

# Gröbner-Fächer für lineare Codes

Daniel Rembold

Technische Universität Hamburg Harburg

*daniel.rembold@tuhh.de*

August 28, 2014

- ① Einleitung
- ② Mathematische Grundlagen
- ③ Aufzählen von Gröbner-Fächern
- ④ Ergebnisse
- ⑤ Fazit
- ⑥ Vorführung

- Gröbnerbasen
-

## Monom

- Produkt von Variablen über ein endliches Feld  $\mathbb{K}[X_1, X_2, \dots, X_n]$
- Schreibweise  $m = X_1^{u_1} X_2^{u_2} \cdots X_n^{u_n}$  und  $u_i \in \mathbb{N}_0$

Grad eines Monoms:  $\deg(m) = \sum_{i=1}^n u_i$ .

## Termordnung $>$

- Relation  $>$  zu der Menge von allen Monomen in  $\mathbb{K}[X_1, X_2, \dots, X_n]$

## Termordnung

- Lexikographische Ordnung  $>_{lex}$
- grad  $>_{grlex}$
- Ordnung mit Gewichtsvektor  $c = (c_1, \dots, c_n) \in \mathbb{R}_+^n$

## Leitterm $LT(f)$

- Polynom  $p \in \mathbb{K}[X_1, X_2, \dots, X_n]$  besitzt Term höchster Ordnung in Bezug auf  $>$

## Beispiel

Sei  $f = x^2 + 3xyz + y^3$

- lex-Order :  $f = \underline{x^2} + 3xyz + y^3$
- grlex-Order :  $f = \underline{3xyz} + y^3 + x^2$
- $(1, 2, 1)$  :  $f = \underline{y^3} + 3xyz + x^2$

## Ideal

- Kollektion von Polynomen  $f_1, \dots, f_s$  :

$$\langle f_1, \dots, f_s \rangle = \left\{ \sum_{i=1}^s h_i f_i \mid h_1, \dots, h_s \in \mathbb{K}[X_1, \dots, X_n] \right\}.$$

## Beispiel

Sei  $I = \langle f_1, f_2 \rangle = \langle x^2 + y, x + y + 1 \rangle$  und  $f = x^2y + x^2 + y^2 + xy + x$   
Dann gilt  $f = y \cdot f_1 + x \cdot f_2$ ,  $f \in I$ .

# Divisionsalgorithmus (1)

- Notwendig zum Lösen des Idealzugehörigkeitsproblems

## Ziel des Algorithmus

Polynom  $g$  durch Ideal  $I = \langle f_1, \dots, f_s \rangle$  teilen, so dass  
 $g = a_1 f_1 + \dots + a_s f_s + r, \quad a_i, g, I, r \in \mathbb{K}[X_1, \dots, X_n]$

Sei Polynom  $p$  und Ideal  $I$

- Wenn  $p \% I = 0$ , dann gilt  $p \in I$



---

**Algorithm 1** Divisionsalgorithmus (Header)

---

**Require:** Basis  $I = \langle f_1, \dots, f_m \rangle$  of nonzero polynomials **and** a fixed termorder  $LT$

**Ensure:**  $r = 0$  **or** none of the terms in  $r$  are divisible by  $LT_{\leq}(f_1), \dots, LT_{\leq}(f_m)$

---

- Reihenfolge der Polynome in  $I$  beeinflusst Ergebnis
- $r \neq 0$  möglich, obwohl  $p \in I$

# Gröbnerbasis (1)

## Gröbnerbasis

Sei Termordnung  $>$  und Ideal  $I$ , dann hat Gröbnerbasis  $G = \{f_1, \dots, f_m\}$  (in Bezug auf  $>$ ) von  $I$  die Eigenschaft:

- Von jedem Polynom  $p \in I$  ist  $LT_{>}(p)$  teilbar durch  $LT_{>}(f_i)$
- Divisionsrest eindeutig bestimmt und unabhängig von Reihenfolge
- Gröbnerbasis aus jedem Ideal mit Hilfe des Buchberger-Algorithmus und einer festen Termordnung

- Gröbnerbasen sind nicht eindeutig
- Reduzierte Gröbnerbasen jedoch eindeutig für Ideale

## Reduzierte Gröbnerbasis

Alle Leiterterme von Gröbnerbasis  $G$  in Bezug auf Termordnung  $>$  monisch und relativ prim zueinander

- Unendlich viele Termordnungen, endlich viele reduzierte Gröbnerbasen

## Gröbner-Fächer

- Vielflächiges Komplex welches Kegel im  $\mathbb{R}_+^n$  enthält
- Flächen werden durch Ebenenungleichungen der Polynome bestimmt

# Beispiel Gröbner-Fächer (1)

Sei

①  $I = \langle x^2 - z, y - x \rangle \in \mathbb{K}[x, y, z]$

②  $G_{>_{lex}} = \{ \underline{y}^2 - z, \underline{x} - y \}$

③  $\mathbf{w} = (a, b, c) \in \mathbb{R}_+^3$

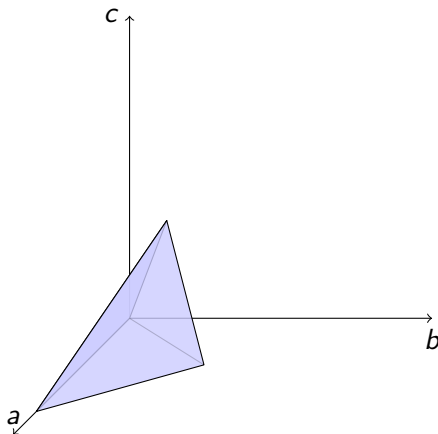
•  $\mathbf{w} \in G_{>_{lex}}$  genau dann, wenn

①  $(0, 2, 0) \cdot (a, b, c) \geq (0, 0, 1) \cdot (a, b, c) \vee 2b \geq c$

②  $(1, 0, 0) \cdot (a, b, c) \geq (0, 1, 0) \cdot (a, b, c) \vee a \geq b$

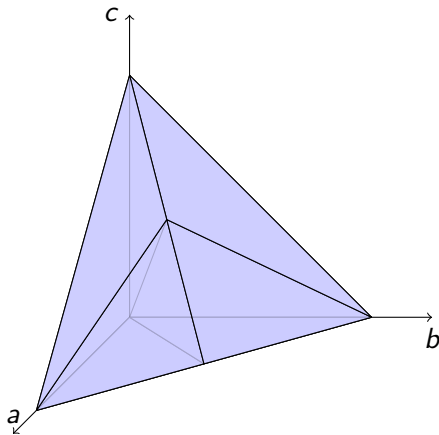
# Beispiel Gröbner-Fächer (2)

Figure: Gröbner-Kegel für  $G_{>_{lex}}$



# Beispiel Gröbner-Fächer (3)

Figure: Kompletter Gröbner-Fächer



- Obermenge von Code-Idealen
- Besteht nur aus Binomen

## Torisches Ideal

Gegeben seien  $A = [a_1, \dots, a_n] \in \mathbb{Z}^{d \times n}$  und  $u \in \mathbb{Z}^n$  zerlegbar in  $u^+$  und  $u^-$ , dann ist das torische Ideal  $I_A$  definiert durch

$$I_A = \langle \mathbf{x}^{u^+} - \mathbf{x}^{u^-} \mid u \in \ker(A) \rangle.$$



- Gröbner-Fächer als Graph
  - Gröbnerbasen als Knoten
  - Gröbnerbasen adjazent wenn im Gröbner-Fächer benachbart

## Facet Binomials

Bestimmen Grenzen zu anderen Gröbnerbasen

Sei  $x^{\alpha_k} - x^{\beta_k} \in \mathcal{G}_c$ ,  $\mathcal{G}_c$  Gröbnerbasis mit Gewichtsvektor  $c$

$x^{\alpha_k} - x^{\beta_k}$  ist Facet binomial wenn ein Vektor  $u \in \mathbb{R}$  folgendes erfüllt :

- $\{\alpha_i \cdot u > \beta_i \cdot u : i = 1, \dots, t, i \neq k\}$
- $\{\beta_k \cdot u > \alpha_k \cdot u\}.$

# Beispiel Facet Binomials

- Sei  $\mathcal{G} = \{x_1 - x_5, x_2^2 - 1, x_2x_4 - x_5x_6, x_2x_6 - x_4x_5, x_3 - x_5, x_4^2 - 1, x_4x_5x_6 - x_2, x_5^2 - 1, x_6^2 - 1\}$
- Prüfe ob  $x_2x_4 - x_5x_6$  ein facet binomial ist mit linearer Programmierung

$$\begin{array}{rcl} Ax & = & b \\ \text{subject to} & x & \geq 0 \end{array}$$

①  $b = (0, 1, 0, 1, -1, -1)^T$

②

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 2 & 1 & 0 & 0 \\ -1 & 0 & -1 & 1 & 0 & 1 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 2 \end{pmatrix}$$

- Gröberbasis erhalten durch andere Gröbnerbasis
- Facet Binomial "umdrehen" (flip)

---

**Algorithm 2** Local change of reduced Gröbner bases in  $I_A$

---

**Input:** Reduced Gröbner basis  $\mathcal{G}$  of  $I_A$  **and**

A prescribed facet binomial  $\underline{x}_i^a - x_i^b \in \mathcal{G}$

**Output:** The reduced Gröbner basis adjacent to  $\mathcal{G}$  in which  $\underline{x}_i^b - x_i^a$  is a facet binomial.

---

- Keine expliziten Termordnung nötig

---

**Algorithm 3** Enumerating the edge graph of the Gröbner fan via breath-first search

---

**Input:** Any reduced Gröbner basis  $\mathcal{G}_0$  of  $I_A$

**Output:** All reduced Gröbner bases of  $I_A$ , (all vertices of the edge graph)

---

- Einfacher und intuitiver Algorithmus
- Alle Gröbnerbasen müssen für den Vergleich gespeichert werden

- Gröbner-Fächer mit umgekehrter Tiefensuche nummerieren
- Ergebnis ist ein gerichteter Subgraph  $\longrightarrow$  Umgekehrter Suchbaum
- Nachteil von Breitensuche kompensiert



## University of Bristol

Optimizing OpenCL performance

[http://www.cs.bris.ac.uk/home/simonm/workshops/OpenCL\\_lecture3.pdf](http://www.cs.bris.ac.uk/home/simonm/workshops/OpenCL_lecture3.pdf)



## NVIDIA

OpenCL SDK Code Samples

<https://developer.nvidia.com/opencv>



Vasily Volkov (UC Berkeley , September 22, 2010)

Better Performance at Lower Occupancy

<http://www.cs.berkeley.edu/~volkov/volkov10-GTC.pdf>

Vielen Dank für eure Aufmerksamkeit!