

Folien zum Vortrag
*Ordnen und Reduzieren von Termen in
mehreren Variablen*

aus der Vortragsreihe
Gröbner Basen für polynomiale Ideale
im Rahmen des Seminars
Symbolische Algorithmen
von Prof. Rauber und Dr. Schenzel
im Wintersemester 1998/99 an der MLU Halle/Wittenberg

Patrick Reichert
November 1998

Ordnen und Reduzieren von Termen in mehreren Variablen

Definition eines Ideals:

Ist R ein kommutativer Ring und $I \subseteq R$ eine Teilmenge von R , so heißt I ein Ideal, wenn

1. $p, q \in I \Rightarrow p - q \in I$
2. $p \in I, r \in R \Rightarrow rp \in I$

Jede endliche Menge von Polynomen $P = \{p_1, \dots, p_k\} \subset F[x_1, \dots, x_n]$ erzeugt ein Ideal:

$$\langle P \rangle = \langle p_1, \dots, p_k \rangle = \left\{ \sum_{i=1}^k a_i p_i : a_i \in F[x_1, \dots, x_n] \right\}$$

P heißt **Basis** des Ideals.

Bezeichnungen:

$F[x]$ Polynomring
 x (geordnete) Menge von Variablen $x = (x_1, \dots, x_n)$
 T_x Menge der Terme in $x : T_x = \{x_1^{i_1} \cdots x_n^{i_n} : i_1, \dots, i_n \in \mathbb{N}\}$

Ordnen von Termen

Für jede zulässige **totale** Ordnung $<_T$ in T_x muß gelten:

1. $1 \leq_T t$
2. $s <_T t \Rightarrow s \cdot u <_T t \cdot u$

für alle $s, t, u \in T_x$, wobei $1 = x_1^0 \cdots x_n^0$ gilt.

Die **lexikographische** Ordnung $<_L$ ist definiert durch:

$$x_1^{i_1} \cdots x_n^{i_n} <_L x_1^{j_1} \cdots x_n^{j_n} \Leftrightarrow \exists k : i_k < j_k, i_m = j_m \text{ für } 1 \leq m < k$$

Die **graduelle** Ordnung (*degree term ordering*) ist definiert durch:

$$s = x_1^{i_1} \cdots x_n^{i_n} <_D x_1^{j_1} \cdots x_n^{j_n} = t \Leftrightarrow \deg(s) < \deg(t) \text{ oder } \deg(s) = \deg(t) \text{ und } \exists k : i_k > j_k, i_m = j_m \text{ für } k < m \leq n$$

Das **führende Monom** (*leading monomial*) von $p \in F[x]$ bezüglich einer Ordnung $<_T$ ist der maximale Term in p .

Schreibweisen:

$M_T(p) = M(p)$: führendes Monom
 $\text{hterm}(p)$: der maximale Term (*head term*) in p
 $\text{hcoeff}(p)$: der zugehörige Koeffizient
 $\Rightarrow M(p) = \text{hcoeff}(p) \cdot \text{hterm}(p)$

Konvention: $\text{hcoeff}(0) = 0$, $\text{hterm}(0) = 1$

Reduktion von Termen in mehreren Variablen

Für von Null verschiedene Polynome $p, q \in F[x]$ sagt man, p läßt sich **modulo q reduzieren** (bezüglich einer festen Termordnung), wenn es in p ein **Monom** gibt, das durch **hterm(q)** teilbar ist.

$$p = \lambda t + r \text{ mit } \lambda \in F - \{0\}, t \in T_x, r \in F[x].$$

$$\frac{t}{\text{hterm}(q)} = u \in T_x$$

$$\text{Schreibweise: } p \mapsto_q p - \frac{\lambda t}{\text{M}(q)} q = p'$$

Ein Polynom p läßt sich modulo einer Menge von Polynomen $Q = \{q_1, \dots, q_m\}$ reduzieren, wenn sich p modulo q_i für (mindestens) ein i reduzieren läßt.

Schreibweise: $p \mapsto_Q p'$

Andernfalls heißt p **irreduzibel** oder **reduziert** modulo Q .

Konvention: 0 ist stets irreduzibel

Satz 1: Für eine feste Menge Q und Ordnung $<_T$ gibt es **keine** unendliche Sequenz von Reduktionen $p_0 \mapsto_Q p_1 \mapsto_Q p_2 \mapsto_Q \dots$

Weitere Bezeichnungen:

\mapsto_Q^+ bezeichnet die reflexive, transitive Hülle von \mapsto_Q

D.h. $p \mapsto_Q^+ q \Leftrightarrow p \mapsto_Q p_1 \mapsto_Q \dots \mapsto_Q p_m = q$

Wenn $p \mapsto_Q^+ q$ und q **irreduzibel** ist, schreibt man: $p \mapsto_Q^* q$.

Ein Algorithmus zur vollen Reduktion von p modulo Q

Gegeben: Polynom p , Menge von Polynomen Q aus $F[x]$

Gesucht: Polynom q mit $p \mapsto_Q^* q$

Grundidee: Zuerst Reduktion der höherwertigen Monome, dadurch werden die niederen Monome auch verändert

Bezeichnung $R(p, Q)$: Menge aller möglichen Reduktionspolynome in Q , die das höchstwertigste Monom in p reduzieren können

$$R(p, Q) = \{q \in Q - \{0\} \text{ mit } \text{hterm}(q) | \text{hterm}(p)\}$$

$$R(0, Q) = \emptyset$$

Es wird aus dieser Menge der möglichen Reduktionspolynome durch die Funktion **SelectPoly($R(p, Q)$)** ein **beliebiges** ausgewählt.

Schematische Beschreibung des Algorithmus

```

procedure Reduziere(p, Q)
  r = p // r: noch zu reduzierendes Polynom
  q = 0 // q: Ergebnispolynom
  while r ≠ 0 do {
    while R(r, Q) ≠ ∅ do {
      f = SelectPoly(R(r, Q)) // f: Reduktionspolynom
      r = r -  $\frac{M(r)}{M(f)}$  f
    }
    q = q + M(r); r = r - M(r)
  }
  return q
end

```

Ein Beispiel zum Algorithmus

Zu reduzierendes Polynom: $p = 3x^3y + 2x^2y^2 - 3xy + 5x$

Zugrundeliegende Termordnung: $<_L$ (lexikographisch)

Reduktionspolynome: $Q = \{q_1, q_2\} \subset \mathbf{Q}[x, y]$

$q_1 = x^2y + 5x^2 + y^2$, $q_2 = 7xy^2 - 2y^3 + 1$

Der Algorithmus liefert:

$$\begin{aligned}
 p &= 3x^3y + 2x^2y^2 - 3xy + 5x \\
 p &\mapsto_{q_1} p - 3x \cdot q_1 = -15x^3 + \underline{2x^2y^2} - 3xy^2 - 3xy + 5x \\
 &\mapsto_{q_1} -15x^3 - \underline{10x^2y} - 3xy^2 - 3xy + 5x - 2y^3 \\
 &\mapsto_{q_1} -15x^3 + \underline{50x^2} - \underline{3xy^2} - 3xy + 5x - 2y^3 + 10y^2 \\
 &\mapsto_{q_1} -15x^3 + 50x^2 - 3xy + 5x - \frac{20}{7}y^3 + 10y^2 + \frac{3}{7}
 \end{aligned}$$

Das ist die **vollständige** Reduktion von p modulo Q .

Ein weiteres Beispiel zum Algorithmus

Zugrundeliegende Termordnung: $<_G$ (graduell)

Reduktionspolynome: $Q = \{q_1, q_2\} \subset \mathbf{Q}[x, y, z]$

$q_1 = xy^2z - xyz$, $q_2 = x^2y^2 - z^2$

Zu reduzierende Polynome: $p_1 = x^2y^2z - z^3$, $p_2 = -x^2y^2z + x^2yz$

$$\begin{aligned}
 p_1 &\mapsto_{q_2} x^2y^2z - z^3 - z(x^2y^2 - z^2) = 0 \\
 p_2 &\mapsto_{q_1} -x^2y^2z + x^2yz - (-x)(xy^2z - xyz) = 0
 \end{aligned}$$

Aber: $p_1 + p_2 = x^2yz - z^3$ ist **irreduzibel** modulo Q .

D.h. p_1 und p_2 lassen sich modulo Q zu 0 reduzieren, ihre Summe ist aber irreduzibel modulo Q .

Nachteile des Algorithmus, Verbesserungen

- **Größtes Problem:** Reduktion ist nicht **eindeutig**, da die Auswahl des nächsten Reduktionspolynoms aus Q nicht determiniert erfolgt, sondern *zufällig*
- Lassen sich zwei Polynome p, q jeweils zu 0 reduzieren, so muß dies nicht für ihre Summe $p + q$ zutreffen
- Algorithmus kann terminieren, wenn alle Terme in r reduziert wurden, die größer sind als der kleinste "head term" in Q (nur minimale Verbesserung)
- Verbesserung in der innersten Schleife kann erreicht werden, wenn vor dem Algorithmus jedes Polynom in Q durch seinen *head term* dividiert wird (nur arithmetische, keine prinzipielle Verbesserung)

Weitere Sätze über Reduktionen

Satz 2: Sind p, q und r Polynome aus $F[x]$ und $S \subset F[x]$ und gilt $p - q \mapsto_S r$, dann existieren Polynome \bar{p} und \bar{q} , so daß $p \mapsto_S^+ \bar{p}$, $q \mapsto_S^+ \bar{q}$, $r = \bar{p} - \bar{q}$ gilt.

Satz 3: Sind p, q Polynome aus $F[x]$ mit $p - q \mapsto_S^+ 0$ für $S \subset F[x]$, dann gibt es ein $r \in F[x]$, so daß $p \mapsto_S^+ r$ und $q \mapsto_S^+ r$ gilt.

Satz 4: Sind p_1, p_2 Polynome mit $p_1 \mapsto_Q p_2$, dann existiert für jedes Polynom r ein Polynom s mit $p_1 + r \mapsto_Q^+ s, p_2 + r \mapsto_Q^+ s$.

Literatur

- [1] Geddes, Gabor, *Algorithms for Computer Algebra*, Kluwer Academic Publ. 1992