Martin Dlugosch
Universität Bremen
Seminararbeit im Rahmen der WE
ALZAGK :
Einführung in Gröbnerbasen
Betreuer: Michael Hortmann
Bremen, den 05.08.08

1 Vorwort

Mein Vortrag gibt zum einen eine ausführliche Einführung zum Thema Gröbnerbasen. Anschließend werden zwei verschiedene Anwendungen dieser Theorie angerissen. Grundlage dafür war Chapter 1 des Buches „Some Tapas of Computer Algebra“ von Cohen, Arjeh M.; Cuypers, Hans; Sterk, Hans (Eds.), erschienen im Springer-Verlag, New York 1999. Darauf möchte ich auch bei allen nicht bewiesenen Sätzen verweisen.nnn

2 Gröbnerbasen

Sei R ab jetzt immer ein kommutativer Ring mit Einselement.

Gröbnerbasen sind Erzeuger von Polynomidealen, die besondere Eigenschaften haben. Dank dieser eignen sie sich für manche Berechnungen in diesen Polynomringen besonders gut.

Um die Algorithmen die hinter den Gröbnerbasen stecken zu verstehen, müssen zunächst einmal Monomialordnungen definiert werden.

Definition
Eine totale Ordnung auf der Menge M der Monome eines Polynomrings $R[x_1, ..., x_n]$ heißt **Monomialordnung** genau dann wenn,

- Jede nichtleere Teilmenge von M besitzt ein minimales Element.
- $\forall\ m, n, k \in M$ gilt $\ m \prec n \Rightarrow m \cdot k \prec n \cdot k$

Ein Beispiel für eine solche Monomialordnung ist die so genannte *Lexikographische Ordnung*. Diese ist wie folgt definiert :

$$x_1^{a_1} x_2^{a_2} ... x_n^{a_n} \prec_{lex} x_1^{b_1} x_2^{b_2} ... x_n^{b_n} \Leftrightarrow \exists\ k \text{ mit } a_i = b_i\ \forall\ i < k \ \text{ und } \ a_k < b_k$$

Ein weiteres verbreitetes Beispiel ist die sogenannte *gradierte Lexikographische Ordnung*, welche folgendermaßen definiert ist:

$$x_1^{a_1} x_2^{a_2} ... x_n^{a_n} \prec_{grlex} x_1^{b_1} x_2^{b_2} ... x_n^{b_n} \Leftrightarrow \sum_{i=1}^{n} a_i < \sum_{i=1}^{n} b_i \ \text{ oder}$$

$$\sum_{i=1}^{n} a_i = \sum_{i=1}^{n} b_i \ \text{ und } \ x_1^{a_1} x_2^{a_2} ... x_n^{a_n} \prec_{lex} x_1^{b_1} x_2^{b_2} ... x_n^{b_n} \ \text{ bzgl. der lexikographischen Ordnung}$$

Def.
Sei $f \in R[x_1, ..., x_n]$ und M_f sei die Menge der Monome in f.
Dann heißt

$lm(f) := \max\{m \in M_f\}$ $bzgl. \prec$	das **Leitmonom** von f
$lc(f) := c_{lm(f)}$	der **Leitkoeffizient** von f
$lt(f) := lc(f) \cdot lt(f)$	der **Leitterm** von f.

Ab sofort behandeln wir nur noch den Fall eines Polynomrings über einem Körper K. Dieser Spezialfall ist analog zum allgemeinen Fall zu behandeln, erleichert allerdings die Dinge insofern, als dass man nicht mehr zwischen $lt(f)|lt(g)$ und $lm(f)|lm(g)$ unterscheiden muss, da man die Koeffizienten anpassen kann.

Da wir nun eine Ordnung auf den Monomen der Polynome erstellt haben, können wir uns dem ersten Algorithmus, der uns den Gröbnerbasen näher bringt, zuwenden.
Diesen nennen wir den *Reduce*-Algorithmus:

Sei B eine endliche Menge von Polynomen und f ebenfalls ein Polynom aus $K[x_1, ..., x_n]$.

Reduce(B,f)=
$J := \{b \in B \mid lt(b)|lt(f)\}$;
`if` $J \neq \emptyset$
`then choose` $b \in J$;
`return` Reduce(B,$f - (lt(f)/lt(b))b$)
`else` return f
`end`

Am Ende dieses Algorithmusses steht dann eine **reduzierte Form** des Polynoms f, dessen Leitterm von keinem Leitterm eines Elements aus B geteilt wird. Zu beachten ist dabei, dass vorher eine Monomordnung zu wählen ist, da diese entscheiden was das Leitmonom eines Polynoms ist. Daher ist bei unterschiedlichen Monomordnungen auch mit unterschiedlichen Ergebnissen zu rechnen.

Bemerkung
Reduce(B, f)=0 bedeutet, dass f zum Nullpolynom reduziert worden ist. Dies impliziert, dass

$$f = \sum_{b \in B} g_b \cdot b \qquad \text{und} \qquad \forall\, b \in B: \quad lm(g_b \cdot b) \leqslant lm(f)$$

Das wiederrum bedeutet, dass wir f als $R-$Linearkombination von Elementen aus B darstellen können, demnach ist f im von B erzeugten Ideal.

Ist die reduzierte Form eines Polynoms f also das Nullpolynom, so ist dieses im von B erzeugten Ideal. Die Umkehrung dieser Aussage gilt allerdings nicht

immer, was folgendes Beispiel demonstriert

Beispiel
Sei $f = x - y \in \mathbb{Q}[x, y]$, die Monomordnung sei $\prec_{lex}$ und$B = \{x^2y - 1, xy^2 - 1\}$. Offensichtlich ist f bereits in reduzierter Form, da die Leitterme von Elementen aus B den Leitterm x von f nicht teilen. Allerdings kann f dargestellt werden als:

$$f = y \cdot (x^2y - 1) + x \cdot (xy^2 - 1) = x - y$$

Also ist f doch im von B erzuegten Ideal. Es mussten allerdings höhere Monome aufgebaut werden, was der Algorithmus nicht abdecken konnte.

Definition
Das **S-Polynom** von f und g wird wie folgt definiert:

$$S(f, g) = \frac{kgV(lm(f), lm(g))}{lt(f)} \cdot f - \frac{kgV(lm(f), lm(g))}{lt(g)} \cdot g$$

Es ist so konstruiert, dass sich die Leitterme gegenseitig aufheben und so ist das **S-Polynom** von f ung g die einfachste Linearkombination von $\{f, g\}$ von der nicht klar ist, ob ihr Leitterm durch $\{f, g\}$ noch reduziert werden kann.

Beispiel
Seien wieder $f = x^2y - 1\ ,\quad g = xy^2 - 1 \quad \in \mathbb{Q}[x, y]$.

$$\begin{aligned} S(f, g) &= \frac{kgV(x^2y, xy^2)}{x^2y} \cdot (x^2y - 1) - \frac{kgV(x^2y, xy^2)}{xy^2} \cdot (xy^2 - 1) \\ &= y \cdot (x^2y - 1) - x \cdot (xy^2 - 1) \\ &= x - y \end{aligned}$$

Kommen wir nun zum zweiten Algorithmus. M_f bezeichne die Menge der Monome in f und c_m den Koeffizienten des Monoms m in f.

StronglyReduce(B,f)=
$J := \{b \in B \ \mid \ lt(b)|m \ \text{ für ein } \ m \in M_f\}$;
`if` $J \neq \emptyset$
`then choose` $b \in J$;
`return` StronglyReduce(B,$f - (c_m \cdot m/lt(b))b$)
`else` return f
`end`

Das Ergebnis dieses Algorithmusses nennen wir eine **stark reduzierte Form** von f. Es ist ein Polynom dessen Monome allesamt keine vielfachen von Leitmonomen der Elemente von B sind.

Definition Sei I ein Ideal im Polynomring $K[x_1, ..., x_n]$. Eine endliche Teilmenge B dieses Ideals heißt Gröbnerbasis, wenn für alle Polynome f im Ideal gilt $Reduce(B, f) = 0$.

Satz
Folgende Aussagen sind Äquivalent:

1. B ist eine Gröbnerbasis von I
2. $\forall\ b, c \in B : Reduce(B, S(b, c)) = 0$
3. $< lt(f) | f \in I > = < lt(b) | b \in B >$

Insbesondere letztere Bedingung erweist sich als äußerst nützich. Deshalb folgt nun der Beweis der Äquivalenz zwischen erster und dritter Aussage.

Beweis
1. $\Rightarrow$ 3.
Vor. : B ist eine Gröbnerbasis von I
zz. : $< lt(f) | f \in I > = < lt(b) | b \in B >$
" $\supseteq$ " trivial
" $\subseteq$ " Es reicht zu zeigen, dass jeder Erzeuger $lt(f)$ von $< lt(f) | f \in I >$ auch in $< lt(b) | b \in B >$ liegt.
Da B eine Gröbnerbasis von I ist , gilt für jedes f aus I $Reduce(B, f) = 0$. Dies impliziert wiederrum $f = \sum_{b \in B} g_b \cdot b$ und $lm(g_b \cdot b) \preceq lm(f)\ \ \forall\ b \in B$. Dann muss es einen Summanden der Form $g_b \cdot b$ geben mit $lt(f) = lt(g_b \cdot b) = lt(g_b) \cdot lt(b)$. Also ist $lt(f)$ in $< lt(b) | b \in B >$.

3. $\Leftarrow$ 1.
Vor. : $< lt(f) | f \in I > = < lt(b) | b \in B >$
zz.: B ist eine Gröbnerbasis von I.
Angenommen es gibt ein $f \in I$, dass nicht zwangsläufig zu Null reduziert wird. Ohne Beschränkung der Allgemeinheit kann der Einfachheit halber angenommen werden, dass f bereits reduzierte Form hat.
Weil der Leitterm von f nach Vorrausetzung in $< lt(b) | b \in B >$ liegt, muss es mindestens ein $b \in B$ und ein $g \in K[x_1, ..., x_n]$ geben, so dass $lt(f) = lt(g \cdot b)$. Dann muss das Leitmonom von $f - g \cdot b$ bzgl. der gewählten Monomialordnung echt kleiner sein, da sich die beiden Leitterme von f und $g \cdot b$ gegenseitig aufheben. Dies bedeutet aber, dass f doch weiter reduziert werden konnte. Somit haben wir einen Widerspruch zu der Annahme f sei nicht das Nullpolynom gewesen.

Mit eben jener Beschreibung einer Gröbnerbasis durch die so genannten Leittermideale $< lt(b)|b \in B >$ lässt sich einiges anfangen. Unter Anderem wird uns dadurch erlaubt einzelne Elemente der Gröbnerbasis fallen zu lassen, wenn ihre Leitterme keine notwendigen Erzeuger für das Leittermideal darstellen.

Der Buchberger Algorithmus

Der folgende Algorithmus wird Buchberger Algorithmus genannt. Er rechnet eine beliebige Basis eines Ideals unseres Polynomrings in eine Gröbnerbasis dieses Ideals um.

GröbnerBasis(B)=
$P :=$ {ungeordneten Paare von Elementen aus B}
`while` $P \neq \emptyset$ do
`choose` $\{f, g\} \in P$;
$P := P \backslash \{f, g\}$;
$c := Reduce(B, S(f, g))$;
`if` $c \neq 0$
`then` $B := B \cup \{c\}$;
$P := P \cup \{(b, c)|b \in B\}$;
`fi`;
`od; return` B.

Definition
Eine Gröbnerbasis B heißt **reduzierte Gröbnerbasis**, wenn

$$\forall\ b \in B :\ b = StronglyReduce(\ B\backslash\{b\}\ ,\ b)\ \text{ und }\ lc(b) = 1$$

Satz
Jedes Polynomideal I aus $K[x_1, ..., x_n]$ hat eine eindeutig bestimmte reduzierte Gröbnerbasis.

Da der Aufwand des Berechnens dieser reduzierten Gröbnerbasis im allgemeinen bereits bei relativ harmlos aussehenden Idealen mit nur 2 Erzeugern schon sehr hoch ist, wird in der Praxis immer der Computer eingesetzt um diese auszurechnen.
Ein Programm, das sich dafür eignet ist **Cocoa**, was für Computations in Commutative Algebra steht. Zu finden ist dieses Gratis-Programm unter `http://cocoa.dima.unige.it` . In dem Programm sind eben jene Algorithmen, die zu unseren Gröbnerbasen geführt haben, implementiert.

Der Befehl zum Errechnen einer Gröbnerbasis lautet folgendermaßen:

```
Use R ::= Q[x,y], Lex;
G := GBasis(Ideal(x^2y-1,xy^2-1));
```

Lässt man sich dann mittels "G;"die Gröbnerbasis ausgeben, so erscheint folgende Ausgabe:

```
[-x + y, y^3-1]
```

3 Standardmonome

Definition
Sei B eine Gröbnerbasis. Ein Monom m heißt **Standardmonom** wenn

$$lm(b) \nmid m \quad \forall\, b \in B$$

Sei nun $U := span(\ \{\ \ m \in M\ \ |\ \ m\ \ \text{ist ein Standardmonom}\}\ )$, also der K-Linear erzeugte Untervektorraum des Polynomrings $K[x_1, ..., x_n]$ aufgefasst als Vektorraum.

Satz
Folgende Abbildung ist dann ein Vektorraumisomorphismus:

$$\begin{aligned} \Phi\ :\ K[x_1, ..., x_n] / < B > &\longrightarrow U \\ f + < B > &\longrightarrow StronglyReduce(B, f) \end{aligned}$$

Folgerung
Sei B eine Gröbnerbasis eines Polynomideals aus $K[x_1, ..., x_n]$ und U das linear Erzeugte der Standardmonome des Polynomrings bzgl. B. Dann gilt:

$$K[x_1, ..., x_n] \ \cong\ U \oplus < B >$$

Demnach gibt es für jede Nebenklasse $f + < B >$ in $K[x_1, ..., x_n] / < B >$ genau einen Repräsentanten in U. Dieser wird **Normalform** genannt.

Auf diese Weise haben wir nun mit U ein Modell für unseren Restklassenring $K[x_1, ..., x_n] / < B >$. Wir können nun ausgewählte Operationen über den Isomorphismus in U durchführen.

So lässt sich z.B. Gleichheit zweier Nebenklassen bei unübersichtlich großen Repräsentanten dadurch testen lassen, dass wir ihre Bilder unter Φ vergleichen, denn

$$f+ < B >= g+ < B > \Leftrightarrow \quad StronglyReduce(B, f) = StronglyReduce(B, g)$$

Ein weiteres Beispiel ist die Multiplikation.

$$(f+ < B >) \cdot (g+ < B >) = StronglyReduce(B, f \cdot g) + < B >$$

Beispiel

Sei in diesem Beispiel $\mathbb{R}[x, y]$ unser Polynomring. Das betrachtete Ideal sei $I = \langle x^2 \cdot y - 1, x \cdot y^2 - 1 \rangle$. Eine Gröbnerbasis dieses Ideals ist $B = \{x - y, y^3 - 1\}$. Nun wollen wir die Nebenklasse von $y^2 + \langle B \rangle \cdot y^2 + \langle B \rangle$ in ihrer eindeutigen Darstellung mit Repräsentant aus U errechnen.

$$(y^2+ < B >) \cdot (y^2+ < B >) = StronglyReduce(B, y^2 \cdot y^2)+ < B >$$
$$= y^4 - \frac{y^4}{y^3} \cdot (y^3 - 1) + < B >= y+ < B >$$

4 Effektivität von Ringen

Definition

Ein Ring R heißt *effektiv* wenn,

- Elemente der Rings sind auf dem Computer beschreibbar
- Gleichheit zweier Elemente sind durch Algorithmen überprüfbar
- Die Ringoperationen Addition und Multiplikation sind durch Algorithmen durchführbar
- Die Lineare Gleichung $\sum_{i=1}^{n} a_i \cdot x_i = b$ mit $a_i, b, x_i \in R$ muss vollständdig lösbar sein

Wobei Letzteres bedeutet, dass im Falle der Lösbarkeit alle Lösungen, d.h. spezielle Lösung und Erzeuger des Kerns der homogenen Gleichung algorithmisch berechnet werden können.

Ein Körper K heißt effektiv, wenn K ein effektiver Ring ist und man algorithmisch die multiplikativ inversen Elemente finden kann.

Als Beispiele für effektive Ringe sind z.B. endliche Ringe oder $\mathbb{Z}$ zu nennen. Ein effektiver Körper ist z.B. $\mathbb{Q}$.

Zu Klären ist die Frage, ob sich die Eigenschaft eines Körpers K effektiv zu sein auf den Poylnomring $K[x]$ und damit über Induktion auf den Poylnomring $K[x_1, .., x_n]$ vererbt.
Als einzig interessanter Punkt erweist sich das Lösen der Linearen Gleichungen.

Beispiel

$R[X] = \mathbb{Q}[x_1, y_2]$ und $(x_1^2 x_2 - 1) \cdot y_1 + (x_1 x_2^2 - 1) \cdot y_2 = x_1 - x_2$
Wie wir schon im Vorfeld an diesem Beispiel gesehen haben lässt sich dieses Problem mittels Gröbnerbasen angehen.

Wie wir nun bereits wissen sind folgende Aussagen äquivalent :

1. $\exists\, y = (y_1, ..., y_l) \in (K[x_1, ..., x_n])^l$ mit $\sum_{i=1}^{l} a_i \cdot y_i = b$, $a_i, b \in K[x_1, ..., x_n]$

2. $b \in < a_1, ..., a_l >$

3. Reduce(Gröbnerbasis($\{a_1, ..., a_l\}$) , $b) = 0$

Dank letzterer Aussage ist die Lösbarkeit einer solchen Linearen Gleichung algorithmisch überprüfbar. Damit der Polynomring. Es stellt sich nun die Frage ob man auch durch Algorithmen auf sämtliche Lösungen der Gleichung kommen kann.
Wie bereits erwähnt will man den Lösungsraum durch eine Spezielle Lösung und Erzeuger der Lösung der Homogenen Gleichung angeben.

Sei $A = \{a_1, ..., a_l\}$ ein Erzeugendensystem eines Polynomideals und b aus dem betrachteten Polynomring.

Die nötigen Algorithmen zum Berechnen aller Lösungen kann man wie folgt zusammenfassen:
Zunächst findet man einen speziellen Koeffizientenvektor, der eine spezielle Lösung der gegebenen Linearen Gleichung darstellt, indem man zunächst eine spezielle Lösung für eine Gröbnerbasis $B = \{b_1, ..., b_k\}$ errechnet. Dies ist relativ simpel, da man nur eine Abwandlung des Reduce Algorithmus braucht, die sich merkt um wieviel reduziert wurde. Dies aufsummiert ergibt dann den gesuchten Koeffizientenvektor. Nun nutzt man einen Moduln-Homomorphismus (d.h. eine $l \times k$ Matrix G mit $G \cdot (a_1, ..., a_k) = (b_1, ..., b_l)$) um den gefunden Koeffizientenvektor in einen Koeffizientenvektor des ursprünglichen Erzeugendensystems A umzurechnen.

Im zweiten Schritt wird nach den Erzeugern des Lösungsraums der homogenen Gleichung $\sum_{i=1}^{l} a_i \cdot x_i = 0$ gesucht. Diese sind also der Kern der folgender $K[x_1, ..., x_n]$-linearen Abbildung :

$$\Phi_A \ : \ (K[x_1, ..., x_n])^l \ \rightarrow \ K[x_1, ..., x_n]$$
$$x \ \rightarrow \ A \cdot x$$

Auch hier errechnet man erst die Erzeuger des Kerns von Φ_B wobei B eine Gröbnerbasis desselben Ideals entspricht. Dabei wird der Kern von folgenden Elementen erzeugt:

$$\frac{kgV(lm(a_i), lm(a_j))}{lt(a_i)} \cdot e_i - \frac{kgV(lm(a_i), lm(a_j))}{lt(a_j)} \cdot e_j \ - \ h^{ij} \qquad , \ 1 \leqslant i < j \leqslant k$$

wobei $(h_1^{ij}, ..., h_k^{ij}) = h^{ij} \in (K[x_1, ..., x_n])^k$ der Vektor ist, für den gilt:

$\sum_{m=1}^{k} \cdot h_m^{ij} \ = \ S(a_i, A_j)$ und zusätzlich $\forall m \ : \ lm(a_m \cdot h_k^{ij}) \ < \ kgV(a_i, a_j)$ bzgl. der gewählten Monomialordnung.

Bemerkung
Die Existenz dieses Vektors ist bereits im Vorfeld gesichert worden, da das S-Polynom von a_i und a_j im von A erzeugten Ideal liegt und dadurch unter der Gröbnerbasis A zu 0 reduziert wird. Dann greift Bemerkung (Seite 3 oben) und der zugehörige Koeffizientenvektor ist das h_{ij}

Ist H eine Matrix deren Zeilen $kern(\Phi_B)$ erzeugen, dann gilt :

$$kern \ (\Phi_A) \ \text{ wird von den Zeilen der beiden Matrizen}$$
$$G \cdot F - E \ \text{ und } \ H \cdot F \ \text{ erzeugt}$$

Damit lässt sich also eine beliebige lineare Gleichung $\sum_{m=1}^{l} a_m \cdot x_m = b$ vollständig lösen. Somit wissen wir nun, dass Polynomring eines effektiven Körpers $K[x_1, ..., x_n]$ ebenfalls effektiv ist.

Satz
Ist K ein effektiver Körper, dann ist jeder Quotientenring von $K[x_1, ..., x_n]$ effektiv.
Der hilbertsche Basissatz besagt, dass jedes Ideal J eines Polynomrings über einem Körper endlich erzeugt ist. Demnach lässt sich für eine gewählte Monomialordnung eine Gröbnerbasis $C = \{c_1, ..., c_k\}$ dieses Ideals errechnen. Geht man die Bedingungen für Effektivität durch, so stellt sich das Lösen der linearen Gleichung wieder als einzig interessantes Problem heraus.

Die lineare Gleichung hat in unserem Fall die Form:

$$\sum_{i=1}^{l}(a_i + J)(x_i + J) = b + J \qquad \text{wobei} \quad a_i, b \in K[x_1, ..., x_n]$$

Ohne Beschränkung der Allgemeinheit kann man davon ausgehen, dass die Monome der Polynome $a_1, ..., a_l, b$ unter C bzgl. der gewählten Monomialordnung unter C stark reduziert (Strongly Reduce Algorithmus) sind, also Standardmonome sind.
Daher können wir das Problem so darstellen, dass man folgende größere lineare Gleichung lösen will:

$$\sum_{i=1}^{l} a_i \cdot x_i + \sum_{j=1}^{k} c_j \cdot y_j = b$$

Da $K[x_1, ..., x_n]$ ein effektiver Ring ist, lässt sich diese große lineare Gleichung vollständig lösen.
Sowohl von der speziellen Lösung als auch von den Erzeugern des Kerns der Homogenen Gleichung betrachtet man die jeweils die ersten l Komponenten. Diese erfüllen dann die gesuchten Eigenschaften der vollständigen Lösung der ursprünglichen Gleichung.