

Die approbierte Originalversion dieser Diplom-/Masterarbeit ist an der
Hauptbibliothek der Technischen Universität Wien aufgestellt
(<http://www.ub.tuwien.ac.at>).

The approved original version of this diploma or master thesis is available at the
main library of the Vienna University of Technology
(<http://www.ub.tuwien.ac.at/englweb/>).

Technische Universität Wien

DIPLOMARBEIT

Algorithmen in der Computeralgebra für Polynomideale und -moduln

ausgeführt am Institut für
Diskrete Mathematik und Geometrie
der Technischen Universität Wien

unter Anleitung von Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Alois Panholzer

durch

Alexander Zapletal
Bürgergasse 17-19/6/7
1100 Wien

Datum

Unterschrift

Tolle lege, tolle lege!
(Nimm und lies, nimm und lies!)

AUGUSTINUS

Vorwort

Die vorliegende Arbeit entstand als meine Diplomarbeit am Institut für Diskrete Mathematik und Geometrie an der Technischen Universität Wien unter der Betreuung von Prof. Alois Panholzer. Sie ist thematisch an einem Schnittpunkt zwischen Informatik und Mathematik angesiedelt. Dabei werden hauptsächlich Teilgebiete der Computeralgebra behandelt. Zentrale Themen sind Gröbnerbasen, bestimmte Operationen auf Polynomidealen und das Lösen von Polynomgleichungssystemen.

Ich möchte mich an dieser Stelle bei meinen Eltern bedanken, die mir durch ihre Unterstützung mein Studium ermöglicht haben. Außerdem gilt mein Dank meinem Betreuer Prof. Panholzer, der mir von Anfang an sehr hilfreich zur Seite stand.

Wien, März 2004

Alexander Zapletal

Inhaltsverzeichnis

Einleitung	1
I Grundlagen und Gröbnerbasen	3
1 Grundlagen	5
1.1 Polynome	5
1.2 Ideale	6
1.3 Noethersche Ringe	8
1.4 Moduln	10
2 Gröbnerbasen	11
2.1 Termordnungen	11
2.2 Der Algorithmus MULTIVARDIV	14
2.3 Normalformen	15
2.4 Der Algorithmus NORMALF	15
2.5 Gröbnerbasen	16
2.6 Reduzierte Gröbnerbasen	18
3 Berechnung von Gröbnerbasen	20
3.1 Der Algorithmus BUCHBERGER	22
3.2 Der Algorithmus ISTGRÖBNER	23
3.3 Der Algorithmus REDGB	23
3.4 Verbesserung des BUCHBERGER-Algorithmus	25
II Operationen auf Polynomidealen	27
4 Idealgleichheitstest	29
4.1 Der Algorithmus IDEALGLEICHHEIT	29
5 Idealmitgliedschaft	30
5.1 Idealmitgliedschaftstest	30
5.2 Der Algorithmus IDEALMITGLIED	31
5.3 Darstellung bzgl. eines Erzeugendensystems	32
5.4 Der Algorithmus IDEALBASIS	33

6	Radikalmitgliedschaftstest	35
6.1	Der Algorithmus RADIKALMITGLIED	36
7	Berechnung von Idealschnitt und Idealquotient	38
7.1	Der Algorithmus IDEALSCHNITT	39
7.2	Der Algorithmus IDEALQUOT	40
III	Systeme von Polynomgleichungen	42
8	Test: $\mathcal{N}(F) = \emptyset$	44
8.1	Der Algorithmus KEINELSG	45
9	Test: $\mathcal{N}(F) = \mathcal{N}(G)$	46
9.1	Der Algorithmus LSGMENGENGLEICH	47
10	Schranke für $\mathcal{N}(F)$	48
10.1	Der Algorithmus LSGMENGESCHRANKE	50
11	Berechnung von $\mathcal{N}(F)$	52
11.1	Der Algorithmus LSG	53
IV	Systeme linearer Gleichungen mit Koeffizienten in Polynomringen	55
12	Noethersche Moduln	57
13	Termordnungen	59
14	Gröbnerbasen von Untermoduln	61
15	Lineare Gleichungen mit Koeffizienten in $K[x]$	62
15.1	Der Algorithmus INHOMOGENELSG	65
15.2	Der Algorithmus KERNGAMMA	65
15.3	Der Algorithmus KERNPHI	66
V	Primärzerlegung von Idealen	69
A	Algorithmen in MAPLE	76
A.1	Der Algorithmus MULTIVARDIV	76
A.2	Der Algorithmus NORMALF	77
A.3	Der Algorithmus BUCHBERGER	79
A.4	Der Algorithmus ISTGRÖBNER	79
A.5	Der Algorithmus REDGB	80
A.6	Der Algorithmus IDEALGLEICHHEIT	81
A.7	Der Algorithmus IDEALMITGLIED	81
A.8	Der Algorithmus IDEALBASIS	81

A.9	Der Algorithmus IDEALSCHNITT	83
A.10	Der Algorithmus IDEALQUOT	83
A.11	Der Algorithmus KEINELSG	84
A.12	Der Algorithmus RADIKALMITGLIED	84
A.13	Der Algorithmus LSGMENGENGLEICH	84
A.14	Der Algorithmus LSGMENGESCHRANKE	85
A.15	Der Algorithmus LSG	86
B	Funktionen und Algorithmen in MAPLE zu Teil IV	88
B.1	Die Funktion Mbasecoeffindex	88
B.2	Die Funktion Mtestorder	88
B.3	Die Funktion Msupp	89
B.4	Die Funktion MsuppL	90
B.5	Die Funktion Mleadterm	90
B.6	Die Funktion Mleadcoeff	91
B.7	Die Funktion Msvector	91
B.8	Die Funktion Msvector2	92
B.9	Der Algorithmus MMULTIVARDIV	93
B.10	Der Algorithmus MNORMALF	94
B.11	Der Algorithmus MBUCHBERGER	95
B.12	Die Funktion MbuchbergerKoeff	95
B.13	Der Algorithmus MISTGRÖBNER	97
B.14	Der Algorithmus MREDGB	97
B.15	Der Algorithmus INHOMOGENELSG	98
B.16	Der Algorithmus KERNGAMMA	98
B.17	Der Algorithmus KERNPHI	99
	Symbolverzeichnis	101
	Literaturverzeichnis	102
	Index	103

Liste der Algorithmen

1	MULTIVARDIV	14
2	NORMALF	16
3	BUCHBERGER	23
4	ISTGRÖBNER	24
5	REDGB	25
6	IDEALGLEICHHEIT	29
7	IDEALMITGLIED	31
8	IDEALBASIS	34
9	RADIKALMITGLIED	37
10	IDEALSCHNITT	40
11	IDEALQUOT	41
12	KEINELSG	45
13	LSGMENGENGLEICH	47
14	LSGMENGESCHRANKE	51
15	LSG	53
16	INHOMOGENELSG	66
17	KERNGAMMA	67
18	KERNPHI	68

Einleitung

*So eine Arbeit wird eigentlich nie fertig,
man muss sie für fertig erklären,
wenn man nach Zeit und Umständen das Mögliche getan hat.*

JOHANN WOLFGANG VON GOETHE, Italienische Reise.

Die vorliegende Arbeit behandelt verschiedene Aspekte von Polynomidealen in der Computeralgebra. Dabei steht vor allem die algorithmische Umsetzung im Vordergrund. Jedes beschriebene Verfahren und jeder beschriebene Test wird als Algorithmus mit genauer Spezifikation der Ein- und Ausgabewerte formuliert, dessen Termination und Korrektheit bewiesen und mit einem Beispiel dessen Verwendung illustriert. Darüber hinaus findet sich im Anhang eine Implementierung aller Algorithmen in MAPLE 8.

Die Arbeit ist in fünf große Teile gegliedert, die inhaltlich wie folgt aufgebaut sind:

Der erste Teil enthält alle wichtigen und grundlegenden Definitionen, die in den weiteren Kapiteln benötigt werden. Insbesondere werden Gröbnerbasen und deren wichtigste Eigenschaften eingeführt und erläutert. In diesem Zusammenhang wird auch der BUCHBERGER-Algorithmus angeführt, der dazu dient, eine Gröbnerbasis eines vorgegebenen Polynomideals zu berechnen. Gröbnerbasen werden in weiterer Folge für die Implementierung aller behandelten Algorithmen verwendet.

Im zweiten Teil wird gezeigt, wie verschiedene Tests auf Polynomidealen mit Hilfe von Gröbnerbasen algorithmisch durchgeführt werden können. Außerdem werden in diesem Teil zwei Algorithmen zur Berechnung des Durchschnitts und des Quotienten zweier Ideale vorgestellt.

Der dritte Teil behandelt eine der Hauptanwendungen von Gröbnerbasen: Systeme von Polynomgleichungen. Dabei werden Tests für die folgenden Fragestellungen formuliert und implementiert: hat ein vorgegebenes Polynomgleichungssystem Lösungen, haben zwei vorgegebene Polynomgleichungssysteme die selben Lösungen und wie viele Lösungen hat ein vorgegebenes Polynomgleichungssystem maximal. Außerdem wird gezeigt, wie man alle Lösungen eines Systems von Polynomgleichungen berechnen kann, sofern die Lösungsmenge endlich ist.

Im vierten Teil werden Systeme linearer Gleichungen mit Koeffizienten in Polynomringen behandelt. Statt der Ideale, die in den bisherigen Teilen die zentrale algebraische Struktur

waren, haben dort die Moduln diese Rolle. Es zeigt sich, dass die meisten Begriffe und Sätze aus dem ersten Teil analog übernommen werden können.

Der fünfte Teil zeigt, dass jedes Ideal in einem noetherschen Ring in sogenannte primäre Ideale zerlegt werden kann. Diese Primärzerlegung ist außerdem in einem gewissen Sinn eindeutig.

Anhang A beinhaltet die Implementierung sämtlicher behandelten Algorithmen aus den ersten drei Teilen in MAPLE 8.

In Anhang B finden sich die Implementierungen der Funktionen und Algorithmen, die im vierten Teil behandelt werden.

Bemerkung: Es hat sich gezeigt, dass der MAPLE-Befehl `gbasis`, der eigentlich eine reduzierte Gröbnerbasis (siehe Abschnitt 2.6 ab Seite 18) zurückliefern sollte, nicht immer gemäß seiner Spezifikation funktioniert: MAPLE normiert in gewissen Fällen die zurückgelieferte Gröbnerbasis nicht richtig, sodass der Leitkoeffizient nicht immer 1 ist. `gbasis` dürfte in MAPLE so implementiert sein, dass alle Koeffizienten in $\mathbb{Z}$ liegen.

Beispiel:

```
f1:=x1*x2+x1^2*x2+x2*x3:
f2:=x1*x2^2+x2^2+x1*x3:
f3:=x1+x1*x2^2+x3:
F:={f1,f2,f3}:
B:=gbasis(F,tdeg(x1,x2,x3)):
leadcoeff(B[4],tdeg(x1,x2,x3));
2
```

Teil I

Grundlagen und Gröbnerbasen

Im 1. Teil werden die Grundlagen für die weiteren Kapitel behandelt.

Dabei werden im 1. Kapitel Polynomringe über einem Körper K und Ideale (spezielle Unterringe eines Rings R) definiert. Dann wird der für die ganze Arbeit zentrale Begriff der noetherschen Ringe eingeführt und der ebenso wichtige HILBERTsche Basissatz formuliert und bewiesen.

Im 2. Kapitel wird auf den spezielleren Fall der Ideale eines multivariaten Polynomrings eingegangen. Nach der Einführung von Begriffen wie Termordnung, Leitterm und Division mit Rest werden Gröbnerbasen definiert und deren wichtigste Eigenschaften behandelt. Zum Abschluss dieses Kapitels wird gezeigt, dass zu jedem Polynomideal eine eindeutig bestimmte reduzierte Gröbnerbasis existiert.

Wie Gröbnerbasen berechnet werden können, wird im 3. Kapitel gezeigt. Der Algorithmus, der dabei zum Einsatz kommt, ist der BUCHBERGER-Algorithmus. Außerdem wird gezeigt, wie dieser Algorithmus erweitert werden kann, um aus einer beliebigen Gröbnerbasis eines Polynomideals dessen reduzierte Gröbnerbasis zu berechnen.

Kapitel 1

Grundlagen

Die Grundlage ist das Fundament der Basis.

LE CORBUSIER

In diesem Kapitel werden wichtige Begriffe und Notationen definiert und eingeführt, von denen in den weiteren Kapiteln häufig Gebrauch gemacht wird. Außerdem wird der HILBERTsche Basissatz, der in der Theorie der Gröbnerbasen (siehe Kapitel 2 ab Seite 11) eine zentrale Rolle spielt, formuliert und bewiesen.

1.1 Polynome

Aus kleinem Anfang entspringen alle Dinge.

MARCUS TULLIUS CICERO

Sei R ein kommutativer Ring mit Einselement.

Definition 1.1 *Seien*

$$R[x] := \{a_0 + a_1x + \cdots + a_nx^n \mid n \in \mathbb{N}_{\geq 0}, a_i \in R\},$$

$$+ : \begin{cases} R[x] \times R[x] \rightarrow R[x] \\ \sum_{k=0}^n a_k x^k + \sum_{k=0}^n b_k x^k \mapsto \sum_{k=0}^n (a_k + b_k) x^k \end{cases}$$

und

$$\cdot : \begin{cases} R[x] \times R[x] \rightarrow R[x] \\ \sum_{k=0}^n a_k x^k \cdot \sum_{k=0}^n b_k x^k \mapsto \sum_{k=0}^{2n} \left(\sum_{i=0}^k a_i b_{k-i} \right) x^k \end{cases}.$$

Dann heißt $R[x]$ mit den gerade definierten Operationen $+$ und $\cdot$ **Polynomring in x über R** .

Lemma 1.1 $R[x]$ ist ein kommutativer Ring mit Einselement.

Beweis: trivial. $\square$

Definition 1.2 Sei $f(x) = \sum_{k=0}^n a_k x^k \in R[x]$ mit $a_n \neq 0$. Dann heißt n der **Grad** von $f(x)$ (in Zeichen $n = \text{grad}(f(x))$) und a_n heißt der **führende Koeffizient** von $f(x)$.

Definition 1.3 Allgemein wird der Polynomring in n Variablen $x_1, \dots, x_n$ mit $n \geq 1$ über einem Ring R induktiv definiert durch

$$(1) \ n = 1: R[x_1] := R[x],$$

$$(2) \ n \geq 2: R[x_1, \dots, x_n] := (R[x_1, \dots, x_{n-1}])[x_n].$$

Satz 1.1 Mit der Konvention $\forall 1 \leq i \leq n : x_i^0 = 1$ gilt

$$R[x_1, \dots, x_n] = \left\{ \sum_{0 \leq i_1, \dots, i_n \leq m} a_{i_1 \dots i_n} x_1^{i_1} \cdots x_n^{i_n} \mid m \in \mathbb{N}, a_{i_1 \dots i_n} \in R \right\}.$$

Beweis: Durch vollständige Induktion nach n . $\square$

Definition 1.4 Seien $m \in \mathbb{N}$ und $0 \neq f(x_1, \dots, x_n) = \sum_{0 \leq i_1, \dots, i_n \leq m} a_{i_1 \dots i_n} x_1^{i_1} \cdots x_n^{i_n} \in R[x_1, \dots, x_n]$. Dann ist der **Grad** von $f(x_1, \dots, x_n)$ definiert als:

$$\text{grad}(f(x_1, \dots, x_n)) := \max_{0 \leq i_1, \dots, i_n \leq m} \left\{ \sum_{k=1}^n i_k \mid a_{i_1 \dots i_n} \neq 0 \right\}.$$

1.2 Ideale

*Ideale sind wie Sterne.
Wir erreichen sie niemals,
aber wie die Seefahrer auf dem Meer,
richten wir unseren Kurs nach ihnen.*

CARL SCHURZ

Definition 1.5 Sei R ein kommutativer Ring mit Einselement und $\emptyset \neq I \subseteq R$. Dann heißt I ein **Ideal** von R (in Zeichen $I \trianglelefteq R$) genau dann, wenn

$$(1) \ \forall a, b \in I : a + b \in I.$$

$$(2) \ \forall a \in I, r \in R : ar \in I.$$

Lemma 1.2 Seien R ein kommutativer Ring mit Einselement und I ein Ideal von R . Dann gilt: $1 \in I \iff I = R$.

Beweis: trivial. $\square$

Satz 1.2 Sei R ein kommutativer Ring mit Einselement, $a \in R$ und

$$\langle a \rangle_R := \{ar \mid r \in R\}.$$

Dann ist $\langle a \rangle_R$ das kleinste Ideal von R , welches a enthält.

Beweis: $\langle a \rangle_R$ ist ein Ideal: Seien $r_1, r_2 \in R$. $ar_1, ar_2 \in \langle a \rangle_R \implies ar_1 + ar_2 = a(r_1 + r_2) \in \langle a \rangle_R$; für ein $r \in R$ ist $r(ar_1) = a(rr_1) \in \langle a \rangle_R$.

Da jedes Ideal, welches a enthält, alle ar mit $r \in R$ und damit $\langle a \rangle_R$ enthalten muss, ist $\langle a \rangle_R$ das kleinste Ideal, welches a enthält. $\square$

Bemerkungen:

- (1) $I \trianglelefteq R$ ist ein Unterring von R .
- (2) $\langle a \rangle_R$ heißt **das von a erzeugte Ideal** von R .
- (3) Analog bezeichnet

$$\langle a_1, \dots, a_n \rangle_R := \{a_1r_1 + \dots + a_nr_n \mid r_1, \dots, r_n \in R\}$$

das von $a_1, \dots, a_n$ erzeugte Ideal von R und ist das kleinste Ideal von R , welches $a_1, \dots, a_n$ enthält. Das Ideal $\langle a_1, \dots, a_n \rangle_R$ heißt dann **endlich erzeugt** und $a_1, \dots, a_n$ heißt **endliches Erzeugendensystem** von $\langle a_1, \dots, a_n \rangle_R$.

- (4) Sei $A \subseteq R$. Dann heißt

$$\langle A \rangle_R := \left\{ \sum_{i=1}^n a_i r_i \mid n \in \mathbb{N}_{>0}, r_i \in R \text{ und } a_i \in A \text{ für } 1 \leq i \leq n \right\}$$

das von A erzeugte Ideal von R und ist das kleinste Ideal von R , welches A enthält. Ist A endlich, dann heißt das Ideal $\langle A \rangle_R$ **endlich erzeugt** und A heißt **endliches Erzeugendensystem** von $\langle A \rangle_R$.

- (5) Wenn klar ist, in welchem Ring R gerechnet wird, wird für $\langle a_1, \dots, a_n \rangle_R$ bzw. $\langle A \rangle_R$ abkürzend $\langle a_1, \dots, a_n \rangle$ bzw. $\langle A \rangle$ geschrieben.

Definition 1.6 Seien R ein kommutativer Ring mit Einselement und $I_1, I_2 \trianglelefteq R$.

- (1) $I_1 + I_2 := \{a + b \mid a \in I_1, b \in I_2\}$ heißt **Summe der Ideale I_1 und I_2** .
- (2) $I_1 \cdot I_2 := \left\{ \sum_{i=1}^k a_i b_i \mid a_i \in I_1, b_i \in I_2, k \in \mathbb{N}_{>0} \right\}$ heißt **Produkt der Ideale I_1 und I_2** .
- (3) $I_1 : I_2 := \{r \in R \mid \forall b \in I_2: rb \in I_1\}$ heißt **Quotient der Ideale I_1 und I_2** .

Lemma 1.3 Sei R ein kommutativer Ring mit Einselement. Für $a_i \in R$ gilt: $\langle a_1, \dots, a_n \rangle = \langle a_1 \rangle + \dots + \langle a_n \rangle$.

Beweis: trivial. $\square$

Satz 1.3 Seien R ein kommutativer Ring mit Einselement und $I_1, I_2 \trianglelefteq R$. Dann gilt:

- (1) $I_1 + I_2 \trianglelefteq R$,
- (2) $I_1 \cap I_2 \trianglelefteq R$,
- (3) $I_1 \cdot I_2 \trianglelefteq R$ und
- (4) $I_1 : I_2 \trianglelefteq R$.

Beweis: (1)-(3): trivial.

(4): Seien $r_1, r_2 \in I_1 : I_2$. Dann ist $\forall b \in I_2 : (r_1 + r_2)b = r_1b + r_2b \in I_1$, da $\forall b \in I_2 : r_1b, r_2b \in I_1$. Also: $r_1 + r_2 \in I_1 : I_2$.

Sei $r \in R$. $\forall b \in I_2 : (rr_1)b = r(r_1b) \in I_1$. Also: $rr_1 \in I_1 : I_2$. $\square$

Lemma 1.4 Seien R ein kommutativer Ring mit Einselement und $I_1, I_2, I_3 \trianglelefteq R$. Dann gilt:

$$I_1 : (I_2 + I_3) = (I_1 : I_2) \cap (I_1 : I_3).$$

Beweis: $\subseteq$: Sei $a \in I_1 : (I_2 + I_3)$. Dann gilt: $\forall a_2 \in I_2, a_3 \in I_3 : a(a_2 + a_3) \in I_1$. Insbesondere: $\forall a_2 \in I_2 : aa_2 \in I_1$ und $\forall a_3 \in I_3 : aa_3 \in I_1$, da $0 \in I_2$ und $0 \in I_3$. Daher gilt: $a \in I_1 : I_2$ und $a \in I_1 : I_3 \implies a \in (I_1 : I_2) \cap (I_1 : I_3)$.

$\supseteq$: Sei $a \in (I_1 : I_2) \cap (I_1 : I_3)$. Dann gilt: $\forall a_2 \in I_2 : aa_2 \in I_1 \wedge \forall a_3 \in I_3 : aa_3 \in I_1$.
 $\implies \forall a_2 \in I_2, a_3 \in I_3 : a(a_2 + a_3) = aa_2 + aa_3 \in I_1 \implies a \in I_1 : (I_2 + I_3)$. $\square$

1.3 Noethersche Ringe

*Wie weit diese Methoden reichen werden,
muss erst die Zukunft zeigen.*

EMMY NOETHER

Definition 1.7 Ein kommutativer Ring mit Einselement R heißt **noethersch**, wenn alle Ideale von R endlich erzeugt sind. Also: $\forall I \trianglelefteq R : \exists a_1, \dots, a_k \in R : I = \langle a_1, \dots, a_k \rangle_R$.

Satz 1.4 Sei R ein kommutativer Ring mit Einselement. Die folgenden Aussagen sind äquivalent:

- (1) R ist noethersch.
- (2) Jede echt aufsteigende Folge von Idealen in R ist endlich,
d.h.: Sind $I_1, I_2, \dots$ Ideale von R mit $I_1 \subsetneq I_2 \subsetneq \dots$, dann gilt: $\exists k \in \mathbb{N} : I_k$ maximal in R .

(3) Für alle Teilmengen $F \subseteq R$ gibt es endliche Teilmengen $E \subseteq F$ mit $\langle E \rangle = \langle F \rangle$.

Beweis: (1) $\implies$ (2): Sei $I_1 \subsetneq I_2 \subsetneq \dots$ eine echt aufsteigende Folge von Idealen. Dann ist

$$I := \bigcup_{j \geq 1} I_j$$

ein Ideal. Nach (1) gibt es $a_1, \dots, a_k \in R$ mit $I = \langle a_1, \dots, a_k \rangle_R$. Nach Definition von I gibt es ein m so, dass $a_1, \dots, a_k \in I_m$. Also gilt $I = I_m$ und I_m ist das letzte Element der Folge $I_1 \subsetneq I_2 \subsetneq \dots$.

(2) $\implies$ (3): Sei $F \subseteq R$. Wähle $a_1, a_2, a_3, \dots \in F$ so, dass $a_2 \notin \langle a_1 \rangle, a_3 \notin \langle a_1, a_2 \rangle, \dots$. Dann ist $\langle a_1 \rangle \subsetneq \langle a_1, a_2 \rangle \subsetneq \dots$ eine echt aufsteigende Folge von Idealen. Nach (2) gibt es ein k so, dass $\langle a_1, \dots, a_k \rangle = \langle F \rangle$.

(3) $\implies$ (1): trivial. $\square$

Satz 1.5 (HILBERTScher Basissatz) Sei R ein kommutativer Ring mit Einselement. Wenn R noethersch ist, dann ist auch $R[x_1, \dots, x_n]$ noethersch.

Beweis: Vollständige Induktion nach n . Für $n = 1$: Annahme: Sei $I \trianglelefteq R[x_1]$ ein Ideal, das nicht endlich erzeugt wird. Dann ist I nicht das von 0 erzeugte Ideal.

Man definiert nun eine Folge $(f_i)_{i \geq 1}$ von Elementen aus I induktiv wie folgt: Wähle $f_1 \in I$ mit minimalem Grad d_1 und $f_1 \neq 0$. Seien $f_1, \dots, f_i \in I$ bereits gewählt, so wähle $f_{i+1} \in I \setminus \langle f_1, \dots, f_i \rangle$ mit minimalem Grad d_{i+1} .

Dann gilt $d_1 \leq d_2 \leq \dots$.

Sei nun $c_i \in R$ der führende Koeffizient von f_i . Weil R noethersch ist, gibt es ein $k \in \mathbb{N}$ mit: $\langle c_1, \dots, c_k \rangle_R = \langle c_1, \dots, c_{k+1} \rangle_R$.

Wähle $r_1, \dots, r_k \in R$ so, dass

$$c_{k+1} = \sum_{i=1}^k r_i c_i$$

und definiere

$$g := \underbrace{\sum_{i=1}^k r_i x^{d_{k+1}-d_i} f_i}_{\in \langle f_1, \dots, f_k \rangle} - \underbrace{f_{k+1}}_{\in I \setminus \langle f_1, \dots, f_k \rangle}.$$

Dann gilt: $g \in I \setminus \langle f_1, \dots, f_k \rangle$ (Denn wäre $g \in \langle f_1, \dots, f_k \rangle$, würde aus der Definition von g folgen, dass auch $f_{k+1} \in \langle f_1, \dots, f_k \rangle$. Das wäre ein Widerspruch zur Konstruktion der Folge (f_i) .) und $\text{grad}(g) < d_{k+1} = \text{grad}(f_{k+1})$. Dies ist aber ein Widerspruch zur Minimalität von d_{k+1} .

Für $n > 1$: $R[x_1, \dots, x_n] = (R[x_1, \dots, x_{n-1}])[x_n]$. $R[x_1, \dots, x_{n-1}]$ ist noethersch nach Induktionsannahme und daher nach Fall $n = 1$ auch $(R[x_1, \dots, x_{n-1}])[x_n]$. $\square$

Bemerkung: Jeder Hauptidealring (also auch jeder Körper) ist noethersch und somit auch Polynomringe über diesen. Beispiele dafür sind $\mathbb{Z}[x_1, \dots, x_n]$, $\mathbb{Q}[x_1, \dots, x_n]$, $\mathbb{C}[x_1, \dots, x_n]$ und $\mathbb{Z}_m[x_1, \dots, x_n]$.

1.4 Moduln

Be wise! Generalize!

PICCAYNE SENTINEL

Definition 1.8 Sei R ein kommutativer Ring mit Einselement. Ein **R -Modul** V ist eine kommutative Gruppe, deren Verknüpfung mit $+$ bezeichnet wird, zusammen mit einer Skalarmultiplikation

$$\cdot : \begin{cases} R \times V \rightarrow V \\ (r, v) \mapsto r \cdot v \end{cases},$$

die für alle $r, s \in R$ und $v, v' \in V$ die folgenden Axiome erfüllt:

- (1) $1 \cdot v = v$.
- (2) $(rs) \cdot v = r \cdot (s \cdot v)$.
- (3) $(r + s) \cdot v = r \cdot v + s \cdot v$.
- (4) $r \cdot (v + v') = r \cdot v + r \cdot v'$.

Moduln sind also eine Verallgemeinerung von Vektorräumen auf Ringe; ein K -Modul über einem Körper K ist ein K -Vektorraum.

Definition 1.9 Ein **Unterm modul** eines R -Moduls V ist eine nichtleere Teilmenge von V , die unter Addition und Skalarmultiplikation abgeschlossen ist.

Definition 1.10 Sei R ein kommutativer Ring mit Einselement und $n \in \mathbb{N}_{>0}$. Dann heißt der R -Modul R^n **freier Modul über R** .

Satz 1.6 Die Untermoduln des R -Moduls R sind die Ideale von R .

Beweis: Nach Definition ist ein Ideal eine nichtleere Teilmenge von R , die unter Addition und Multiplikation mit Elementen von R abgeschlossen ist. $\square$

Kapitel 2

Gröbnerbasen

*Nun ging mir eine neue Welt auf.
Ich näherte mich den Gebirgen,
die sich nach und nach entwickelten.*

JOHANN WOLFGANG VON GOETHE, Italienische Reise.

In diesem Kapitel wird nun auf Polynomideale eingegangen. Dabei werden vor allem Gröbnerbasen (das sind spezielle Erzeugendensysteme eines Ideals) behandelt. Gröbnerbasen sind Grundlage für alle Algorithmen in den nächsten Kapiteln. Außerdem wird der Algorithmus MULTIVARDIV formuliert, der eine Verallgemeinerung des Euklidischen Algorithmus auf den multivariaten Fall darstellt. Am Ende dieses Kapitels wird gezeigt, wie für jedes Polynomideal eine eindeutige reduzierte Gröbnerbasis konstruiert werden kann.

Es seien in diesem und allen folgenden Kapiteln K ein Körper und für $n \in \mathbb{N}_{>0}$ $K[\underline{x}] := K[x_1, \dots, x_n]$. Außerdem sei für $i = (i_1, \dots, i_n) \in \mathbb{N}^n$: $\underline{x}^i := x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}$ und $|i| := i_1 + \dots + i_n$.

2.1 Termordnungen

*Gebraucht der Zeit, sie geht so schnell von hinnen,
doch Ordnung lehrt Euch Zeit gewinnen.*

JOHANN WOLFGANG VON GOETHE, Faust I.

Definition 2.1 Die Menge $P \subseteq K[\underline{x}]$ der **Potenzprodukte** (oder **Terme**) in $K[\underline{x}]$ ist definiert durch

$$P := \{\underline{x}^i \mid i \in \mathbb{N}_{\geq 0}^n, n \in \mathbb{N}_{>0}\}.$$

Bemerkung: P ist eine K -Vektorraumbasis von $K[\underline{x}]$; d.h. $f \in K[\underline{x}] \implies f = \sum_{p \in P} c_p p$ mit $c_p \in K$ und nur endlich viele c_p sind ungleich null.

Definition 2.2 Eine Totalordnung $\prec$ auf P heißt **Termordnung** $:\Leftrightarrow$

- (1) $\forall p \in P \setminus \{1\} : 1 \prec p$,
- (2) $\forall p, q, r \in P : p \prec q \implies rp \prec rq$.

Beispiele:

- (1) Lexikographische Ordnung mit $x_1 > x_2 > \dots > x_n$:

$$\underline{x}^i \prec_{lex} \underline{x}^j : \Leftrightarrow \exists k \in \mathbb{N}_{>0} \text{ so, dass } i_1 = j_1, i_2 = j_2, \dots, i_k = j_k \text{ und } i_{k+1} < j_{k+1}.$$

$$\textbf{Beispiel: } 1 \prec_{lex} x_n \prec_{lex} x_n^2 \prec_{lex} x_1 \prec_{lex} x_1^2 \text{ und } x_1^2 x_2 x_3^2 \prec_{lex} x_1^2 x_2 x_3^3 \prec_{lex} x_1^2 x_2^2.$$

- (2) Graduiert-lexikographische Ordnung mit $x_1 > x_2 > \dots > x_n$:

$$\underline{x}^i \prec_{glex} \underline{x}^j : \Leftrightarrow |i| < |j| \vee (|i| = |j| \wedge x^i \prec_{lex} x^j).$$

$$\textbf{Beispiel: } x_1^2 x_2^2 \prec_{glex} x_1^2 x_2 x_3^2 \prec_{glex} x_1^2 x_2 x_3^3.$$

- (3) Graduiert-invers-lexikographische Ordnung mit $x_1 > x_2 > \dots > x_n$:

$$\underline{x}^i \prec_{ilex} \underline{x}^j : \Leftrightarrow |i| < |j| \vee (|i| = |j| \wedge \exists k \in \mathbb{N}_{>0} : i_k > j_k \text{ und } i_{k+1} = j_{k+1}, \dots, i_n = j_n).$$

$$\textbf{Beispiel: } 1 \prec_{ilex} x_3 \prec_{ilex} x_2 \prec_{ilex} x_1 \prec_{ilex} x_3^2 \prec_{ilex} x_2 x_3 \prec_{ilex} x_2 x_3^2 \prec_{ilex} x_2^3 \prec_{ilex} x_3^4 \text{ und } x_1 x_2 x_3^4 x_4^2 \prec_{ilex} x_1 x_2^2 x_3^3 x_4^2.$$

Definition 2.3 Seien $\prec$ eine Termordnung auf P , $0 \neq f = \sum_{p \in P} c_p p \in K[x]$ und $F \subseteq K[x]$.

- (1) $\text{supp}(f) := \{p \in P \mid c_p \neq 0\}$ heißt **Träger von f** .
- (2) $\text{lt}(f) := \max_{\prec} \text{supp}(f)$ heißt **Leitterm von f (bezüglich $\prec$)**.
- (3) $\text{lk}(f) := c_{\text{lt}(f)}$ heißt **Leitkoeffizient von f (bezüglich $\prec$)**.
- (4) $\text{lt}(F) := \{\text{lt}(f) \mid f \in F \setminus \{0\}\}$.

Lemma 2.1 Seien $p, q \in P$ und $p \mid q$. Dann gilt: $p \preceq q$.

Beweis: $p \mid q \implies \exists r \in P : 1 \preceq r \wedge q = p \cdot r \implies p \preceq p \cdot r = q$. $\square$

Lemma 2.2 Seien $f, g \in K[x] \setminus \{0\}$. Dann gilt $\text{lt}(f g) = \text{lt}(f) \text{lt}(g)$.

Inbesondere: $p \in P \implies \text{lt}(p f) = p \text{lt}(f)$.

Beweis: Seien $f = \sum_{p \in P} c_p p$ und $g = \sum_{q \in P} d_q q$. Dann ist $\text{lt}(f g) = \max_{\prec} \{p q \mid \sum_{rs=pq} d_r c_s \neq 0\}$.

Seien $p \in \text{supp}(f) \setminus \{\text{lt}(f)\}$ und $q \in \text{supp}(g) \setminus \{\text{lt}(g)\}$, dann gilt: $p \prec \text{lt}(f)$ und $q \prec \text{lt}(g)$. Daher: $p q \prec \text{lt}(f) \text{lt}(g)$. Das Maximum wird also für $p = \text{lt}(f)$ und $q = \text{lt}(g)$ angenommen. $\square$

Satz 2.1 Sei $Q \subseteq P$, $p \in P$. Dann gilt

$$p \in \langle Q \rangle \iff \exists q \in Q : q \mid p.$$

Beweis: $p \in \langle Q \rangle \iff \exists (f_q)_{q \in Q}$ in $K[\underline{x}]$ so, dass $p = \sum_{q \in Q} f_q q$. Wegen $f_q \in K[\underline{x}]$ gilt $f_q = \sum_{t \in \text{supp}(f_q)} c_{qt} t$ mit $c_{qt} \in K$. Da P eine K -Basis von $K[\underline{x}]$ ist, gilt nun

$$p = \sum_{q \in Q} \sum_{t \in \text{supp}(f_q)} c_{qt} t q \iff \exists t \in P : p = tq. \quad \square$$

Satz 2.2 Jede absteigende Folge in P ist endlich. Insbesondere: Jede Teilmenge von P enthält ein kleinstes Element.

Beweis: Sei $p_1 \succ p_2 \succ p_3 \succ \dots$ eine absteigende Folge in P . Dann ist die Folge $\langle p_1 \rangle \subsetneq \langle p_1, p_2 \rangle \subsetneq \langle p_1, p_2, p_3 \rangle \subsetneq \dots$ aufsteigend, denn nach Lemma 2.1 (Seite 12) wird p_i von $p_1, \dots, p_{i-1}$ nicht geteilt; also folgt aus Satz 2.1 $p_i \notin \langle p_1, \dots, p_{i-1} \rangle$. Nach Satz 1.5 (Seite 9) und Satz 1.4 (Seite 8) ist diese Folge endlich. $\square$

Satz 2.3 (Division mit Rest) F sei eine endliche Teilmenge von $K[\underline{x}] \setminus \{0\}$, $\prec$ sei eine Termordnung auf $P \subseteq K[\underline{x}]$ und $g \in K[\underline{x}]$. Dann gibt es eine Familie $(h_f)_{f \in F}$ in $K[\underline{x}]$ so, dass entweder

$$g = \sum_{f \in F} h_f f$$

oder

$$g \neq \sum_{f \in F} h_f f \quad \text{und} \quad \text{lt}\left(g - \sum_{f \in F} h_f f\right) \notin \langle \text{lt}(F) \rangle_{K[\underline{x}]}$$

gilt. Außerdem gilt $\max_{\prec} \{\text{lt}(h_f f) \mid f \in F, h_f \neq 0\} = \text{lt}(g)$.

Beweis: Siehe Terminations- und Korrektheitsbeweis des Algorithmus MULTIVARDIV (Seite 14). $\square$

Definition 2.4 Es gelten die gleichen Voraussetzungen wie in Satz 2.3. Dann heißt

$$r_{\prec}(g, F) := g - \sum_{f \in F} h_f f$$

ein Rest von g nach Division durch F bzgl. $\prec$.

Bemerkungen:

- (1) $r_{\prec}(g, F)$ ist durch g , F und $\prec$ nicht eindeutig bestimmt.
- (2) Wenn klar oder unbedeutend ist, bzgl. welcher Termordnung $\prec$ der Rest zu berechnen ist, wird für $r_{\prec}(g, F)$ abkürzend $r(g, F)$ geschrieben.

Beispiel: $f_1 := x_1 x_2 + 1$, $f_2 := x_2^2 - 1$, $\prec := \prec_{\text{lex}}$ mit $x_1 > x_2$ und $g := x_1 x_2^2 - x_1$.
 $g = x_2 \cdot f_1 + 0 \cdot f_2 + (-x_1 - x_2) \implies r_{\prec}(g, \{f_1, f_2\}) = -x_1 - x_2$.
 $g = 0 \cdot f_1 + x_1 \cdot f_2 + 0 \implies r_{\prec}(g, \{f_1, f_2\}) = 0$.

2.2 Der Algorithmus MULTIVARDIV

Der Algorithmus MULTIVARDIV berechnet einen Rest von g nach Division durch F bzgl. einer Termordnung $\prec$.

Ähnlich der univariaten Division wird dabei ein $f \in F$ gesucht, für das $\text{lt}(f) \mid \text{lt}(g)$ gilt. Wird so ein f gefunden, wird die Reduktion $g - c \cdot \frac{\text{lt}(g)}{\text{lt}(f)} \cdot f$ mit einem geeignet gewählten Koeffizienten $c \in K$ durchgeführt und das Ergebnis analog weiter reduziert.

Spezifikation: $((q_1, \dots, q_k), r) \leftarrow \text{MULTIVARDIV}(g, F, \prec)$

Führt die Division von g durch F aus.

Eingabe: $g \in K[x]$, $F = (f_1, \dots, f_k)$ mit $f_i \in K[x] \setminus \{0\}$ und eine Termordnung $\prec$.

Ausgabe: $q_1, \dots, q_k, r \in K[x]$ so, dass $g = q_1 f_1 + \dots + q_k f_k + r$ und $\max_{1 \leq i \leq k, q_i \neq 0} \{\text{lt}(q_i f_i) \mid 1 \leq i \leq k, q_i \neq 0\} = \text{lt}(g)$.

begin

$h \leftarrow g$

for $1 \leq i \leq k$ **do** $q_i \leftarrow 0$

while $h \neq 0$ **do**

$red \leftarrow \text{false}$

for $1 \leq i \leq k$ **do**

if $\text{lt}(f_i) \mid \text{lt}(h)$ **then**

$h \leftarrow h - \frac{\text{lk}(h)\text{lt}(h)}{\text{lk}(f_i)\text{lt}(f_i)} f_i, \quad q_i \leftarrow q_i + \frac{\text{lk}(h)\text{lt}(h)}{\text{lk}(f_i)\text{lt}(f_i)}, \quad red \leftarrow \text{true}$

goto A

end

end

A: **if** $red = \text{false}$ **then return** $((q_1, \dots, q_k), h)$

end

return $((q_1, \dots, q_k), h)$

end

Algorithmus 1: MULTIVARDIV

Korrektheit: Terminiert der Algorithmus, so ist entweder $h = g - \sum_{i=0}^k q_i f_i = 0$ oder

$\forall 1 \leq i \leq k : \text{lt}(f_i) \nmid \text{lt}(h) = \text{lt}(g - \sum_{i=0}^k q_i f_i) \quad (\iff \text{lt}(g - \sum_{i=0}^k q_i f_i) \notin \langle \text{lt}(F) \rangle).$ $\square$

Termination: Existiert kein i mit $1 \leq i \leq k$ so, dass $\text{lt}(f_i) \mid \text{lt}(h)$, so terminiert der Algorithmus.

Existiert so ein i , so gilt $\text{lt}(h - \frac{\text{lk}(h)\text{lt}(h)}{\text{lk}(f_i)\text{lt}(f_i)} f_i) \prec \text{lt}(h)$. Nach Satz 2.2 (Seite 13) tritt dann in endlich vielen Schritten der Fall $h = 0$ ein und der Algorithmus terminiert. $\square$

Beispiel: $K = \mathbb{Q}$, $F := (x_1^2, x_1 x_2 + 1)$, $g := x_1 x_2^4 + x_1 x_2^3 + x_1^2 x_2$.
 $((x_2, x_2^3 + x_2^2), -x_2^3 - x_2^2) \leftarrow \text{MULTIVARDIV}(g, F, \prec_{lex})$.

2.3 Normalformen

Die Notwendigkeit schafft die Form.

WASSILY KANDINSKY

Satz 2.4 Sei F eine endliche Teilmenge von $K[\underline{x}] \setminus \{0\}$ und $g \in K[\underline{x}]$. Dann gibt es eine Familie $(h_f)_{f \in F}$ in $K[\underline{x}]$ so, dass

$$\text{supp}(g - \sum_{f \in F} h_f f) \cap \langle \text{lt}(F) \rangle_{K[\underline{x}]} = \emptyset$$

und die Potenzprodukte $p \cdot \text{lt}(f)$ mit $p \in \text{supp}(h_f)$ und $f \in F$ paarweise verschieden sind.

$$\text{normalf}(g, F, \prec) := g - \sum_{f \in F} h_f f$$

heißt dann **eine Normalform von g bzgl. F und $\prec$** und kann mit dem Algorithmus **NORMALF** berechnet werden.

Beweis: Siehe Terminations- und Korrektheitsbeweis des Algorithmus **NORMALF**. $\square$

Beispiel: $F := \{x_1\} \subseteq \mathbb{Q}[x_1, x_2]$, $g := x_1 + x_2^2$.

$$r \prec_{\text{glex}}(g, F) = g.$$

$$\text{normalf}(g, F, \prec_{\text{glex}}) = x_2^2.$$

Bemerkung: $\text{normalf}(g, F, \prec)$ ist durch g , F und $\prec$ im allgemeinen nicht eindeutig bestimmt.

Beispiel: $f_1 := x_1^2$, $f_2 := x_1 x_2 + 1$, $F := \{f_1, f_2\} \subseteq \mathbb{Q}[x_1, x_2]$, $g := x_1^2 x_2 + 1$.

Sowohl $g - x_2 f_1 = 1$ als auch $g - x_1 f_2 = -x_1 + 1$ sind Normalformen von g bzgl. F und $\prec_{\text{glex}}$.

2.4 Der Algorithmus NORMALF

Korrektheit: trivial. $\square$

Termination: Es gilt $\max_{\prec} (\text{supp}(r - c \text{lk}(f)^{-1} p f) \cap \langle \text{lt}(F) \rangle) \prec \max_{\prec} (S)$; also folgt aus Satz 2.2 (Seite 13), dass **NORMALF** terminiert. $\square$

Beispiele:

$$\begin{aligned} (1) \quad & f_1 := x_1 x_2 + x_2, \\ & f_2 := x_1 x_2^2 + x_1 x_2 + x_1^2, \\ & g := 2x_1^3 x_2^2 + x_1 x_2^3 + x_1^2 x_2 + x_2, \\ & -2x_1^4 + x_1^2 + 2x_2 \leftarrow \text{NORMALF}(g, \{f_1, f_2\}, \prec_{\text{glex}}). \end{aligned}$$

Spezifikation: $r \leftarrow \text{NORMALF}(g, F, \prec)$

Berechnet eine Normalform von g bzgl. F und $\prec$.

Eingabe: $g \in K[x]$, eine endliche Menge $F \subseteq K[x] \setminus \{0\}$ und eine Termordnung $\prec$.

Ausgabe: Eine Normalform $\text{normalf}(g, F, \prec)$.

begin

$r \leftarrow g$

repeat

$S \leftarrow \text{supp}(r) \cap \langle \text{lt}(F) \rangle$

if $S \neq \emptyset$ **then**

Wähle $f \in F$ und $p \in P$ so, dass $p \text{ lt}(f) = \max_{\prec} S$.

Sei c der Koeffizient von r bei $p \text{ lt}(f)$.

$r \leftarrow r - c \text{ lk}(f)^{-1} p f$

end

until $S = \emptyset$

return r

end

Algorithmus 2: NORMALF

- (2) $f_1 := x_2^2 + x_3$,
 $f_2 := x_2^3 + x_2 x_3^2$,
 $g := x_1 x_2 + x_2^3 + x_2 x_3$.
 $((0, 0), x_1 x_2 + x_2^3 + x_2 x_3) \leftarrow \text{MULTIVARDIV}(g, (f_1, f_2), \prec_{lex})$,
 $x_1 x_2 \leftarrow \text{NORMALF}(g, \{f_1, f_2\}, \prec_{lex})$.

2.5 Gröbnerbasen

*Erst dann, wenn sich der Mensch
neue Begriffe formt, befreit er sich.*

ANTOINE DE SAINT-EXUPÉRY, Gesammelte Schriften.

Der in Abschnitt 2.2 (Seite 14) vorgestellte Algorithmus für eine multivariate Division hat im Vergleich zum Divisionsalgorithmus für univariate Polynome zwei entscheidende Nachteile:

- (1) Der Rest bei der multivariaten Division gemäß Definition 2.4 (Seite 13) ist nicht eindeutig bestimmt.

Bemerkung: Der Algorithmus MULTIVARDIV ist dennoch deterministisch, da dort F als Tupel übergeben wird und immer mit jenem der möglichen Polynome f_i reduziert wird, das den kleinsten Index i hat.

- (2) Ist der Divisionsrest $r(g, F)$ gleich null, so gilt $g \in \langle F \rangle$. Die Umkehrung dieser Aussage gilt jedoch im allgemeinen nicht, wie das Beispiel nach der Definition 2.4 (Seite

13) zeigt. Die zweite Berechnung liefert dort, dass $g \in \langle F \rangle$; anhand der ersten Berechnung sieht man jedoch, dass $g \in \langle F \rangle$ keine hinreichende Bedingung dafür ist, dass der Divisionsrest null ist.

Es stellen sich nun die Fragen, ob und wie man diese Probleme lösen kann. Die Antwort liegt in der Wahl eines „günstigen“ Erzeugendensystems des Ideals, einer Gröbnerbasis.

$\prec$ sei eine Termordnung auf P .

Definition 2.5 Eine endliche Teilmenge F eines Ideals $I \trianglelefteq K[x]$ heißt genau dann **Gröbnerbasis von I (bezüglich $\prec$)**, wenn

$$\langle \text{lt}(F) \rangle_{K[x]} = \langle \text{lt}(I) \rangle_{K[x]}.$$

Eine endliche Teilmenge F von $K[x]$ heißt genau dann **Gröbnerbasis (bezüglich $\prec$)**, wenn F eine Gröbnerbasis von $\langle F \rangle_{K[x]}$ ist.

Bemerkungen:

- (1) Gröbnerbasen sind nicht eindeutig: $F \subseteq F' \subseteq I$, F' endlich; F ist Gröbnerbasis von $I \implies F'$ ist Gröbnerbasis von I .
- (2) Jedes Ideal besitzt eine Gröbnerbasis: Wähle ein endliches Erzeugendensystem $M \subseteq P$ von $\langle \text{lt}(I) \rangle_{K[x]}$ (HILBERTscher Basissatz, Satz 1.5, Seite 9), wähle eine endliche Menge $G \subseteq I$ so, dass $\text{lt}(G) = M$. Dann ist G eine Gröbnerbasis von I .
- (3) Für ein $f \in K[x]$ ist die Normalform $\text{normalf}(f, G, \prec)$ für eine Gröbnerbasis G bzgl. $\prec$ eindeutig bestimmt.

Lemma 2.3 Sei F eine Gröbnerbasis eines Ideals $I \trianglelefteq K[x]$. Dann ist F ein Erzeugendensystem von I , d.h.: $\langle F \rangle_{K[x]} = I$.

Beweis: Die Inklusion „ $\subseteq$ “ ist klar, da $F \subseteq I$. Sei nun $g \in I$. Für $g = 0$ ist die Aussage trivial. Sei also $g \neq 0$. Es gilt: $\text{lt}(g) \in \langle \text{lt}(I) \rangle$. Da F eine Gröbnerbasis von I ist, gilt außerdem $\text{lt}(g) \in \langle \text{lt}(F) \rangle$. Daher existiert ein $f \in F$ mit $\text{lt}(f) \mid \text{lt}(g)$.

1.Fall: $f \mid g$: Dann gilt $g \in \langle F \rangle$.

2.Fall: $f \nmid g$: Dann gilt: $\exists p \in P : \text{lt}(g - p f) \prec \text{lt}(g)$. Da $g - p f \in I$, kann man nun mit $g - p f$ genauso verfahren wie mit g . Wegen Satz 2.2 (Seite 13) tritt nach endlich vielen Schritten der 1.Fall ein. $\square$

Lemma 2.4 Sei F eine endliche Teilmenge von $I \trianglelefteq K[x]$. Dann gilt:

$$\langle \text{lt}(F) \rangle_{K[x]} = \langle \text{lt}(I) \rangle_{K[x]} \iff \text{lt}(I) = P \cdot \text{lt}(F).$$

Beweis: $\implies \subseteq$: $p \in \text{lt}(I) \implies p \in \langle \text{lt}(I) \rangle \implies p \in \langle \text{lt}(F) \rangle \implies \exists f \in F : \text{lt}(f)|p \implies \exists q \in P : \text{lt}(f) \cdot q = p \implies p \in P \text{lt}(F)$.

$\supseteq$: trivial.

$\Leftarrow \subseteq$: trivial.

$\supseteq$: $g \in \langle \text{lt}(I) \rangle \implies g \in \langle P \text{lt}(F) \rangle$. Wegen $P \subseteq K[x]$ folgt $g \in \langle \text{lt}(F) \rangle$. $\square$

Das Problem (1), das zu Beginn dieses Abschnittes angeführt wurde, wird von der Normalform bzgl. einer Gröbnerbasis gelöst (siehe Bemerkung (3) auf Seite 17). Dass Gröbnerbasen auch das Problem (2) lösen, zeigt der folgende

Satz 2.5 *F sei eine Gröbnerbasis eines Ideals $I \trianglelefteq K[x]$. Dann gilt:
Für alle $g \in I$ sind alle Reste von g nach Division durch F gleich null.*

Beweis: Sei $r := g - \sum_{f \in F} h_f f$ ein Rest von $g \in I$ nach Division durch F . Dann ist $r \in I$, also gilt $r = 0$ oder $\text{lt}(r) \in \text{lt}(I)$. Wegen $\text{lt}(r) \notin P \text{lt}(F)$ und $P \text{lt}(F) = \text{lt}(I)$ (siehe Lemma 2.4), gilt $\text{lt}(r) \notin \text{lt}(I)$, also $r = 0$. $\square$

2.6 Reduzierte Gröbnerbasen

Lieber weniger, aber besser.

WLADIMIR ILJITSCH LENIN

Definition 2.6 *Seien $\prec$ eine Termordnung auf P und $\emptyset \neq F$ eine endliche Teilmenge von $K[x] \setminus \{0\}$. F heißt genau dann **reduzierte Gröbnerbasis**, wenn gilt:*

- (1) F ist eine Gröbnerbasis.
- (2) $\forall f \in F : \text{lk}(f) = 1$.
- (3) $\forall f \in F : \text{normalf}(f, F \setminus \{f\}, \prec) = f$.

Bemerkungen: Sei F eine reduzierte Gröbnerbasis. Dann gilt:

- (1) $\forall f \in F : \text{supp}(f) \cap \text{lt}(\langle F \rangle) = \{\text{lt}(f)\}$.
- (2) Die Leiterterme von F sind paarweise verschieden.

Satz 2.6 *Jedes Ideal $I \neq \{0\}$ in $K[x]$ hat genau eine reduzierte Gröbnerbasis. Diese kann in endlich vielen Schritten berechnet werden.*

Beweis: Existenz: Siehe Terminations- und Korrektheitsbeweis des Algorithmus REDGB (Seite 23).

Eindeutigkeit: Annahme: Seien F und F' zwei verschiedene reduzierte Gröbnerbasen von I . Sei f ein Element aus der symmetrischen Differenz $F \triangle F'$ so, dass $\text{lt}(f)$ in $\text{lt}(F \triangle F')$ minimal ist. O.B.d.A. sei $f \in F \setminus F'$. Wegen $r(f, F') = 0$ existiert ein $f' \in F'$ mit $\text{lt}(f')| \text{lt}(f)$.

Da F eine reduzierte Gröbnerbasis ist und $f \neq f'$, gilt $f' \in F' \setminus F \subseteq F \triangle F'$; denn wäre $f' \in F \cap F'$, so wäre wegen $\text{lt}(f') \mid \text{lt}(f)$ $\text{normalf}(f, F \setminus \{f\}, <) < f$. Da f minimal in $F \triangle F'$ gewählt wurde, folgt aus $\text{lt}(f') \mid \text{lt}(f)$, dass $\text{lt}(f') = \text{lt}(f)$ gilt.

Wegen Bemerkung (2) nach Definition 2.6 gibt es daher für jedes $f \in F$ ein eindeutig bestimmtes $f' \in F'$ so, dass $\text{lt}(f) = \text{lt}(f')$. Daher existiert eine eineindeutige Zuordnung zwischen den Elementen von F und F' ; insbesondere gilt: $|F| = |F'|$.

Seien nun f und f' wie oben gewählt:

$f \neq f' \implies 0 \neq f - f' \in I \implies \text{lt}(f - f') \in \text{lt}(I)$. Aber $\text{supp}(f' - \text{lt}(f')) \subseteq \text{supp}(f - \text{lt}(f)) \cup \text{supp}(f' - \text{lt}(f')) =: S$ und $S \cap \text{lt}(I) = \emptyset$, also $\text{lt}(f - f') \notin \text{lt}(I)$. Das ist ein Widerspruch zu $\text{lt}(f - f') \in \text{lt}(I)$. $\square$

Kapitel 3

Berechnung von Gröbnerbasen (BUCHBERGER-Algorithmus)

*Es ist nicht genug, zu wissen, man muss auch anwenden;
es ist nicht genug, zu wollen, man muss auch tun.*

JOHANN WOLFGANG VON GOETHE

In diesem Kapitel wird ein Verfahren behandelt, mit dem Gröbnerbasen algorithmisch berechnet werden können. Das entscheidende Kriterium dabei liefert Satz 3.1 (Seite 21), der einerseits aussagt, ob ein Erzeugendensystem eines Ideals eine Gröbnerbasis ist (siehe Algorithmus ISTGRÖBNER), und andererseits auch eine Möglichkeit zeigt, wie Gröbnerbasen schrittweise berechnet werden können (siehe Algorithmus BUCHBERGER). Satz 3.1 und der nach ihm benannte Algorithmus stammen von BRUNO BUCHBERGER. Die behandelte Implementierung des BUCHBERGER-Algorithmus ist dabei einfach und recht ineffizient. Abschnitt 3.4 (ab Seite 25) zeigt eine erste Verbesserungsmöglichkeit. Weitere effizientere Varianten finden sich in [1] und [5].

Außerdem wird in diesem Kapitel gezeigt, wie aus einer gegebenen Gröbnerbasis eine reduzierte Gröbnerbasis berechnet werden kann (siehe Algorithmus REDGB).

$\prec$ sei eine Termordnung auf P .

Definition 3.1 Für $i, j \in \mathbb{N}^n$ seien

$$\text{kgV}(\underline{x}^i, \underline{x}^j) := x_1^{\max(i_1, j_1)} x_2^{\max(i_2, j_2)} \dots x_n^{\max(i_n, j_n)}$$

und

$$\text{ggT}(\underline{x}^i, \underline{x}^j) := x_1^{\min(i_1, j_1)} x_2^{\min(i_2, j_2)} \dots x_n^{\min(i_n, j_n)}.$$

Definition 3.2 Für $f, g \in K[\underline{x}] \setminus \{0\}$ sei

$$S(f, g) := \text{lk}(g) \cdot p \cdot f - \text{lk}(f) \cdot q \cdot g,$$

wobei $p, q \in P$ so, dass $p \cdot \text{lt}(f) = q \cdot \text{lt}(g) = \text{kgV}(\text{lt}(f), \text{lt}(g))$.
 $S(f, g)$ heißt **S-Polynom von f und g** .

Hilfssatz 3.1 Seien $k \in \mathbb{N}_{>1}$, $c_1, \dots, c_k \in K$ mit $\sum_{i=1}^k c_i = 0$, $f_j \in K[x] \setminus \{0\}$ und $p_j \in P$ so, dass $\text{lk}(f_j) = 1$ und $\text{lt}(p_j f_j) = \text{lt}(p_1 f_1)$, $1 \leq j \leq k$.

Dann gibt es $d_j \in K$, $q_j \in P$ so, dass $\sum_{j=1}^k c_j p_j f_j = \sum_{j=1}^{k-1} d_j q_j S(f_j, f_{j+1})$ und $\forall j \in \{1, 2, \dots, k-1\}$: $\text{lt}(q_j S(f_j, f_{j+1})) \prec \text{lt}(p_1 f_1)$.

Beweis: Wegen $\text{lt}(p_i f_i) = \text{lt}(p_{i+1} f_{i+1})$ gibt es ein $q_i \in P$ so, dass $\text{lt}(p_i f_i) = q_i \text{kgV}(\text{lt}(f_i), \text{lt}(f_{i+1}))$. Dann ist

$$q_i \cdot S(f_i, f_{i+1}) = q_i \cdot (\bar{p} f_i - \bar{q} f_{i+1})$$

mit $\bar{p}, \bar{q}$ so, dass

$$q_i \bar{p} \text{lt}(f_i) = q_i \bar{q} \text{lt}(f_{i+1}) = q_i \text{kgV}(\text{lt}(f_i), \text{lt}(f_{i+1})) = \text{lt}(p_i f_i) = \text{lt}(p_{i+1} f_{i+1}).$$

Daraus folgt wegen Lemma 2.2 (Seite 12)

$$q_i \bar{p} = p_i \quad \text{und} \quad q_i \bar{q} = p_{i+1}.$$

Es gilt also für alle $1 \leq i \leq k-1$:

$$q_i S(f_i, f_{i+1}) = p_i f_i - p_{i+1} f_{i+1}.$$

Definiert man $d_i := \sum_{m=1}^i c_m$, so erhält man

$$\begin{aligned} \sum_{j=1}^k c_j p_j f_j &= \underbrace{c_1}_{d_1} (p_1 f_1 - p_2 f_2) + \underbrace{(c_1 + c_2)}_{d_2} (p_2 f_2 - p_3 f_3) + \dots + \underbrace{(c_1 + \dots + c_{k-1})}_{d_{k-1}} (p_{k-1} f_{k-1} - p_k f_k) \\ &+ \underbrace{\left(\sum_{i=1}^k c_i \right)}_0 p_k f_k = \sum_{j=1}^{k-1} d_j q_j S(f_j, f_{j+1}). \quad \square \end{aligned}$$

Satz 3.1 F sei eine endliche Teilmenge von $K[x] \setminus \{0\}$. Dann sind die folgenden Aussagen äquivalent:

- (1) F ist eine Gröbnerbasis.
- (2) Für alle $f, g \in F$ ist ein Rest von $S(f, g)$ nach Division durch F gleich null.

Bemerkung: Wenn F eine Gröbnerbasis ist, so sind gemäß Satz 2.5 (Seite 18) alle Reste von $S(f, g)$ bei Division durch F gleich null.

Beweis: (1) $\implies$ (2): folgt aus Satz 2.5 (Seite 18), da $S(f, g) \in \langle F \rangle$.

(2) $\implies$ (1): O.B.d.A. gilt $\forall f \in F : \text{lk}(f) = 1$. Sei $0 \neq h \in \langle F \rangle$. Zeige: $\text{lt}(h) \in \langle \text{lt}(F) \rangle$.

Sei $\mathcal{U} := \{(c_{p,f})_{\substack{p \in P \\ f \in F}} \mid c_{p,f} \in K, \sum_{p \in P, f \in F} c_{p,f} p f = h\}$.

Wegen $h \in \langle F \rangle$ ist $\mathcal{U} \neq \emptyset$.

Sei

$$\lambda : \begin{cases} \mathcal{U} \rightarrow P \\ (c_{p,f})_{\substack{p \in P \\ f \in F}} \mapsto \max_{\prec} \{\text{lt}(p f) \mid c_{p,f} \neq 0\}. \end{cases}$$

$\lambda(\mathcal{U}) \subseteq P$ enthält ein kleinstes Element m . Man wähle $(c_{p,f})_{\substack{p \in P \\ f \in F}} \in \mathcal{U}$ so, dass

$$\lambda((c_{p,f})_{\substack{p \in P \\ f \in F}}) = m.$$

Seien

$$h' := \sum_{\substack{p,f \\ \text{lt}(pf)=m}} c_{p,f} p f \quad \text{und} \quad h'' := \sum_{\substack{p,f \\ \text{lt}(pf) \prec m}} c_{p,f} p f.$$

Dann ist $h = h' + h''$.

Fall 1: $\text{lt}(h') = m \implies \text{lt}(h) = \text{lt}(h') = m \in \langle \text{lt}(F) \rangle$.

Fall 2: $\text{lt}(h') \neq m$. Dann ist $\sum_{\substack{p,f \in P \\ \text{lt}(pf)=m}} c_{p,f} = 0$. Nach dem Hilfssatz 3.1 (Seite 21) gibt es da-

her eine Familie $(d(q, f, g))_{\substack{q \in P \\ f, g \in F}}$ in K so, dass $h' = \sum_{q,f,g} d(q, f, g) q S(f, g)$ und für alle

q, f, g mit $d(q, f, g) \neq 0$ ist $\text{lt}(q S(f, g)) \prec m$. Nach (2): Für alle $f, g \in F$ gibt es Familien $(b(r, e, f, g))_{\substack{e \in F \\ r \in P}}$ in K so, dass $S(f, g) = \sum_{r,e} b(r, e, f, g) r e$ und entweder $S(f, g) = 0$ oder

$$\max_{\prec} \{\text{lt}(r e) | b(r, e, f, g) \neq 0\} = \text{lt}(S(f, g)).$$

Somit gilt:

$$\begin{aligned} h = h' + h'' &= \sum_{\substack{q,r \in P \\ e,f,g \in F}} d(q, f, g) b(r, e, f, g) (qr) e + \sum_{\substack{p,f \\ \text{lt}(pf) \prec m}} c_{p,f} p f \\ &= \sum_{\substack{t \in P \\ e \in F}} \left(\underbrace{\sum_{\substack{f,g,q,r \\ qr=t}} d(q, f, g) b(r, e, f, g)}_{=: a_{t,e}} \right) t e + h'' \end{aligned}$$

und $\max_{\prec} \{\text{lt}(te) | a_{t,e} \neq 0\} \prec m$. Das ist ein Widerspruch zur Minimalität von m . $\square$

3.1 Der Algorithmus BUCHBERGER

Der Algorithmus BUCHBERGER nützt die Aussage des letzten Satzes, um aus einer endlichen Menge F schrittweise eine Gröbnerbasis von $\langle F \rangle$ zu berechnen. Der Algorithmus BUCHBERGER ist nach seinem Erfinder BRUNO BUCHBERGER benannt.

Korrektheit: Bei Termination des Algorithmus gilt $S = \emptyset$. Daher ist die Aussage (2) aus Satz 3.1 erfüllt und G eine Gröbnerbasis. $\square$

Termination: Sei G_k die Menge G und S_k die Menge S nach dem k -ten Durchlauf beider for-Schleifen. Ist $S_k = \emptyset$, so terminiert der Algorithmus.

Andernfalls gilt $G_{k+1} = G_k \cup S_k \implies G_{k+1} \not\subseteq G_k \implies \langle \text{lt}(G_k) \rangle \subsetneq \langle \text{lt}(G_{k+1}) \rangle$. Da $K[x]$ noethersch ist, bricht diese echt aufsteigende Kette von Idealen nach Satz 1.4 (Seite 8) ab. $\square$

Beispiel: $K = \mathbb{Q}$, $F := \{x_1 + x_2, x_1 x_2^2 + 1\}$.

$$\{x_1 + x_2, x_1 x_2^2 + 1, x_2^3 - 1\} \leftarrow \text{BUCHBERGER}(F, \prec_{lex}).$$

Spezifikation: $G \leftarrow \text{BUCHBERGER}(F, \prec)$
 Berechnet eine Gröbnerbasis von $\langle F \rangle$ bzgl. $\prec$.

Eingabe: $F = \{f_1, \dots, f_k\} \subseteq K[x]$ und eine Termordnung $\prec$.
Ausgabe: Eine Gröbnerbasis von $\langle F \rangle$ bzgl. $\prec$.

```

begin
     $G \leftarrow (f_1, \dots, f_k)$ 
    repeat
         $S \leftarrow \emptyset$ 
        Sei  $G = (g_1, \dots, g_s)$ .
        for  $1 \leq i \leq s$  do
            for  $i + 1 \leq j \leq s$  do
                 $(Q, r) \leftarrow \text{MULTIVARDIV}(S(g_i, g_j), G, \prec)$ 
                if  $r \neq 0$  then  $S \leftarrow S \cup \{r\}$ 
            end
        end
        if  $S \neq \emptyset$  then
            Sei  $S = \{s_1, \dots, s_t\}$ .
             $G \leftarrow (g_1, \dots, g_s, s_1, \dots, s_t)$ 
        end
    until  $S = \emptyset$ 
    return  $\{g_1, \dots, g_s\}$ 
end
    
```

Algorithmus 3: BUCHBERGER

3.2 Der Algorithmus ISTGRÖBNER

Der Algorithmus ISTGRÖBNER benützt die Aussage des Satzes 3.1 (Seite 21), um festzustellen, ob eine gegebene Menge F eine Gröbnerbasis ist.

Korrektheit: Folgt aus Satz 3.1 (Seite 21) und der Bemerkung nach diesem Satz. $\square$

Termination: trivial. $\square$

Beispiel: $K = \mathbb{Q}$, $F := \{x_1 + x_2, x_1x_2^2 + 1\}$, $G := \{x_1 + x_2, x_1x_2^2 + 1, x_2^3 - 1\}$.
 $false \leftarrow \text{ISTGRÖBNER}(F, \prec_{lex})$.
 $true \leftarrow \text{ISTGRÖBNER}(G, \prec_{lex})$.

3.3 Der Algorithmus REDGB

Der Algorithmus REDGB berechnet die reduzierte Gröbnerbasis eines Ideals. Sowohl im Korrektheits- als auch im Terminationsbeweis wird folgendes Lemma benötigt:

Spezifikation: $b \leftarrow \text{ISTGRÖBNER}(F, \prec)$
 Testet, ob F eine Gröbnerbasis bzgl. $\prec$ ist.

Eingabe: $F = \{f_1, \dots, f_k\} \subseteq K[x]$ und eine Termordnung $\prec$.
Ausgabe: *true*, wenn F eine Gröbnerbasis bzgl. $\prec$ ist, sonst *false*.

```

begin
  for  $1 \leq i \leq k$  do
    for  $i + 1 \leq j \leq k$  do
       $(Q, r) \leftarrow \text{MULTIVARDIV}(S(f_i, f_j), (f_1, \dots, f_k), \prec)$ 
      if  $r \neq 0$  then return false
    end
  end
  return true
end
    
```

Algorithmus 4: ISTGRÖBNER

Lemma 3.1 Seien G eine Gröbnerbasis bzgl. $\prec$ und $\bar{g} \in G$ mit $h := \text{normalf}(\bar{g}, G \setminus \{\bar{g}\}, \prec) \neq 0$. Dann gilt: $\text{lt}(h) = \text{lt}(\bar{g})$.

Beweis: Durch indirekten Beweis wird die äquivalente Aussage $\forall g \in G \setminus \{\bar{g}\} : \text{lt}(g) \nmid \text{lt}(\bar{g})$ gezeigt.

Annahme: $\exists g' \in G \setminus \{\bar{g}\} : \text{lt}(g') \mid \text{lt}(\bar{g})$.

Daraus folgt: $\exists p \in P : p \cdot \text{lt}(g') = \text{lt}(\bar{g}) \implies P \text{lt}(G) = P \text{lt}(G \setminus \{\bar{g}\}) \implies G \setminus \{\bar{g}\}$ ist eine Gröbnerbasis. Daher gilt: $r_{\prec}(\bar{g}, G \setminus \{\bar{g}\}) = 0 \implies \text{normalf}(\bar{g}, G \setminus \{\bar{g}\}, \prec) = 0$. Das ist jedoch ein Widerspruch zur Voraussetzung $h \neq 0$. $\square$

Korrektheit: Eigenschaften (2) und (3) aus Definition 2.6 (Seite 18) gelten nach Termination des Algorithmus trivialerweise. Zu zeigen bleibt, dass G nach jedem Schleifendurchlauf immer noch eine Gröbnerbasis ist.

1.Fall: $h = 0$. Daraus folgt: $\exists g' \in G \setminus \{g\} : \text{lt}(g') \mid \text{lt}(g)$. Daher gilt: $P \text{lt}(G) = P \text{lt}(G \setminus \{g\})$.

Wegen Lemma 2.4 (Seite 17) ist daher auch $G \setminus \{g\}$ eine Gröbnerbasis.

2.Fall: $h \neq 0$. Wegen Lemma 3.1 gilt $\text{lt}(h) = \text{lt}(g)$. Daher gilt $\text{lt}(G) = \text{lt}((G \setminus \{g\}) \cup \{h\})$, und $(G \setminus \{g\}) \cup \{h\}$ ist ebenfalls eine Gröbnerbasis. $\square$

Termination: Bezeichne G'_{neu} die Menge G' im nächsten Schleifendurchlauf.

Behauptung: $|G'_{\text{neu}}| = |G'| - 1$.

Beweis: 1.Fall: $h = 0$. Daraus folgt: $\exists g' \in G \setminus \{g\} : \text{lt}(g') \mid \text{lt}(g)$. Daher gilt: $G'_{\text{neu}} = G' \setminus \{g\}$.

2.Fall: $h \neq 0$. Aus Lemma 3.1 folgt: $\text{lt}(G) = \text{lt}((G \setminus \{g\}) \cup \{h\})$. Daher gilt hier ebenfalls $G'_{\text{neu}} = G' \setminus \{g\}$. $\square$

Beispiel: $K = \mathbb{Q}$, $F := \{x_1^2 x_2 + x_1, x_1 x_2 + x_2 + x_3, x_1 + x_3\}$.

$\{x_1 + x_3, -x_2 - x_3 + x_2 x_3, x_3^2 + x_2, x_2^2 + x_3\} \leftarrow \text{REDGB}(F, \prec_{\text{lex}})$.

Spezifikation: $G \leftarrow \text{REDGB}(F, \prec)$
 Berechnet eine reduzierte Gröbnerbasis von $\langle F \rangle$ bzgl. $\prec$.

Eingabe: Eine endliche Menge $F \subseteq K[x]$ und eine Termordnung $\prec$.
Ausgabe: Eine reduzierte Gröbnerbasis von $\langle F \rangle$ bzgl. $\prec$.

```

begin
     $H \leftarrow \text{BUCHBERGER}(F, \prec)$ 
     $G \leftarrow \{ \frac{h}{\text{lk}(h)} \mid h \in H \}$ 
    repeat
         $G' \leftarrow \{ g \in G \mid \text{normalf}(g, G \setminus \{g\}, \prec) \neq g \}$ 
        if  $G' \neq \emptyset$  then
            Wähle ein  $g \in G'$ .
             $G \leftarrow G \setminus \{g\}$ 
             $h \leftarrow \text{normalf}(g, G, \prec)$ 
            if  $h \neq 0$  then  $G \leftarrow G \cup \{ \frac{h}{\text{lk}(h)} \}$ 
        end
    until  $G' = \emptyset$ 
    return  $G$ 
end
    
```

Algorithmus 5: REDGB

3.4 Verbesserung des BUCHBERGER-Algorithmus

*Le mieux est l'ennemi du bien.
 (Das Bessere ist der Feind des Guten.)*

VOLTAIRE

Der Algorithmus BUCHBERGER aus Abschnitt 3.1 (Seite 22) ist nicht sehr effizient. Das Aufwändigste dabei ist die Durchführung der zahlreichen Polynomdivisionen mittels MULTIVARDIV. Es ist daher ein Ziel, die Anzahl der S-Polynome, die durch G dividiert werden, zu senken. Eine erste Möglichkeit dies umzusetzen ist das Erste BUCHBERGER Kriterium, das im folgenden behandelt wird.

Definition 3.3 Die Terme p und q aus P heißen **disjunkt**, wenn $\text{ggT}(p, q) = 1$ bzw. gleichbedeutend $\text{kgV}(p, q) = p \cdot q$ gilt.

Satz 3.2 (Erstes BUCHBERGER Kriterium) Seien $f, g \in K[x]$ mit disjunkten Leittermen und $\prec$ eine Termordnung. Dann gilt:

$$\text{normalf}(S(f, g), \{f, g\}, \prec) = 0.$$

Beweis: Sei

$$f = \sum_{i=1}^k a_i p_i \quad \text{und} \quad g = \sum_{j=1}^l b_j q_j$$

mit $a_i, b_j \in K$, $a_i, b_j \neq 0$ und $p_i, q_j \in P$. O.B.d.A. sei $p_1 > p_2 > \dots > p_k$ und $q_1 > q_2 > \dots > q_l$. Da p_1 und q_1 disjunkt sind, ist $\text{kgV}(p_1, q_1) = p_1 q_1$ und daher

$$(*) \quad S(f, g) = b_1 q_1 f - a_1 p_1 g = b_1 q_1 \sum_{i=2}^k a_i p_i - a_1 p_1 \sum_{j=2}^l b_j q_j.$$

Die beiden Summen in $(*)$ haben keine gemeinsamen Terme, denn wäre $p_i q_1 = p_1 q_j$ für ein i und ein j mit $2 \leq i \leq k$ und $2 \leq j \leq l$, so wäre $p_i q_1$ (ein gemeinsames Vielfaches von p_1 und q_1) durch $\text{kgV}(p_1, q_1) = p_1 q_1$ teilbar und es würde $p_1 q_1 \preceq p_i q_1$ und daraus $p_1 \preceq p_i$ folgen.

Weiters gilt, dass jeder Term der 2. Summe durch $\text{lt}(f) = p_1$ teilbar ist.

Man kann daher die Darstellung auf der rechten Seite von $(*)$ weiter reduzieren: Im ersten Reduktionsschritt addiert man $b_l q_l f$. Dies ergibt:

$$\begin{aligned} & b_1 q_1 \sum_{i=2}^k a_i p_i - a_1 p_1 \sum_{j=2}^l b_j q_j + b_l q_l f \\ &= b_1 q_1 \sum_{i=2}^k a_i p_i - a_1 p_1 \sum_{j=2}^l b_j q_j + b_l q_l (a_1 p_1 + \sum_{i=2}^k a_i p_i) \\ &= b_1 q_1 \sum_{i=2}^k a_i p_i - a_1 p_1 \sum_{j=2}^{l-1} b_j q_j + b_l q_l \sum_{i=2}^k a_i p_i. \end{aligned}$$

Im nächsten Schritt wird $b_{l-1} q_{l-1} f$ addiert usw. bis im letzten Schritt $b_2 q_2 f$ addiert wird.

Jede dieser Additionen ist auch wirklich ein Reduktionsschritt, denn für alle j mit $2 \leq j \leq l$ gilt: nach der Addition von $b_l q_l f + \dots + b_j q_j f$ sind immer noch alle Terme $p_1 q_{j-1}, \dots, p_1 q_2$ präsent, da jeder dieser Terme echt größer ist als jeder Term in $b_l q_l f + \dots + b_j q_j f$.

Am Ende dieser $l - 1$ Reduktionsschritte erhält man:

$$b_1 q_1 \sum_{i=2}^k a_i p_i - \sum_{j=2}^l b_j q_j \sum_{i=2}^k a_i p_i = g \sum_{i=2}^k a_i p_i.$$

Ein weiterer Reduktionsschritt modulo g ergibt den Rest null. $\square$

Aus dem Ersten BUCHBERGER Kriterium folgt also, dass zwei Polynome mit disjunkten Leittermen im BUCHBERGER-Algorithmus nicht weiter betrachtet werden müssen, da ihr S-Polynom modulo jedes Erzeugendensystems verschwindet. Würde man für diese zwei Polynome die Polynomdivision dennoch durchführen, so würde diese nicht notwendigerweise den Rest null ergeben, und der Rest würde daher unnötigerweise in die Gröbnerbasis aufgenommen werden.

Das Zweite BUCHBERGER Kriterium, das eine weitere Effizienzsteigerung des BUCHBERGER-Algorithmus ermöglicht, und eine algorithmische Umsetzung des Ersten und Zweiten BUCHBERGER Kriteriums findet man in [1] und [5].

Teil II

Operationen auf Polynomidealen

Im 2. Teil werden wichtige Operationen auf Polynomidealen behandelt, die mit Hilfe von Gröbnerbasen durchgeführt werden können. Unter anderem werden dabei ein Idealgleichheitstest, der mittels der reduzierten Gröbnerbasen überprüft, ob zwei endliche Mengen von Polynomen ein Erzeugendensystem des selben Ideals bilden, und ein Idealmitgliedschaftstest, der feststellt, ob ein gegebenes Polynom in dem von einer Menge von Polynomen erzeugten Ideal liegt, vorgestellt. Außerdem werden Algorithmen zur Berechnung des Durchschnitts und des Quotienten zweier Ideale angegeben.

Der Radikalmitgliedschaftstest, der ebenfalls in diesem Teil behandelt wird, findet im 3. Teil Verwendung.

Kapitel 4

Idealgleichheitstest

Gleich und gleich gesellt sich gern.

MARCUS TULLIUS CICERO, Cato maior.

In diesem Kapitel soll ein Algorithmus behandelt werden, der überprüft, ob zwei endliche Teilmengen von $K[x]$ das gleiche Ideal erzeugen.

Nach Satz 2.6 (Seite 18) hat jedes Ideal eine eindeutige reduzierte Gröbnerbasis. Aus dieser Eigenschaft lässt sich nun ein Test auf Idealgleichheit formulieren, der in endlich vielen Schritten feststellt, ob zwei Ideale gleich sind: zwei Ideale sind genau dann gleich, wenn ihre reduzierten Gröbnerbasen gleich sind.

4.1 Der Algorithmus IDEALGLEICHHEIT

Spezifikation: $b \leftarrow \text{IDEALGLEICHHEIT}(F, G)$

Testet, ob $\langle F \rangle = \langle G \rangle$.

Eingabe: F und G , zwei endliche Teilmengen von $K[x]$.

Ausgabe: *true*, wenn $\langle F \rangle = \langle G \rangle$, sonst *false*.

begin

Sei $\prec$ eine Termordnung.

if REDGB($F, \prec$) = REDGB($G, \prec$) **then return true**

else return false

end

Algorithmus 6: IDEALGLEICHHEIT

Korrektheit: Folgt unmittelbar aus Satz 2.6 (Seite 18). $\square$

Termination: Folgt aus der Termination von REDGB. $\square$

Beispiel: $K = \mathbb{Q}$.

$\text{true} \leftarrow \text{IDEALGLEICHHEIT}(\{x_1^2x_2 + x_1, x_1x_3 + x_2, x_3\}, \{x_1, x_2, x_3\})$.

Kapitel 5

Idealmitgliedschaft

Dabei sein ist alles.

Olympisches Motto

In diesem Kapitel werden 2 Probleme zur Idealmitgliedschaft mittels Gröbnerbasen gelöst: einerseits wird gezeigt, wie mit Gröbnerbasen überprüft werden kann, ob ein gegebenes Polynom g Element eines Ideals I ist; andererseits wird gezeigt, wie man eine Darstellung des Elementes g bezüglich eines vorgegebenen Erzeugendensystems von I berechnen kann.

5.1 Idealmitgliedschaftstest

*Drum prüfe, wer sich ewig bindet,
ob sich das Herz zum Herzen findet!*

FRIEDRICH SCHILLER, Das Lied von der Glocke.

Dieser Abschnitt erörtert die Frage, wie festgestellt werden kann, ob ein Element $g \in K[x]$ in einem gegebenen Ideal $I \trianglelefteq K[x]$ liegt. Dabei sei F ein endliches Erzeugendensystem von I .

Es ist klar, dass aus $r(g, F) = 0$ $g \in \langle F \rangle$ folgt. $r(g, F) = 0$ ist zwar eine hinreichende Bedingung für die Idealmitgliedschaft von g , aber keine notwendige, wie das Beispiel zur Definition 2.4 (Seite 13) zeigt.

Es gilt jedoch der folgende

Satz 5.1 *Es seien $g \in K[x]$ und F eine Gröbnerbasis eines Ideals $I \trianglelefteq K[x]$ bzgl. der Termordnung $\prec$. Dann gilt:*

$$g \in I \iff r_{\prec}(g, F) = 0.$$

Bemerkung: Diese Äquivalenz ist so zu verstehen, dass im Falle $g \in I$ alle Reste von g bei Division durch F verschwinden; d.h., wenn F eine Gröbnerbasis ist, kommt es bei der Berechnung von $r_{\prec}(g, F)$ nicht mehr auf die Reduktionsreihenfolge an und alle Reste $r_{\prec}(g, F)$ sind gleich null.

Ist $r_{\prec}(g, F) \neq 0$, sind nicht alle Reste gleich, sondern nur die Normalform $\text{normalf}(g, F, \prec)$

ist eindeutig bestimmt.

Beweis: $\implies$: Ist äquivalent zu Satz 2.5 (Seite 18).

$\impliedby$: trivial. $\square$

5.2 Der Algorithmus IDEALMITGLIED

Aus Satz 5.1 folgt, dass nach Berechnung einer Gröbnerbasis von F (mittels BUCHBERGER, siehe Seite 23) nur mehr ein beliebiger Rest der Division von g durch F berechnet werden muss (mittels MULTIVARDIV, siehe Seite 14), um die Idealmitgliedschaft von g in $\langle F \rangle$ zu entscheiden.

Spezifikation: $b \leftarrow \text{IDEALMITGLIED}(g, F)$

Testet, ob $g \in \langle F \rangle$.

Eingabe: $g \in K[x]$ und eine endliche Menge $F \subseteq K[x]$.

Ausgabe: *true*, wenn $g \in \langle F \rangle$, sonst *false*.

begin

Sei $\prec$ eine Termordnung.

$G \leftarrow \text{BUCHBERGER}(F, \prec)$

Sei $G = \{g_1, \dots, g_s\}$.

$(Q, r) \leftarrow \text{MULTIVARDIV}(g, (g_1, \dots, g_s), \prec)$

if $r = 0$ **then return true**

else return false

end

Algorithmus 7: IDEALMITGLIED

Korrektheit: Folgt unmittelbar aus Satz 5.1. $\square$

Termination: Folgt aus der Termination von BUCHBERGER und MULTIVARDIV. $\square$

Beispiel: $K = \mathbb{Q}$, $g := x_1x_2^3 - x_1$, $F := \{x_1 + x_2, x_1x_2^2 + 1\}$.

$\text{true} \leftarrow \text{IDEALMITGLIED}(g, F)$.

5.3 Darstellung bzgl. eines Erzeugendensystems

Ist dies schon Tollheit, hat es doch Methode.

WILLIAM SHAKESPEARE, Hamlet.

Sei $v \in K[x]$ und F eine endliche Teilmenge von $K[x]$. Gilt $v \in \langle F \rangle$, so gibt es eine Familie $(h_f)_{f \in F}$ in $K[x]$ so, dass $v = \sum_{f \in F} h_f f$. In diesem Abschnitt wird behandelt, wie der BUCHBERGER-Algorithmus adaptiert werden kann, um diese Familie $(h_f)_{f \in F}$ zu berechnen.

Sei G eine Gröbnerbasis von $\langle F \rangle$ und alle $a_{fg} \in K[x]$ so, dass für $g \in G$

$$(*) \quad g = \sum_{f \in F} a_{fg} f.$$

Dividiert man nun $v \in \langle F \rangle$ mit Rest durch G , so ist der Rest wegen Satz 5.1 (Seite 30) null, und man erhält

$$(**) \quad v = \sum_{g \in G} b_g g$$

mit einer Familie $(b_g)_{g \in G}$ in $K[x]$.

Insgesamt also:

$$(***) \quad v = \sum_{g \in G} \sum_{f \in F} b_g a_{fg} f = \sum_{f \in F} \underbrace{\left(\sum_{g \in G} b_g a_{fg} \right)}_{=: h_f} f = \sum_{f \in F} h_f f.$$

Die Darstellung (**) kann mit Hilfe des Algorithmus MULTIVARDIV berechnet werden. Zur Berechnung der a_{fg} in (*) muss der Algorithmus BUCHBERGER etwas erweitert werden: Zu jedem Element g der als Ergebnis gelieferten Gröbnerbasis G soll ein Vektor $c_g \in K[x]^{|F|}$ so mitberechnet werden, dass c_g die Koeffizienten von g bzgl. des Erzeugendensystems F von $\langle F \rangle$ enthält.

Für $G = \{g_1, \dots, g_s\}$ liefert die Zeile

$$(Q, r) \leftarrow \text{MULTIVARDIV}(S(g_i, g_j), (g_1, \dots, g_s), \prec)$$

im Algorithmus BUCHBERGER (Seite 23) die Darstellung (Q sei $\{q_1, \dots, q_s\}$)

$$r = S(g_i, g_j) - \sum_{k=1}^s q_k g_k = h_i f_i + h_j f_j - \sum_{k=1}^s q_k g_k$$

mit $h_i = \text{lk}(g_j) \frac{\text{kgV}(\text{lt}(g_i), \text{lt}(g_j))}{\text{lt}(g_i)}$ und $h_j = -\text{lk}(g_i) \frac{\text{kgV}(\text{lt}(g_i), \text{lt}(g_j))}{\text{lt}(g_j)}$. Ist $r \neq 0$ und kennt man die Darstellung aller g_i bzgl. des Erzeugendensystems F , so liefert dies die Darstellung des neuen Elements r bzgl. F :

$$c_r := h_i c_{f_i} + h_j c_{f_j} - \sum_{k=1}^s q_k c_{g_k}.$$

5.4 Der Algorithmus IDEALBASIS

Die im vorigen Abschnitt gemachten Überlegungen können nun dazu verwendet werden, den Algorithmus BUCHBERGER so zu erweitern, dass die Koeffizientenvektoren c_g mitberechnet werden. Zusätzlich wird am Ende des Algorithmus die Formel (***) ausgewertet, um die Familie $(h_f)_{f \in F}$ zu berechnen.

Korrektheit: Folgt aus der Korrektheit des Algorithmus BUCHBERGER und Abschnitt 5.3. $\square$

Termination: Folgt aus der Termination des Algorithmus BUCHBERGER. $\square$

Beispiel: $K = \mathbb{Q}$, $F := (x_1^2 x_2 + x_1, x_1 x_2^2 + 1, x_1 x_2 x_2^2)$.
 $(x_2, -x_1) \leftarrow \text{IDEALBASIS}(x_1 x_2 - x_1, F, \prec_{lex})$.

Spezifikation: $b \leftarrow \text{IDEALBASIS}(v, (f_1, \dots, f_k))$
 Berechnet die Koordinaten von $v \in \langle \{f_1, \dots, f_k\} \rangle$ bzgl. $(f_1, \dots, f_k)$.

Eingabe: $v \in \langle \{f_1, \dots, f_k\} \rangle$ und ein Tupel $(f_1, \dots, f_k)$ mit $f_i \in K[x]$.

Ausgabe: $b = (b_1, \dots, b_k)$ mit $b_i \in K[x]$ so, dass $v = \sum_{i=1}^k b_i f_i$. (Ist $v \notin \langle \{f_1, \dots, f_k\} \rangle$, so bricht der Algorithmus mit **abort** ab.)

begin
 Sei $\prec$ eine Termordnung.
 $G \leftarrow (f_1, \dots, f_k)$
 $C \leftarrow ((1, 0, \dots, 0), (0, 1, 0, \dots, 0), \dots, (0, \dots, 0, 1)) \in \{0, 1\}^{k \times k}$
repeat
 $S \leftarrow ()$
 Sei $G = (g_1, \dots, g_s)$.
 for $1 \leq i \leq s$ **do**
 for $i + 1 \leq j \leq s$ **do**
 $h_1 \leftarrow \text{lk}(g_j) \frac{\text{kgV}(\text{lt}(g_i), \text{lt}(g_j))}{\text{lt}(g_i)}$
 $h_2 \leftarrow -\text{lk}(g_i) \frac{\text{kgV}(\text{lt}(g_i), \text{lt}(g_j))}{\text{lt}(g_j)}$
 $((q_1, \dots, q_s), r) \leftarrow \text{MULTIVARDIV}(h_1 g_i + h_2 g_j, G, \prec)$
 if $r \neq 0$ **then**
 Seien $S = (s_1, \dots, s_m)$ und $C = (c_1, \dots, c_w)$.
 $S \leftarrow (s_1, \dots, s_m, r)$
 $C \leftarrow (c_1, \dots, c_w, h_1 c_i + h_2 c_j - \sum_{l=1}^s q_l c_l)$
 end
 end
 end
 if $S \neq ()$ **then**
 Sei $S = (s_1, \dots, s_t)$.
 $G \leftarrow (g_1, \dots, g_s, s_1, \dots, s_t)$
 end
until $S = ()$

 $((q_1, \dots, q_s), r) \leftarrow \text{MULTIVARDIV}(v, G, \prec)$
if $r \neq 0$ **then abort**
 Sei $C = ((c_{11}, \dots, c_{1k}), \dots, (c_{s1}, \dots, c_{sk}))$.
return $(\sum_{i=1}^s q_i c_{i1}, \dots, \sum_{i=1}^s q_i c_{ik})$
end

Algorithmus 8: IDEALBASIS

Kapitel 6

Radikalmitgliedschaftstest

Radikal sein ist die Sache an der Wurzel fassen.

KARL MARX, Kritik der Hegelschen Rechtsphilosophie.

In diesem Kapitel wird zuerst für ein Polynomideal I das Radikal $\text{Rad}(I)$, das ein spezielles Oberideal von I ist, definiert. Dann wird gezeigt, wie die Radikalmitgliedschaft eines beliebigen Elementes aus dem Polynomring mit einem Idealmitgliedschaftstest entschieden werden kann.

Definition 6.1 Sei I ein Ideal von $K[\underline{x}]$. Dann heißt

$$\text{Rad}(I) := \{f \in K[\underline{x}] \mid \exists e \in \mathbb{N} : f^e \in I\}$$

das **Radikal von I** .

Bemerkung: Es gilt: $I \subseteq \text{Rad}(I)$.

Lemma 6.1 Sei $I \trianglelefteq K[\underline{x}]$. Dann gilt: $\text{Rad}(I) \trianglelefteq K[\underline{x}]$.

Beweis: $f_1, f_2 \in \text{Rad}(I) \implies \exists n_1, n_2 \in \mathbb{N} : f_1^{n_1} \in I \wedge f_2^{n_2} \in I$.

$$(f_1 + f_2)^{n_1+n_2-1} = \sum_{k=0}^{n_1+n_2-1} \binom{n_1+n_2-1}{k} f_1^k f_2^{n_1+n_2-1-k}.$$

Dabei gilt für alle k mit $0 \leq k \leq n_1 + n_2 - 1$: $k \geq n_1 \vee n_1 + n_2 - 1 - k \geq n_2$. Daher ist jeder Summand Element von I und daher auch die Summe selbst. Also gilt: $f_1 + f_2 \in \text{Rad}(I)$.

Sei $g \in K[\underline{x}]$. $(gf_1)^{n_1} = g^{n_1} f_1^{n_1} \in I \implies gf_1 \in \text{Rad}(I)$. $\square$

Beispiele:

(1) $f := f_1^{e_1} f_2^{e_2} \cdots f_n^{e_n} \in K[\underline{x}]$, f_i irreduzibel; $\text{Rad}(\langle f \rangle) = \langle f_1 f_2 \cdots f_n \rangle$.

(2) $\text{Rad}(\langle x^a, y^b \rangle) = \langle x, y \rangle$, mit $a, b \in \mathbb{N}_{>0}$.

(3) $\text{Rad}(\langle x^2, xy \rangle) = \langle x \rangle$.

Satz 6.1 Seien $g \in K[\underline{x}]$ und $I \trianglelefteq K[\underline{x}]$. Dann gilt:

$$g \in \text{Rad}(I) \iff \langle I \cup \{gt - 1\} \rangle_{K[\underline{x}, t]} = K[\underline{x}, t].$$

Beweis: Sei $J := \langle I \cup \{gt - 1\} \rangle \trianglelefteq K[\underline{x}, t]$.

$$\implies: g \in \text{Rad}(I) \implies \exists e \in \mathbb{N} : g^e \in I \implies (gt)^e = g^e t^e \in J.$$

$$1 = \underbrace{(gt)^e}_{\in J} - \underbrace{((gt)^e - 1)}_{\substack{\in J \\ \underbrace{(gt - 1) \left(\sum_{i=0}^{e-1} (gt)^i \right) \in J}}}} \in J \implies J = K[\underline{x}, t].$$

$$\Longleftarrow: J = K[\underline{x}, t] \implies \exists (h_f(x, t))_{f \in I} \subseteq K[\underline{x}, t] \text{ und } \exists h(x, t) \in K[\underline{x}, t] \text{ so, dass}$$

$$1 = \sum_{f \in I} h_f(x, t) f + h(x, t)(gt - 1).$$

Ersetze t durch $\frac{1}{g}$:

$$1 = \sum_{f \in I} h_f(x, \frac{1}{g}) f + 0.$$

Wähle $e \in \mathbb{N}_{>0}$ so, dass $\forall f \in I : h_f(x, \frac{1}{g}) \cdot g^e \in K[\underline{x}]$. Dann gilt:

$$g^e = \sum_{f \in I} \underbrace{h_f(x, \frac{1}{g}) g^e}_{\in K[\underline{x}]} f \in I. \quad \square$$

Satz 6.2 Seien $F \subseteq K[\underline{x}]$ endlich, $g \in K[\underline{x}]$ und G eine Gröbnerbasis von $\langle F \cup \{gt - 1\} \rangle \trianglelefteq K[\underline{x}, t]$ (bzgl. einer beliebigen Termordnung). Dann gilt:

$$g \in \text{Rad}(\langle F \rangle) \iff G \cap (K \setminus \{0\}) \neq \emptyset.$$

Beweis: Folgt unmittelbar aus Satz 6.1. $\square$

6.1 Der Algorithmus RADIKALMITGLIED

Aus Satz 6.2 lässt sich nun unmittelbar der Algorithmus RADIKALMITGLIED zum Radikalmitgliedschaftstest ableiten.

Korrektheit: Folgt unmittelbar aus Satz 6.2. $\square$

Termination: Folgt aus der Termination von BUCHBERGER. $\square$

Beispiel: $K = \mathbb{Q}$, $g := x_2 - x_1^2 + 1$, $F := \{x_1 x_2^2 + 2x_2^2, x_1^4 - 2x_1^2 + 1\}$.
 $\text{true} \leftarrow \text{RADIKALMITGLIED}(g, F)$.

Spezifikation: $b \leftarrow \text{RADIKALMITGLIED}(g, F)$
 Testet, ob $g \in \text{Rad}(\langle F \rangle)$.

Eingabe: $g \in K[\underline{x}]$ und eine endliche Menge $F \subseteq K[\underline{x}]$.

Ausgabe: *true*, wenn $g \in \text{Rad}(\langle F \rangle)$, sonst *false*.

begin

Sei $\prec$ eine Termordnung auf den Termen von $K[\underline{x}, t]$.

$G \leftarrow \text{BUCHBERGER}(F \cup \{gt - 1\}, \prec)$

if $G \cap (K \setminus \{0\}) \neq \emptyset$ **then return true**

else return false

end

Algorithmus 9: RADIKALMITGLIED

Kapitel 7

Berechnung von Idealschnitt und Idealquotient

*In nichts zeigt sich der Mangel an mathematischer Bildung mehr
als in einer übertrieben genauen Rechnung.*

CARL FRIEDRICH GAUSS

In diesem Kapitel wird behandelt, wie Gröbnerbasen für den Schnitt und den Quotienten von Idealen berechnet werden können.

Satz 7.1 Für alle i mit $1 \leq i \leq k$ sei $I_i := \langle f_{i,1}, \dots, f_{i,m_i} \rangle$ ein Ideal von $K[\underline{x}]$. Außerdem sei

$$S := \langle y_1 f_{1,1}, \dots, y_1 f_{1,m_1}, \\ y_2 f_{2,1}, \dots, y_2 f_{2,m_2}, \\ \vdots \\ y_k f_{k,1}, \dots, y_k f_{k,m_k}, 1 - \sum_{i=1}^k y_i \rangle$$

ein Ideal in $K[\underline{x}, y_1, \dots, y_k] =: K[\underline{x}, \underline{y}]$.

Dann gilt: $\bigcap_{i=1}^k I_i = S \cap K[\underline{x}]$.

Beweis: $\subseteq$: Sei $f \in \bigcap_{i=1}^k I_i$. Dann ist wegen

$$f = \underbrace{\left(1 - \sum_{i=1}^k y_i\right)f}_{\in S} + \underbrace{\sum_{i=1}^k y_i f}_{\forall i: y_i f \in y_i I_i \subseteq S}$$

$f \in S \cap K[\underline{x}]$.

$\supseteq$: Sei $f \in S \cap K[\underline{x}]$. Dann ist

$$f = \sum_{i=1}^k \sum_{j=1}^{m_i} g_{i,j}(\underline{x}, \underline{y}) y_i f_{i,j} + g(\underline{x}, \underline{y}) \left(1 - \sum_{i=1}^k y_i\right)$$

mit $g(\underline{x}, \underline{y}), g_{i,j}(\underline{x}, \underline{y}) \in K[\underline{x}, \underline{y}]$. Für alle $1 \leq i \leq k$ setze $y_i = 1$ und $y_j = 0$ für $i \neq j$. Das ergibt:

$$f = \sum_{j=1}^{m_i} g_{i,j}(\underline{x}, 0, \dots, 0, 1, 0, \dots, 0) f_{i,j} \in I_i. \quad \square$$

Lemma 7.1 Seien $I_1 \trianglelefteq K[\underline{x}]$ und $I_2 = \langle f_1, \dots, f_s \rangle \trianglelefteq K[\underline{x}]$. Dann gilt:

$$I_1 : I_2 = \bigcap_{i=1}^s I_1 : \langle f_i \rangle.$$

Beweis: Folgt unmittelbar aus Lemma 1.3 (Seite 7) und Lemma 1.4 (Seite 8). $\square$

Bei der Berechnung des Quotienten $I_1 : \langle f_1, \dots, f_s \rangle$ reicht es daher, die Quotienten $I_1 : \langle f_i \rangle$ zu betrachten und diese dann zu schneiden.

Satz 7.2 Seien $I \trianglelefteq K[\underline{x}]$ und $f \in K[\underline{x}]$. Dann gilt:

$$I \cap \langle f \rangle = \langle h_1, \dots, h_t \rangle \implies I : \langle f \rangle = \left\langle \frac{h_1}{f}, \dots, \frac{h_t}{f} \right\rangle.$$

Beweis: Sei $g \in I : \langle f \rangle$. Dann gilt: $gf \in I \cap \langle f \rangle$.

Sei $h \in I \cap \langle f \rangle$. Dann existiert ein $q \in I : \langle f \rangle$ so, dass $h = q \cdot f$. Daraus folgt: $\frac{h}{f} \in I : \langle f \rangle$. $\square$

7.1 Der Algorithmus IDEALSCHNITT

Aus Satz 7.1 (Seite 38) und mit Hilfe von Satz 11.1 (Seite 52) lässt sich der Algorithmus IDEALSCHNITT formulieren, der eine Gröbnerbasis (bzgl. der lexikographischen Termordnung, siehe Satz 11.1) eines Schnittideals berechnet: berechne eine Gröbnerbasis (bzgl. der lexikographischen Termordnung) von S (Bezeichnung aus Satz 7.1) und entferne daraus alle Polynome, die von einem y_i abhängig sind.

Korrektheit: Folgt aus Satz 7.1 (Seite 38). $\square$

Termination: Folgt aus der Termination von BUCHBERGER. $\square$

Beispiel: $K = \mathbb{Q}$, $F_1 := \{x_1^2 + x_2^2, x_1 x_2\}$, $F_2 := \{x_1^2 - x_2^2\}$.
 $\{x_1^2 x_2 - x_2^3, x_1^3 - x_1 x_2^2\} \leftarrow \text{IDEALSCHNITT}(F_1, F_2)$.

Spezifikation: $G \leftarrow \text{IDEALSCHNITT}(F_1, \dots, F_k)$

Berechnet eine Gröbnerbasis bzgl. $\prec_{lex}$ mit $x_1 > \dots > x_n$ von $\bigcap_{i=1}^k \langle F_i \rangle$.

Eingabe: $F_i = \{f_{i,1}, \dots, f_{i,m_i}\} \subseteq K[x]$ für $1 \leq i \leq k$.

Ausgabe: Eine Gröbnerbasis bzgl. $\prec_{lex}$ des Ideals $\langle F_1 \rangle \cap \dots \cap \langle F_k \rangle$.

begin

Sei $\prec_{lex}$ die lexikographische Termordnung auf den Termen von $K[\underline{x}, \underline{y}]$ mit

$y_1 > y_2 > \dots > y_k > x_1 > \dots > x_n$.

$M \leftarrow \{1 - y_1 - y_2 - \dots - y_k\} \cup \bigcup_{i=1}^k \{y_i f_{i,1}, \dots, y_i f_{i,m_i}\}$

$B \leftarrow \text{BUCHBERGER}(M, \prec_{lex})$

return $B \cap K[\underline{x}]$

end

Algorithmus 10: IDEALSCHNITT

7.2 Der Algorithmus IDEALQUOT

Aus Satz 7.2 (Seite 39) und mit Hilfe des Algorithmus IDEALSCHNITT lässt sich der Algorithmus IDEALQUOT zur Berechnung einer Gröbnerbasis (bzgl. der lexikographischen Termordnung) des Quotienten zweier Ideale formulieren.

Korrektheit: Folgt aus Lemma 7.1 (Seite 39) und Satz 7.2 (Seite 39). $\square$

Termination: Folgt aus der Termination von IDEALSCHNITT. $\square$

Beispiel: $K = \mathbb{Q}$, $F_1 := \{x_1^2 + x_2^2, x_1 x_2\}$, $F_2 := \{x_1^2 - x_2^2\}$.
 $\{x_1, x_2\} \leftarrow \text{IDEALQUOT}(F_1, F_2)$.

Spezifikation: $G \leftarrow \text{IDEALQUOT}(F, Q)$

Berechnet eine Gröbnerbasis bzgl. $\prec_{lex}$ mit $x_1 > \dots > x_n$ von $\langle F \rangle : \langle Q \rangle$.

Eingabe: Eine endliche Menge $F \subseteq K[\underline{x}]$ und $Q = \{q_1, \dots, q_s\} \subseteq K[\underline{x}]$.

Ausgabe: Eine Gröbnerbasis bzgl. $\prec_{lex}$ des Ideals $\langle F \rangle : \langle Q \rangle$.

begin

for $1 \leq i \leq s$ **do**

$H \leftarrow \text{IDEALSCHNITT}(F, \{q_i\})$

$M_i \leftarrow \left\{ \frac{h}{q_i} \mid h \in H \right\}$

end

return $\text{IDEALSCHNITT}(M_1, \dots, M_s)$

end

Algorithmus 11: IDEALQUOT

Teil III

Systeme von Polynomgleichungen

In diesem Teil werden Polynomgleichungssysteme der Form

$$\begin{aligned} f_1(x_1, \dots, x_n) &= 0 \\ f_2(x_1, \dots, x_n) &= 0 \\ &\vdots \\ f_m(x_1, \dots, x_n) &= 0 \end{aligned}$$

mit $f_i \in K[\underline{x}]$ behandelt.

Sei X die Menge aller Lösungen dieses Systems, d.h.

$$X = \{(c_1, \dots, c_n) \in K^n \mid \forall 1 \leq i \leq m : f_i(c_1, \dots, c_n) = 0\}.$$

Dann verschwindet auch jede Linearkombination $g_1 f_1 + \dots + g_m f_m$ mit $g_i \in K[\underline{x}]$ auf X . Daher ist X die Nullstellenmenge des ganzen Ideals $\langle f_1, \dots, f_m \rangle$.

Wählt man nun ein anderes Erzeugendensystem $h_1, \dots, h_k$ dieses Ideals, sodass also

$$\langle f_1, \dots, f_m \rangle = \langle h_1, \dots, h_k \rangle$$

gilt, dann haben die beiden Polynomgleichungssysteme $f_1 = \dots = f_m = 0$ und $h_1 = \dots = h_k = 0$ die gleiche Nullstellenmenge X .

In den folgenden Kapiteln werden algorithmische Tests entwickelt, um folgende Fragen zu beantworten: wann besitzt ein vorgegebenes Polynomgleichungssystem überhaupt Lösungen, wann hat es endlich viele Lösungen und wann haben zwei vorgegebene Polynomgleichungssysteme die gleiche Lösungsmenge. Außerdem wird gezeigt, wie alle Lösungen eines Polynomgleichungssystems über einem Körper K berechnet werden können, sofern dies endlich viele sind und alle Nullstellen aller univariaten Polynome über K berechnet werden können.

Kapitel 8

Test: $\mathcal{N}(F) = \emptyset$

Manche Menschen haben einen Gesichtskreis vom Radius null und nennen ihn ihren Standpunkt.

DAVID HILBERT

In diesem Kapitel wird die Frage nach der Existenz von Lösungen eines Polynomgleichungssystems behandelt. Im Mittelpunkt steht dabei der HILBERTsche Nullstellensatz, mit dem diese Frage auf einen Idealmitgliedschaftstest zurückgeführt werden kann.

Definition 8.1 Es seien K ein Körper, $n \in \mathbb{N}_{>0}$ und $F \subseteq K[x]$. Dann heißt

$$\mathcal{N}(F) := \{a \in K^n \mid \forall f \in F : f(a) = 0\}$$

die Menge der gemeinsamen Nullstellen von F in K^n .

Definition 8.2 K heißt **algebraisch abgeschlossen**, wenn jedes nichtkonstante Polynom aus $K[x]$ in K eine Nullstelle hat.

Beispiele: $\mathbb{Q}$ und $\mathbb{R}$ sind nicht algebraisch abgeschlossen; $\bar{\mathbb{Q}} := \{a \in \mathbb{C} \mid a \text{ algebraische Zahl}\}$ und $\mathbb{C}$ sind algebraisch abgeschlossen.

Satz 8.1 (Schwacher HILBERTscher Nullstellensatz) K sei algebraisch abgeschlossen und $F \subseteq K[x]$. Dann gilt:

$$\mathcal{N}(F) = \emptyset \iff \langle F \rangle = K[x].$$

Bemerkung: Nach Lemma 1.2 (Seite 6) gilt daher: $\mathcal{N}(F) = \emptyset \iff 1 \in \langle F \rangle$.

Beweis: Findet sich in [1] und [5]. Für den Spezialfall $K = \mathbb{C}$ ist ein Beweis in [2] zu finden.
 $\square$

Die Frage, ob das Polynomgleichungssystem F Lösungen hat, kann also auf einen Idealmitgliedschaftstest zurückgeführt werden. Außerdem gilt der folgende

Satz 8.2 Sei K algebraisch abgeschlossen, $F \subseteq K[x]$ eine endliche Menge und G eine Gröbnerbasis von $\langle F \rangle$. Dann gilt:

$$\mathcal{N}(F) = \emptyset \iff G \cap (K \setminus \{0\}) \neq \emptyset.$$

Beweis: Nach Lemma 2.4 (Seite 17) und Satz 8.1 gilt: $\mathcal{N}(F) = \emptyset \iff 1 \in \langle F \rangle \iff 1 \in \text{lt}(\langle F \rangle) \iff 1 \in P \text{lt}(G) \iff 1 \in \text{lt}(G) \iff G \cap K \setminus \{0\} \neq \emptyset. \square$

8.1 Der Algorithmus KEINELSG

Aus Satz 8.2 lässt sich nun unmittelbar der Algorithmus KEINELSG ableiten, um auf $\mathcal{N}(F) = \emptyset$ zu testen.

Spezifikation: $b \leftarrow \text{KEINELSG}(F)$
 Testet, ob $\mathcal{N}(F) = \emptyset$.

Eingabe: Eine endliche Menge $F \subseteq K[x]$ (K sei algebraisch abgeschlossen).

Ausgabe: *true*, wenn $\mathcal{N}(F) = \emptyset$, sonst *false*.

begin

Sei $\prec$ eine Termordnung.

$G \leftarrow \text{BUCHBERGER}(F, \prec)$

if $G \cap (K \setminus \{0\}) \neq \emptyset$ **then return true**

else return false

end

Algorithmus 12: KEINELSG

Korrektheit: Folgt unmittelbar aus Satz 8.2. $\square$

Termination: Folgt aus der Termination von BUCHBERGER. $\square$

Beispiel: $K = \mathbb{C}$, $F := \{x_1^2 - 1, x_2^2 - 1, x_1 x_2\}$.

true $\leftarrow \text{KEINELSG}(F)$.

Kapitel 9

Test: $\mathcal{N}(F) = \mathcal{N}(G)$

*Das Ideal der Gleichheit ist deshalb so schwer,
weil die Menschen Gleichheit nur mit jenen wünschen,
die über ihnen stehen.*

JOHN B. PRIESTLEY

In diesem Kapitel soll behandelt werden, wann zwei vorgegebene Polynomgleichungssysteme die selben Nullstellen haben.

Generell gilt für $F, G \subseteq K[x]$: $\langle F \rangle = \langle G \rangle \implies \mathcal{N}(F) = \mathcal{N}(G)$. Die Umkehrung gilt aber im allgemeinen nicht: $\mathcal{N}(\{x_1\}) = \mathcal{N}(\{x_1^2\})$, aber $\langle x_1 \rangle \neq \langle x_1^2 \rangle$.

Definition 9.1 Sei $M \subseteq K^n$. Dann heißt

$$\mathcal{I}(M) := \{f \in K[x] \mid M \subseteq \mathcal{N}(f)\}$$

das **Ideal von M**.

Bemerkungen:

- (1) $\mathcal{I}(M) \trianglelefteq K[x]$.
- (2) $\text{Rad}(\mathcal{I}(M)) = \mathcal{I}(M)$.
- (3) Für $M = \mathcal{N}(F)$ mit $F \subseteq K[x]$: $\mathcal{I}(M)$ ist die Menge aller Polynome, die zu dem durch F gegebenen Gleichungssystem dazugenommen werden können, ohne die Lösungsmenge zu verändern.

Satz 9.1 (Starker HILBERTscher Nullstellensatz) K sei algebraisch abgeschlossen und $F \subseteq K[x]$. Dann gilt:

$$\mathcal{I}(\mathcal{N}(F)) = \text{Rad}(\langle F \rangle).$$

Beweis: $\supseteq$: trivial.

$\subseteq$: Sei $g \in K[x]$, $\mathcal{N}(F) \subseteq \mathcal{N}(\{g\})$. Zeige: $g \in \text{Rad}(\langle F \rangle)$.

Sei $J := \langle F \cup \{gt - 1\} \rangle_{K[x,t]} \trianglelefteq K[x,t]$. Dann ist $\mathcal{N}(J) = \emptyset$; denn wären $a \in K^n$, $b \in K$ so, dass $(a, b) \in \mathcal{N}(J) \subseteq K^{n+1}$, dann wäre $a \in \mathcal{N}(F)$ und $g(a)b - 1 = 0$. Wegen $g(a) = 0$ (folgt aus $a \in \mathcal{N}(F) \subseteq \mathcal{N}(\{g\})$) würde dies jedoch zum Widerspruch $-1 = 0$ führen.

Aus Satz 8.1 (Seite 44) folgt $J = K[x, t]$ und nach Satz 6.1 (Seite 36) also $g \in \text{Rad}(\langle F \rangle)$.

□

Satz 9.2 Seien K algebraisch abgeschlossen, $I, I' \trianglelefteq K[x]$ und F, G endliche Teilmengen von $K[x]$. Dann gilt:

$$(1) \mathcal{N}(I) = \mathcal{N}(I') \iff \text{Rad}(I) = \text{Rad}(I').$$

$$(2) \mathcal{N}(F) = \mathcal{N}(G) \iff (F \subseteq \text{Rad}(\langle G \rangle)) \wedge (G \subseteq \text{Rad}(\langle F \rangle)).$$

Beweis: (1): $\implies$: $\mathcal{N}(I) = \mathcal{N}(I') \implies \mathcal{I}(\mathcal{N}(I)) = \mathcal{I}(\mathcal{N}(I')) \xrightarrow{\text{Satz 9.1}} \text{Rad}(I) = \text{Rad}(I')$.
 $\impliedby$: trivial wegen $\mathcal{N}(I) = \mathcal{N}(\text{Rad}(I))$.

$$(2): \implies: \mathcal{N}(F) = \mathcal{N}(G) \xrightarrow{(1)} \underbrace{\text{Rad}(\langle F \rangle)}_{\supseteq \langle F \rangle} = \underbrace{\text{Rad}(\langle G \rangle)}_{\supseteq \langle G \rangle}.$$

$$\impliedby: \langle F \rangle \subseteq \text{Rad}(\langle G \rangle) \implies \mathcal{N}(G) \subseteq \mathcal{N}(F), \langle G \rangle \subseteq \text{Rad}(\langle F \rangle) \implies \mathcal{N}(F) \subseteq \mathcal{N}(G). \quad \square$$

9.1 Der Algorithmus LSGMENGENGLEICH

Aus Satz 9.2 lässt sich nun unmittelbar der Algorithmus LSGMENGENGLEICH ableiten, um auf $\mathcal{N}(F) = \mathcal{N}(G)$ zu testen.

Spezifikation: $b \leftarrow \text{LSGMENGENGLEICH}(F, G)$
 Testet, ob $\mathcal{N}(F) = \mathcal{N}(G)$.

Eingabe: F und G , zwei endliche Teilmengen von $K[x]$ (K sei algebraisch abgeschlossen).

Ausgabe: *true*, wenn $\mathcal{N}(F) = \mathcal{N}(G)$, sonst *false*.

begin

Seien $F = \{f_1, \dots, f_m\}$ und $G = \{g_1, \dots, g_k\}$.

for $1 \leq i \leq m$ **do**

if $\neg \text{RADIKALMITGLIED}(f_i, G)$ **then return false**

end

for $1 \leq i \leq k$ **do**

if $\neg \text{RADIKALMITGLIED}(g_i, F)$ **then return false**

end

return true

end

Algorithmus 13: LSGMENGENGLEICH

Korrektheit: Folgt unmittelbar aus Satz 9.2, (2). $\square$

Termination: Folgt aus der Termination von RADIKALMITGLIED. $\square$

Beispiel: $K = \mathbb{C}$, $F := \{x_1\}$, $G := \{x_1^2\}$.
 $\text{true} \leftarrow \text{LSGMENGENGLEICH}(F, G)$.

Kapitel 10

Schranke für $|\mathcal{N}(F)|$

*Nichts ist mächtiger
als eine Idee zur richtigen Zeit.*

VICTOR HUGO

In diesem Kapitel wird behandelt, wann ein gegebenes Polynomgleichungssystem F nur endlich viele Lösungen hat. Außerdem wird im Fall der Endlichkeit eine obere Schranke für $|\mathcal{N}(F)|$ angegeben. Der Algorithmus LSGMENGESCHRANKE berechnet diese Schranke.

Hilfssatz 10.1 *Es seien V ein K -Untervektorraum von $K[x]$, $F \subseteq V$ und $0 \notin F$. Falls $\text{lt}(F) = \text{lt}(V)$, ist F ein K -Erzeugendensystem von V . Für $g \in V$ kann dann eine Familie $(c_f)_{f \in F}$ in K mit $g = \sum_{f \in F} c_f f$ wie folgt berechnet werden:*

(*) Wähle $f \in F$ so, dass $\text{lt}(f) = \text{lt}(g) \in V$. Setze $c_f := \text{lk}(g)\text{lk}(f)^{-1}$. Wenn $g \neq c_f f$, ersetze g durch $g - c_f f$ und gehe zu (*).

Beweis: Zeige: Das Verfahren zur Berechnung der c_f , $f \in F$, liefert nach endlich vielen Schritten das gewünschte Ergebnis.

Wegen $\text{lk}(c_f f) = \text{lk}(g)\text{lk}(f)^{-1}\text{lk}(f) = \text{lk}(g)$ ist entweder $g = c_f f$ oder $\text{lt}(g - c_f f) \prec \text{lt}(g)$. Nach Satz 2.2 (Seite 13) tritt nach endlich vielen Schritten der Fall $g = c_f f$ ein. $\square$

Hilfssatz 10.2 *Es seien $I \trianglelefteq K[\underline{x}]$ und G eine Gröbnerbasis von I bzgl. einer Termordnung $\prec$. Außerdem gelte $I \neq K[\underline{x}]$. Die folgenden Aussagen sind äquivalent:*

- (1) $\dim_K(K[\underline{x}]/I) \in \mathbb{N}$.
- (2) $|P \setminus \text{lt}(I)| \in \mathbb{N}$.
- (3) $\forall 1 \leq i \leq n : \exists e_i \in \mathbb{N}_{>0} : x_i^{e_i} \in \text{lt}(I)$.
- (4) $\forall 1 \leq i \leq n : \exists e'_i \in \mathbb{N}_{>0} : x_i^{e'_i} \in \text{lt}(G)$.

Gelten (1)-(4), dann ist $\dim_K(K[\underline{x}]/I) = |P \setminus \text{lt}(I)|$.

Beweis: (1) $\iff$ (2): Sei B ein K -Erzeugendensystem von I mit $\text{lt}(B) = \text{lt}(I)$. Nach Satz 10.1 ist dann $B \cup (P \setminus \text{lt}(I))$ ein K -Erzeugendensystem von $K[x]$, also

$$K[x] = \underbrace{\bigoplus_{b \in B} Kb}_I \oplus \bigoplus_{p \in P \setminus \text{lt}(I)} Kp = I \oplus \bigoplus_{p \in P \setminus \text{lt}(I)} Kp.$$

Daher gilt: $K[x]/I \cong \bigoplus_{p \in P \setminus \text{lt}(I)} Kp$ (als K -Vektorraumisomorphismus).

(2) $\iff$ (3): $\implies$: **Annahme:** $\exists i_0 : \forall e \in \mathbb{N} : x_{i_0}^e \notin \text{lt}(I)$. Daraus folgt: $\forall e \in \mathbb{N} : x_{i_0}^e \in P \setminus \text{lt}(I) \implies |P \setminus \text{lt}(I)| \notin \mathbb{N}$. Das ist ein Widerspruch zur Voraussetzung von (2).

$\impliedby$: $x_1^{e_1}, \dots, x_n^{e_n} \in \text{lt}(I)$. Wegen Lemma 2.2 (Seite 12) gilt daher für alle k mit $1 \leq k \leq n$:

$$\{x_1^{i_1} \cdots x_{k-1}^{i_{k-1}} \cdot x_k^s \cdot x_{k+1}^{i_{k+1}} \cdots x_n^{i_n} \mid i_1, \dots, i_{k-1}, i_{k+1}, \dots, i_n \in \mathbb{N}, s \geq e_k\} \subseteq \text{lt}(I).$$

Daher sind fast alle Terme Elemente von $\text{lt}(I)$. Daraus folgt: $|P \setminus \text{lt}(I)| \in \mathbb{N}$.

(3) $\iff$ (4): $\implies$: Sei i fest gewählt. Dann gilt: $\exists e_i \in \mathbb{N} : x_i^{e_i} \in \text{lt}(I) \implies x_i^{e_i} \in P \text{lt}(I) = P \text{lt}(G) \xrightarrow{1 \notin G} \exists e'_i \in \mathbb{N} : x_i^{e'_i} \in \text{lt}(G)$.

$\impliedby$: analog. $\square$

Lemma 10.1 Sei G eine Gröbnerbasis eines Ideals $I \trianglelefteq K[x]$ bzgl. einer Termordnung $\prec$ mit $I \neq K[x]$. Außerdem gelte $\forall 1 \leq i \leq n : \exists g_i \in G : \text{lt}(g_i) = x_i^{e_i}$ für ein $e_i \in \mathbb{N}_{>0}$. Dann folgt:

$$|P \setminus \text{lt}(I)| \leq e_1 \cdots e_n.$$

Beweis: Nach dem Beweis von Satz 10.2 ist $P \setminus \text{lt}(I)$ eine Basis eines Vektorraums, der K -isomorph zu $K[x]/I$ ist. Für alle i mit $1 \leq i \leq n$ gilt: $\forall p \in P \setminus \text{lt}(I) : (p = x_1^{m_1} \cdots x_n^{m_n} \implies m_i < e_i)$. Es gibt genau $e_1 \cdots e_n$ Terme, die diese Bedingung für alle i erfüllen. $\square$

Hilfssatz 10.3 Es sei $I \trianglelefteq K[x]$. Dann gilt:

$$\dim_K(K[x]/I) \in \mathbb{N} \iff \dim_K(K[x]/\text{Rad}(I)) \in \mathbb{N}.$$

Beweis: $\implies$: Wegen $I \subseteq \text{Rad}(I)$ ist π mit

$$\pi : \begin{cases} K[x]/I \rightarrow K[x]/\text{Rad}(I) \\ \bar{f} \mapsto \bar{f} \end{cases}$$

wohldefiniert und surjektiv.

$\impliedby$: $\dim_K(K[x]/\text{Rad}(I)) \in \mathbb{N} \xrightarrow{\text{Satz 10.2}} \forall 1 \leq i \leq n \exists e_i \in \mathbb{N} : x_i^{e_i} \in \text{lt}(\text{Rad}(I))$. Sei $f_i \in \text{Rad}(I)$ so, dass $\text{lt}(f_i) = x_i^{e_i}$. Dann gibt es ein $m_i \in \mathbb{N}$ so, dass $f_i^{m_i} \in I$ und $\text{lt}(f_i^{m_i}) = x_i^{m_i e_i} \in \text{lt}(I)$. Mit Satz 10.2 ((3) $\implies$ (1)) folgt nun $\dim_K(K[x]/I) \in \mathbb{N}$. $\square$

Hilfssatz 10.4 Es seien K algebraisch abgeschlossen, $I \trianglelefteq K[x]$ und E eine endliche Teilmenge von $\mathcal{N}(I)$.

Der Algebramorphismus

$$\psi : \begin{cases} K[x] \rightarrow \text{Abb}(E, K) \\ f \mapsto [e \mapsto f(e)] \end{cases}$$

ist wohldefiniert und surjektiv. Außerdem gilt $\text{Rad}(I) \subseteq \text{Kern}(\psi)$.
 Insbesondere: Der Algebramorphismus

$$\overline{\psi} : K[\underline{x}]/\text{Rad}(I) \rightarrow \text{Abb}(E, K)$$

ist wohldefiniert und surjektiv, also $|E| \leq \dim_K(K[\underline{x}]/\text{Rad}(I))$.

Beweis: Lagrange-Interpolation: Für $e, e' \in E$ mit $e \neq e'$ sei $f_{e,e'} \in K[\underline{x}]$ so, dass $\text{grad}(f_{e,e'}) = 1$, $f_{e,e'}(e) = 1$ und $f_{e,e'}(e') = 0$.
 Setze $g_e := \prod_{e' \in E \setminus \{e\}} f_{e,e'} \in K[\underline{x}]$. Dann gilt für alle $e' \in E \setminus \{e\}$: $g_e(e) = 1$ und $g_e(e') = 0$.
 Ist $h \in \text{Abb}(E, K)$, dann ist $h = \psi(\sum_{e \in E} h(e)g_e)$, also ist ψ surjektiv. $\square$

Satz 10.1 Es seien K algebraisch abgeschlossen und $I \trianglelefteq K[\underline{x}]$. Dann gilt:

$$\mathcal{N}(I) \text{ endlich} \iff P \setminus \text{lt}(I) \text{ endlich.}$$

Ist $\mathcal{N}(I)$ endlich, gilt außerdem $|\mathcal{N}(I)| \leq |P \setminus \text{lt}(I)|$.

Beweis: $P \setminus \text{lt}(I)$ endlich $\xLeftrightarrow{HS10.2} \dim_K(K[\underline{x}]/I)$ endlich $\xLeftrightarrow{HS10.3} \dim_K(K[\underline{x}]/\text{Rad}(I))$ endlich $\xLeftrightarrow{HS10.4} \mathcal{N}(I)$ endlich.

Falls $\mathcal{N}(I)$ (und daher $P \setminus \text{lt}(I)$) endlich ist, gilt:

$$|\mathcal{N}(I)| \xLeftrightarrow{HS10.4} \dim_K(K[\underline{x}]/\text{Rad}(I)) \leq \dim_K(K[\underline{x}]/I) \xLeftrightarrow{HS10.2} |P \setminus \text{lt}(I)|. \square$$

Korollar 10.1 Es gelten die gleichen Voraussetzungen wie in Lemma 10.1 (Seite 49) und $\mathcal{N}(I)$ sei endlich. Dann ist $|\mathcal{N}(I)| \leq e_1 \cdots e_n$.

Beweis: : Folgt unmittelbar aus Lemma 10.1 (Seite 49) und Satz 10.1. $\square$

Beispiel: $K = \mathbb{C}$.

$$f_1 := x_1^5 - 2x_1x_2^2x_3 + x_2^4x_3^2 - 37,$$

$$f_2 := x_2^3 + 2x_2^2x_3 - x_3^3,$$

$$f_3 := x_3^4 - x_3^3 - x_3^2 - x_3 - 2,$$

$$F := \{f_1, f_2, f_3\}, F \text{ ist Gröbnerbasis bzgl. } \prec_{lex}.$$

$$\text{lt}(F) = \{x_1^5, x_2^3, x_3^4\}.$$

$$\text{Für } K = \mathbb{C} \text{ gilt: } |\mathcal{N}(F)| \leq 5 \cdot 3 \cdot 4 = 60.$$

10.1 Der Algorithmus LSGMENGESCHRANKE

Der Algorithmus LSGMENGESCHRANKE berechnet eine Schranke für $|\mathcal{N}(F)|$ und unterscheidet zusätzlich zwei Sonderfälle. Die möglichen Rückgabewerte dieses Algorithmus sind:

- (1) Ist $\mathcal{N}(F) = \emptyset$, so ist der Rückgabewert 0 (siehe Kapitel 8 und Algorithmus KEINELSG).

Spezifikation: $s \leftarrow \text{LSGMENGESCHRANKE}(F)$
 Berechnet eine Schranke für $|\mathcal{N}(F)|$.

Eingabe: Eine endliche Menge $F \subseteq K[x]$ (K sei algebraisch abgeschlossen).

Ausgabe: 0, wenn $\mathcal{N}(F) = \emptyset$, und ∞ , wenn $\mathcal{N}(F)$ nicht endlich ist; sonst wird eine obere Schranke für $|\mathcal{N}(F)|$ geliefert.

```

begin
    Sei  $\prec$  eine Termordnung.
     $G \leftarrow \text{BUCHBERGER}(F, \prec)$ 
    if  $G \cap (K \setminus \{0\}) \neq \emptyset$  then return 0
    if  $\{x_1^{e_1}, \dots, x_n^{e_n}\} \subseteq \text{lt}(G)$  then
        return  $\prod_{i=1}^n e_i$ 
    then return  $\infty$ 
end
    
```

Algorithmus 14: LSGMENGESCHRANKE

(2) Ist $\mathcal{N}(F)$ unendlich, so ist der Rückgabewert ∞ .

(3) Sonst ist der Rückgabewert eine Schranke für $|\mathcal{N}(F)|$.

Korrektheit: Ist $G \cap (K \setminus \{0\}) \neq \emptyset$, so ist nach Satz 8.2 (Seite 45) $\mathcal{N}(F) = \emptyset$, und der Rückgabewert ist daher 0.

Ist $\{x_1^{e_1}, \dots, x_n^{e_n}\} \subseteq \text{lt}(G)$, dann gilt Korollar 10.1 (Seite 50). Andernfalls ist $\mathcal{N}(F)$ wegen Hilfssatz 10.2 (Seite 48) und Satz 10.1 (Seite 50) unendlich. $\square$

Termination: Folgt aus der Termination von BUCHBERGER. $\square$

Beispiel: $K = \mathbb{C}$, $F := \{x_1^5 - 2x_1x_2^2x_3 + x_2^4x_3^2 - 37, x_2^3 + 2x_2^2x_3 - x_3^3, x_3^4 - x_3^3 - x_3^2 - x_3 - 2\}$.
 $60 \leftarrow \text{LSGMENGESCHRANKE}(F)$.

Kapitel 11

Berechnung von $\mathcal{N}(F)$

Das also war des Pudels Kern!

JOHANN WOLFGANG VON GOETHE, Faust I.

In diesem Kapitel wird für den Fall, dass $\mathcal{N}(F)$ endlich ist, ein Algorithmus behandelt, der die Lösungen des Polynomgleichungssystems F (also die Elemente von $\mathcal{N}(F)$) berechnet. Es ist zu beachten, dass für das in diesem Kapitel behandelte Verfahren zur Lösungsberechnung immer die lexikographische Termordnung vorausgesetzt wird und davon ausgegangen wird, dass alle Lösungen aller univariaten Polynome über K berechnet werden können.

Satz 11.1 *I sei ein Ideal von $K[x]$ und G sei eine Gröbnerbasis von I bzgl. der lexikographischen Ordnung $\prec_{lex}$ mit $x_1 > x_2 > \dots > x_n$. Für $1 \leq k \leq n$ gilt dann:*

Wenn $I \cap K[x_k, \dots, x_n] \neq \{0\}$, dann ist $G \cap K[x_k, \dots, x_n]$ eine Gröbnerbasis von $I \cap K[x_k, \dots, x_n]$.

Beweis: Es sei $h \in I \cap K[x_k, \dots, x_n]$. Dann ist $\text{lt}(h) \in K[x_k, \dots, x_n]$. Weil G eine Gröbnerbasis ist, gibt es eine Familie $(h_g)_{g \in G}$ in $K[x]$ mit $h = \sum_{g \in G} h_g g$ und

$$\text{lt}(h) = \max \{ \text{lt}(h_g g) \mid g \in G, h_g g \neq 0 \}.$$

Zeige: $h_g \neq 0 \implies g \in G \cap K[x_k, \dots, x_n]$.

Sei $h_g \neq 0$. Dann ist $\text{lt}(g) \preceq_{lex} \text{lt}(h_g g) \preceq_{lex} \text{lt}(h) \in K[x_k, \dots, x_n]$. Weil $\prec_{lex}$ die lexikographische Ordnung mit $x_1 > x_2 > \dots > x_n$ ist, folgt: $\text{lt}(g) \in K[x_k, \dots, x_n]$ und $g \in K[x_k, \dots, x_n]$. $\square$

Bemerkungen:

- (1) Die Ideale $I \cap K[x_k, \dots, x_n] \trianglelefteq K[x_k, \dots, x_n]$, $1 \leq k \leq n$, heißen **Eliminationsideale von I** .
- (2) Wenn $I \cap K[x_n] = \{0\}$, dann gibt es in $I \setminus \{0\}$ kein Polynom, das nur von x_n abhängig ist.
- (3) Wenn $I \cap K[x_n] \neq \{0\}$ und G eine Gröbnerbasis von I bzgl. der lexikographischen Ordnung mit $x_1 > \dots > x_n$ ist, dann wird $I \cap K[x_n]$ von $G \cap K[x_n]$ erzeugt, also: $I \cap K[x_n] = \langle \text{ggT}(G \cap K[x_n]) \rangle_{K[x]}$.

- (4) Bezüglich der lexikographischen Termordnung gilt: Wenn $P \setminus \text{lt}(I)$ endlich ist, dann ist $I \cap K[x_n] \neq \{0\}$. Denn wäre $I \cap K[x_n] = \{0\}$, so würde $\{x_n, x_n^2, x_n^3, \dots\} \not\subseteq \text{lt}(I)$ gelten und $P \setminus \text{lt}(I)$ wäre nicht endlich.

11.1 Der Algorithmus LSG

Mit Hilfe von Satz 11.1 und den darauf folgenden Bemerkungen lässt sich nun ein Algorithmus zur Lösungsberechnung formulieren: Gibt es nur endlich viele Lösungen, so existiert in jeder Gröbnerbasis bzgl. der lexikographischen Termordnung ein Polynom f_n , das nur von x_n abhängig ist, ein Polynom f_{n-1} , das nur von x_n und x_{n-1} abhängig ist, usw. Zuerst werden nun alle Nullstellen von f_n berechnet. Dann wird in alle Polynome der Gröbnerbasis für x_n der Wert der ersten Nullstelle eingesetzt und mit der so erhaltenen Gröbnerbasis rekursiv weiterverfahren. Danach wird mit den weiteren Nullstellen von f_n (falls solche existieren) genauso vorgegangen.

Wenn ein Polynomgleichungssystem nur endlich viele Lösungen hat und man über K alle Nullstellen eines univariaten Polynoms berechnen kann, so kann man auf diese Weise alle Nullstellen des Polynomgleichungssystems berechnen.

Spezifikation: $\mathcal{L} \leftarrow \text{LSG}(G, j)$

Berechnet $\mathcal{N}(G)$ in $K[x_1, \dots, x_j]$.

Eingabe: Eine Gröbnerbasis $G \subseteq K[x_1, \dots, x_j]$ bzgl. der lexikographischen Ordnung $\prec_{lex}$ mit $x_1 > x_2 > \dots > x_j$ und $\text{LSGMENGESCHRANKE}(G) \notin \{0, \infty\}$.

Ausgabe: Alle Lösungen des Polynomgleichungssystems G über K , sofern alle Nullstellen aller auftretenden univariaten Polynome in K berechnet werden können.

begin

$N \leftarrow \mathcal{N}(\{\text{ggT}(G \cap K[x_j, \dots, x_n])\})$

if $N = \emptyset$ **then return** $\emptyset$

Sei $N = \{n_1, \dots, n_s\}$.

if $j = 1$ **then**

return $\bigcup_{i=1}^s \{(x_1, n_i)\}$

end

$\mathcal{L} \leftarrow \emptyset$

for $1 \leq i \leq s$ **do**

$G' \leftarrow \{f|_{x_j=n_i} \mid f \in G\} \setminus \{0\}$

$\mathcal{L}' \leftarrow \text{LSG}(G', j-1)$

$\mathcal{L} \leftarrow \mathcal{L} \cup \bigcup_{L' \in \mathcal{L}'} \{L' \cup \{(x_j, n_i)\}\}$

end

return $\mathcal{L}$

end

Algorithmus 15: LSG

Korrektheit: Wegen der Voraussetzung $\text{LSGMENGESCHRANKE}(G) \notin \{0, \infty\}$ und Bemerkung (3) auf Seite 52 ist $G \cap K[x_j, \dots, x_n] \neq \{0\}$ und N daher in jedem Schritt endlich. $\square$

Termination: Die Rekursion terminiert trivialerweise. $\square$

Beispiel: $K = \mathbb{Q}$, $G := \{x_1^5 - 2x_1x_2^2x_3 + x_2^4x_3^2 - 37, x_2^3 + 2x_2^2x_3 - x_3^3, x_3^4 - x_3^3 - x_3^2 - x_3 - 2\}$.
 $true \leftarrow \text{ISTGRÖBNER}(G, \prec_{lex})$.

$60 \leftarrow \text{LSGMENGESCHRANKE}(G)$.

$\{\{(x_1, 2), (x_2, 1), (x_3, -1)\}\} \leftarrow \text{LSG}(G, 3)$.

Teil IV

Systeme linearer Gleichungen mit Koeffizienten in Polynomringen

In diesem Teil wird die folgende Situation behandelt:

Seien $m \in \mathbb{N}_{>0}$, F eine endliche Teilmenge von $K[\underline{x}]^m = K[x_1, \dots, x_n]^m$ und $f_0 = (f_{0,1}, \dots, f_{0,m}) \in K[\underline{x}]^m$ mit $f_{0,i} \in K[\underline{x}]$.

Gesucht ist eine Familie $(h_f)_{f \in F}$ in $K[\underline{x}]$ so, dass

$$\sum_{f \in F} h_f f = f_0.$$

Mit anderen Worten: Man berechne das Urbild von f_0 unter der $K[\underline{x}]$ -linearen Abbildung

$$\varphi : \begin{cases} K[\underline{x}]^F \rightarrow K[\underline{x}]^m \\ (y_f)_{f \in F} \mapsto \sum_{f \in F} y_f f \end{cases}.$$

Dazu muss zuerst überprüft werden, ob f_0 überhaupt im Bild von φ (also dem von F erzeugten $K[\underline{x}]$ -Untermodul von $K[\underline{x}]^m$) enthalten ist. Wenn dies der Fall ist, berechnet man ein Urbild u von f_0 und ein endliches Erzeugendensystem E von $\text{Kern}(\varphi)$.

Die Lösungsmenge des Gleichungssystems ist dann

$$\left\{ u + \sum_{e \in E} c_e e \mid c_e \in K[\underline{x}] \right\} \subseteq K[\underline{x}]^F.$$

In den Kapiteln 12, 13 und 14 werden zuerst die Grundlagen dieses Teils erörtert. Danach wird im Kapitel 15 gezeigt, wie man alle Lösungen eines Systems linearer Gleichungen mit Koeffizienten in $K[\underline{x}]$ bestimmt.

Die in diesem Teil behandelte Theorie ist sehr analog zu jener aus Teil I. Deshalb werden in den folgenden Kapiteln die Beweise nicht immer vollständig ausformuliert, sondern auf den entsprechenden Beweis aus Teil I verwiesen.

Außerdem werden die Algorithmen, die zu denen aus den Teilen I und II analog sind, hier nicht nochmals angeführt. Benützt ein in diesem Teil neu vorgestellter Algorithmus so einen analogen Algorithmus, so wird dem Namen zur Unterscheidung das Präfix „M“ (für Modul) vorgestellt.

In Anhang B (ab Seite 88) finden sich die MAPLE-Implementierungen sowohl der im folgenden neu vorgestellten Algorithmen, als auch jener, die analog zu einem Algorithmus aus Teil I sind.

Kapitel 12

Noethersche Moduln

*Alles kommt zwar wieder,
aber auf eine andere Weise.*

SØREN KIERKEGAARD

R sei ein kommutativer Ring mit Einselement.

Definition 12.1 Ein R -Modul heißt **noethersch**, wenn alle seine R -Untermodule endlich erzeugt sind.

Beispiele:

- (1) Sei K ein Körper. Dann ist jeder endlichdimensionale K -Vektorraum noethersch.
- (2) R ist als Ring genau dann noethersch, wenn R als R -Modul noethersch ist.

Lemma 12.1 Sei V ein noetherscher R -Modul und W ein R -Untermodule von V . Dann ist auch V/W noethersch.

Beweis: Die Abbildung

$$\text{kan} : \begin{cases} V \rightarrow V/W \\ v \mapsto \bar{v} \end{cases}$$

ist R -linear und surjektiv.

Sei $M \leq_R V/W$. Dann ist $V' := \text{kan}^{-1}(M)$ ein R -Untermodule von V . Also gibt es $v_1, \dots, v_k \in V$ so, dass $V' = \langle v_1, \dots, v_k \rangle_R$. Dann ist $M = \langle \bar{v}_1, \dots, \bar{v}_k \rangle_R$. $\square$

Satz 12.1 Sei R noethersch. Dann ist jeder endlich erzeugte R -Modul noethersch. Insbesondere: Endlich erzeugte $K[\underline{x}]$ -Moduln sind noethersch.

Beweis: Sei $V = \langle v_1, \dots, v_k \rangle$ ein R -Modul. Vollständige Induktion über k :
Sei $k = 1$: Die Abbildung

$$\alpha : \begin{cases} R \rightarrow V = Rv_1 \\ r \mapsto rv_1 \end{cases}$$

ist surjektiv, und daher gilt $R/\text{Kern}(\alpha) \cong_R V$. Nach Lemma 12.1 (Seite 57) ist $R/\text{Kern}(\alpha)$ noethersch und daher auch V .

Sei $k > 1$: Die Abbildung

$$\text{kan} : \begin{cases} V \rightarrow V/Rv_k \\ v \mapsto \bar{v} \end{cases}$$

ist surjektiv und R -linear. $V/Rv_k = \langle \bar{v}_1, \dots, \bar{v}_{k-1} \rangle_R$ ist nach Induktionsannahme noethersch. Sei $W \leq_R V$. Dann ist $\text{kan}(W)$ endlich erzeugt und es gibt $w_1, \dots, w_l \in W$ mit $\text{kan}(W) = \langle \bar{w}_1, \dots, \bar{w}_l \rangle_R$. Es gibt ein endliches Erzeugendensystem $u_1, \dots, u_m$ von $(Rv_k) \cap W \leq_R Rv_k$ (Fall $k = 1$). Dann ist $W = \langle u_1, \dots, u_m, w_1, \dots, w_l \rangle_R$, denn: $w \in W \implies \exists c_i \in R : \bar{w} = \sum_{i=1}^l c_i \bar{w}_i \implies \exists d \in R : w = \sum_{i=1}^l c_i w_i + dv_k \implies dv_k = w - \sum_{i=1}^l c_i w_i \in W \cap (Rv_k)$. Insgesamt gilt daher: $\exists d_i \in R : w = \sum_{i=1}^l c_i w_i + \sum_{i=1}^m d_i u_i$. $\square$

Bemerkung: Aus Satz 12.1 folgt: Seien R ein noetherscher Ring, $m, n \in \mathbb{N}_{>0}$ und $\varphi : R^m \rightarrow R^n$ eine R -lineare Abbildung. Dann ist $\text{Kern}(\varphi)$ ein endlich erzeugter R -Untermodul von R^m . Mit anderen Worten: Der Lösungsraum eines homogenen linearen Gleichungssystems mit Koeffizienten in einem noetherschen Ring ist endlich erzeugt.

Kapitel 13

Termordnungen

Zum zehnten Mal wiederholt, wird es gefallen.

QUINTUS FLACCUS HORAZ

Für die nun folgenden Kapitel gelten folgende Bezeichnungen:

- (1) K ist ein Körper, $n \in \mathbb{N}_{>0}$, $K[\underline{x}] = K[x_1, \dots, x_n]$.
- (2) $\underline{x}^i = x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}$ mit $i = (i_1, \dots, i_n) \in \mathbb{N}^n$.
- (3) $P := \{\underline{x}^i \mid i \in \mathbb{N}_{\geq 0}^n, n \in \mathbb{N}_{\geq 0}\}$.
- (4) V ist ein freier, endlich erzeugter $K[\underline{x}]$ -Modul und B eine Basis von V .
- (5) $\mathcal{U} := \{p b \mid p \in P, b \in B\}$.

Bemerkung: $\mathcal{U}$ ist eine K -Basis von V , d.h. $\forall v \in V : \exists (c_u)_{u \in \mathcal{U}} \in K^{\mathcal{U}} : v = \sum_{u \in \mathcal{U}} c_u u$.

Definition 13.1 Eine Totalordnung $\prec$ auf $\mathcal{U}$ heißt **Termordnung** $:\Leftrightarrow$

- (1) $\forall b \in B \forall p \in P : b \preceq p b$,
- (2) $\forall u, v \in \mathcal{U} \forall p \in P : (u \prec v \implies pu \prec pv)$.

Beispiel: Seien $B = \{b_1, \dots, b_m\}$ und $\prec_P$ eine Termordnung auf P . Dann ist durch

$$p b_i \prec q b_j :\Leftrightarrow ((i > j) \vee (i = j \wedge p \prec_P q))$$

eine Termordnung auf $\mathcal{U}$ gegeben.

Diese Termordnung heißt lexikographisch bzw. graduiert-lexikographisch, wenn $\prec_P$ die lexikographische bzw. graduiert-lexikographische Ordnung ist.

Im weiteren sei $\prec$ eine Termordnung auf $\mathcal{U}$.

Definition 13.2 Sei $0 \neq v = \sum_{u \in \mathcal{U}} c_u u \in V$ und $F \subseteq V$. Dann sind die Begriffe $\text{supp}(v)$, $\text{lt}(v)$, $\text{lk}(v)$ und $\text{lt}(F)$ analog wie in Definition 2.3 (Seite 12) definiert.

Beispiel: Seien $V = \mathbb{Q}[x_1, x_2]^3$, $B := \{e_1 := (1, 0, 0), e_2 := (0, 1, 0), e_3 := (0, 0, 1)\}$ und $\prec := \prec_{\text{glex}}$ mit $x_1 > x_2$ und $e_1 > e_2 > e_3$.

$$f_1 := (2x_1^2x_2 + 2x_1^2 - 3, x_1^4x_2, x_1 + x_2),$$

$$f_2 := (x_1x_2^3 + x_1x_2^2, 0, 0),$$

$$f_3 := (0, 0, -2x_1 + 3).$$

$$f_1 = 2x_1^2x_2e_1 + 2x_1^2e_1 - 3e_1 + x_1^4x_2e_2 + x_1e_3 + x_2e_3.$$

$$\text{supp}(f_1) = \{x_1^2x_2e_1, x_1^2e_1, e_1, x_1^4x_2e_2, x_1e_3, x_2e_3\},$$

$$\text{lt}(f_1) = x_1^2x_2e_1, \text{lk}(f_1) = 2.$$

$$\text{lt}(f_2) = x_1x_2^3e_1.$$

$$\text{lt}(f_3) = x_1e_3.$$

Satz 13.1 Jede absteigende Folge in $\mathcal{U}$ ist endlich.

Beweis: Analog zu Satz 2.2 (Seite 13). $\square$

Satz 13.2 (Division mit Rest) F sei eine endliche Teilmenge von $V \setminus \{0\}$, $\prec$ sei eine Termordnung auf $\mathcal{U}$ und $v \in V$.

Dann gibt es eine Familie $(h_f)_{f \in F}$ in $K[x]$ so, dass entweder

$$v = \sum_{f \in F} h_f f$$

oder

$$v \neq \sum_{f \in F} h_f f \quad \text{und} \quad \text{lt}\left(v - \sum_{f \in F} h_f f\right) \notin \langle \text{lt}(F) \rangle_{K[x]}.$$

Außerdem gilt $\max_{\prec} \{\text{lt}(h_f f) \mid f \in F, h_f \neq 0\} = \text{lt}(v)$.

Beweis: Analog zu Satz 2.3 (Seite 13). $\square$

Kapitel 14

Gröbnerbasen von Untermoduln

Strukturen sind die Waffen der Mathematiker.

NICOLAS BOURBAKI

Definition 14.1 Seien V ein freier, endlich erzeugter $K[x]$ -Modul und $\prec$ eine Termordnung auf $\mathcal{U}$. Eine endliche Teilmenge F eines Untermoduls $W \leq_{K[x]} V$ heißt genau dann **Gröbnerbasis von W (bezüglich $\prec$)**, wenn

$$\langle \text{lt}(F) \rangle_{K[x]} = \langle \text{lt}(W) \rangle_{K[x]}.$$

Definition 14.2 Für $v, w \in V \setminus \{0\}$ seien $s, t \in P$ und $b, c \in B$ so, dass $\text{lt}(v) = sb$ und $\text{lt}(w) = tc$. Dann heißt

$$S(v, w) := \begin{cases} 0 & , \text{ falls } b \neq c \\ \text{lk}(w) \frac{\text{kgV}(s,t)}{s} v - \text{lk}(v) \frac{\text{kgV}(s,t)}{t} w & , \text{ falls } b = c \end{cases}$$

der **S-Vektor von v und w** .

Beispiel: Seien $V = \mathbb{Q}[x_1, x_2]^3$, $B := \{e_1 := (1, 0, 0), e_2 := (0, 1, 0), e_3 := (0, 0, 1)\}$ und $\prec := \prec_{\text{glex}}$ mit $x_1 > x_2$ und $e_1 > e_2 > e_3$.

$$f_1 := (x_1^2 + 1, 0, 1), f_2 := (x_1 x_2 + x_2, 0, 0), f_3 := (0, 0, x_1 x_2).$$

$$W := \langle f_1, f_2, f_3 \rangle_{\mathbb{Q}[x]} \leq \mathbb{Q}[x_1, x_2]^3.$$

Dann ist $S(f_1, f_3) = S(f_2, f_3) = 0$ und $S(f_1, f_2) = x_2 f_1 - x_1 f_2 = (-x_1 x_2 + x_2, 0, x_2)$.

Die Aussagen für Gröbnerbasen von Idealen gelten analog für Gröbnerbasen von Untermoduln. Insbesondere gilt der folgende

Satz 14.1 Sei F eine endliche Teilmenge von $W \leq_{K[x]} V$. Dann sind die folgenden Aussagen äquivalent:

- (1) F ist eine Gröbnerbasis von W .
- (2) Für alle $v, w \in F$ ist ein Rest von $S(v, w)$ nach Division durch F gleich null.

Beweis: Analog zu Satz 3.1 (Seite 21). $\square$

Kapitel 15

Lineare Gleichungen mit Koeffizienten in $K[x]$

*Du wolltest doch Algebra,
da hast du den Salat.*

JULES VERNE

In diesem Kapitel gelten folgende Bezeichnungen:

- (1) $V := K[x]^m$ mit der Basis $B := \{e_1 := (1, 0, \dots, 0), e_2 := (0, 1, 0, \dots, 0), \dots, e_m := (0, \dots, 0, 1)\}$.
- (2) F ist eine endliche Teilmenge von V .
- (3) $f_0 \in V$.
- (4) Die Abbildung φ sei definiert durch

$$\varphi : \begin{cases} K[x]^F \rightarrow V \\ (y_f)_{f \in F} \mapsto \sum_{f \in F} y_f f \end{cases} .$$

- (5) G ist eine Gröbnerbasis von $\varphi(K[x]^F) = \langle F \rangle_{K[x]}$; außerdem gilt für $g \in G$: $g = \sum_{f \in F} c(g)_f f$ mit $c(g)_f \in K[x]$.
- (6) Die Abbildung γ sei definiert durch

$$\gamma : \begin{cases} K[x]^G \rightarrow V \\ (y_g)_{g \in G} \mapsto \sum_{g \in G} y_g g \end{cases} .$$

Satz 15.1 *Das durch φ und f_0 gegebene Gleichungssystem hat genau dann eine Lösung (also $f_0 \in \langle F \rangle_{K[x]}$), wenn ein Rest (oder alle Reste) von f_0 nach Division durch G null ist. Existiert eine Lösung und ist $f_0 = \sum_{g \in G} d_g g$, dann ist*

$$\left(\sum_{g \in G} c(g)_f d_g \right)_{f \in F}$$

eine Lösung.

Beweis: Der Beweis des ersten Teils des Satzes ist analog zum Beweis von Satz 2.5 (Seite 18).

$$f_0 = \sum_{g \in G} d_g g = \sum_{\substack{f \in F \\ g \in G}} c(g)_f d_g f = \sum_{f \in F} \left(\sum_{g \in G} c(g)_f d_g \right) f. \quad \square$$

Beispiel: $V = \mathbb{Q}[x_1, x_2]^2$, $f_1 := (x_1 + x_2, x_1^2 x_2)$, $f_2 := (x_1 - x_2, x_1 x_2^2)$, $f_3 := (0, x_1 x_2 + x_1)$, $f_4 := (0, x_1 x_2)$, $F := \{f_1, \dots, f_4\}$, $f_0 := (2x_1^2 + 2x_2^2, x_1 x_2)$.
 $G := \{g_1, \dots, g_3\}$ mit $g_1 := (2x_1, 0)$, $g_2 := (2x_2, 0)$ und $g_3 := (0, x_1)$ ist eine Gröbnerbasis von $\langle F \rangle$ bzgl. $\prec_{\text{plex}}$.

Außerdem gilt:

$$\begin{aligned} g_1 &= f_1 + f_2 - (x_1 x_2 + x_2^2) f_3 + (x_1 x_2 + x_2^2) f_4, \\ g_2 &= f_1 - f_2 + (x_2^2 - x_1 x_2) f_3 + (x_1 x_2 - x_2^2) f_4, \\ g_3 &= f_3 - f_4. \end{aligned}$$

$((x_1, x_2, x_2), 0) \leftarrow \text{MMULTIVARDIV}(f_0, (g_1, g_2, g_3), \prec_{\text{plex}})$.

Das liefert für f_0 die Darstellung:

$$f_0 = x_1 g_1 + x_2 g_2 + x_2 g_3 = (x_1 + x_2) f_1 + (x_1 - x_2) f_2 + (x_2^3 + x_2 - x_1^2 x_2 - 2x_1 x_2^2) f_3 + (x_1^2 x_2 + 2x_1 x_2^2 - x_2^3 - x_2) f_4.$$

Also ist $(x_1 + x_2, x_1 - x_2, x_2^3 + x_2 - x_1^2 x_2 - 2x_1 x_2^2, x_1^2 x_2 + 2x_1 x_2^2 - x_2^3 - x_2)$ eine Lösung des durch φ und f_0 gegebenen inhomogenen Gleichungssystems.

Satz 15.2 O.B.d.A. gelte $\forall g \in G : \text{lk}(g) = 1$. Für $g_1, g_2 \in G$ mit $\text{lt}(g_1) = p_1 b$, $\text{lt}(g_2) = p_2 c$ ($p_1, p_2 \in P$, $b, c \in B$) und $b = c$ seien $p(g_1, g_2), q(g_1, g_2) \in P$ und $t(g_1, g_2)_h \in K[x]$ so, dass

$$S(g_1, g_2) = p(g_1, g_2) g_1 - q(g_1, g_2) g_2 = \sum_{h \in G} t(g_1, g_2)_h h$$

mit $\text{lt}(S(g_1, g_2)) = \max_{\prec} \{\text{lt}(t(g_1, g_2)_h h) \mid h \in G, t(g_1, g_2)_h h \neq 0\}$. Weiters sei $u(g_1, g_2) := (u(g_1, g_2)_h)_{h \in G}$ mit

$$u(g_1, g_2)_h := \begin{cases} t(g_1, g_2)_{g_1} - p(g_1, g_2) & , \text{ falls } h = g_1 \\ t(g_1, g_2)_{g_2} + q(g_1, g_2) & , \text{ falls } h = g_2 \\ t(g_1, g_2) & , \text{ sonst.} \end{cases}$$

Dann ist $\{u(g_1, g_2) \mid g_1, g_2 \in G\}$ ein $K[x]$ -Erzeugendensystem von $\text{Kern}(\gamma)$, d.h. ein $K[x]$ -Erzeugendensystem des durch G gegebenen homogenen Gleichungssystems.

Beweis: Sei $W := \langle u(g_1, g_2) \mid g_1, g_2 \in G \rangle_{K[x]}$.

Zu zeigen ist dann: $W = \text{Kern}(\gamma)$.

Wegen $\sum_{h \in G} u(g_1, g_2)_h h = \sum_{g \in G} t(g_1, g_2)_h h - p(g_1, g_2) g_1 + q(g_1, g_2) g_2 = 0$ ist $W \leq_{K[x]} \text{Kern}(\gamma)$.

$\text{Kern}(\gamma)$.

Annahme: $W \neq \text{Kern}(\gamma)$.

Wähle $v = (v_h)_{h \in G} \in \text{Kern}(\gamma) \setminus W$ so, dass für alle $v' \in (\text{Kern}(\gamma) \setminus W) \setminus \{v\}$ entweder

$$M(v) := \max_{\prec} \{\text{lt}(v_h h) \mid h \in G, v_h h \neq 0\} \prec M(v')$$

oder

$$M(v) = M(v') \quad \wedge \quad A(v) := |\{h \in G \mid v_h h \neq 0, \text{lt}(v_h h) = M(v)\}| < A(v')$$

gilt.

Wegen $v \in \text{Kern}(\gamma)$ ist $\sum_{h \in G} v_h h = 0$. Also gibt es $g_1, g_2 \in G$ mit $g_1 \neq g_2$ und $\text{lt}(v_{g_1} g_1) = \text{lt}(v_{g_2} g_2) = M(v)$. Dann gibt es ein $z \in P$ so, dass $z \cdot p(g_1, g_2) \cdot \text{lt}(g_1) = z \cdot q(g_1, g_2) \cdot \text{lt}(g_2) = M(v)$. Es sei nun $w := v + \text{lk}(v_{g_1}) z u(g_1, g_2)$. Wegen $v \in \text{Kern}(\gamma) \setminus W$ und $u(f, g) \in W \leq \text{Kern}(\gamma)$ ist $w \in \text{Kern}(\gamma) \setminus W$. Es ist

$$\begin{aligned} w_h &= (v + \text{lk}(v_{g_1}) z u(g_1, g_2))_h \\ &= \begin{cases} v_{g_1} + \text{lk}(v_{g_1}) z t(g_1, g_2)_{g_1} - \text{lk}(v_{g_1}) z p(g_1, g_2) & , \text{ falls } h = g_1 \\ v_{g_2} + \text{lk}(v_{g_1}) z t(g_1, g_2)_{g_2} + \text{lk}(v_{g_1}) z q(g_1, g_2) & , \text{ falls } h = g_2 \\ v_h + \text{lk}(v_{g_1}) z t(g_1, g_2)_h & , \text{ sonst.} \end{cases} \end{aligned}$$

Dann gilt: $\text{lt}(w_h h) \preceq M(v)$ für $h \neq g_1$ und

$$\text{lt}(w_{g_1} g_1) \preceq \max_{\prec} \left\{ \underbrace{\text{lt}((v_{g_1} - \text{lk}(v_{g_1}) z p(g_1, g_2)) \text{lt}(g_1))}_{\prec M(v)}, \underbrace{\text{lt}(z t(g_1, g_2)_{g_1} g_1)}_{\prec \text{lt}(z p(g_1, g_2) g_1)} \right\} \preceq M(v).$$

Also gilt: $M(w) \prec M(v) \quad \vee \quad (M(w) = M(v) \wedge A(w) < A(v))$. Das ist jedoch ein Widerspruch zur Minimalität von v . $\square$

Beispiel: Seien V, G und F wie in dem Beispiel auf Seite 63 definiert.

Dann ist $S(g_1, g_2) = x_2 g_1 - x_1 g_2 = 0$ und $S(g_1, g_3) = S(g_2, g_3) = 0$.

Also ist $u(g_1, g_2) = (x_2, -x_1, 0)$ und $\text{Kern}(\gamma) = \langle (x_2, -x_1, 0) \rangle_{\mathbb{Q}[x]}$.

Lemma 15.1 Für $f \in F$ sei $f = \sum_{g \in G} d(f)_g g$. Außerdem sei $z(f) := (z(f)_h)_{h \in F} \in K[x]^F$ mit

$$z(f)_h := \sum_{g \in G} d(f)_g c(g)_h - \delta_{f h}.$$

Dann gilt: $z(f) \in \text{Kern}(\varphi)$.

Beweis: Wegen $g = \sum_{f \in F} c(g)_f f$ ist $f = \sum_{g \in G} d(f)_g \sum_{h \in F} c(g)_h h = \sum_{h \in F} \left(\sum_{g \in G} d(f)_g c(g)_h \right) h$.

Da $f = \sum_{h \in F} \delta_{f h} h$, ist $\sum_{h \in F} \underbrace{\left(\sum_{g \in G} d(f)_g c(g)_h - \delta_{f h} \right)}_{z(f)_h} h = 0$. $\square$

Lemma 15.2 Sei $u := (u_g)_{g \in G} \in \text{Kern}(\gamma)$. Außerdem sei $y(u) := (y(u)_h)_{h \in F} \in K[x]^F$ mit

$$y(u)_h := \sum_{g \in G} u_g c(g)_h.$$

Dann gilt: $y(u) \in \text{Kern}(\varphi)$.

Beweis: $0 = \sum_{g \in G} u_g g = \sum_{g \in G} u_g \sum_{f \in F} c(g)_f f = \sum_{f \in F} \underbrace{\left(\sum_{g \in G} u_g c(g)_f \right)}_{y(u)_f} f$. $\square$

Satz 15.3 Es gelten die gleichen Bezeichnungen wie in den Lemmata 15.1 und 15.2. Es sei ein endliches Erzeugendensystem von $\text{Kern}(\gamma)$.

Dann ist $\{z(f) | f \in F\} \cup \{y(u) | u \in E\}$ ein $K[x]$ -Erzeugendensystem von $\text{Kern}(\varphi)$.

Beweis: Sei $v \in \text{Kern}(\varphi)$. Dann ist $\sum_{f \in F} v_f f = 0$ und daher gilt $\sum_{g \in G} \left(\sum_{f \in F} v_f d(f)_g \right) g = 0$. Es gibt daher Polynome $b_u \in K[x]$ mit $\left(\sum_{f \in F} d(f)_g v_f \right)_{g \in G} = \sum_{u \in E} b_u u$.

Behauptung: $v = \sum_{u \in E} b_u y(u) - \sum_{f \in F} v_f z(f)$.

$$\begin{aligned}
 \text{Beweis: } & \left(\sum_{u \in E} b_u y(u) - \sum_{f \in F} v_f z(f) \right)_h \\
 &= \sum_{u \in E} b_u \sum_{g \in G} u_g c(g)_h - \sum_{f \in F} \left(\sum_{g \in G} d(f)_g c(g)_h - \delta_{fh} \right) v_f \\
 &= \sum_{g \in G} c(g)_h \sum_{u \in E} b_u u_g - \sum_{g \in G} c(g)_h \sum_{f \in F} d(f)_g v_f + v_h \\
 &= \sum_{g \in G} c(g)_h \left(\sum_{u \in E} b_u u_g - \sum_{f \in F} d(f)_g v_f \right) + v_h \\
 &= \sum_{g \in G} c(g)_h \left(\sum_{f \in F} d(f)_g v_f - \sum_{f \in F} d(f)_g v_f \right) + v_h \\
 &= v_h \quad \square
 \end{aligned}$$

In den folgenden Abschnitten werden die in den Sätzen 15.1, 15.2 und 15.3 gezeigten Berechnungsverfahren als Algorithmen formuliert. Um diese möglichst einfach zu notieren, werden dabei die meisten Berechnungen mit Matrizenmultiplikationen durchgeführt.

Der im folgenden verwendete Algorithmus MODULBASIS entspricht dabei dem Algorithmus IDEALBASIS aus Abschnitt 5.4 (Seite 33).

15.1 Der Algorithmus INHOMOGENELSG

In diesem Abschnitt wird der Algorithmus INHOMOGENELSG behandelt, der eine Lösung des durch φ und f_0 gegebenen inhomogenen Gleichungssystems berechnet.

Bemerkung: Die Berechnung einer Gröbnerbasis mittels BUCHBERGER-Algorithmus und die Berechnung der Darstellung der g_i bzgl. F (in der for-Schleife mittels MODULBASIS) können zur Effizienzsteigerung algorithmisch in einem Schritt durchgeführt werden, indem der BUCHBERGER-Algorithmus etwas erweitert wird. Dies ist in der MAPLE-Funktion `MbuchbergerKoeff` in Anhang B (ab Seite 88) durchgeführt.

Korrektheit: Folgt unmittelbar aus Satz 15.1 (Seite 62). $\square$

Termination: Folgt aus der Termination von MBUCHBERGER, MODULBASIS und MMULTIVARDIV. $\square$

Beispiel: Seien F und f_0 wie in dem Beispiel auf Seite 63 definiert.

$(x_1 + x_2, x_1 - x_2, x_2^3 + x_2 - x_1^2 x_2 - 2x_1 x_2^2, x_1^2 x_2 + 2x_1 x_2^2 - x_2^3 - x_2)$ —INHOMOGENELSG(F, f_0).

15.2 Der Algorithmus KERNGAMMA

Der Algorithmus KERNGAMMA berechnet gemäß Satz 15.2 (Seite 63) ein Erzeugendensystem von $\text{Kern}(\gamma)$.

Spezifikation: $v \leftarrow \text{INHOMOGENELSG}(F, f_0)$
 Berechnet eine Lösung des durch F und f_0 gegebenen inhomogenen Gleichungssystems.

Eingabe: Eine endliche Teilmenge F von $V = K[x]^m$ und $f_0 \in V$.

Ausgabe: $v = (v_f)_{f \in F}$ mit $\sum_{f \in F} v_f f = f_0$.

begin

Sei $\prec$ eine Termordnung.

$G \leftarrow \text{MBUCHBERGER}(F, \prec)$

Seien $G = \{g_1, \dots, g_s\}$ und $C = (c_{ij})_{\substack{1 \leq i \leq s \\ 1 \leq j \leq |F|}}$.

for $1 \leq i \leq s$ **do**

$(c_{i1}, \dots, c_{i|F|}) \leftarrow \text{MODULBASIS}(g_i, F)$

end

$((d_1, \dots, d_s), r) \leftarrow \text{MMULTIVARDIV}(f_0, (g_1, \dots, g_s), \prec)$

return $(d_1, \dots, d_s) \cdot C$

end

Algorithmus 16: INHOMOGENELSG

Korrektheit: Folgt unmittelbar aus Satz 15.2 (Seite 63). $\square$

Termination: Folgt aus der Termination von MMULTIVARDIV. $\square$

Beispiel: Sei G wie in dem Beispiel auf Seite 63 definiert.

$\{(-x_2, x_1, 0)\} \leftarrow \text{KERNGAMMA}(G, \prec_{\text{glex}})$.

15.3 Der Algorithmus KERNPHI

Der Algorithmus KERNPHI berechnet gemäß Satz 15.3 (Seite 64) ein Erzeugendensystem von $\text{Kern}(\varphi)$.

Bemerkungen:

- (1) I_m ist die $m \times m$ Einheitsmatrix.
- (2) Die Berechnung einer Gröbnerbasis mittels BUCHBERGER-Algorithmus und die Berechnung der Darstellung der g_i bzgl. F (in der for-Schleife mittels MODULBASIS) können zur Effizienzsteigerung algorithmisch in einem Schritt durchgeführt werden, indem der BUCHBERGER-Algorithmus etwas erweitert wird. Dies ist in der MAPLE-Funktion `MbuchbergerKoeff` in Anhang B (ab Seite 88) durchgeführt.

Korrektheit: Folgt unmittelbar aus Satz 15.3 (Seite 64). $\square$

Spezifikation: $E \leftarrow \text{KERNGAMMA}(G, \prec)$

Berechnet ein Erzeugendensystem von $\text{Kern}(\gamma)$.

Eingabe: Eine Gröbnerbasis $G = \{g_1, \dots, g_k\} \subseteq V = K[\underline{x}]^m$ bzgl. der Termordnung $\prec$.

Ausgabe: Ein Erzeugendensystem von $\text{Kern}(\gamma)$, d.h. ein Erzeugendensystem der Lösungsmenge des durch G gegebenen homogenen Gleichungssystems.

begin

$E \leftarrow \emptyset$

for $1 \leq i \leq k$ **do**

for $i + 1 \leq j \leq k$ **do**

Seien $\text{lt}(g_i) = s_i e_{m_i}$ und $\text{lt}(g_j) = s_j e_{m_j}$ mit $s_i, s_j \in P, e_{m_i}, e_{m_j} \in B$.

if $m_i = m_j$ **then**

$p \leftarrow \text{lk}(g_j) \frac{\text{kgV}(s_i, s_j)}{s_i}$

$q \leftarrow \text{lk}(g_i) \frac{\text{kgV}(s_i, s_j)}{s_j}$

$((t_1, \dots, t_k), r) \leftarrow \text{MMULTIVARDIV}(pg_i - qg_j, (g_1, \dots, g_k), \prec)$

$E \leftarrow E \cup \{(t_1, \dots, t_{i-1}, t_i - p, t_{i+1}, \dots, t_{j-1}, t_j - q, t_{j+1}, \dots, t_k)\}$

end

end

end

return E

end

Algorithmus 17: KERNGAMMA

Termination: Folgt aus der Termination von MBUCHBERGER, MODULBASIS, MMULTIVARDIV und KERNGAMMA. $\square$

Beispiel: Sei F wie in dem Beispiel auf Seite 63 definiert.

$\{((0, 0, x_2, -x_2 - 1), (x_1 - x_2, -x_1 - x_2, 2x_1x_2 + x_2^2 - x_2, -x_1^2 - 2x_1 + 1))\} \leftarrow \text{KERNPHI}(F)$.

Spezifikation: $E \leftarrow \text{KERNPHI}(F)$

Berechnet ein Erzeugendensystem von $\text{Kern}(\varphi)$.

Eingabe: Eine endliche Teilmenge F von $V = K[x]^m$.

Ausgabe: Ein Erzeugendensystem von $\text{Kern}(\varphi)$, d.h. ein Erzeugendensystem der Lösungsmenge des durch F gegebenen homogenen Gleichungssystems.

begin

Sei $\prec$ eine Termordnung.

$G \leftarrow \text{MBUCHBERGER}(F, \prec)$

Seien $G = \{g_1, \dots, g_s\}$ und $C = (c_{ij})_{\substack{1 \leq i \leq s \\ 1 \leq j \leq |F|}}$.

for $1 \leq i \leq s$ **do**

$(c_{i1}, \dots, c_{i|F|}) \leftarrow \text{MODULBASIS}(g_i, F)$

end

Sei $D = (d_{ij})_{\substack{1 \leq i \leq |F| \\ 1 \leq j \leq s}}$.

for $1 \leq i \leq |F|$ **do**

$((d_{i1}, \dots, d_{is}), r) \leftarrow \text{MMULTIVARDIV}(f_i, (g_1, \dots, g_s), \prec)$

end

$(z_{ij})_{\substack{1 \leq i \leq |F| \\ 1 \leq j \leq |F|}} \leftarrow D \cdot C - I_{|F|}$

$E \leftarrow \{(z_{i1}, \dots, z_{i|F|}) \mid 1 \leq i \leq |F|\}$

$E_\gamma \leftarrow \text{KERNGAMMA}(G, \prec)$

if $E_\gamma \neq \emptyset$ **then**

Seien $E_\gamma = \{(a_{i1}, \dots, a_{i|G|}) \mid 1 \leq i \leq t, t \in \mathbb{N}_{>0}\}$ und $A = (a_{ij})_{\substack{1 \leq i \leq t \\ 1 \leq j \leq s}}$.

$(y_{ij})_{\substack{1 \leq i \leq t \\ 1 \leq j \leq |F|}} \leftarrow A \cdot C$

return $E \cup \{(y_{i1}, \dots, y_{i|F|}) \mid 1 \leq i \leq t\}$

end

return E

end

Algorithmus 18: KERNPHI

Teil V

Primärzerlegung von Idealen

In diesem Teil wird gezeigt, dass jedes Ideal eines noetherschen Rings als endlicher Durchschnitt sogenannter primärer Ideale (siehe Definition 15.3 auf Seite 71) dargestellt werden kann. Diese Aussage wird dann noch verstärkt: Jedes Ideal hat eine sogenannte Primärzerlegung (siehe Definition 15.5 auf Seite 72), bei der noch Anforderungen an die Radikale der einzelnen Ideale des Durchschnitts gestellt werden.

Dieser Teil bietet nur eine Einführung in die Primärzerlegung von Idealen. Anwendungen der hier vorgestellten Theorie, Erweiterungen und deren algorithmische Umsetzungen finden sich in [2] und vor allem in [1].

Sei R ein noetherscher Ring.

Definition 15.1 Ein Ideal $I \trianglelefteq R$ heißt **prim**, wenn

$$I \neq R \wedge (\forall r, s \in R : (rs \in I \implies r \in I \vee s \in I))$$

gilt.

Definition 15.2 Ein Ideal $I \trianglelefteq R$ heißt **reduzibel**, wenn zwei Ideale $I_1, I_2 \trianglelefteq R$ so existieren, dass

$$I = I_1 \cap I_2 \quad \text{und} \quad I_1, I_2 \neq I.$$

Andernfalls heißt I **irreduzibel**.

Satz 15.4 Jedes Ideal $I \trianglelefteq R$ ist ein endlicher Durchschnitt von irreduziblen Idealen.

Beweis: Sei N die Menge aller Ideale, die nicht ein endlicher Durchschnitt von irreduziblen Idealen sind.

Annahme: $N \neq \emptyset$.

Jedes $I \in N$ ist reduzibel, da es sonst selbst ein endlicher Durchschnitt von irreduziblen Idealen wäre. Daher gibt es Ideale I_1 und I_2 so, dass $I = I_1 \cap I_2$ und $I_1, I_2 \neq I$. Zumindest eines der beiden Ideale I_1 und I_2 ist nun selbst wiederum Element von N , da sonst $I_1 \cap I_2$ ein endlicher Durchschnitt von irreduziblen Idealen wäre, was zu einem Widerspruch zu $I \in N$ führen würde. O.B.d.A. sei $I_1 \in N$. Mit I_1 wird nun genauso verfahren. Dies führt zu einer unendlichen, echt aufsteigenden Folge von Idealen $I_1 \subset I_2 \subset I_3 \subset \dots$ in R . Da R noethersch ist, ist dies ein Widerspruch zu Satz 1.4 (Seite 8). $\square$

Definition 15.3 Ein Ideal $I \trianglelefteq R$ heißt **primär**, wenn

$$\forall r, s \in R : ((rs \in I \wedge r \notin I) \implies \exists n \in \mathbb{N} : s^n \in I)$$

gilt.

Das folgende Lemma liefert eine symmetrischere Formulierung dieser Definition.

Lemma 15.3 Ein Ideal $I \trianglelefteq R$ ist genau dann primär, wenn

$$\forall r, s \in R : ((rs \in I \wedge r, s \notin I) \implies \exists n, m \in \mathbb{N} : r^n, s^m \in I)$$

gilt.

Beweis: trivial. $\square$

Lemma 15.4 Ist $I \trianglelefteq R$ primär, so ist $\text{Rad}(I)$ ein primes Ideal.

Beweis: Sei $rs \in \text{Rad}(I)$ und o.B.d.A. $s \in \text{Rad}(I)$. Dann gilt: $\exists n \in \mathbb{N} : (rs)^n = r^n s^n \in I$ und $\nexists m \in \mathbb{N} : s^m \in I$. Da I primär ist, folgt daraus $r^n \in I$, also $r \in \text{Rad}(I)$. $\square$

Definition 15.4 Ist I ein primäres Ideal mit $\text{Rad}(I) = J$, so heißt I **J -primär** und J das assoziierte Primideal von I .

Satz 15.5 Ein irreduzibles Ideal ist primär.

Beweis: Annahme: Sei $I \subseteq R$ ein irreduzibles Ideal, das nicht primär ist. Dann gilt:

$$\exists a, b \in R \setminus I : (ab \in I) \wedge (\nexists n, m \in \mathbb{N} : a^n, b^m \in I).$$

Man betrachte nun die aufsteigende Folge der Ideale $I : \langle b^1 \rangle, I : \langle b^2 \rangle, I : \langle b^3 \rangle, \dots$. Da R noethersch ist, folgt aus Satz 1.4 (Seite 8), dass diese Folge endlich ist; d.h. es gibt ein $n \in \mathbb{N}$ so, dass $I : \langle b^n \rangle = I : \langle b^{n+1} \rangle$. Da $a, b^n \notin I$, gilt: $I + \langle a \rangle, I + \langle b^n \rangle \supset I$.

Für einen Widerspruch zur Irreduzibilität von I zeigt man nun: $(I + \langle a \rangle) \cap (I + \langle b^n \rangle) = I$. Sei $r \in (I + \langle a \rangle) \cap (I + \langle b^n \rangle)$, dann ist $r = q_1 + r_1 a = q_2 + r_2 b^n$ mit $q_1, q_2 \in I$ und $r_1, r_2 \in R$. Multiplikation mit b ergibt: $rb = q_1 b + r_1 ab = q_2 b + r_2 b^{n+1}$. Aus $ab, q_1 b \in I$ folgt $rb \in I$ und aus $q_2 b \in I$ folgt $r_2 b^{n+1} \in I$. Daher gilt: $r_2 \in I : \langle b^{n+1} \rangle = I : \langle b^n \rangle$; also: $r_2 b^n \in I$. Daraus folgt: $r = q_2 + r_2 b^n \in I$. $\square$

Satz 15.6 Seien I_1 und I_2 J -primär, dann ist auch $I_1 \cap I_2$ J -primär.

Beweis: Sei $ab \in I_1 \cap I_2$ und o.B.d.A. $a \notin I_1 \cap I_2$. O.b.d.A. sei $a \notin I_1$. Da I_1 primär ist, existiert ein $m \in \mathbb{N}$ so, dass $b^m \in I_1$. Aus $I_1 \subseteq \text{Rad}(I_1) = \text{Rad}(I_2)$ folgt, dass ein $n \in \mathbb{N}$ so existiert, dass $(b^m)^n \in I_1 \cap I_2$. $\square$

Satz 15.7 Jedes Ideal eines noetherschen Rings ist ein endlicher Durchschnitt von primären Idealen.

Beweis: Folgt unmittelbar aus den Sätzen 15.4, 15.5 und 15.6. $\square$

Definition 15.5 Seien $I \subseteq R$ und $I_1, \dots, I_r$ primäre Ideale von R . Dann heißen die I_i eine **Primärzerlegung von I** , wenn

$$(1) \quad I = \bigcap_{i=1}^r I_i,$$

$$(2) \quad \nexists i \in \{1, \dots, r\} : \bigcap_{\substack{1 \leq j \leq r \\ j \neq i}} I_j \subseteq I_i,$$

(3) Die assoziierten Primideale der I_i sind paarweise verschieden.

gilt. Die I_i heißen dann **primäre Komponenten von I** .

Satz 15.8 Jedes Ideal $I \subsetneq R$ hat eine Primärzerlegung.

Beweis: Seien $I = I_1 \cap \dots \cap I_r$ und $J_i := \text{Rad}(I_i)$. Man kann nun alle Komponenten in dieser Zerlegung zusammensammeln, deren Radikal J_1 ist. Der Durchschnitt aller dieser Ideale sei das J_1 -primäre Ideal $\bar{I}_1$. Analog verfährt man für die anderen Radikale. So erhält man die Zerlegung $I = \bar{I}_1 \cap \dots \cap \bar{I}_t$, wobei jedes $\bar{I}_i$ J_i -primär ist und die Radikale der $\bar{I}_i$ paarweise verschieden sind. Wenn man jetzt alle $\bar{I}_i$ entfernt, die den Durchschnitt der übrigen Ideale

enthalten, so erhält man eine Primärzerlegung des Ideals I . $\square$

Beispiel: Sei $I = I_1 \cap I_2$ mit $I_1 := \langle x \rangle$ und $I_2 := \langle x^2, y \rangle$; I_1 und I_2 sind primär. I_1 besteht aus allen Polynomen, bei denen das konstante Glied und die Koeffizienten aller Potenzen von y gleich null sind. I_2 besteht aus allen Polynomen, bei denen das konstante Glied und der Koeffizient bei x gleich null sind. Daher gilt: $I = I_1 \cap I_2 = \langle x^2, xy \rangle$. Dieses Ideal ist nicht primär. Eine andere Primärzerlegung von I ist $I = \langle x \rangle \cap \langle x^2, xy, y^2 \rangle$. Für beide Zerlegungen gilt: Die Anzahl der primären Komponenten ist zwei und die assoziierten Primideale sind jeweils $\langle x \rangle$ und $\langle x, y \rangle$.

Die Primärzerlegung eines Ideals ist also nicht eindeutig. Dass die Situation aus diesem Beispiel jedoch kein Zufall war und eine gewisse Art der Eindeutigkeit sehr wohl besteht, zeigt der nächste Satz.

Lemma 15.5 Sei $r \in \mathbb{N}_{>1}$ und seien $I_1, \dots, I_r$ eine Primärzerlegung eines Ideals $I \trianglelefteq R$. Dann ist $I = \bigcap_{i=1}^r I_i$ nicht primär.

Beweis: Für $1 \leq i \leq r$ sei $J_i := \text{Rad}(I_i)$.

Annahme: I ist primär.

O.B.d.A. sei J_1 unter den assoziierten Primidealen J_i minimal bzgl. der Inklusion. Daher existieren $a_2, \dots, a_r \in R$ mit $a_i \in J_i \setminus J_1$ für alle $2 \leq i \leq r$. Außerdem gilt: $\forall 2 \leq i \leq r : \exists n_i \in \mathbb{N} : a_i^{n_i} \in I_i$.

Es gilt $I \subsetneq I_1$, da sonst jedes I_i außer I_1 den Schnitt der übrigen primären Komponenten enthalten würde. Sei $a \in I_1 \setminus I$. Man betrachte

$$(*) \quad a \cdot (a_2^{n_2} \cdots a_r^{n_r}) \in \bigcap_{i=1}^r I_i = I.$$

Trivialerweise liegt der erste Faktor a nicht in I . Auch der zweite Faktor liegt nicht in I , da sonst ein $m \in \mathbb{N}$ mit

$$(a_2^{n_2} \cdots a_r^{n_r})^m \in I \subsetneq I_1 \subseteq J_1$$

existieren würde und daher für ein i mit $2 \leq i \leq r$ gelten würde: $a_i \in J_1$. Daher steht die Eigenschaft $(*)$ im Widerspruch dazu, dass I primär ist. $\square$

Satz 15.9 Seien $I = I_1 \cap \cdots \cap I_r = I'_1 \cap \cdots \cap I'_s$ zwei Primärzerlegungen von I mit $J_i := \text{Rad}(I_i)$ und $J'_j := \text{Rad}(I'_j)$ für alle $1 \leq i \leq r$ und $1 \leq j \leq s$. Dann gilt: $\{J_1, \dots, J_r\} = \{J'_1, \dots, J'_s\}$ und insbesondere $r = s$.

Beweis: Der Beweis erfolgt mittels vollständiger Induktion über r .

Ist $r = 1$, dann ist $s = 1$, $I_1 = I'_1$ und $J_1 = J'_1$, da sonst ein Widerspruch zu Lemma 15.5 auftreten würde.

Sei nun $r > 1$. Unter den Idealen $J_1, \dots, J_r, J'_1, \dots, J'_s$ existiert ein maximales bzgl. der Inklusion. Die Behauptung, die nun gezeigt wird, ist, dass dieses bzgl. der Inklusion maximale Ideal in beiden Primärzerlegungen auftritt. O.B.d.A. sei J_1 dieses Ideal in der Zerlegung $I = I_1 \cap \cdots \cap I_r$. Es wird nun durch indirekten Beweis gezeigt, dass eines der Ideale J'_j gleich J_1 ist.

Annahme: $J_1 \neq J'_j$ für alle $1 \leq j \leq s$.

Um diese Annahme auf einen Widerspruch zu führen, wird folgende Gleichheit gezeigt:

$$\bigcap_{i=2}^r I_i = \bigcap_{j=1}^s I'_j.$$

Wäre diese Gleichung nämlich erfüllt, ergäbe sich ein Widerspruch zu Eigenschaft (2) einer Primärzerlegung (Definition 15.5, Seite 72).

Die Inklusion „ $\supseteq$ “ ist trivial. Seien nun $a \in \bigcap_{i=2}^r I_i$ und $1 \leq j \leq s$. Wegen der Voraussetzung über J_1 existiert ein $b_j \in J_1 \setminus J'_j$. Außerdem existiert ein $m \in \mathbb{N}$ so, dass $b_j^m \in I_1$. Daher gilt:

$$a b_j^m \in \bigcap_{i=1}^r I_i \subseteq I'_j.$$

Daraus folgt: $a \in I'_j$, da $b_j \notin J'_j$ und daher keine Potenz von b_j in I'_j liegt.

Also ist der Widerspruch gezeigt und o.B.d.A. sei $J_1 = J'_1$. Um den Induktionsbeweis abzuschließen, wird nun noch

$$\bigcap_{i=2}^r I_i = \bigcap_{j=2}^s I'_j$$

gezeigt. Aus Symmetriegründen reicht es, nur eine der beiden Inklusionsrichtungen zu zeigen. Seien also $a \in \bigcap_{i=2}^r I_i$ und $2 \leq j \leq s$. Da die assoziierten Primideale J'_j paarweise verschieden sind und $J_1 = J'_1$ maximal bzgl. der Inklusion ist, gibt es ein $b_j \in J_1 \setminus J'_j$. Nun kann man analog wie oben beweisen, dass $a \in I'_j$. $\square$

Definition 15.6 Eine primäre Komponente I_i eines Ideals I heißt **isoliert**, wenn das assoziierte Primideal von I_i kein assoziiertes Primideal einer anderen primären Komponente von I enthält. Andernfalls heißt I_i **eingebettet**.

Beispiel: Sei $I := \langle x^2, xy \rangle$ und $I = \langle x \rangle \cap \langle x^2, y \rangle$ eine Primärzerlegung von I . Wegen $\text{Rad}(\langle x \rangle) = \langle x \rangle \subseteq \langle x, y \rangle = \text{Rad}(\langle x, y \rangle)$ ist $\langle x \rangle$ eine isolierte Komponente und $\langle x^2, y \rangle$ eine eingebettete Komponente. Eine andere Primärzerlegung von I ist $I = \langle x \rangle \cap \langle x^2, xy, y^2 \rangle$.

Beide Zerlegungen haben also die selbe isolierte Komponente und der nächste Satz zeigt, dass das kein Zufall war. Zuerst wird noch ein Lemma angeführt, das für den Beweis dieses Satzes notwendig ist.

Lemma 15.6 Seien $I, I_1, \dots, I_r$ Ideale von R mit $J_i := \text{Rad}(I_i)$ für alle $1 \leq i \leq r$. Außerdem sei P ein primes Ideal von R , das keines der Ideale J_i ($1 \leq i \leq r$) enthält. Dann existiert ein $b \in (I_1 \cdots I_r) \setminus P$.

Beweis: Sei $b_i \in J_i \setminus P$ für alle $1 \leq i \leq r$. Dann existieren $n_i \in \mathbb{N}_{>0}$ so, dass $b_i^{n_i} \in I_i$ für alle $1 \leq i \leq r$. Daher gilt: $b := b_1^{n_1} \cdots b_r^{n_r} \in I_1 \cdots I_r$. Es gilt aber $b \notin P$, da sonst zumindest ein b_i in P liegen würde. $\square$

Satz 15.10 Seien $I \trianglelefteq R$ und P das assoziierte Primideal einer isolierten, primären Komponente von I . Dann existiert ein primäres Ideal $I_P \trianglelefteq R$ so, dass in jeder Primärzerlegung von I das primäre Ideal, dessen assoziiertes Primideal P ist, gleich I_P ist.

Beweis: Sei $I = Q \cap \bigcap_{i=1}^r I_i$ eine Primärzerlegung und sei Q eine isolierte Komponente, deren assoziiertes Primideal P ist. Ist $r = 0$, dann ist die Aussage des Satzes trivialerweise erfüllt. Andernfalls sei $M := R \setminus P$ und

$$I_M := \{a \in R \mid \exists m \in M : am \in I\}.$$

Behauptung: $Q = I_M$. In diesem Fall wäre Q eindeutig durch P und I bestimmt.

Beweis: Sei $a \in Q$. Da P und die Ideale $\text{Rad}(I_i)$ paarweise verschieden sind und Q isoliert ist, gilt wegen Lemma 15.6:

$$\exists m \in (I_1 \cdots I_r) \setminus P = (I_1 \cdots I_r) \cap M.$$

Daraus folgt

$$am \in Q \cap \bigcap_{i=1}^r I_i = I$$

und daher $a \in I_M$. Sei andererseits $a \in I_M$ und sei $m \in M$ so, dass $am \in I$. Dann ist $am \in Q$ und da keine Potenz von m in Q liegt (da sonst $m \in P$), gilt $a \in Q$. $\square$

Bemerkung: Satz 15.10 besagt mit anderen Worten: Die isolierten, primären Komponenten einer Primärzerlegung sind eindeutig.

Anhang A

Algorithmen in MAPLE

In Anhang A werden die MAPLE-Implementierungen der Algorithmen aus den Teilen I, II und III angeführt. Alle Algorithmen setzen dabei das Einbinden des Paketes Groebner voraus:

```
with(Groebner):
```

A.1 Der Algorithmus MULTIVARDIV

```
multivardiv:=proc(g,F,tord,q) local h,i,red,qh;

h:=g;
q:=array(1..nops(F));

for i from 1 to nops(F) do
    q[i]:=0;
end; #for

while h<>0 do
    red:=false;

    for i from 1 to nops(F) do
        if divide(leadterm(h,tord),leadterm(F[i],tord),'quot')
            then
                qh:=leadcoeff(h,tord)/leadcoeff(F[i],tord)*quot;
                h:=expand(h-qh*F[i]);
                q[i]:=q[i]+qh;
                red:=true;
                break;
            end if;
    end; #for

    if not(red) then
        return h;
    end if;
end;
```

```
od: #while
```

```
return 0;
```

```
end:
```

Beispiel:

```
f1:=x^2:
```

```
f2:=x*y+1:
```

```
g:=x*y^4+x*y^3+x^2*y:
```

```
multivardiv(g, [f1, f2], plex(x, y), 'q'):
```

```
rest:=expand(%);
```

```
rest := -y^2-y^3
```

```
print(q);
```

```
[y, y^3+y^2]
```

A.2 Der Algorithmus NORMALF

Für die Implementierung des Algorithmus NORMALF in MAPLE wird die Hilfsfunktion `suppL` definiert.

`suppL(f)` gibt $\text{supp}(f)$ als Liste zurück.

```
suppL:=proc(f) local i,s,h,t;
```

```
if f=0 then
```

```
return [];
```

```
end if;
```

```
if type(f, monomial) then
```

```
return [f/lcoeff(f)];
```

```
end if;
```

```
h:=[op(expand(f))];
```

```
s:=[];
```

```
for i from 1 to nops(h) do
```

```
s:=[op(s), h[i]/lcoeff(h[i])];
```

```
end; #for i
```

```
return s;
```

```
end:
```

```
nf:=proc(g, F, tord) local G, i, j, n, Sp, m, q, c, p, f;
```

```
G:=[];
```

```
for i from 1 to nops(F) do
```

```
G:=[op(G), leadterm(F[i], tord)];
```

```
end; #for i
```

```
n:=g;
while 1<>2 do
  Sp:=suppL(n);
  m:=-1;
  for i from 1 to nops(Sp) do
    for j from 1 to nops(G) do
      if divide(Sp[i],G[j], 'q') then
        if m=-1 then
          m:=Sp[i];
          p:=q;
          c:=lcoeff(n,p*G[j]);
          f:=F[j];
          break;
        else
          if testorder(m,Sp[i],tord) then
            m:=Sp[i];
            p:=q;
            c:=lcoeff(n,p*G[j]);
            f:=F[j];
            break;
          end if;
        end if;
      end if;
    end; #for j
  end; #for i

  if m=-1 then
    return n;
  end if;

  n:=simplify(n-(c/leadcoeff(f,tord))*p*f);
  if type(n,'complex') then
    return n;
  end if;
od; #while

end;
```

Beispiel:

```
g:=x1^3*x2+x1*x2:
f1:=x1*x2+x2:
f2:=x1*x2^2+x1*x2+x1^2:
nf(g,{f1,f2},tdeg(x1,x2));
-2*x2
```

A.3 Der Algorithmus BUCHBERGER

```
buchberger:=proc(F,tord) local G,i,j,r,S;

G:=F;
S:={1};

while nops(S)<>0 do
  S:={};
  for i from 1 to nops(G) do
    for j from i+1 to nops(G) do
      r:=reduce(spoly(G[i],G[j],tord),G,tord);
      if r<>0 then
        S:=S union {r};
      end if;
    end; #for j
  end; #for i
  G:=G union S;
od; #while

return G;
end:
```

Beispiel:

```
f1:=x^2*y+x:
f2:=x*y^2+1:
buchberger({f1,f2},plex(x,y));
{-x-1,x*y-x,x^2*y+x,x*y^2+1,y-1,-x^2-x,-y+1,-y^2+1}
```

A.4 Der Algorithmus ISTGRÖBNER

```
IstGroebner:=proc(F,tord) local i,j,r,S;

if nops(F)=0 then return false;end if;

for i from 1 to nops(F) do
  for j from i+1 to nops(F) do
    if reduce(spoly(F[i],F[j],tord),F,tord)<>0 then
      return false;
    end if;
  end; #for j
end; #for i

return true;
end:
```

Beispiel:

```
IstGroebner({x^2*y+x, x*y^2+1, y*x-x, x+x^2, 1+x, 1-y, 1-y^2},
    plex(x,y));
    true
```

A.5 Der Algorithmus REDGB

```
redGB:=proc(F,tord) local H,G,i,g,h,found;

H:=buchberger(F,tord);
G:={};

for i from 1 to nops(H) do
    G:=G union {H[i]/leadcoeff(H[i],tord)};
end; #for i

found:=false;
while 1<>2 do

    found:=false;
    for i from 1 to nops(G) do
        h:=normalf(G[i],G minus {G[i]},tord);
        if h<>G[i] then found:=true;break;end if;
    end; #for i

    if found then
        G:=G minus {G[i]};
        if (h<>0) then
            G:=G union {h/leadcoeff(h,tord)};
        end if;
    else
        break;
    end if;
end; #while

return G;
end;
```

Beispiel:

```
f1:=x1*x2+x1*x3:
f2:=x1*x2+x2*x3+x1:
f3:=x1*x2+x1*x3+x3:
redGB({f1,f2,f3},tdeg(x1,x2,x3,x4));
    {x3, x2^2, x2+x1}
```

A.6 Der Algorithmus IDEALGLEICHHEIT

```
idealggleich:=proc(F1,F2) local vars,G1,G2;

vars:=indets(F) union indets(G);
G1:=gbasis(F1,plex(op(vars)));
G2:=gbasis(F2,plex(op(vars)));

return evalb({op(G1)}={op(G2)});
end;
```

Beispiel:

```
f1:=x1*x2+x1*x3:
f2:=x1*x2+x2*x3+x1:
f3:=x1*x2+x1*x3+x3:
G:=redGB({f1,f2,f3},plex(x1,x2,x3)):
idealggleich({f1,f2,f3},G);
    true
```

A.7 Der Algorithmus IDEALMITGLIED

```
idealmitglied:=proc(g,F,vars) local G;

G:=gbasis(F,plex(op(vars)));
return evalb(reduce(g,G,plex(op(vars)))=0);
end;
```

Beispiel:

```
f1:=x*y+1:
f2:=y^2-1:
g:=x*y^2-x:
idealmitglied(g,[f1,f2],[x,y]);
    true
```

A.8 Der Algorithmus IDEALBASIS

```
idealbasis:=proc(v,F,vars) local G,tord,i,j,r,S,
                                c1,c2,lt1,lt2,l,C,h;

G:=F;
tord:=tdeg(op(vars));
C:=[seq([],i=1..nops(G))];

for i from 1 to nops(G) do
    C[i]:=[seq(0,j=1..nops(G))];
```

```

    C[i][i]:=1;
end; #for i

while nops(S)<>0 do
    S:=[];
    for i from 1 to nops(G) do
        for j from i+1 to nops(G) do
            lt1:=leadterm(G[i],tord);
            lt2:=leadterm(G[j],tord);
            h:=lcm(lt1,lt2);
            c1:=leadcoeff(G[j],tord)*h/lt1;
            c2:=-leadcoeff(G[j],tord)*h/lt2;

            r:=multivardiv(expand(c1*G[i]+c2*G[j]),G,tord,'q');
            if r<>0 then
                h:=evalm(evalm(c1*C[i])&+evalm(c2*C[j]) &+
                    evalm(add(expand(-q[l]*C[l]),l=1..nops(G))));

                for l from 1 to nops(F) do
                    h[l]:=expand(h[l]);
                end; #for l

                S:=[op(S),r];
                C:=[op(C),evalm(h)];
            end if; #r<>0
        end; #for j
    end; #for i
    G:=[op(G),op(S)];
od; #while

r:=multivardiv(v,G,tord,'q');

if r<>0 then
    error "v not in ideal"
end if;

h:=[];
for i from 1 to nops(F) do
    h:=[op(h),add(expand(q[j]*C[j][i]),j=1..nops(G))];
end; #for i

return h;
end:

```

Beispiel:

```

f1:=x^2*y+x:
f2:=x*y^2+1:

```

```
B:=idealbasis(x*y-x,[f1,f2},{x,y});
  B := [y, -x]
expand(B[1]*f1+B[2]*f2);
  x*y-x
```

A.9 Der Algorithmus IDEALSCHNITT

```
idealschnitt:=proc(Fs,vars) local Y,G,i,j,GB,Yseq;

G:={};
for i from 1 to nops(Fs) do #ueber alle Ideale
  for j from 1 to nops(Fs[i]) do #ueber alle Elemente der
                                #i-ten Idealbasis
    G:=G union {Y[i]*Fs[i][j]};
  end; #for j
end; #for i

G:=G union {add(Y[i],i=1..nops(Fs))-1};
Yseq:=seq(Y[i],i=1..nops(Fs));
GB:=gbasis(G,plex(Yseq,op(vars)));
GB:=remove(has,GB,{Yseq});
return {op(GB)};
end;
```

Beispiel:

```
F1:={x^2+y^2,x*y}:
F2:={x^2-y^2}:
idealschnitt({F1,F2});
  {x^2*y-y^3, x^3-y^2*x}
```

A.10 Der Algorithmus IDEALQUOT

```
idealquot:=proc(F,Q,vars) local i,H,M,j,q;

M:=seq({},i=1..nops(Q));
for i from 1 to nops(Q) do
  H:=idealschnitt({F,{Q[i]}},vars);
  for j from 1 to nops(H) do
    divide(H[j],Q[i],'q');
    M[i]:=M[i] union {q};
  end; #for j
end; #for i

return idealschnitt({op(M)},vars);
end;
```

Beispiel:

```
F1:={x^2+y^2,x*y}:
F2:={x^2-y^2}:
idealquot(F1,F2);
      {x, y}
```

A.11 Der Algorithmus KEINELSG

```
keineLsg:=proc(F,vars) local G;

G:=gbasis(F,plex(op(vars)));
return (evalb(G=[1]));
end:
```

Beispiel:

```
f1:=x^2-1:
f2:=y^2-1:
f3:=x*y:
keineLsg({f1,f2,f3},{x,y});
      true
```

A.12 Der Algorithmus RADIKALMITGLIED

```
radikalmitglied:=proc(g,F,vars) local G;

G:=gbasis(F union {g*t-1},plex(op(vars),t));
return (evalb(G=[1]));
end:
```

Beispiel:

```
f1:=x*y^2+2*y^2:
f2:=x^4-2*x^2+1:
g:=y-x^2+1:
radikalmitglied(g,{f1,f2},[x,y]);
      true
```

A.13 Der Algorithmus LSGMENGENGLEICH

```
lsgmengengleich:=proc(F,G,vars) local i;

for i from 1 to nops(F) do
  if not(radikalmitglied(F[i],G,vars)) then
    return false;
  end if;
end for;
```

```
    end if;
end; #for i

for i from 1 to nops(G) do
    if not(radikalmitglied(G[i],F,vars)) then
        return false;
    end if;
end; #for i

return true;
end;
```

Beispiel:

```
F:={x}:
G:={x^2}:
lsgmengengleich(F,G,[x]);
    true
```

A.14 Der Algorithmus LSGMENGESCHRANKE

```
LsgSchrinke:=proc(F,vars) local G,i,schrinke,lt,var,count;

G:=gbasis(F,plex(op(vars)));

if evalb(G=[1]) then
    return 0; #keine Loesung
end if;

if nops(G)<nops(vars) then
    return infinity; #unendlich viele Loesungen
end if;

schrinke:=1;
count:=0;

for i from 1 to nops(G) do
    lt:=leadterm(G[i],plex(op(vars)));
    if nops(indets(lt) intersect {op(vars)})=1 then
        if nops(lt)>1 then #ist nops(lt)=1, zB x1, so wuerde nur
                        #der Faktor 1 dazu
                        #kommen (Exponent von x1)
            schrinke:=schrinke*op(2,lt);
        end if;
        count:=count+1;
    end if;
end if;
end; #for i
```

```

if count<nops(vars) then
  return infinity; #unendlich viele Loesungen
end if;

return schranke;
end:

```

Beispiele:

```

f1:=x1^5-2*x1*x2^2*x3+x2^4*x3^2-37:
f2:=x2^3+2*x2^2*x3-x3^3:
f3:=x3^4-x3^3-x3^2-x3-2:
LsgSchranke({f1,f2,f3},[x1,x2,x3]);
60

```

```

g1:=x+y-1:
g2:=x+y+1:
LsgSchranke({g1},[x,y]);
∞

```

```

LsgSchranke({g1,g2},[x,y]);
0

```

A.15 Der Algorithmus LSG

```

#Voraussetzung: G ist Gröbnerbasis bzgl. der
#lexikographischen Ordnung mit vars[1]>vars[2]>...
Lsg:=proc(G,vars) local i,s,E,v,g,N,L,Gs,loesung,j;

v:=vars[nops(vars)];
E:={};

for i from 1 to nops(G) do
  if (indets(G[i]) intersect {op(vars)})={v} then
    E:=E union {G[i]};
  end if;
end; #for i

if nops(E)=1 then
  g:=E[1];
else
  g:=E[1];
  for i from 2 to nops(E) do
    g:=gcd(g,E[i]);
  end; #for i
end if;

```

```
N:=solve(g=0,op(v));

if N=[] then
  return {};
end if;

if nops(vars)=1 then
  L:={};
  for i from 1 to nops(N) do
    if (not(type(N[i],RootOf))) and (Im(N[i])=0) then
      L:=L union {[op(v),N[i]]};
    end if;
  end; #for i
  return L;
end if;

L:={};

for i from 1 to nops(N) do
  if (Im(N[i])=0) and (not(type(N[i],RootOf))) then
    Gs:=simplify(G,{op(v)=N[i]});
    Gs:={op(Gs)} minus {0};
    loesung:=Lsg(Gs,vars[1..-2]);

    for j from 1 to nops(loesung) do
      L:=L union {loesung[j] union {[op(v),N[i]]}};
    end; #for j
  end if;
end; #for i

return L;
end;
```

Beispiel:

```
f1:=x1^5-2*x1*x2^2*x3+x2^4*x3^2-37:
f2:=x2^3+2*x2^2*x3-x3^3:
f3:=x3^4-x3^3-x3^2-x3-2:
Lsg({f1,f2,f3},[x1,x2,x3]);
  {[x1, 2], [x2, 1], [x3, -1]}
```

Anhang B

Funktionen und Algorithmen in MAPLE zu Teil IV

Anhang B besteht aus 2 Teilen: Zuerst werden Funktionen angeführt, die zum Großteil entweder an Funktionen aus dem Paket `Groebner` angelehnt sind oder andere praktische Funktionalitäten erfüllen. Danach werden die MAPLE-Implementierungen der Algorithmen, die im Teil IV behandelt wurden, angeführt. Um dabei die Algorithmen, die analog zu jenen aus Anhang A sind, von diesen unterscheiden zu können, ist dem Namen das Präfix „M“ (für Modul) vorgestellt.

B.1 Die Funktion `Mbasecoeffindex`

`Mbasecoeffindex` bestimmt den minimalen Index der Basisvektoren $e[i]$, die in f auftreten.

```
Mbasecoeffindex:=proc(f) local i;

i:=1;
while not(has(f,e[i])) do
    i:=i+1;
end;
return i;
end;
```

Beispiel:

```
Mbasecoeffindex(x1^2*e[3]+x2^2*e[2]);
2
```

B.2 Die Funktion `Mtestorder`

Diese Funktion liefert genau dann `true`, wenn t_1 kleiner oder gleich t_2 bzgl. t_{ord} ist.

```
Mtestorder:=proc(t1,t2,tord) local i;
#true <=> t1 "kleiner gleich" t2
```

```

i:=1;
while 1<>2 do
  if has(t1,e[i]) then
    if has(t2,e[i]) then
      return testorder(lcoeff(t1,e[i]),lcoeff(t2,e[i]),tord);
    else
      return false;
    end if;
  else
    if has(t2,e[i]) then
      return true;
    end if;
  end if;
  i:=i+1;
end; #while

return true;
end:

```

Beispiele:

```

Mtestorder(x1*e[3],x2*e[2],plex(x1,x2));
  true
Mtestorder(x1*e[2],x2*e[2],plex(x1,x2));
  false

```

B.3 Die Funktion Msupp

Diese Funktion liefert den Träger von v als Menge. Die verwendete Funktion `suppL` ist in Abschnitt A.2 (Seite 77) definiert.

```

Msupp:=proc(v) local i,S,H,j;
S:={};
for i from 1 to nops(v) do
  if v[i]<>0 then
    H:=suppL(v[i]);
    for j from 1 to nops(H) do
      S:=S union {H[j]*e[i]};
    end; #for j
  end if;
end; #for i

return S;
end:

```

Beispiel:

```

Msupp([x1*x2,x1,x2*x3]);
  {x1*e2,x1*x2*e1,x2*x3*e3}

```

B.4 Die Funktion MsuppL

Diese Funktion liefert den Träger von v als Liste. Die verwendete Funktion `suppL` ist in Abschnitt A.2 (Seite 77) definiert.

```
MsuppL:=proc(v) local i,S,H,j;

S:=[];
for i from 1 to nops(v) do
  if v[i]<>0 then
    H:=suppL(v[i]);
    for j from 1 to nops(H) do
      S:=[op(S),e[i]*H[j]];
    end; #for j
  end if;
end; #for i

return S;
end;
```

Beispiel:

```
MsuppL([x1*x2,x1,x2*x3]);
      [x1*x2*e1,x1*e2,x2*x3*e3]
```

B.5 Die Funktion Mleadterm

Diese Funktion liefert den Leitterm von f bzgl. der Termordnung `tord`.

```
Mleadterm:=proc(f,tord) local i,m,s;

s:=Msupp(f);
m:=s[1];
for i from 2 to nops(s) do
  if Mtestorder(m,s[i],tord) then
    m:=s[i];
  end if;
end; #for i

return m;
end;
```

Beispiel:

```
Mleadterm([x1*x2,x1,x2*x3],plex(x1,x2,x3));
      x1*x2*e1,
```

B.6 Die Funktion `Mleadcoeff`

Diese Funktion liefert den Leitkoeffizienten von f bzgl. der Termordnung `tord`.

```
Mleadcoeff:=proc(f,tord) local lt,i;
lt:=Mleadterm(f,tord);
i:=Mbasecoeffindex(lt);
if type(f[i],numeric) then
    return f[i];
else
    return lcoeff(f[i],lcoeff(lt,e[i]));
end if;
end;
```

Beispiel:

```
Mleadcoeff([4*x1*x2,x1,x2*x3],plex(x1,x2,x3));
4
```

B.7 Die Funktion `Msvector`

Diese Funktion liefert den S-Vektor von v und w bzgl. der Termordnung `tord`.

```
Msvector:=proc(v,w,tord) local i,j,ltv,ltw,
                                coeff_ltv,coeff_ltw,h;

ltv:=Mleadterm(v,tord);
ltw:=Mleadterm(w,tord);
i:=Mbasecoeffindex(ltv);
j:=Mbasecoeffindex(ltw);

if i<>j then
    return [seq(0,i=1..nops(v))];
end if;

coeff_ltv:=lcoeff(ltv,e[i]);
coeff_ltw:=lcoeff(ltw,e[j]);
h:=lcm(coeff_ltv,coeff_ltw);
return
    expand(Mleadcoeff(w,tord)*h/coeff_ltv*v-
        Mleadcoeff(v,tord)*h/coeff_ltw*w);
end;
```

Beispiele:

```
f1:=[x1^2+1,0,1]:
f2:=[x1*x2+x2,0,0]:
f3:=[0,0,x1*x2]:
```

```

Msvector(f1,f2,plex(x1,x2));
      [-x1*x2+x2, 0, x2]
Msvector(f1,f3,plex(x1,x2));
      [0, 0, 0]
Msvector(f2,f3,plex(x1,x2));
      [0, 0, 0]

```

B.8 Die Funktion Msvector2

Diese Funktion liefert den S-Vektor von v und w bzgl. der Termordnung $tord$ und die dabei berechneten Koeffizienten; d.h. `Msvector2(v,w,tord,'cv','cw')` liefert cv und cw so, dass der S-Vektor gleich $cv*v+cw*w$ ist.

```

Msvector2:=proc(v,w,tord,cv,cw)
local i,j,ltv,ltw,coeff_ltv,coeff_ltw,h,_cv,_cw;

ltv:=Mleadterm(v,tord);
ltw:=Mleadterm(w,tord);
i:=Mbasecoeffindex(ltv);
j:=Mbasecoeffindex(ltw);

if i<>j then
  cv:=0;
  cw:=0;
  return [seq(0,i=1..nops(v))];
end if;

coeff_ltv:=lcoeff(ltv,e[i]);
coeff_ltw:=lcoeff(ltw,e[j]);

h:=lcm(coeff_ltv,coeff_ltw);
_cv:=Mleadcoeff(w,tord)*h/coeff_ltv;
_cw:=-Mleadcoeff(v,tord)*h/coeff_ltw;
cv:=_cv;
cw:=_cw;

return expand(_cv*v+_cw*w);
end;

```

Beispiele:

```

f1:=[x1^2+1,0,1]:
f2:=[x1*x2+x2,0,0]:
f3:=[0,0,x1*x2]:
Msvector2(f1,f2,plex(x1,x2),'cv','cw');
      [-x1*x2+x2, 0, x2]
print(cv);

```

```

      x2
print(cw);
      -x1

```

B.9 Der Algorithmus MMULTIVARDIV

```

Mmultivardiv:=proc(g,F,tord,q) local h,i,red,qh,null;

h:=g;
null:=[seq(0,i=1..nops(F[1]))];
q:=array(1..nops(F));

for i from 1 to nops(F) do
  q[i]:=0;
end; #for i

while h<>null do
  red:=false;

  for i from 1 to nops(F) do
    if divide(Mleadterm(h,tord),Mleadterm(F[i],tord),'quot')
    then
      qh:=Mleadcoeff(h,tord)/Mleadcoeff(F[i],tord)*quot;
      h:=expand(h-qh*F[i]);
      q[i]:=q[i]+qh;
      red:=true;
      break;
    end if;
  end; #for

  if not(red) then
    return h;
  end if;
od:

return null;
end:

```

Beispiel:

```

f1:=[x1^2+1,0,1]:
f2:=[x1*x2+x2,0,0]:
f3:=[0,0,x1*x2]:
Mmultivardiv([x1*x2,x2,0],[f1,f2,f3],plex(x1,x2),'q');
      [-x2, x2, 0]
print(q);
      [0, 1, 0]

```

B.10 Der Algorithmus MNORMALF

```

Mnormalf:=proc(g,F,tord) local G,i,j,n,Sp,m,q,c,p,f,m_j,ei;

G:=[];
for i from 1 to nops(F) do
    G:=[op(G),Mleadterm(F[i],tord)];
end; #for i

n:=g;
while 1<>2 do
    Sp:=MsuppL(n);
    m:=-1;

    for i from 1 to nops(Sp) do
        for j from 1 to nops(G) do
            if divide(Sp[i],G[j],'q') then
                if m=-1 then
                    m:=Sp[i];
                    p:=q;
                    m_j:=j;
                    break;
                else
                    if Mtestorder(m,Sp[i],tord) then
                        m:=Sp[i];
                        p:=q;
                        m_j:=j;
                        break;
                    end if;
                end if;
            end if;
        end; #for j
    end; #for i

    if m=-1 then
        return n;
    end if;

    ei:=Mbasecoeffindex(m);
    c:=lcoeff(n[ei],m/e[ei]);
    f:=F[m_j];

    n:=expand(n-(c/Mleadcoeff(f,tord))*p*f);

    if type(n,'complex') then
        return n;
    end if;
end while;
end proc;

```

```
od; #while
```

```
end:
```

Beispiel:

```
Mnormalf([x1^2, x1*x2], [[x1*x2, 0], [x1, x2]], plex(x1, x2));
      [0, 0]
```

B.11 Der Algorithmus MBUCHBERGER

```
Mbuchberger:=proc(F,tord) local G,i,j,r,S,null,q;

null:=[seq(0,i=1..nops(F[1]))];
G:=F;
S:={1};
while nops(S)<>0 do
  S:={};

  for i from 1 to nops(G) do
    for j from i+1 to nops(G) do
      r:=Mmultivardiv(Msvector(G[i],G[j],tord),G,tord,'q');
      if r<>null then
        S:=S union {r};
      end if;
    end; #for j
  end; #for i

  G:=G union S;
od; #while

return G;
end:
```

Beispiel:

```
Mbuchberger({[x1*x2, 0], [x1, x2]}, plex(x1, x2));
      {[x1*x2, 0], [x1, x2], [0, -x2^2]}
```

B.12 Die Funktion MbuchbergerKoeff

Diese Funktion ist eine Erweiterung des Algorithmus MBUCHBERGER. Sie liefert nicht nur eine Gröbnerbasis von F zurück, sondern in cv auch die Koeffizienten der Darstellung der gelieferten Gröbnerbasis bzgl. F .

```
MbuchbergerKoeff:=proc(F,tord,cv) local G,i,j,r,S,sp,c1,c2,
      l,k,_cv,null,ch;
```

```

G:=F;
null:=[seq(0,i=1..nops(F[1]))];
_cv:=[seq([],i=1..nops(G))];

for i from 1 to nops(G) do
  _cv[i]:=[seq(0,j=1..nops(G))];
  _cv[i][i]:=1;
end; #for i

k:=nops(G);
while nops(S)<>0 do
  S:=[];
  k:=0;
  for i from 1 to nops(G) do
    for j from i+1 to nops(G) do
      sp:=Msvector2(G[i],G[j],tord,'c1','c2');
      r:=Mmultivardiv(sp,G,tord,'q');
      if r<>null then
        S:=[op(S),r];
        k:=k+1;

        ch:=[seq(0,j=1..nops(G))];
        ch:=evalm(add(expand(-q[l]*_cv[l]),l=1..nops(G)));
        ch:=evalm(ch+evalm(c1*_cv[i]));
        ch:=evalm(ch+evalm(c2*_cv[j]));
        _cv:=[op(_cv),convert(evalm(ch),list)];
        for l from 1 to nops(F) do
          _cv[nops(G)+k][l]:=expand(_cv[nops(G)+k][l]);
        end; #for l
      end if; #r<>0
    end; #for j
  end; #for i
  G:=[op(G),op(S)];
od; #while

cv:=_cv;
return G;
end:

```

Beispiel:

```

MbuechbergerKoeff({[x1*x2,0],[x1,x2]},plex(x1,x2),'cv');
  {[x1*x2, 0], [x1, x2], [0, -x2^2]}
print(cv);
  [[1, 0], [0, 1], [1, -x2]]

```

B.13 Der Algorithmus MISTGRÖBNER

```
MistGroeber:=proc(F,tord) local i,j,r,S,null,q;

if nops(F)=0 then return false;end if;

null:=[seq(0,i=1..nops(F[1]))];
for i from 1 to nops(F) do
  for j from i+1 to nops(F) do
    if Mmultivardiv(Msvector(F[i],F[j],tord),F,tord,'q')<>null
    then
      return false;
    end if;
  end; #for j
end; #for i

return true;
end;
```

Beispiel:

```
MistGroeber({[x1, x2], [0, x2^2]},plex(x1,x2));
true
```

B.14 Der Algorithmus MREDGB

```
MredGB:=proc(F,tord) local null,H,G,i,g,h,found;

null:=[seq(0,i=1..nops(F[1]))];
H:=Mbuchberger(F,tord);
G:={};

for i from 1 to nops(H) do
  G:=G union {H[i]/Mleadcoeff(H[i],tord)};
end; #for i

found:=false;

while 1<>2 do

  found:=false;
  for i from 1 to nops(G) do
    h:=Mnormalf(G[i],G minus {G[i]},tord);
    if h<>G[i] then found:=true;break;end if;
  end; #for i

  if found then
```

```

    G:=G minus {G[i]};
    if h<>null then
        G:=G union {h/Mleadcoeff(h,tord)};
    end if;
else
    break;
end if;
od; #while

return G;
end:

```

Beispiel:

```

MredGB({[x1*x2, 0], [x1, x2]}, plex(x1, x2));
      { [x1, x2], [0, x2^2] }

```

B.15 Der Algorithmus INHOMOGENELSG

```

MinhomLsg:=proc(F, f0, vars) local G, c, d;

G:=MbuechbergerKoeff(F, plex(op(vars)), 'c');
Mmultivardiv(f0, G, plex(op(vars)), 'd');
return evalm(d*matrix(c));
end:

```

Beispiel:

```

f0:=[2*x1^2+2*x2^2, x1*x2]:
f1:=[x1+x2, x1^2*x2]:
f2:=[x1-x2, x1*x2^2]:
f3:=[0, x1*x2+x1]:
f4:=[0, x1*x2]:
F:=[f1, f2, f3, f4]:
L:=MinhomLsg(F, f0, [x1, x2]);
      L := [2*x1, -2*x2, 2*x2^2-2*x2, -2*x1^2+3]
expand(add(L[k]*F[k], k=1..4));
      [2*x1^2+2*x2^2, x1*x2]

```

B.16 Der Algorithmus KERNGAMMA

```

MKernGamma:=proc(G, tord) local i, j, S, c1, c2, t, u, E;

E:={};
for i from 1 to nops(G) do
    for j from i+1 to nops(G) do
        S:=Msvector2(G[i], G[j], tord, 'c1', 'c2');

```

```

    if (c1<>0) or (c2<>0) then
        Mmultivardiv(S,G,tord,'t');
        u:= [seq(t[k],k=1..nops(G))];
        u[i]:=t[i]-c1;
        u[j]:=t[j]-c2;
        E:=E union {u};
    end if;
end; #for j
end; #for i

return E;
end:

```

Beispiel:

```

g1:=[x1,0]:
g2:=[x2,0]:
g3:=[0,x1]:
G:=[g1,g2,g3]:
E:=MKernGamma(G,tdeg(x1,x2));
    E := {[-x2, x1, 0]}
expand(add(G[k]*E[1][k],k=1..nops(G)));
    [0, 0]

```

B.17 Der Algorithmus KERNPHI

```

MKernPhi:=proc(F,vars) local G,c,i,h,d,Egamma,Ephi,cMtx,
                        zMtx,yMtx;

G:=MbuchbergerKoeff(F,plex(op(vars)),'c');

d:= [seq([],i=1..nops(F))];
for i from 1 to nops(F) do
    Mmultivardiv(F[i],G,plex(op(vars)),'h');
    h:=convert(h,list);
    d[i]:=evalm(h);
end; #for i

cMtx:=matrix(c);
zMtx:=evalm(matrix(d)*cMtx-Matrix(1..nops(F),1..nops(F),
Vector([seq(1,i=1..nops(F))]),shape=diagonal));
Ephi:={op(convert(zMtx,listlist))};

Egamma:=MKernGamma(G,plex(op(vars)));
if nops(Egamma)>0 then
    yMtx:=evalm(matrix([op(Egamma)])*cMtx);
    Ephi:=[op(expand(Ephi union {op(convert(yMtx,listlist))})

```

```

        minus {[seq(0,i=1..nops(F))]}]);
end if;

#Zusaetzlich: Leitkoeffizient jedes Elementes von Ephi > 0;
#eliminiert Elemente, die sich nur durch den Faktor (-1)
#unterscheiden

for i from 1 to nops(Ephi) do
    h:=Mleadcoeff(Ephi[i],plex(op(vars)));
    if h<0 then
        Ephi[i]:=(-1)*Ephi[i];
    end if;
end; #for i

return {op(Ephi)};
end:

```

Beispiel:

```

f1:=[x1+x2,x1^2*x2]:
f2:=[x1-x2,x1*x2^2]:
f3:=[0,x1*x2+x1]:
f4:=[0,x1*x2]:
F:=[f1,f2,f3,f4]:
E:=MKernPhi(F,[x1,x2]);
    E := {[x1-x2, -x1-x2, 2*x1*x2+x2^2-x2, -x1^2-2*x1+1],
          [0, 0, x2, -1-x2]}
expand(add(E[1][k]*F[k],k=1..nops(F)));
    [0, 0]
expand(add(E[2][k]*F[k],k=1..nops(F)));
    [0, 0]

```

Symbolverzeichnis

$\mathbb{N}_{>m}$	$\mathbb{N}_{>m} = \{m + 1, m + 2, m + 3, \dots\} \subseteq \mathbb{N}$	5
R	Kommutativer Ring mit Einselement	5
$R[x]$	Polynomring in x über R	5
$\trianglelefteq$	Ideal von	6
$\langle a \rangle_R, \langle a \rangle$	Das von $a \in R$ erzeugte Ideal	7
$\langle a_1, \dots, a_n \rangle_R,$ $\langle a_1, \dots, a_n \rangle$	Das von $a_1, \dots, a_n \in R$ erzeugte Ideal	7
$\langle A \rangle_R, \langle A \rangle$	Das von $A \subseteq R$ erzeugte Ideal	7
$I_1 + I_2$	Summe der Ideale I_1 und I_2	7
$I_1 \cdot I_2$	Produkt der Ideale I_1 und I_2	7
$I_1 : I_2$	Quotient der Ideale I_1 und I_2	7
P	Menge der Potenzprodukte (Terme)	11
K	Körper	11
$K[x]$	Polynomring über K in $x_1, \dots, x_n$	11
$ i $	Abkürzung für $i_1 + \dots + i_n$	11
$\prec$	Termordnung	12
supp	Träger	12
lt	Leitterm	12
lk	Leitkoeffizient	12
$r_{\prec}$	Divisionsrest	13
normalf	Normalform	15
kgV	Kleinstes gemeinsames Vielfaches	20
ggT	Größter gemeinsamer Teiler	20
$S(f, g)$	S-Polynom	20
Rad	Radikal	35
$\mathcal{N}$	Menge der gemeinsamen Nullstellen	44
$\mathcal{I}$	Ideal	46
$S(v, w)$	S-Vektor	61

Literaturverzeichnis

- [1] Thomas Becker und Volker Weispfenning. *Gröbner Bases, A Computational Approach to Commutative Algebra*, Springer (1993).
- [2] Ralf Fröberg. *An Introduction to Gröbner Bases*, John Wiley & Sons (1997).
- [3] Franz Pauer und Marlene Pfeifhofer. *The Theory of Gröbner Bases*, L'Enseignement Mathématique (1988), Seite 215-232.
- [4] Franz Pauer. *Gröbner Basen und ihre Anwendungen (Gröbner Bases and applications)*, Jahresbericht Überblick Mathematik (1991), Seite 127-149.
- [5] David Cox, John Little und Donal O'Shea. *Ideals, Varieties and Algorithms*, Springer (1997).
- [6] Tobias Hofmann. *Berechnung von Gröbnerbasen und eine Implementierung des Buchbergeralgorithmus mit Mathematica*, Universität Kassel, Diplomarbeit (2003).
- [7] Günther Eigenthaler. *Algebra, Begleitmaterial zur Vorlesung* (2000).
- [8] Dietmar Dorninger, Christoph Fabianek und Andreas Traxler. *Algebraische Methoden in den Computerwissenschaften*, Skriptum zur Vorlesung (2002).
- [9] Michael Artin. *Algebra*, Birkhäuser (1991).
- [10] Gerd Baron und Peter Kirschenhofer. *Einführung in die Mathematik für Informatiker, Band 1*, Springer (1989).
- [11] Michael Kofler, Gerhard Bitsch, Michael Komma. *Maple - Einführung, Anwendung, Referenz*, Addison-Wesley Longman Verlag (2002).

Index

- Algebraische Abgeschlossenheit, 44
- Assoziiertes Primideal, 71
- BUCHBERGER-Algorithmus, 20, 22
- Division mit Rest, 13, 60
- Durchschnitt von Idealen, 38
- Eliminationsideal, 52
- Endliche Lösungsmenge, 48
- Endliches Erzeugendensystem, 7
- Erstes BUCHBERGER Kriterium, 25
- Führender Koeffizient, 6
- Gröbnerbasis, 17, 61
- Größter gemeinsamer Teiler, 20
- HILBERTscher Basissatz, 9
- HILBERTscher Nullstellensatz
 - Schwacher, 44
 - Starker, 46
- Ideal, 6, 46
 - endlich erzeugtes, 7
 - irreduzibles, 71
 - primäres, 71
 - primes, 71
 - reduzibles, 71
- Idealgleichheitstest, 29
- Idealmitgliedschaftstest, 30
- Kleinstes gemeinsames Vielfaches, 20
- Komponente
 - eingebettete, 74
 - isolierte, 74
 - primäre, 72
- Leitkoeffizient, 12
- Leitterm, 12
- Menge der gemeinsamen Nullstellen, 44
- Modul, 10
 - freier, 10
 - Noetherscher, 57
- Noetherscher Ring, 8
- Normalform, 15
- Polynom, 5
 - grad, 6
 - ring, 5, 6
- Potenzprodukte, 11
- Primärzerlegung, 72
- Produkt von Idealen, 7
- Quotient von Idealen, 7, 38
- Radikal, 35
- Radikalmitgliedschaftstest, 35
- Reduzierte Gröbnerbasis, 18, 23
- S-Polynom, 20
- S-Vektor, 61
- Summe von Idealen, 7
- Terme, 11
 - disjunkte, 25
- Termordnung, 11, 59
 - graduiert-invers-lexikographische, 12
 - graduiert-lexikographische, 12, 59
 - lexikographische, 12, 59
- Träger, 12
- Untermodul, 10