = f$$

und

$$\tilde{f} = 1$$

Man beachte, dass f nicht quadratfrei ist, denn

$$f = x^p - 1 = \sum_{i=0}^{p} \binom{p}{i} x^i (-1)^{p-i} = (x-1)^p,$$

da p alle $\binom{p}{i}$ für $1 < i < p$ teilt.

Die Lösung für das Problem ist, für $f' = 0$ die p-te Wurzel zu ziehen.

Falls $\operatorname{char}(R) = 0$ allerdings gilt Gleichung 7.1. Wir erinnern uns:

Definition 7.3.8 *Sei R ein kommutativer Ring mit 1 und*

$$\begin{array}{rccc} \chi: & \mathbb{Z} & \longrightarrow & R \\ & n & \longmapsto & n \cdot 1_R \end{array}$$

die charakteristische Abbildung. Der Kern ist ein Ideal

$$\ker \chi = \langle n \rangle$$

mit $n \geq 0$. Zwei Fälle können auftreten:

1) $n = 0$, d.h. χ ist injektiv, d.h. $\mathbb{Z}$ ist ein Unterring von R.

2) $n > 0$. Dann ist

$$\mathbb{Z}/n \hookrightarrow R$$

nach dem Homomorphiesatz ein Unterring von R.

Ist R ein Integritätsring, dann ist n also eine Primzahl.

In beiden Fällen nennt man

$$n \geq 0$$

*die **Charakteristik** von R.*

Bemerkung 7.3.9 *Jeder Körper K enthält also den Unterkörper $\mathbb{Q}$ (für $\operatorname{char}(K) = 0$) oder $\mathbb{F}_p$ (für $\operatorname{char}(K) = p$).*

Beispiel 7.3.10 *Wir haben*

$$\operatorname{char}(\mathbb{Z}) = \operatorname{char}(\mathbb{Q}) = 0.$$

Für p prim ist

$$\operatorname{char}(\mathbb{F}_p) = \operatorname{char}(\mathbb{F}_{p^r}) = p$$

Satz 7.3.11 *Zu jeder Primzahlpotenz p^r gibt es bis auf Isomorphie genau einen Körper $\mathbb{F}_{p^r}$ mit p^r Elementen, nämlich den Zerfällungskörper von*

$$x^{p^r} - x \in \mathbb{F}_p[x]$$

Ist $f \in \mathbb{F}_p[x]$ irreduzibel vom Grad $\deg(f) = r$, dann ist

$$\mathbb{F}_{p^r} \cong \mathbb{F}_p[x]/\langle f \rangle$$

Das in Bemerkung 7.3.5 beschriebene Verfahren funktioniert zumindest für Ringe der Charakteristik 0 (sofern wir einen Algorithmus zur Berechnung des ggT haben):

Lemma 7.3.12 *Sei*

$$f = \prod_{i=1}^{s} f_i^i \in R[x]$$

eine quadratfreie Faktorisierung. Ist $\operatorname{char}(R) = 0$ *dann gilt*

$$\operatorname{ggT}(f, f') = r \cdot \prod_{i=2}^{s} f_i^{i-1}$$

mit $r \in R$. *Insbesondere gilt*

$$f \text{ quadratfrei} \iff \operatorname{ggT}(f, f') \in R.$$

Beweis. Sei $f = r \cdot p_1^{a_1} \cdot \ldots \cdot p_r^{a_r} \in R[x]$ mit p_i irreduzibel, $\deg(p_i) \geq 1$, $r \in R$ und $a_i \geq 1$. Angenommen $p_i^{a_i}$ teilt

$$f' = p_1^{a_1-1} \cdot \ldots \cdot p_r^{a_r-1} \cdot (a_r \cdot p_1 \cdot \ldots \cdot p_{r-1} \cdot p_r' + \ldots + a_1 \cdot p_1' \cdot p_2 \cdot \ldots \cdot p_{r-1})$$

dann $p_i \mid (a_i \cdot p_i')$. Da $\deg(p_i') < \deg(p_i)$ folgt $a_i \cdot p_i' = 0$. Ist $p_i' = 0$, dann folgt für $\operatorname{char}(R) = 0$, dass $\deg(p_i) = 0$. Anderenfalls liegen die Koeffizienten von $a_i \cdot p_i'$ in $\ker \chi$, ein Widerspruch zu $\operatorname{char}(R) = 0$. ■

Für $\operatorname{char}(R) = 0$ erhalten wir aus Bemerkung 7.3.5 und Lemma 7.3.12 den Algorithmus 7.1 zur Berechnung einer quadratfreien Faktorisierung.

Algorithmus 7.1 Quadratfreie Faktorisierung

Input: $f \in R[x]$ und R faktoriell mit $\operatorname{char} R = 0$.

Output: $(f_1, \ldots, f_s)$ sodass $f = \prod_{i=1}^{s} f_i^i$ die quadratfreie Faktorisierung von f.

1: $d = \operatorname{ggT}(f, f')$
2: $\tilde{f} = \frac{f}{d}$
3: $\tilde{d} = \frac{d}{\operatorname{ggT}(d, d')}$
4: $f_1 = \frac{\tilde{f}}{\tilde{d}}$
5: Quadratfreie Faktorisierung von d gibt $(f_2, \ldots, f_s)$.
6: **return** $(f_1, \ldots, f_s)$

Beweis. Mit Lemma 7.3.12 gilt

$$\begin{aligned} d &= \prod_{i=2}^{s} f_i^{i-1} \\ \tilde{f} &= \frac{f}{\operatorname{ggT}(f, f')} = f_1 \cdot \ldots \cdot f_s \\ \tilde{d} &= \frac{d}{\operatorname{ggT}(d, d')} = f_2 \cdot \ldots \cdot f_s \end{aligned}$$

■

In einem perfekten Körper der Charakteristik p haben wir $f = g^p$ falls $f' = 0$. In diesem Fall ziehen wir die p-te Wurzel und fahren mit g fort.

7.4 Berlekamp-Algorithmus

Der Berlekamp-Algorithmus ist der Standardalgorithmus zur Faktorisierung von Polynomen $f \in \mathbb{F}_q[x]$. Nach Abschnitt 7.3 können wir annehmen, dass f quadratfrei ist. Aus dem Chinesischen Restsatz 2.5.3 und Satz 7.3.11 folgt:

Bemerkung 7.4.1 *Sei K ein Körper und $f = p_1 \cdot \ldots \cdot p_r \in K[x]$ quadratfrei mit p_i irreduzibel. Dann ist*

$$\begin{array}{ccc} K[x]/\langle f\rangle & \cong & K[x]/\langle p_1\rangle \times \ldots \times K[x]/\langle p_r\rangle \\ g + \langle f\rangle & \mapsto & (g + \langle p_1\rangle, \ldots, g + \langle p_r\rangle) \end{array}$$

Ist $K = \mathbb{F}_q$ ein endlicher Körper, dann

$$K[x]/\langle p_i\rangle \cong \mathbb{F}_{q^{d_i}}$$

mit $d_i = \deg(p_i)$.

Lemma 7.4.2 *Sei $K = \mathbb{F}_q$ und $f = p_1 \cdot \ldots \cdot p_r \in K[x]$ quadratfrei mit p_i irreduzibel vom Grad $d = \deg(f)$. Die Abbildung*

$$\begin{array}{rccc} L: & K[x]/\langle f\rangle & \to & K[x]/\langle f\rangle \\ & a & \mapsto & a^q - a \end{array}$$

ist ein K-Vektorraumendomorphismus und

$$\ker(L) \cong K^r$$

insbesondere ist $|\ker(L)| = q^r$ und

$$r = d - \operatorname{rang}(L).$$

Beweis. Es ist

$$L(a+b) = (a+b)^q - a - b = a^q + b^q - a - b = L(a) + L(b)$$

wegen

$$(a+b)^q = \sum_{i=0}^{q} \binom{q}{i} a^i b^{q-i} = a^p + b^p$$

denn q teilt jedes $\binom{q}{i}$ mit $1 < i < q$. Weiter gilt

$$L(\lambda a) = (\lambda a)^q - \lambda a = \lambda L(a),$$

denn für alle $\lambda \in K$ gilt $\lambda^q = \lambda$, da $|K^\times| = q - 1$.

Vermöge des Ringisomorphismus

$$\begin{array}{ccc} K[x]/\langle f\rangle & \to & K[x]/\langle p_1\rangle \times \ldots \times K[x]/\langle p_r\rangle \\ a & \mapsto & (a_1, \ldots, a_r) \end{array}$$

aus Lemma 7.4.1 ist $a^q = a$ genau dann, wenn $a_i^q = a_i$ für alle i. Da $x^q - x$ maximal q Nullstellen in dem Körper $K[x]/\langle p_i\rangle$ hat, sind die Nullstellen genau die Elemente von K, also

$$a^q = a \iff a_i \in K \ \forall i.$$

■

Bemerkung 7.4.3 *Sei $M_B^B(L) \in K^{d\times d}$ die darstellende Matrix von L bezüglich der Basis $\bar{1}, \bar{x}, ..., \bar{x}^{d-1}$ von $K[x]/\langle f\rangle$. Eine Basis von* $\ker(L)$ *erhält man mittels des Gauß-Algorithmus 4.1.*

Aus der Basis erhalten wir die Faktorisierung:

Lemma 7.4.4 *Sei $K = \mathbb{F}_q$ und $f = p_1 \cdot ... \cdot p_r \in K[x]$ quadratfrei mit p_i irreduzibel und $r > 1$.*

1) Ist $g \in K[x]$ mit $\bar{g}^q = \bar{g} \in K[x]/\langle f\rangle$, also $f \mid (g^q - g)$, dann gilt

$$f = \prod_{y\in K} \mathrm{ggT}(f, g - y)$$

2) Sei $\bar{g}_1, ..., \bar{g}_r$ eine Basis von $\ker(L)$ *mit $\bar{g}_1 = 1$. Dann gibt es $y_2, ..., y_r \in K$ sodass die*

$$\mathrm{ggT}(f, g_j - y_j)$$

für $j = 2, ..., r$ nichttriviale Teiler von f sind.

Beweis.

1) Wie im Beweis von Lemma 7.4.2 gezeigt, ist $\bar{g} \in \ker(L)$ genau dann, wenn es (vermöge des Ringisomorphismus aus Bemerkung 7.4.1) $y_1, ..., y_r \in K$ gibt mit

$$g \equiv y_j \bmod p_j$$

Für jedes $y \in K$ ist damit

$$\mathrm{ggT}(f, g - y) = \prod_{y_j = y} p_j$$

2) Sei $r \geq 2$. Für jedes g_j mit $j \geq 2$ gibt es ein $y_j \in K$ mit

$$g_j \equiv y_j \bmod p_1$$

und es gibt ein $i \geq 2$ mit

$$g_j \not\equiv y_j \bmod p_i$$

denn sonst wäre

$$(g_j, ..., g_j) = y_j(1, ..., 1) \in K[x]/\langle p_1\rangle \times ... \times K[x]/\langle p_r\rangle$$

also

$$\bar{g}_j = y_j\bar{1} = y_j \cdot \bar{g}_1 \in K[x]/\langle f\rangle$$

im Widerspruch zur K-Basiseigenschaft.

Der $\mathrm{ggT}(f, g_j - y_j)$ ist somit durch p_1 teilbar, aber nicht durch p_i, ist also ein nichttrivialer Teiler von f.

Algorithmus 7.2 Berlekamp

Input: $K = \mathbb{F}_q$, $f \in K[x]$ quadratfrei vom Grad $n > 1$.
Output: $f = \prod_i f_i$ mit f_i irreduzibel.

1: Berechne für $M_B^B(L) \in K^{n\times n}$ mit $B = (1, ..., x^{n-1})$ eine Basis

$$v_1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, v_2, ..., v_r \in K^n$$

des Kerns.
2: $F = \{f\}$
3: **for all** $j = 1, ..., r$ **do**
4: $g_j = \sum_i B_i \cdot v_{j,i}$ (konvertiere Vektor in Polynom)
5: **for all** $y \in K$ **do**
6: **for all** $h \in F$ **do**
7: $d = \mathrm{ggT}(h, g_j - y)$
8: **if** d nichttrivialer Teiler von h **then**
9: $F = (F \backslash \{h\}) \cup \{d, \frac{h}{d}\}$
10: **if** $|F| = r$ **then**
11: **return** F

■

Dass wir den ggT mit den Polynomen h in der Menge F bilden dürfen, ohne die Basis $v_1, ..., v_r$ neu aufzustellen zeigt (Beweis Übung):

Lemma 7.4.5 *Mit der Notation wie oben: Ist $h \in K[x]$ mit $h \mid f$ sodass $\mathrm{ggT}(h, g_j - y)$ ein trivialer Teiler von h für alle $y \in K$, dann ist h irreduzibel.*

Beispiel 7.4.6 *Sei $K = \mathbb{F}_2$ und $f = x^3 + 1$. Für $B = (1, x, x^2)$ ist*

$$x^0 \equiv 1 \bmod f$$
$$x^2 \equiv x^2 \bmod f$$
$$x^4 \equiv x \bmod f$$

also

$$M_B^B(a \mapsto a^q) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

d.h.

$$M_B^B(L) = M_B^B(a \mapsto a^q) - E = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

und somit eine Basis des Kerns

$$v_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \qquad v_2 = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

Damit ist

$$g_1 = 1$$
$$g_2 = x^2 + x$$

also

$$\mathrm{ggT}(g_2 - 0, f) = x + 1$$
$$\mathrm{ggT}(g_2 - 1, f) = x^2 + x + 1$$

und somit

$$x^3 + 1 = (x + 1)(x^2 + x + 1).$$

7.5 Polynomfaktorisierung über $\mathbb{Z}$ mit LLL

Zum Abschluss wollen wir noch die Grundidee eines Faktorisierungsverfahrens für Polynome über $\mathbb{Z}$ vorstellen, das den LLL-Algorithmus verwendet und dem Berlekamp-Algorithmus nachgeschaltet werden kann.

Lemma 7.5.1 *Sei $f \in \mathbb{Z}[x]$. Für die Abbildung*

$$\begin{array}{rccc} \phi: & \mathbb{Z}[x]\backslash\{0\} & \to & \mathbb{Z}(x) \\ & g & \mapsto & \frac{fg'}{g} \end{array}$$

gilt

$$\phi(gh) = \phi(g) + \phi(h)$$

Ist g ein Teiler von f, so gilt $\phi(g) \in \mathbb{Z}[x]$.

Beweis. Klar mit Produktregel. ■

Für $n = \deg(f)$ identifizieren wir Elemente von $\mathbb{Z}[x]$ vom Grad $\leq n$ mit Vektoren in $\mathbb{Z}^{n+1}$. Die Abbildung ϕ bildet dann Teiler von f also auf Elemente von $\mathbb{Z}^n$ ab. Das Produkt der Teiler ist im Bild eine Summe. Die Teiler sollten kleine Koeffizienten haben. Haben wir eine Faktorisierung

$$f = g_1 \cdot \ldots \cdot g_r \in \mathbb{F}_p[x]$$

(oder allgemeiner ein sogenanntes Hensel-Lifting von einer solchen Faktorisierung nach $\mathbb{Z}/p^r$), suchen wir $S \subset \{1, ..., r\}$ mit $\prod_{i \in S} g_i \in \mathbb{Z}[x]$, d.h. welche Faktoren müssen wir in $\mathbb{Z}$ kombinieren? Vermöge dem Lemma müssen wir also folgendes Problem lösen: Gegeben

$$G_i = \phi\left(\frac{fg'}{g}\right)$$

finde $e_i \in \{0,1\}$, sodass $\sum e_i G_i$ kurz modulo p. Dazu suchen wir in dem Gitter erzeugt von den Spalten der Matrix

$$\left(\begin{array}{ccc|c} G_1 & \cdots & G_r & p \cdot E_n \\ \hline & E_r & & 0 \end{array}\right)$$

kurze Vektoren.

Beispiel 7.5.2 *Für*

$$f = x^4 + 3x + 1$$

erhalten wir in $\mathbb{F}_{10009}[x]$ *die Faktorisierung*

$$f = (x + 2971) \cdot (x + 7038) \cdot (x + 7041) \cdot (x + 2968).$$

Somit ist

$$\begin{aligned} G_1 &= x^3 + 7038 \cdot x^2 + 8905 \cdot x + 7041 \\ G_2 &= x^3 + 2971 \cdot x^2 + 8905 \cdot x + 2968 \\ G_3 &= x^3 + 2968 \cdot x^2 + 1097 \cdot x + 2971 \\ G_4 &= x^3 + 7041 \cdot x^2 + 1097 \cdot x + 7038 \end{aligned}$$

In dem Bild von

$$\left(\begin{array}{cccc|ccccc} 0 & 0 & 0 & 0 & 10009 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 10009 & 0 & 0 & 0 \\ 7038 & 2971 & 2968 & 7041 & 0 & 0 & 10009 & 0 & 0 \\ 8905 & 8905 & 1097 & 1097 & 0 & 0 & 0 & 10009 & 0 \\ 7041 & 2968 & 2971 & 7038 & 0 & 0 & 0 & 0 & 10009 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

sind die kürzesten Vektoren

$$\left(\begin{array}{c} 0 \\ 2 \\ 3 \\ -7 \\ -3 \\ \hline 0 \\ 1 \\ 0 \\ 1 \end{array}\right), \left(\begin{array}{c} 0 \\ 2 \\ -3 \\ -7 \\ 3 \\ \hline 1 \\ 0 \\ 1 \\ 0 \end{array}\right)$$

entsprechend den Produkten

$$\begin{aligned} (x + 7038) \cdot (x + 2968) &= x^2 + 3x + 1 \\ (x + 2971) \cdot (x + 7041) &= x^2 - 3x + 1 \end{aligned}$$

in $\mathbb{F}_{10009}[x]$ *und somit der Faktorisierung*

$$f = x^4 - 7x^2 + 1 = (x^2 + 3x + 1) \cdot (t^2 - 3x + 1)$$

in $\mathbb{Z}[x]$.

7.6 Übungen

Übung 7.1 *Sei*

$$R = \mathbb{Z}\left[\sqrt{-5}\right] = \left\{a + b\sqrt{-5} \mid a, b \in \mathbb{Z}\right\} \subset \mathbb{C}$$

1) *Bestimmen Sie die Einheitengruppe* $R^\times$.

2) *Zeigen Sie, dass* R *nicht faktoriell ist.*

3) *Zeigen Sie, dass* R *Noethersch ist.*

   Hinweis: Corollar 4.2.4.

Übung 7.2 *Sei* p *eine Primzahl,* $\mathbb{F}_p = \mathbb{Z}/p = \{\overline{0}, ..., \overline{p-1}\}$ *der Körper mit* p *Elementen und* $f \in \mathbb{F}_p[x_1, ..., x_n]$ *ein nicht-konstantes Polynom. Die Anzahl der möglichen Werte* $f(x)$*, die* f *für jeden Punkt* $x \in \mathbb{F}_p^n$ *annehmen kann, ist* $|\mathbb{F}_p| = p$*. Setzen Sie voraus, dass* f *für zufällig gewähltes* x *alle diese Werte mit gleicher Wahrscheinlichkeit* $\frac{1}{p}$ *annimmt.*

1) *Wie hoch ist die Wahrscheinlichkeit, dass* $f(x) = 0$ *für zufälliges* x*, falls* f *das Produkt von zwei irreduziblen Polynomen ist.*

2) *Entwickeln Sie ein Verfahren, das Irreduzibilität probabilistisch testet.*

   Hinweis: Betrachten Sie die relative Häufigkeit der Nullstellen von f *für eine Stichprobe aus* $N \geq 0$ *zufällig gewählten Punkten* $x \in \mathbb{F}_p^n$.

3) *Erproben Sie Ihre Methode an einfachen Beispielen und vergleichen Sie mit dem* SINGULAR*-Kommando* `factorize`.

Übung 7.3 *Sei* R *ein kommutativer Ring mit* 1*. Zeigen Sie, dass für die formale Ableitung*

$$(-)' : R[x] \to R[x],\ \textstyle\sum_{i=0}^{d} a_i x^i \mapsto \sum_{i=0}^{d} i a_i x^{i-1}$$

gilt $f' = 0$ *für* $f \in R$ *und*

$$\begin{aligned}(f+g)' &= f' + g' \\ (f \cdot g)' &= f' \cdot g + f \cdot g' \\ (f^n)' &= n \cdot f^{n-1} \cdot f'\end{aligned}$$

für alle $f, g \in R[x]$, $n \in \mathbb{N}$.

7.7 Praktische Aufgaben

Übung 7.4 *1) Schreiben Sie eine Funktion, die alle normierten, irreduziblen Polynome vom Grad $\leq d$ in $\mathbb{F}_p[x]$ aufzählt.*

2) Implementieren Sie die Faktorisierung von Polynomen $f \in \mathbb{F}_p[x]$ mittels Probedivision.

Übung 7.5 *Implementieren Sie das Verfahren aus Aufgabe 7.2.*

Übung 7.6 *Schreiben Sie eine Funktion, die für Polynome f in $\mathbb{Q}[x]$ oder in $\mathbb{F}_p[x]$ die quadratfreie Faktorisierung bestimmt.*

Übung 7.7 *Implementieren Sie den Algorithmus von Berlekamp zur Faktorisierung eines quadratfreien Polynoms $f \in \mathbb{F}_p[x]$.*

Vergleichen Sie die Laufzeit anhand von Beispielen mit der Probedivision.

Index

Literaturverzeichnis

[1] Alford, W. R.; Granville, A.; Pomerance, C.: *There are infinitely many Carmichael numbers*. Ann. Math. 139, 703-722 (1994).

[2] The Axiom Group: *Axiom*, http://www.axiom-developer.org/ (2012).

[3] Bezanson, J.; Edelman, A.; Karpinski, S.; Shah, V.: *Julia: A Fresh Approach to Numerical Computing*, SIAM Review, 59 (2017), 65–98, https://julialang.org/.

[4] Bosma, W.; Cannon J.; Playoust C.: *The Magma algebra system. I. The user language*, J. Symbolic Comput., 24 (1997), 235–265.

[5] Decker, W.; Greuel, G.-M.; Pfister, G.; Schönemann, H.: SINGULAR *4-1-1 — A computer algebra system for polynomial computations*, available at http://www.singular.uni-kl.de (2018).

[6] Endrass, S.: *Surf*, http://surf.sourceforge.net

[7] Grayson, D. R.; Stillman, M. E.: *Macaulay2, a software system for research in algebraic geometry*, available at http://www.math.uiuc.edu/Macaulay2/ (2009).

[8] Greuel, G.-M.; Meyer, H.; Stussak, Ch.: *Surfer*, http://www.imaginary-exhibition.com/surfer.php (2008).

[9] Greuel, G.-M.; Pfister, G: A Singular introduction to commutative algebra, Springer (2002).

[10] Hearn, A. C.: *REDUCE 3.8*, available at http://reduce-algebra.com/ (2009).

[11] Maple (Waterloo Maple Inc.): *Maple 16*, http://www.maplesoft.com/ (2012).

[12] Maxima: *Maxima, a Computer Algebra System*. Version 5.25.1, available at http://maxima.sourceforge.net/ (2011).

[13] The GAP Group, *GAP – Groups, Algorithms, and Programming, Version 4.5.6*, available at http://www.gap-system.org (2012).

[14] The PARI-Group, *PARI/GP, version 2.5.3*, available at http://pari.math.u-bordeaux.fr/ (2012).

[15] Gawrilow, E.; Joswig, M.: *polymake: a framework for analyzing convex polytopes. Polytopes—combinatorics and computation* (Oberwolfach, 1997), 43–73, DMV Sem., 29, Birkhäuser, Basel, 2000, available at http://www.polymake.org/.

[16] Wolfram Research, Inc.: *Mathematica Edition: Version 7.0* (2008).