Computeralgebra

Prof. Dr. K. Madlener

20. Juni 2008



Einführung "Computer Algebra"

- Berechnungen in algebraischen Strukturen: Monoide, Gruppen, Ringe, Polynomringe, Körper, Körpererweiterungen, Moduln, Vektorräume ...
- Lineare Algebra: Vektorräume, Matrizen, Determinanten, Gleichungssysteme.
- Analysis: Grenzwerte, Funktionen, Differentiation, Integration, Differentialgleichungen.
- Algebra: Gruppen, Ringe, Körper, Moduln, Konstruktionen: Produkte, Quotienten, Unterstrukturen.
- ► Zahlentheorie: Primzahlen, Faktorisierung, Kryptographie.
- ▶ Numerische Berechnungen vs. Symbolische Berechnungen.
- Computer Algebra Systeme, seit 60'er Jahren.
- Inhalt–Umriss



Einführung

Moderne CA-Systeme

Derive, Macsyma, Maple, Mathematica, Reduce, Scratchpad, Mupad, Mumath, Axiom, Magma, Mathlab

Ziele:

- ▶ Breite Funktionalität
- ► Einfache Bedienung
- Effizienz
- Frweiterbarkeit

Probleme:

- ▶ Darstellung der Strukturen und ihrer Elemente
- ▶ Effiziente Lösungen: Darstellungsabhängig
- ▶ Effiziente Transformationen zwischen Darstellungen

Vorteile CA-Systeme

Verarbeitung großer algebraischer Berechnungen

→ genaue Berechnungen "fehlerfrei"

Grundoperationen: Multiplikation, Division, Addition, Substraktion,

Exponentation

 \leadsto Arithmetik, Langzahlarithmetik

► GGT, KGV:

Euklidischer Algorithmus (Ringe euklidisch z. B. $\mathbb{Z}, \mathbb{Q}[x]$)

► Faktorisierung:

UFD (ZPE)-Ringe, Prim-Elemente (z. B. $\mathbb{Z}, \mathbb{Z}[x, y, z], .)$

► Klassische Algorithmen sind nicht immer effizient.

Problem: Zwischengrößenwachstum

Kosten arithmetischer Operatoren hängt von der Länge der Operanden ab.



- Allgemeine Systeme: Sprachumgebung, Notationen, Ein/Ausgaben,...
- ► Erfordern oft spezielle Programmiersprachen und Umgebungen
- ▶ Spezielle Systeme, z. B. Gruppen oder Gröbnerbasen, können oft nicht in andere Systeme verwendet werden.
- ▶ Vielzahl algorithmischer Lösungen, Vergleich schwer.
- ▶ Analyse der Algorithmen erfordert oft tiefe mathematische Ergebnisse.
- ▶ Wahl der Implementierungs- und Programmiersprachen



Prof. Dr. K. Madlener: Computeralgebra

Einführung Algebraische Grundlagen: Polynome, rationale Funktionen und Potenzreihe

Symbolische Numerische Berechnungen

Numerische Berechnungen Symbolische

1.1 Beispiel Chebyshev-Polynome. Rekursive Definition.

$$T_0(x) = 1$$
; $T_1(x) = x$; $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$ für $k \ge 2$.

Liste der Polynome: $1, x, 2x^2 - 1, 4x^3 - 3x, 8x^4 - 8x^2 + 1, ...$

Werte die Polynome an bestimmten Stellen aus.

Etwa für x = 0.3: 1, 0.3, -0.82, -0.792, 0.3448, . . .

Programm: Berechnung der 5 ersten Werte an einer Stelle x.

Für 0.3 sollte das Programm die Ausgabe:

$$T_0[0.3] = 1.0; \ T_1[0.3] = 0.3; \ T_2[0.3] = -0.82; \ T_3[0.3] = -0.792;$$

 $T_4[0.3] = 0.3448$ liefern.

Prof. Dr. K. Madlener: Computeralgebra

Symbolische

Einführung

Symbolische

Numerische Berechnungen

Standard Algorithmus in Programmiersprachen

```
procedure Chebyshev (input, output);
begin
var x: real; T: array[0..4] of real; n: integer;
writeln(x eingeben);
read(x);
T[0] := 1; T[1] := x;
for n := 2 to 4 do
    T[n] := 2 \cdot x \cdot T[n-1] - T[n-2];
for n := 0 to 4 do
    writeln('T'<sub>n</sub>[x] = T[n])
end.
```



Maple Version für Chebyshev Polynome

Numerische Berechnungen

$$T[0] := 1; \\ T[1] := x; \\ T[1] := x \\ \text{for } n = 2 \text{ to } 4 \text{ do} \\ T[n] := \operatorname{expand}(2 \cdot x \cdot T[n-1] - T[n-2]); \\ T[2] := 2x^2 - 1 \\ T[3] := 4x^3 - 3x \\ T[4] := 8x^4 - 8x^2 + 1 \\ \text{Interne Darstellung} \\ \text{Externe Darstellung}.$$



Historische Entwicklung der CASE

Historische Entwicklung der Case

Faktoren:

- Systeme (Programmiersprachen, HW.)
- Algorithmen (spezielle Lösungen)
- Anwendungen (Erweiterungen)

Höhere Programmiersprachen: Ende der 50er Anfang 60er. Fortran (58), Algol (60), Lisp (61).



Systeme **1961-1966**:

- ▶ I. Slagle (MIT): Lisp-Programm SAINT (Symbolic Automatic Integration): Lösen von unbestimmten Integralen unter Ausnutzung von Heuristiken.
- ▶ J. Sammet, R. Tobey (IBM): FORMAC (Fortran-Preprozessor): Symbolisches Rechnen mit elementaren Funktionen: Polynome, rationale Funktionen,u.a.
- W.S. Brown (Bell Labs): ALPAK (in Assembler geschriebene Subroutinen für Fortran): Symbolisches Rechnen mit Polynomen und rationalen Funktionen.
- G. Collins (IBM, University of Wisconsin at Madison): PM: symbolisches Rechnen mit Polynomen.
- ► C. Engelman (MIT): MATHLAB (LISP-basiert): symbolisches Rechnen mit Polynomen und rationale Funktionen, erstes interaktives System.

Systeme **1966-1971**:

- ▶ J. Moses (MIT): LISP-Programm SIN (Symbolic INtegrator).
- ► T. Hearn (Stanford University): REDUCE (LISP-basiert, interaktiv): für physikalische Berechnungen, hohe Portabilität.
- ► C. Engelman (MIT):MATHLAB-68 (graphische Ausgaben).
- ▶ A.D. Hall: ALTRAN (ALgebraic TRANslator): Sprache und System für das symbolische Rechnen mit Polynomen und rationale Funktionen.
- ▶ G. Collins: SAC-1 (Symbolic and Algebraic Calculations).
- ▶ D. Barton, S. Bourne, J. Fitch (University of Cambridge): CAMAL (CAMbridge ALgebra system: für astronomische Berechnungen und für Berechnungen der allgemeinen Relativitätstheorie.
- ► T. Hearn: REDUCE-2: allgemeines System mit Schwerpunkt für Berechnungen in der Hochenergie-Physik, geschrieben in RLISP (ALGOL-ähnlich).

Prof. Dr. K. Madlener: Computeralgebra 11

Einführung

○○○○○○○○○○○○○

Historische Entwicklung der CASE

Systeme **1971-1981**:

Alle bisherigen Systeme rein experimenteller Natur, wurden auch außerhalb der Gruppe der Entwickler verwendet. Insbesondere REDUCE weite Verbreitung aufgrund der leichten Portierbarkeit.

- ▶ J. Griesmer, R. Jenks (IBM Research): SCRATCHPAD: LISP-basiert, interaktiv, beinhaltet MATHLAB-68, REDUCE-2 und SIN.
- ▶ J. Moses, W. Martin (MIT): MACSYMA: algebraische Berechnungen, Grenzwert-Berechnungen, symbolisch Integrieren, Lösen von Gleichungen.
- ▶ G. Collins, R. Loos: SAC/ALDES: Bibliothek von Modulen, die in ALDES (ALgebraic DEScription language) geschrieben sind, zusammen mit einem Übersetzer nach ANSI FORTRAN. Alle verwendeten Algorithmen waren vollständig und ausführlich dokumentiert.
- ▶ D. Stautemeyer, A. Rich (University of Hawai): muMATH: eigene Programmiersprache, lief auf PC.

Spezielle Systeme:

- I. Frick (University of Stockholm): SHEEP: Berechnungen von Tensorprodukten.
- ► W. Jeffreys (University of Texas at Austin): TRIGMAN: in FORTRAN geschrieben, zur Berechnung von Poisson-Reihen.
- ► H. Veltman (NL): SCHOONSHIP: für Berechnungen in der Hochenergie-Physik.
- ▶ V.M. Glushkov (Hiev): ANALYTIK: Implementierung in Hardware.

CASe, die portabel sind meistens C-basiert. Wegen der stark angestiegenen Rechenleistung der Computer finden CASe mehr und mehr Anwendungen und Benutzer. Insbesondere entstehen nun auch kommerzielle CASe



Systeme **1981-1991**:

- ► G. Gonnet, K. Geddes (University of Waterloo): MAPLE: modulare Struktur, bestehend aus einem kleinen kompilierten Kern in C, und einer großen Library von mathematischen Subroutinen, die alle in der eigenen MAPLE Sprache geschrieben sind. Interpreter für die Kommandos, Integer und rationale Arithmetik, Polynom-Routinen und ein effizientes Speicherverwaltungssystem.
- ► S. Wolfram (Caltech): SMP (Symbolic Manipulation Program): in C geschrieben, Regel-basiert.
- ► S. Wolfram: MATHEMATICA: symbolische und numerische Berechnungen, graphische Wiedergabe (2-D und 3-D, inkl. Animation), C-basiert mit eigener Programmiersprache.
- ▶ D. Stoutemeyer, A. Rich: DERIVE: interaktiv, nicht als Programmierumgebung.
- ▶ weitere allgemeine Systeme: REDUCE 3, DOE-MACSYMA, MuPAD, AXIOM (SCRATCHPAD II).

Spezielle Systeme:

- J. Cannon (University of Sydney): CAYLEY: Gruppentheoretische Berechnungen. Mittlerweile MAGMA.
- ▶ J. Neubüser (RWTH Aachen): GAP (Group Algorithms and Programming). Mittlerweile in St. Andrews neu implementiert.
- ▶ J. Vermaseren: FORM: Berechnungen in der Hochenergie-Physik.
- ▶ A.M. Cohen: Lie: Berechnungen in Lie Algebren.
- M. Stillman: MACAULAY: Algebraische Geometrie und komm. Algebra.
- ► H. Cohen: PARI: Zahlentheorie.
- ► Greuel, Pfister (KL): SINGULAR: Gröbner Basen, Algebraische Geometrie, Singularitäten.
- ► COCOA (Genova) Kommutative Algebra. MAGNUS Gruppen.



Literatur

- ▶ von zur Gathen/Gerhard: Modern Computer Algebra, 1999, Cambridge University Press, ISBN 0-521-64176-4, INF 235/167 und L inf 92
- ► Geddes/Czapor/Labahn: Algorithms for Computer Algebra, INF 235/132, L inf 694.
- ▶ Davenport/Siret/Tournier: Computer Algebra, INF 235/116.
- ▶ Buchberger et al. (Eds.): Computer Algebra, INF 235/095.
- ▶ Mignotte: Mathematics for Computer Algebra, INF 235/126.
- Mignotte/Stefanescu: Polynomials: An Algorithmic Approach, INF 246/057.
- ▶ Winkler: Polynomial Algorithms in Computer Algebra, INF 235/132.
- ▶ Zippel: Effective Polynomial Computation, INF 246/054.
- ▶ Kreuzer, Robbiano Computational Commutative Algebra (0,1,2)

4□ > 4問 > 4 = > 4 = > 9 Q ○

Einführung

Einführung

Einleitung

Symbolische Numerische Berechnungen

Historische Entwicklung der CASE

Literatur

Algebraische Grundlagen: Polynome, rationale Funktionen und

Potenzreihen

Grundlagen ZPE (UFD)-Bereiche

Euklidische Bereiche

Ringkonstruktionen: Polynomring Ringkonstruktionen: Quotientenkörper Ringkonstruktionen: Potenzreihen

Prof. Dr. K. Madlener: Computeralgebra

Einführung

Algebraische Grundlagen: Polynome, rationale Funktionen und Potenzreiher

Grundlagen

Grundlegende Konzepte der Algebra

Axiome:

Assoziativität: $a \circ (b \circ c) = (a \circ b) \circ c$ $e \circ a = a \circ e = a$ (alle a) Neutrales Element: $a \circ a^{-1} = a^{-1} \circ a = e$ (alle a) Inverse:

 $a \circ b = b \circ a$ (alle a, b) Kommutativität:

 $a \circ (b + c) = (a \circ b) + (a \circ c)$ Distributivität:

 $(a+b) \circ c = (a \circ c) + (b \circ c)$ $a \circ b = 0 \Rightarrow a = 0$ oder b = 0Nullteilerfreiheit:

Additiv neutrales Element: 0 null. Multiplikativ neutrales Element: 1 eins.

Struktur Notation Axiome Gruppe $(G:\circ)$ $A_1: A_2: A_3$ Abelsche Gruppe $(G; \circ)$ A_1 ; A_2 ; A_3 ; A_4

Ring $(R:+,\cdot)$ A_1 ; A_2 ; A_3 ; A_4 bzgl. +, A_1 ; A_2 bzgl. ·, A_5

Kommutative Ringe $(R;+,\cdot)$ $+A_4$ bzgl. \cdot Integritätsbereich $(D;+,\cdot)$ $+A_6$

Körper $+A_3$ für $F\setminus\{0\}$ bzgl. \cdot (A_6 folgt daraus)

Grundlegende Konzepte der Algebra

2.1 Beispiel Strukturen

 \mathbb{Z} Integritätsbereich, $\mathbb{Q}, \mathbb{R}, \mathbb{C}$ Körper,

 \mathbb{Z}_n Ring, $n \in \mathbb{N}$, Primzahl $p \mathbb{Z}_n$ Körper, sonst kein Integritätsbereich.

Beachte: jeder endliche Integritätsbereich ist Körper.

Teilbarkeit und Faktorisierung in Integritätsbereichen

 $a, b \in D$ $a \mid b$ (a teilt b) gdw b = ax für ein $x \in D$.

a Teiler von b, b Vielfaches von a

GGT (GCD) größter gemeinsamer Teiler von $a, b \in D$ ist $c \in D$ mit $c \mid a$ und $c \mid b$, wobei c Vielfaches von jedem d mit $d \mid a$ und $d \mid b$ ist.

KGV (LCM) kleinstes gemeinsames Vielfaches von $a, b \in D$ ist $c \in D$ mit $a \mid c \text{ und } b \mid c$, wobei c Teiler von jedem d mit $a \mid d \text{ und } b \mid d \text{ ist.}$

Beachte: nicht eindeutig.

Assoziierte Elemente: $a, b \in D$ sind assoziiert, falls $a \mid b \land b \mid a$

Einheiten: $a \in D$ ist Einheit, falls a eine multiplikative Inverse besitzt.

(D) (A A) (B) (B)

Prof. Dr. K. Madlener: Computeralgebra

Algebraische Grundlagen: Polynome, rationale Funktionen und Potenzreihe

Einführung Grundlager

Grundlegende Konzepte der Algebra

2.2 Beispiel Einheiten, assoziierte Elemente:

 \mathbb{Z} :: Einheiten 1. -16. -6 sind GGT von 18 und 30.

6, -6 sind assoziiert.

Assoziert sein ist Äquivalenzrelation, Klassen assoziierter Elemente.

 $\mathbb{Z} :: \{0\}, \{1, -1\}, \{2, -2\}, \dots$

Auswahl von Repräsentanten aus assoziierten Klassen: ↔

Finheits-normale Flemente:: $n: D \rightarrow D$ n(a) EN.

 $\mathbb{Z}::\mathbb{N}$ als Einheits-normale Elemente

F Körper:: 0.1 Einheits-normal.

Einheits-normaler GGT und KGV sind dann eindeutig.

In der Regel: 0 ist EN, 1 ist EN, a, b EN, so auch ab.

Für $a \in D$, so a = u(a)n(a) eindeutig, wobei u(a) Einheit und n(a) EN.

Diese Begriffe, geeignet angepasst, sind auch für Ringe und Halbgruppen sinnvoll.

Nullteiler: $0 \neq a \in R$ ist Nullteiler, falls ab = 0 für ein $0 \neq b \in D$.

ZPE (UFD)-Bereiche

Prim-Elemente (irreduzible Elemente):: $p \in D - \{0\}$ ist Prim, falls

- b) p = ab, so a oder b Einheit. a) p ist keine Einheit,
- $a, b \in D$ heißen relativ Prim (teilerfremd), falls GGT(a, b) = 1.

Ein Integritätsbereich heißt ZPE-Ring (UFD), falls für $a \in D - \{0\}$ gilt, a ist Einheit oder a kann als (endliches Produkt) von Primelementen dargestellt werden und diese Darstellung ist eindeutig bis auf Assoziativität und Umordnung:

 $a = p_1 \cdots p_n$ Primelemente p_i $1 \le i \le n$ und ist $a = q_1 \cdots q_m$ mit Primelemente q_i , so m = n und bei geeigneter Umordnung der q_i gilt p_i ist assoziert mit q_i .

Wähle Einheitsnormale Primelemente: Eindeutige Primfaktorisierung: $a = u(a)p_1^{e_1}p_2^{e_2}\cdots p_n^{e_n}$ p_i Einheitsnormale Primelemente, $p_i \neq p_i$ für $i \neq j$. 2.3 Satz

Ist D ein ZPE-Ring $\{a,b\} \neq \{0\}$, so existiert GGT(a,b) und ist eindeutig.

21

Prof. Dr. K. Madlener: Computeralgebra

ZPE (UFD)-Bereiche

Einführung

Algebraische Grundlagen: Polynome, rationale Funktionen und Potenzreihe

Polynomringe

Polynomring über R: R[x]

$$a(x) = \sum_{k=0}^{m} a_k x^k$$
, $a_k \in R$, $m \ge 0$, Koeffizienten a_k

Grad von a(x): größtes n mit $a_n \neq 0$:: grad(a(x))

Standardform
$$\sum_{k=0}^{n} a_k x^k$$
, $a_n \neq 0$: a_n Leitkoeffizient

0 Polynom: $a_k = 0$ für alle k:: grad(0) := $-\infty$

Monisch, falls $a_n = 1$

Addition, Multiplikation definiert wie üblich.

Eigenschaften von R[x]

- i) R ist kommutativ, so auch R[x], 0 add. Einheit, 1 mult. Einheit.
- ii) D Integritätsbereich, so auch D[x]. Einheiten sind die konstanten Polynome a_0 mit a_0 ist Einheit in D.
- iii) Ist D ein ZPE-Ring, so auch D[x].

Polynome mit einheitsnormalen Leitkoeffizienten heißen einheitsnormal.

Polynomringe: Beispiele

2.4 Beispiel

Einführung

ZPE (UFD)-Bereiche

 $\mathbb{Z}[x]$: Einheiten 1, -1, einheitsnormale Polynome sind solche mit positiven Leitkoeffizienten.

 $\mathbb{Q}[x]$: Einheiten sind konstante Polynome ($\neq 0$). Einheitsnormale Polynome: Leitkoeffizienten 1 bzw. 0.

 $\mathbb{Z}_p[x]$ p prim: wie in $\mathbb{Q}[x]$.

Beachte

```
grad(a(x) + b(x)) \le max(grad(a(x), grad(b(x)))
(= falls grad(a(x)) \neq grad(b(x))
grad(a(x) \cdot b(x)) = grad(a(x)) + grad(b(x))
```

Prof. Dr. K. Madlener: Computeralgebra

Einführung

Euklidische Bereiche

Algebraische Grundlagen: Polynome, rationale Funktionen und Potenzreihe

Euklidische Bereiche

2.5 Definition

Euklidischer Ring ist Integritätsbereich D mit einer Bewertung $v: D - \{0\} \to \mathbb{N}$, die folgende Eigenschaft hat:

- 1) für alle $a, b \in D \{0\} : v(ab) > v(a)$
- 2) für alle $a, b \in D$ mit $b \neq 0$ gibt es $q, r \in D$: a = bq + r, wobei r = 0oder v(r) < v(b).

2.6 Beispiel

- $ightharpoonup \mathbb{Z}$ mit v(a) = |a| erfüllt 1), 2).
- F Körper v(a) = 1, $a \neq 0$ (r immer 0).
- $ightharpoonup \mathbb{Q}[x]$ (allgemeiner F[x] F Körper) $v(a(x)) = \operatorname{grad}(a(x)) \operatorname{erfüllt}(1) + 2)$

Beachte:

q (Quotient), r (Rest) in 2) müssen nicht eindeutig bestimmt sein (falls $r \neq 0$).



Einführung

Euklidische Bereiche

2.7 Beispiel

$$\mathbb{Z}$$
: $a = -8$ $b = 3$, so $-8 = 3 \cdot (-2) - 2 = 3 \cdot (-3) + 1$, d. h. $q = -2$, $r = -2$ und $q = -3$, $r = 1$ erfüllen 2).

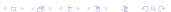
Vereinbarungen um Eindeutigkeit zu erreichen:

- ▶ In ℤ
 - a) Wähle q, r mit r = 0 oder sign(r) = sign(a)
 - b) Wähle q, r mit r = 0 oder sign(r) = sign(b)
- ▶ In F[x] sind q, r eindeutig. (warum?)

Euklidische Ringe sind ZPE-Ringe.

 $g = \mathsf{GGT}(a, b)$, so gibt es $s, t \in D$ mit g = sa + tb (nicht eindeutig!) s, t heißen Bezout-Koeffizienten.

Annahme: In "effektiven" Euklidischen Ringen sein zu a, b stets eindeutige q, r berechenbar.



Prof. Dr. K. Madlener: Computeralgebra

Einführung

Algebraische Grundlagen: Polynome, rationale Funktionen und Potenzreiher

Euklidische Bereiche

Euklidischer Algorithmus

2.8 Beispiel

In \mathbb{Z} :: GGT-Berechnung von 126 35

 $126 = 3 \cdot 35 + 21$

 $35 = 1 \cdot 21 + 14$

7 ist GGT(126, 35) $21 = 1 \cdot 14 + 7$

 $14 = 2 \cdot 7 + 0$

Anwendung: Simplifikation rationaler Ausdrücke: 35/126 → 5/18

Nutzen: Zahlen "klein" halten.

Sei D euklidischer Bereich $a, b \in D, b \neq 0$. Seien q, r Quotient und Rest mit a = bq + r, wobei r = 0 oder v(r) < v(b) setze

> quo(a, b) = q (auch a quo b) und rem(a, b) = r (auch a rem b oder a mod b)

Es gilt dann GGT(a, b) = GGT(b, r)

4□ > 4問 > 4 = > 4 = > 9 Q ○

Grundlage für Euklidischen Algorithmus

```
2.9 Lemma GGT(a, b) = GGT(b, r)
Beweis: Sei a = bq + r, dann gilt
GGT(b,r)|a \text{ und } |b \rightsquigarrow GGT(b,r)| GGT(a,b), \text{ wegen } r=a-bq \text{ folgt}
GGT(a, b)|r \text{ und } |b \rightsquigarrow GGT(a, b)|GGT(b,r), d. h.
GGT(a, b) und GGT(b, r) sind assoziiert, da EN sind sie gleich.
Seien a, b \in D, b \neq 0, v(a) > v(b).
Eine Restefolge für a, b ist definiert durch die Folge \{r_i\} mit
r_0 := a, r_1 := b \text{ und } r_i = \text{rem}(r_{i-2}, r_{i-1}), i = 2, 3, 4...
Es gilt v(r_0) \ge v(r_1) > v(r_2) > v(r_3) \dots
Es gibt ein k mit r_{k+1} = 0 (k < v(b)) und GGT(a, b) = n(r_k).
```

Prof. Dr. K. Madlener: Computeralgebra

Algebraische Grundlagen: Polynome, rationale Funktionen und Potenzreihe

Euklidische Bereiche

Einführung

Procedure Euclid

```
procedure Euclid (a,b)
    {Berechne g = \mathsf{GGT}(a, b) \ a, b \in D \text{ euklid. Bereich}}
begin
c := n(a); d := n(b);
while d \neq 0 do
    begin
    r := \text{rem}(c, d):
   c := d;
    d := r:
    end
g := n(c); return g
end.
```

Korrektheit und Terminierung folgen aus Lemma und Restefolgeneigenschaften. Komplexitätsanalyse folgt.



Erweiterter euklidischer Algorithmus (EEA)

```
procedure EEA(a, b; s, t)
       {Berechne g = \mathsf{GGT}(a, b) \text{ und } s, t \in D \text{ mit } g = sa + tb}
   begin
   c := n(a); d := n(b); c_1 := 1; d_1 := 0; c_2 := 0; d_2 := 1;
   while d \neq 0 do
       begin
       q := quo(c, d); r := c - q \cdot d;
       r_1 := c_1 - q \cdot d_1; r_2 := c_2 - q \cdot d_2;
                                                                          {Invariante::}
                                                         \{c = c_1 n(a) + c_2 n(b) \land \}
       c := d; c_1 := d_1; c_2 := d_2;
                                                          \{d = d_1 n(a) + d_2 n(b)\}
       d := r; d_1 := r_1; d_2 := r_2;
       end
   g := n(c);
   s := c_1/(u(a) \cdot u(c)); t := c_2/(u(b) \cdot u(c)); return (g, s, t);
   end.
Beachte:
n(c) = c_1 \cdot \frac{n(a)}{u(c)} + c_2 \cdot \frac{n(b)}{u(c)}: d.h. s, t sind die Bezout-Koeffizienten.
```

Prof. Dr. K. Madlener: Computeralgebra

Einführung Euklidische Bereiche Algebraische Grundlagen: Polynome, rationale Funktionen und Potenzreiher

Erweiterter euklidischer Algorithmus: Beispiel

2.10 Beispiel In Z:: a = 18b = 30

Wertefolge::

Prof. Dr. K. Madlener: Computeralgebra

Iteration	q	С	c_1	<i>c</i> ₂	d	d_1	d_2
0	_	18	1	0	30	0	1
1	0	30	0	1	18	1	0
2	1	18	1	0	12	-1	1
3	1	12	-1	1	6	3	-1
4	2	6	2	-1	0	0 1 -1 3 -5	3

$$g = 6$$
, $s = 2$, $t = -1$, $GGT(18, 30) = 2 \cdot 18 - 1 \cdot 30 = 6$

4□ > 4問 > 4 = > 4 = > 9 Q ○

Erweiterter euklidischer Algorithmus: Beispiel

$$\begin{array}{llll} & \ln \mathbb{Q}[x] & \text{in } \mathbb{Q}[x] &$$

Prof. Dr. K. Madlener: Computeralgebra

Einführung

Algebraische Grundlagen: Polynome, rationale Funktionen und Potenzreiher

Euklidische Bereiche

Kostenanalyse von EAA für \mathbb{Z} und $\boldsymbol{F}[x]$

Seien $a, b \in R$ mit n = v(a) > v(b) = m > 0.

Die Anzahl / der Durchläufe der While-Schleife wird durch l < v(b) + 1beschränkt. Die wesentliche Operation ist die Division mit Rest.

Diese ist *I*-mal durchzuführen: $I \le v(b) + 1 = m + 1$.

Sei R = F[x], F Körper, dann v(a) = grad(a).

Zähle Grundoperationen (go) in F:

Kosten der Division mit Rest: Seien grad(a) = n, grad(b) = m.

Ein Durchgang der Division kostet: Eine Division, m Multiplikationen, m Additionen in F, n-m+1 Durchläufe, d. h.

$$(2m+1)(n-m+1) = (2 \operatorname{grad}(b) + 1)(\operatorname{grad}(q) + 1) \in O(n^2)$$

Operationen in F. Ist b monisch, so spart man die Division.

Sei $n_i = \operatorname{grad}(c)$ in Durchlauf i (0 < i < l + 1), wobei d in Durchlauf l

Null wird. Dann gilt $n_0 = n \ge n_1 = m > n_2 > \cdots > n_l$ und $grad(q_i) = n_{i-1} - n_i$ für $1 \le i \le I$ (q_i Wert von q in Durchlauf i). Kosten

der Division mit Rest: $(2n_i + 1)(n_{i-1} - n_i + 1)$ arithm. Operationen in F.

Prof. Dr. K. Madlener: Computeralgebra

Kostenanalyse von EAA für F[x]: Kosten für s und t

Die Kosten für die r_i und q_i sind $\sum_{1 \le i \le l} (2n_i + 1)(n_{i-1} - n_i + 1)$ Operationen in F. Normaler Fall: $n_i = n_{i-1} - 1 = \cdots = m - i + 1$ 2 < i < l = m + 1 < 2mn + 2m.

2.11 Lemma Sei s_i Wert von c_1 in Durchgang i und t_i Wert von c_2 in Durchgang i. Dann gilt

1. grad
$$s_i = \sum_{2 \leq j < i} \operatorname{grad} q_j = n_1 - n_{i-1}$$
 $2 \leq i \leq l+1$

2. grad
$$t_i = \sum_{1 \le j \le i} \operatorname{grad} q_j = n_0 - n_{i-1} \quad 1 \le i \le l+1$$

Beweis: Wir zeigen nur 1) und grad $s_{i-1} < \text{grad } s_i$ (2 < i < l) durch Induktion nach i.

$$i=2$$
:: $s_2=(s_0-q_1s_1)=1-q_1\cdot 0$, grad $s_1=-\infty<0=$ grad s_2 . Sei $i>2$ Behauptung richtig für $2< i< i$, dann

$$\operatorname{grad} s_{i-1} < \operatorname{grad} s_i < n_{i-1} - n_i + \operatorname{grad} s_i = \operatorname{grad} (q_i s_i)$$



Prof. Dr. K. Madlener: Computeralgebra

Euklidische Bereiche

Einführung

Algebraische Grundlagen: Polynome, rationale Funktionen und Potenzreiher

Kostenanalyse von EAA für F[x]: Kosten für s und t

Also grad $s_{i+1} = \text{grad } (s_{i-1} - a_i s_i) = \text{grad } a_i + \text{grad } s_i > \text{grad } s_i$

$$ext{grad } s_{i+1} = ext{grad } q_i + ext{grad } s_i = \sum_{2 \le i \le i} ext{grad } q_j + ext{grad } q_i = \sum_{2 \le i \le i+1} ext{grad } q_j$$

Die Berechnung $t_{i+1} = (t_{i-1} - q_i t_i)$ bzw. $s_{i+1} = (s_{i-1} - q_i s_i)$. Multiplikation von Pol grad n, m : < 2(n+1)(m+1) Operationen. $2(\text{grad } t_i + 1)(\text{grad } q_i + 1) + (\text{grad } t_{i+1} + 1), d. h.$

$$\sum_{2 \le i \le l} 2(n_0 - n_{i-1} + 1)(n_{i-1} - n_i + 1) + (n_0 - n_i + 1)$$

Normalfall

$$\sum_{2 \le i \le m+1} 2(n-m+i-1)2 + n - (m-i+1) + 1 =$$

$$\sum_{2 \le i \le m+1} 5n - 5m + 5i - 4 = 5nm - 5mm + \frac{5}{2}m(m+1) + O(m)$$

Kostenanalyse für \mathbb{Z} : Langzahlarithmetik

Darstellung von Zahlen: Wort 64 Bits. 2⁶⁴-Standard Darstellung: Zahl als Feld von Wörtern. Erstes Wort für Vorzeichen und Länge des Feldes, d. h. $a \in \mathbb{Z}$

$$a = (-1)^s \sum_{0 \le i \le n} a_i 2^{64i}$$

$$s \in \{0,1\}, 0 \le n+1 < 2^{63}, a_i \in \{0,\dots,2^{64}-1\}$$

Als Feld: $s2^{63} + n + 1, a_0, ..., a_n$ von 64 Bit-Wörtern, z. B. $-1:2^{63}+1,1$ und 1:1,1.

Bereich: $-2^{64 \cdot 2^{63}} + 1$ bis $2^{64 \cdot 2^{63}} - 1$.

Länge von a: $\lambda(a) = \lfloor \log_{2^{64}} |a| \rfloor + 1 = \left\lfloor \frac{\log_2 |a|}{64} \right\rfloor + 1$.

Allgemein: Darstellung zur Basis b mit $2 \le b < \frac{|w|}{2}$, wobei |w|

Wortlänge ist (Multiplikation der Koeffizienten in Wort).

$$a = (u_1 \dots u_n)_b$$
 $0 \le u_i < b$, d. h. $a = \sum_{i=1}^n u_i b^{n-i}$
= $u_n + u_{n-1}b + \dots + u_1b^{n-1}$ a ist n-stellig zur Basis b.

$$a < b^n \rightsquigarrow a$$
 hat Länge $< n$.

Prof. Dr. K. Madlener: Computeralgebra

Einführung Euklidische Bereiche Algebraische Grundlagen: Polynome, rationale Funktionen und Potenzreihe

Langzahlarithmetik: Klassische Algorithmen

Klassische Algorithmen für: $+, -, \cdot,$ quo, Exponentation

Maß in Grundoperationen (go):

- ▶ Addition, Substraktion von 1-stelligen Zahlen
- ► Multiplikation von 1-stelligen Zahlen
- ▶ Division von 1-stelligen Zahlen

Einführung

A: Addition nicht negativer ganzer Zahlen zur Basis b.

```
Eingabe: (u_1 \cdots u_n)_b (v_1 \cdots v_n)_b
Ausgabe: (w_0 \cdots w_n)_b w_0 Übertrag mit
         (u_0 \cdots u_n)_b + (v_1 \cdots v_n)_b = (w_0 \cdots w_n)_b
  begin
                                                                   \{k = Übertrag\}
  i := n; k := 0
  while j > 0 do
      begin
      w_i := (u_i + v_i + k) \mod b;
                                                                       \{k \in \{0,1\}\}
      k := |(u_i + v_i + k)/b|;
      j := j - 1;
      end
   w_0 := k;
                                                 Korrektheit! Aufwand \approx 2n go
  end.
```

Prof. Dr. K. Madlener: Computeralgebra

Einführung Euklidische Bereiche Algebraische Grundlagen: Polynome, rationale Funktionen und Potenzreihen

Algorithmen: Substraktion

S: Substraktion nicht negativer ganzer Zahlen.

```
Eingabe: (u_1 \cdots u_n)_b > (v_1 \cdots v_n)_b
Ausgabe: Nichtnegative Differenz: u - v = (w_1 \cdots w_n)_b
  begin
  i := n; k := 0
  while j > 0 do
       w_i := (u_i - v_i + k) \mod b;
                                                                  \{k \in \{0, -1\}\}
      k := |(u_i - v_i + k)/b|
      i := i - 1;
      end
                                               Korrektheit! Aufwand \approx 2n go.
  end.
```

Algorithmen: Multiplikation

```
M: Multiplikation nicht negativer ganzer Zahlen Basis b.
Eingabe: (u_1 \cdots u_n)_b \geq (v_1 \cdots v_m)_b, d. h. n \geq m
Ausgabe: Produkt u \cdot v = (w_1 \cdots w_{m+n})_b
  for i from 1 to n do
       w_{m+i} := 0;
                                                           {Initialisierung m + i - te Stelle}
  j := m;
  while j > 0 do
       begin
       if v_i = 0 then
            w_i := 0
       else
           begin
           i := n; k := 0;
           while i > 0 do
                t := u_i v_i + w_{i+j} + k; w_{i+j} := t \mod b; k := \lfloor t/b \rfloor; i := i - 1;
            end
       j := j - 1;
       end
                                                         {Korrektheit! Aufwand \approx 3nm \text{ go}}
```

Prof. Dr. K. Madlener: Computeralgebra

Einführung Euklidische Bereiche Algebraische Grundlagen: Polynome, rationale Funktionen und Potenzreiher

Algorithmen: Motivation für Multiplikationsalg.

```
(u_1 \cdots u_n)(v_1 \cdots v_m)
   \begin{pmatrix} (u_1v_m)\cdots & (u_{n-1}v_m)(u_nv_m) \\ (u_1v_{m-1})\cdots & (u_nv_{m-1}) \end{pmatrix} m 
(u_1v_1)\cdots(u_nv_1)
                                                    \cdots w_{n+m}
W_1 \cdot \cdot \cdot W_m W_{m+1}
```

Einführung

D: Division mit Rest nicht negativer ganzer Zahlen Basis b.

Eingabe: (m+n) stellige Zahl, n stellige Zahl.

Ausgabe: (m+1) stelliger Quotient, *n* stelliger Rest.

Reduktion auf: Division mit Rest einer (n+1) stelligen Zahl u durch *n*-stellige Zahl v, mit $0 \le \left| \frac{u}{v} \right| < b$

Rest r ist jeweils kleiner als v, d. h. rb + (nächste Stelle des Dividenden) als "neues" u,

z.B.

$$3142: 47 = 66$$
 Rest 40

282

322

282

40



Prof. Dr. K. Madlener: Computeralgebra

Einführung Euklidische Bereiche Algebraische Grundlagen: Polynome, rationale Funktionen und Potenzreiher

Algorithmen: Division

Eingabe: $u = (u_0 u_1 \cdots u_n)_b$ $v = (v_1 \cdots v_n)_b$ mit $\left| \frac{u}{v} \right| < b$ (einstellig) **Bestimme**: $q = \left| \frac{u}{v} \right|$ mit u = qv + r, wobei $0 \le r < v$.

Schätzung für q: $\hat{q} = \min\left(\left|\frac{u_0b + u_1}{v_1}\right|, b - 1\right)$ erste Stelle für q.

2.12 Lemma (Übung): Es gilt

1) $\hat{q} > q$

2) Für $v_1 \ge \left| \frac{b}{2} \right|$ gilt $\hat{q} - 2 \le q \le \hat{q}$

D: Division mit Rest nicht negativer ganzer Zahlen Basis t.

Eingabe:
$$u=(u_1\cdots u_{m+n})_b \quad v=(v_1\cdots v_n)_b, \ v_1\neq 0, n>1$$

Ausgabe: Quotient $\lfloor \frac{u}{v}\rfloor=(q_0\cdots q_m)_b$, Rest $u \bmod v=(r_1\cdots r_n)_b$

Algorithmen: Division

```
begin
d:=\left|\frac{b}{(v_1+1)}\right|;
                                                                                 \{d \in \{|b/2|, \dots 1\}\}
(u_0 \cdots u_{m+n})_b := (u_1 \cdots u_{m+n}) \cdot d; (v_1 \cdots v_n)_b := (v_1 \cdots v_n) \cdot d; \{Normierung\}
for j from 0 to m do
      begin
     if u_i = v_1 then
            \hat{a} := b - 1
     while v_2\hat{q} > (u_j b + u_{j+1} - \hat{q}v_1)b + u_{j+2} do
            \hat{q} := \hat{q} - 1:
     if (u_j \cdots u_{j+n})_b < \hat{q} \cdot (v_1 \cdots v_n)_b then
     (u_j\cdots u_{j+n})_b:=(u_j\cdots u_{j+m})_b-\hat{q}\cdot (v_1\cdots v_n)_b; q_j:=\hat{q};
(r_1\cdots r_n)_b:=(u_{m+1}\cdots u_{m+m})_b/d;
end.
                                                              Korrektheit! Aufwand O(m \cdot n) go.
```

Prof. Dr. K. Madlener: Computeralgebra

◆ロト 4回ト 4 重ト 4 重ト 重 り Q C

Algebraische Grundlagen: Polynome, rationale Funktionen und Potenzreiher

Euklidische Bereiche

Einführung

Algorithmen: Exponentation

```
E: Exponentation:: Eingabe: x Basis b, n \in \mathbb{N}. Ausgabe: x^n
Naive Lösung: n-Multiplikationen.
Durch Quadrieren: \log n Multiplikationen, d. h. x^2, x^4, x^8, \dots
Länge der Zahlen: \lambda(x) = h \rightsquigarrow \lambda(x^n) = n \cdot h
  begin
                                              {Ergebnis in z, y \rightsquigarrow x, x^2, x^4, \dots}
  v := x; z := 1;
  while n > 1 do
       begin
       m:=\left|\frac{n}{2}\right|;
       if n > 2m then
           z := zy;
       y := yy; n := m;
       end
  z := zy;
   end.
```

Einführung

Algorithmen: Exponentation Beispiel

Grundlage: Ist
$$n = \sum_{i=0}^{k} e_i 2^i$$
 $e_i \in \{0, 1\}$, so

$$x^{n} = x^{\sum_{i=0}^{k} e_{i} 2^{i}} = \prod_{i=0}^{k} x^{e_{i} \cdot 2^{i}} = \prod_{i: e_{i} \neq 0} x^{2^{i}}$$

Anzahl der Multiplikationen:

$$N = k + e_0 + e_1 + \cdots + e_k - 1 \le 2k = 2 \log n$$

Problem:

Naiver Algorithmus x^n $\lambda(x)$ fest $x^i \cdot x$ kostet $c \cdot i \cdot \lambda(x)^2$ Hingegen $y \cdot y$ kostet $c \cdot \lambda(y) \cdot \lambda(y)$. D.h. es kommen größere Zahlen vor!



Prof. Dr. K. Madlener: Computeralgebra

45

OOOOOOOOOOO

Algebraische Grundlagen: Polynome, rationale Funktionen und Potenzreiher

Euklidische Bereiche

Algorithmen: Exponentation Analyse

$$c_{\text{exp}}(n) \approx c \cdot \lambda(x)^2 \sum_{i=0}^{k-1} 2^{2^{i+1}} + c \cdot \lambda(x)^2 \sum_{i=1}^{k} e_i \left(\sum_{j=0}^{i-1} e_j 2^j \right) 2^i$$

$$c_{\mathsf{naiv}}(n) \approx \frac{1}{2}c \cdot n^2 \cdot \lambda(x)^2 = c \cdot \lambda(x) \cdot \sum_{i=1}^{n-2} i \cdot \lambda(x)$$

d. h.
$$n = 2^k$$

$$c_{\text{exp}}(n) \cong \frac{4}{3}c \cdot n^2 \lambda(x)^2 \cong \frac{8}{3}c_{\text{naiv}}(n)$$

Für
$$n = 2^k + 2^{k-1}$$

$$c_{\text{exp}}(n) \cong \frac{4}{3}c \cdot 2^{2k}\lambda(x)^2 + c \cdot 2^{2k-1}\lambda(x)^2 \simeq \frac{11}{6}c \cdot 2^k\lambda(x)^2$$

$$c_{\mathsf{naiv}}(n) = \frac{9}{4}c \cdot 2^{2k}\lambda(x)^2 \simeq \frac{27}{12}c_{\mathsf{exp}}(n)$$

Falls $x \in R$, R endlich, so können die Kosten der Multiplikation als konstant gesehen werden und exp ist erheblich schneller als naiv.

Anwendungen: Cryptographie: Kodierung und Decodierung

RSA-Methode: $y = x^n \mod a$, $n > 10^{50}$,

Rekurrenzgleichungen, Potenzreihenentwicklungen.

GGT Kosten für \mathbb{Z} : $\mathbf{v}(\mathbf{a}) = |\mathbf{a}|$

$$a=r_0\geq b=r_1>r_2>\cdots>r_l\geq 0$$
 $q_i\geq 0$ alle i Darstellung der Zahlen z. B. 2^{64} -Standard Darstellung Länge $\lambda(a)=\left\lfloor\frac{\log_2|a|}{64}\right\rfloor+1$ Verwendet man $l\leq v(b)+1=b+1\leq 2^{64\lambda(b)} \Longrightarrow \exp$ in $\lambda(b)$. Polynomiale Schranke für $l:1\leq i\leq l$ $r_{i-1}=q_ir_i+r_{i+1}\geq r_i+r_{i+1}>2r_{i+1}$, d. h.

$$\prod_{\substack{2 \le i < l}} r_{i-1} > 2^{l-2} \prod_{\substack{2 \le i < l}} r_{i+1} = r_l + r_{l+1} > 2r_{l+1}, \text{ d. ii.}$$

$$\sum_{\substack{2 \le i < l}} r_{i-1} > 2^{l-2} \prod_{\substack{2 \le i < l}} r_{i+1} \text{ für } l \ge 2 \quad r_{l-1} \ge 2 \text{ folgt}$$

$$2^{l-2} < \frac{r_1 \cdot r_2}{r_1 \cdot r_1} < \frac{r_1^2}{2} \text{ oder } l \le \lfloor 2 \log r_1 \rfloor + 1 \approx 128\lambda(b)$$

2.13 Satz Lamé 1845

Sei $n \in \mathbb{N}^+$ und u kleinste positive Zahl, für die der EA für Eingabe u, v' n Iterationen benötigt für mindestens eine Zahl v' mit $v' \leq u$. Dann gilt $u = F_{n+1}$ und $v' = F_n$, wobei F_k k-te Fibonacci Zahl.

Prof. Dr. K. Madlener: Computeralgebra

Einführung 000000000 Algebraische Grundlagen: Polynome, rationale Funktionen und Potenzreihe

Euklidische Bereiche

GGT Kosten für \mathbb{Z} : $\mathbf{v}(\mathbf{a}) = |\mathbf{a}|$

Alle Quotienten gleich 1, z. B. (a, b) = (13, 8) EA

$$\begin{array}{lll} 13 = & 1 \cdot 8 + 5 & \textit{I für } (a,b) = (F_{n+1},F_n) \\ 8 = & 1 \cdot 5 + 3 & \leadsto \textit{I} = \textit{n} - 1 \approx 1.44 \log F_n + \textit{O}(1) \\ 5 = & 1 \cdot 3 + 2 & \text{für } \textit{b} \text{ fest und } \textit{a} \text{ Var gilt} \\ 3 = & 1 \cdot 2 + 1 & \text{im Mittel } \textit{I} \approx 0.584 \log \textit{b} \\ 2 = & 2 \cdot 1 & \end{array}$$

Beachte: Dirichlet / Lejeune 1849 Cesaro 1881

Für zufällig gewählte Zahlen a,b gilt $PR(\mathsf{GGT}(a,b)=1)=\frac{6}{\pi^2}\approx 0.6079$ Verwende: $PR(p\nmid n\land p\nmid m)=1-\frac{1}{\rho^2}$ $\prod_p (1-\frac{1}{\rho^2})\approx \frac{6}{\pi^2}$



Aufwand für EEA über \mathbb{Z}

Sei $n = \lambda(a), m = \lambda(b) \rightsquigarrow O(nm)$ für EA (Kosten der Div mit Rest a = qb + r $O((\lambda(a) - \lambda(b)) \cdot \lambda(b))$ go) Für die Bezout Koeffizienten gilt analog $|s_i| \leq \frac{b}{r_{i-1}}$ und $|t_i| \leq \frac{a}{r_{i-1}}$ $1 \leq i \leq l+1$

2.14 Satz Der EEA für Zahlen $a, b \in \mathbb{N}$ $\lambda(a) = n \ge \lambda(b) = m$, kann mit O(nm) go durchgeführt werden.

Weitere Ergebnisse und Bemerkungen siehe von zu Gathen, Gerhard bzw. Mignotte. Siehe auch Knuth Kap. 4.5.3, Bach/Shallit 4.2, 4.3. Viele Varianten zur Berechnung vom GGT (z.B. ohne Division). KGV Kleinste gemeinsamer Vielfache (LCM)

$$KGV(a, b) = \frac{|ab|}{GGT(a, b)}$$

Reduktion auf GGT-Berechnung.



Prof. Dr. K. Madlener: Computeralgebra

Einführung

Algebraische Grundlagen: Polynome, rationale Funktionen und Potenzreiher

Ringkonstruktionen: Polynomring

Ringkonstruktionen: R[x] Polynomring

R ZPE, so R[x] ZPE-Ring. R euklidisch $\neq R[x]$ euklidisch z. B. $\mathbb{Z}[x]$ nicht euklidisch, da kein Hauptidealring (z. B. $\langle 2, x \rangle$ wird nicht von $a(x) \in \mathbb{Z}[x]$ erzeugt oder $\mathbb{Q}[x, y]$ nicht euklidisch, da kein Hauptidealring (z. B. $\langle x, y \rangle$).

Vorteile E-Ringe: Euklidischer Algorithmus für GGT Berechnung.

Anwendungen: Lösung diophantischer Gleichungen in F[x]: a(x), b(x), c(y) gesucht $\sigma(x)$ und $\tau(x)$ mit

$$\sigma(x)a(x) + \tau(x)b(x) = c(x)$$

Lösbar für g(x) = GGT(a(x), b(x))|c(x). Eindeutigkeit und Schranken für die Grade von $\sigma(x)$, $\tau(x)$ (Übung).

Zerlegung rationaler Funktionen:

$$\frac{c(x)}{a(x)b(x)} = \frac{\tau(x)}{a(x)} + \frac{\sigma(x)}{b(x)} \quad \rightsquigarrow \text{Integration}$$

Problem: wie berechnet man GGT in $\mathbb{Z}[x]$ oder $\mathbb{Q}[x,y]$.

→ Pseudodivision primitiver EA.

4□ > 4問 > 4 = > 4 = > 9 Q ○

Quotienten-Körper von Integritätsbereichen

Übergang von $\mathbb{Z} \leadsto \mathbb{Q}$. D: Integritätsbereich \leadsto Körper.

Setze:: $S = \{a/b : a \in D, b \in D - \{0\}\}$ formale Quotienten.

 \sim auf S: $a/b \sim c/d \text{ gdw } ad = bc \text{ ist Äquivalenzrelation auf } S$ [a/b]

 $S/\sim = \{[a/b] : a \in D, b \in D - \{0\}\}, a/b \in [a/b] \text{ Repräsentant.}$

Addition + Multiplikation auf S/\sim :

(a/b) + (c/d) = (ad + bc)/bd

 $(a/b) \cdot (c/d) = ac/bd$ wohldefiniert auf Äguivalenzklassen.

 S/\sim ist Körper:: Q(D) (F_D) :: Quotientenkörper von D.

Kleinster Körper, der D enthält, $D \cong \{[a/1] : a \in D\}$

0/1 1/1a/1 mit a identifiziert.

Praxis: eindeutige Repräsentanten für [a/b], Entscheidung für \sim .

Falls GGT in *D* existiert:

 $a/b \in [a/b] \in S$ ist Repräsentant, falls GGT(a, b) = 1, b ist einheitsnormal, a, b in "Normalform".

z. B. \mathbb{Z} Quotientenkörper $Q(\mathbb{Z}) = \mathbb{Q}$ a/b "kanonisch", (b > 0).

-2/4, 2/-4, 100/-200, -600/1200 Kan. repräsentant: -1/2.

Prof. Dr. K. Madlener: Computeralgebra

Einführung

Einführung

Ringkonstruktionen: Quotientenkörpe

Algebraische Grundlagen: Polynome, rationale Funktionen und Potenzreiher

Ringkonstruktionen: Quotientenkörpe

Quotienten-Körper rationaler Funktionen

D[x] mit D ZPE-Ring, Q(D[x]) Körper der rationalen Funktionen (Ausdrücke) in x :: Schreibe D(x).

Beachte: Operationen $+, \cdot$ sind "teuer".

Addition: 3-Multiplikationen + Addition + GGT Berechnung Multiplikation: 2 Multiplikationen und GGT Berechnung.

Wähle geeignete Darstellungen

Fall $\mathbb{Z}[x]$ $\mathbb{Q}[x]$ bzw. $\mathbb{Z}(x)$ $\mathbb{Q}(x)$

in $\mathbb{Q}(x)$: $a(x)/b(x) = \left(\frac{17}{100}x^2 - \frac{3}{113}x + \frac{1}{2}\right)/\left(\frac{5}{9}x^2 + \frac{4}{5}\right)$

Die Äquivalenzlasse enthält Repräsentanten mit ganzzahligen Koeffizienten: z. B.

 $a(x)/b(x) = (4284x^2 - 675x + 12600)/(14000x^2 + 20160) \in \mathbb{Z}(x).$

D mit Quotienten-Körper F_D dann $D(x) \cong F_D(x)$.

Beachte: unterschiedliche kanonische Repräsentanten möglich.

Siehe Beispiel oben.



R[[x]] Potenzreihen mit Koeffizienten in R: Ausdrücke

$$a(x) = \sum_{k=0}^{\infty} a_k x^k \qquad a_k \in R$$

 $\operatorname{ord}(a(x)) = \min\{k : a_k \neq 0\}.$

0 alle $a_k = 0$, $a_k = 0$ für k > 1 Konstante PR.

Addition + Multiplikation wie üblich!

$$d(x) = a(x) \cdot b(x) = \sum_{k=0}^{\infty} d_k x^k$$
 mit $d_k = a_0 b_k + \cdots + a_k b_0$ $k \ge 0$

Eigenschaften:

- 1. $R[x] \hookrightarrow R[[x]]$
- 2. R kommutativ, so auch R[[x]] 0, 1
- 3. R Inthereich, so auch D[[x]]. Einheiten sind PR mit a_0 Einheit in R.
- 4. F Körper, so ist F[[x]] euklidischer Ring mit Bewertung $v(a(x)) = \operatorname{ord}(a(x)).$



Prof. Dr. K. Madlener: Computeralgebra

Algebraische Grundlagen: Polynome, rationale Funktionen und Potenzreiher

Ringkonstruktionen: Potenzreiher

Einführung

Potenzreihen - Einheiten

$$a(x) = \sum_{k} a_k x^k$$
 $b(x) = \sum_{k} b_k x^k$ $a(x) \cdot b(x) = 1$ so $1 = a_0 b_0$

$$0 = a_0 b_1 + a_1 b_0$$

$$b_1 + a_1 b_0$$

 \rightarrow a_0 ist Einheit

 $0 = a_0 b_n + a_1 b_{n-1} + \cdots + a_n b_0$

Ist a_0 Einheit in R, so wird b bestimmt durch

$$b_0 = a_0^{-1}, \ b_1 = -a_0^{-1}(a_1b_0), \cdots, \ b_n = -a_0^{-1}(a_1b_{n-1} + \cdots + a_nb_0)$$

In $\mathbb{Z}[[x]]$ gilt $(1-x)^{-1} = 1 + x + x^2 + x^3 + \cdots$

$$\mathbb{Z}[[x]]$$
 gilt $(1-x)^{-1} = 1 + x + x^2 + x^3 + \cdots$

Beachte

$$\operatorname{ord}(a(x) + b(x)) \ge \min{\operatorname{ord}(a(x)), \operatorname{ord}(b(x))}$$

$$\operatorname{ord}(a(x)\cdot b(x))=\operatorname{ord}(a(x))+\operatorname{ord}(b(x)).$$

Für
$$a(x), b(x) \in F[[x]], a(x) \neq 0 \neq b(x)$$
, so $a(x) \mid b(x)$ oder $b(x) \mid a(x)$.

Sei
$$\operatorname{ord}(a(x)) = I$$
 $\operatorname{ord}(b(x)) = m$, d. h.

$$a(x) = x^{l} \bar{a}(x)$$
 $b(x) = x^{m} \bar{b}(x)$ $\bar{a}(x), \bar{b}(x)$ Einheiten.

$$1 > m$$
, so $a(x)/b(x) = x^{1-m}\bar{a}(x) \cdot \bar{b}(x)^{-1} \in F[[x]].$

4 D > 4 B > 4 E > 4 E > E 900

Einführung

Potenzreihen - Einheiten, GCD in F[[x]]

Für $a(x), b(x) \in F[[x]], b(x) \neq 0$ gibt es q(x), r(x) mit

$$a(x) = b(x) \cdot q(x) + r(x)$$
 mit

$$r(x) = 0$$
 falls $ord(a(x)) \ge ord(b(x))$, $r(x) = a(x)$ falls

$$\operatorname{ord}(a(x)) < \operatorname{ord}(b(x)).$$

Quotientenkörper: Q(D[[x]]) Schreibe D((x)).

Achtung: D ZPE Ring $\not \hookrightarrow D[[x]]$ ZPE Ring, d. h. Normalformen schwer,

assoziierte Elemente!

$$a(x) = 2 + 2x + 2x^2 + 3x^3 + 4x^4 + \cdots$$

$$b(x) = 2 + 4x + 6x^2 + 9x^3 + 13x^4 + \cdots$$

$$c(x) = 2 + x^3 + x^4 + x^5 + x^6 + \cdots$$

sind assoziiert

$$b(x) = a(x)(1 + x + x^2 + x^3 + x^4 + \cdots)$$

$$c(x) = a(x)(1-x)$$

Welche PR soll als einheitsnormal gewählt werden! In F[[x]] geht dies:

$$a(x) = x^{l} \cdot b(x), \ l = \text{ord}(a(x)) \ b(x) = a_{l} + a_{l+1}x + \cdots \ a_{l} \neq 0 \ \text{also} \ b(x)$$

Einheit. Die Monome $x^{I}(I > 0)$ und 0 sind einheitsnormal.

$$GCD(a(x),b(x)) = x^{\min\{\operatorname{ord}(a(x)),\operatorname{ord}(b(x))\}}$$

Prof. Dr. K. Madlener: Computeralgebr

Einführung Ringkonstruktionen: Potenzreiher Algebraische Grundlagen: Polynome, rationale Funktionen und Potenzreiher

Erweiterte Potenzreihen

In
$$F((x))$$

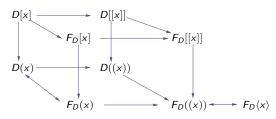
$$\left(\sum_{k=0}^{\infty} a_k x^k\right) / x^n \qquad n \ge 0$$

$$F\langle x \rangle$$
: $a(x) = \sum_{k=m}^{\infty} a_k x^k \quad a_k \in F, k \ge m, m \in \mathbb{Z}$

$$\operatorname{ord}(a(x)) = \min\{k : a_k \neq 0\} \ (< 0!)$$

 $F\langle x\rangle$ ist Körper.

Zusammenhang:



Standard Ringkonstruktionen

- ▶ $i \le R, i$ ideal, so R/i Ring: Quotientenring Idealkongruenz: $x \equiv_i y$ $(x \equiv y \mod i)$ gdw. $x y \in i$.
- ▶ R_1, R_2 Ringe, $R_1 + R_2 = \{(r_1, r_2) : r_1 \in R_1, r_2 \in R_2\}$ mit $(r_1, r_2) + (r'_1, r'_2) = (r_1 + r'_1, r_2 + r'_2)$ $(r_1, r_2) \cdot (r'_1, r'_2) = (r_1 r'_1, r_2 r'_2)$. $(0_{R_1}, 0_{R_2})$ Nullelement, $(1_{R_1}, 1_{R_2})$ Einselement. Produkt
- ▶ Ist *R* bzw. sind *R*₁, *R*₂ effektiv, so stellt sich die Frage ob der Quotientenring bzw. das Produkt effektiv sind.



Prof. Dr. K. Madlener: Computeralgebra

Normalformen - Algebraische Darstellungen

57

Inhalt Kapitel 3

Normalformen - Algebraische Darstellungen

Datenstrukturebene

Datenstrukturen - Algebraische Strukturen
Einfache Simplifikationsregeln in CA-Systemen
Wortproblem - Simplifikation
Formalisierung des Simplifikationsbegriffs
Abstraktionsebenen für algebraische Strukturen
Normalformen für Polynomringe, Quotientenkörper und Potenzreihen

Normalformen - Algebraische Darstellungen

Algebraische Strukturen \leftrightarrow Datenstrukturen (Typen) \simeq "Klassen" Menge und Operationen:: (\mathbb{Z} ; 0, 1, +, -, ·, /, mod , ggt , kgv , exp , . . .)

- ► Darstellung der Objekte: oft Konstruktionsvorschriften "Konstruktoren" für Definitionsbereich
- ▶ Termalgebra: Terme in Konstanten und Operatoren. Grundterme stellen Elemente des Definitionsbereich dar. Terme sind gleich, wenn sie das gleiche Element des Definitionsbereichs darstellen. z. B. $2^3 = (2 \cdot 2) + 4 = ggt(24, 16)$.

Probleme:

- \blacktriangleright Welche Darstellungen sind erlaubt (Operatoren z. B. für Ringe: $0, 1, +, \cdot$)
- ► Transformation von Darstellungen
- ► Eindeutige oder mehrdeutige Darstellungen
- ► Gleichheit von Darstellungen

(□) (□) (□) (□) (□) (□)

4 D > 4 B > 4 B > 4 B > B 990

Prof. Dr. K. Madlener: Computeralgebra

Normalformen - Algebraische Darstellungen

Datenstrukturen - Algebraische Strukturen

Verschiedene Darstellungsebenen

Elemente der algebraischen Struktur, Darstellungen, Rechnerdarstellung.

Objektebene, Formebene, Darstellungsebene

3.1 Beispiel Funktionenringe, Differentiation als Operator $\frac{\partial}{\partial x}(ax + xe^{x^2})$ Regeln (Axiome-Gleichungen)

$$\frac{\partial c}{\partial x} \to 0 \qquad \frac{\partial x}{\partial x} \to 1 \qquad \frac{\partial (u+v)}{\partial x} \to \frac{\partial u}{\partial x} + \frac{\partial v}{\partial x}$$
$$\frac{\partial (uv)}{\partial x} \to u \frac{\partial v}{\partial x} + \frac{\partial u}{\partial x} v$$

$$\frac{\partial (u^{\mathsf{v}})}{\partial x} \to \mathsf{v} u^{\mathsf{v}} \frac{\partial u}{\partial x} + (\log_e u) u^{\mathsf{v}} \frac{\partial \mathsf{v}}{\partial x}$$

 \leadsto Simplifikation symbolischer Ausdrücke \leadsto Reduktionsmethoden.

Einfache Simplifikationsregeln in CA-Systemen

- ▶ Unterdrücken von Klammern: Präfix-Postfix Notationen:: Formebene
- ▶ Identitäten Vereinfachung: z. B. $0 \cdot u \rightarrow 0$, $1 \cdot u \rightarrow u$, $u/1 \rightarrow u$, $v^0 \rightarrow 1$ ($v \neq 0$), $0^w \rightarrow 0$ (w > 0)
- ► Vorzeichenregeln: z. B. $(-u)(-v^3) = uv^3$, $-(u+v) \rightarrow -u-v$?
- Numerische Vereinfachungen: $\frac{5}{8} \frac{1}{8} \rightarrow \frac{1}{2}$, $9! \rightarrow 362880$ Vorsicht! oft nicht einfach: $(33282)\frac{1}{2}\sin\left(\frac{13\pi}{6}\right) \rightarrow \frac{122}{\sqrt{2}}$, e^e , e, π , . . .
- Assoziativ-kommutative Gesetze

$$(uv)w + (p+q) \rightarrow uvw + p + q$$
 $q+p \rightarrow p+q$?

► Anordnung: z. B. Polynomdarstellung



Prof. Dr. K. Madlener: Computeralgebra

Normalformen - Algebraische Darstellungen

Einfache Simplifikationsregeln in CA-Systemen

Einfache Simplifikationsregeln in CA-Systemen

▶ Zusammenfassung gemeinsamer Faktoren

$$u + (\frac{2}{3}) u \to \frac{5}{3} u, \ 2^{x+2} \to 4 \cdot 2^x, \ e^{5 + \log u} \to e^5 e^{\log u}$$

- ▶ Operationen mit Exponenten: $(u^w)^v \rightarrow u^{wv}$, $(uv)^w \rightarrow u^w v^w$
- ▶ Distributiv Gesetze: $(u + v)w \rightarrow uw + vw$
- ▶ Potenzen erweitern: $(a+b)^2 \rightarrow a^2 + 2ab + b^2$, $(1+x)^{100} \rightarrow ?$
- ► GGT-Vereinfachungen: $\frac{4u^2+12u^3+12u^2+4u}{2u^4-2u^3-2u^2+2u} \rightarrow \frac{2u+2}{u-1}$

Normalformen - Algebraische Darstellungen

Wortproblem - Simplifikation

Wortproblem - Simplifikation

 (M, \sim) WP:: Gegeben $u, v \in M$, Frage: $u \sim v$?

Wie ist M gegeben: oft endlich erzeugt, z. B. Termalgebra.

Wie ist \sim gegeben: oft Axiome (Gleichungen)

3.2 Beispiel Monoide, Gruppen: Erzeugende, Definierende Relationen

$$M=(\{a,b\};aba=\lambda,bab=\lambda)$$

 $G=(a,b,\bar{a},\bar{b};a\bar{a}=\bar{a}a=b\bar{b}=\bar{b}b=\lambda)$ freie Gruppe.
Frage: gilt $a=_Mb$ $a\bar{b}b\bar{a}=_G\lambda$?
 $abab$

Wortersetzungssysteme:: $M \cong (a; a^3 = \lambda)$, allg: Termersetzungssysteme

Simplifikation: Terme in "einfachste" Form zu bringen.

Methode: Maß: wohlfundierte (Partial)-Ordnung \succ auf M.

 $rep(u) = min_{v \sim u} v$ sollte eindeutig sein.

Frage: Ist rep effektiv berechenbar? i. Allg. nicht, da WP damit lösbar.



4 D > 4 D > 4 D > 4 D > 3 D 9 O

Prof. Dr. K. Madlener: Computeralgebra

Normalformen - Algebraische Darstellungen

Wortproblem - Simplifikation

Wortproblem - Simplifikation

Termersetzungssysteme: Methoden zur Behandlung von WP: Regeln, Konfluenz, Terminierung, Vervollständigung (KB).

Oft genügt es ein spezielles Wortproblem zu betrachten:

Rolle der Konstanten z. B. 0, 1.

Gruppen: $u = v \text{ gdw } uv^{-1} = 1$

Ringe: u = v gdw u - v = 0

→ Eigenschaften einer speziellen Äquivalenzklasse.

Wortproblem - Simplifikation

Wortproblem - 0-Äquivalenz

Richardson: Some unsolvable problems involving elementary funtions of a real var. J. Symb. Logic 33 (1968). How to recognize zero. J. Symb. Comp. 24 (1997). Funktionsklassen: $\{f: f: \mathbb{R} \to \mathbb{R}\}, +, \cdot, 0, 1$.

3.3 Satz Sei R die Klasse aller Terme, die man aus

- 1. \mathbb{Q} (rationale Zahlen), π Konstanten.
- 2. Einer Variablen x und Funktionen sin(x), |x|.
- 3. Operatoren: add(+), mult(*), $komp(\circ)$. Wie üblich definiert: $t_1(x) \circ t_2(x) = t_1(t_2(x))$

Interpretiert man die Konstanten als konstante Funktionen auf \mathbb{R} , x als Identitätsfunktion und $\sin(x)$, |x| durch die Standardfunktionen, so stellt jeder Term eine eindeutige Funktion aus $(\mathbb{R} \to \mathbb{R})$ dar.

Folgendes Problem ist unentscheidbar:

Eingabe: $E \in R$ **Frage**: Gilt E = 0?

□ ▶ ◆♬ ▶ ◆ 분 ▶ ◆ 분 · ∽ 오

Prof. Dr. K. Madlener: Computeralgebra

Normalformen - Algebraische Darstellungen

Wortproblem - Simplifikation

Wortproblem - 0-Äquivalenz

3.4 Beispiel

Prof. Dr. K. Madlener: Computeralgebra

$$\begin{array}{ll} \frac{1}{2} & \sin\left(\frac{1}{2}\right) = \sin(x) \circ \frac{1}{2} & \sin(|x|) \\ |\sin(x)|, & \sin(\sin(\sin(x))), & \sin(x)^2, & \sin(x) + |x| \end{array}$$

 $\textbf{Beweisidee} \hbox{: Reduziere Hilbert's 10. Problem auf ,,} E = 0 \hbox{".}$

Matiasevic 1970: Unentscheidbarkeit.

Hilbert's 10. Problem

Es gibt eine Menge $\mathcal{P} = \{p(x_1, \dots, x_n) \mid p \text{ Polynome über } \mathbb{Z}\}$ mit Prädikat: $p \in \mathcal{P}$ hat p Nullstelle in \mathbb{N}^n ist nicht rekursiv entscheidbar.

Idee der Reduktion: Sei
$$F : \mathbb{R} \to \mathbb{R}$$
, so $f^0(x) = x$ $f^{(n+1)}(x) = f(f^n(x))$ $n > 0$

4□ > 4□ > 4□ > 4□ > 4□ > 4□

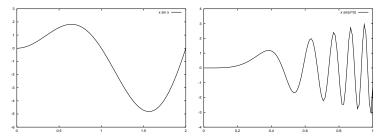
Normalformen - Algebraische Darstellungen

Wortproblem - Simplifikation

Wortproblem - 0-Äquivalenz

3.5 Lemma Jedes Tupel reeller Zahlen kann durch eine reelle Zahl dargestellt werden (bis auf ε). Seien $h(x)=x\sin x, g(x)=x\sin(x^3)$. Dann gibt es für alle $a_1,a_2,\ldots,a_n\in\mathbb{R}, 0<\varepsilon<1$ ein $b\in\mathbb{R}$ mit $|h(g^{(k+1)}(b))-a_k|<\varepsilon$ $1\leq k\leq n$.

$$h(x) = x \sin x \qquad g(x) \qquad = \qquad x \sin(x^3)$$



Prof. Dr. K. Madlener: Computeralgebra 6

Wortproblem - 0-Äquivalenz

3.6 Definition Eine Funktion (Term) $F(x_1,...,x_n) \in R$ wird von $G(x_1,...,x_n) \in R$ dominiert, wenn für alle $x_i \in \mathbb{R}$

i)
$$G(x_1, ..., x_n) > 1$$

ii)
$$\forall \delta_1, \dots, \delta_n \in \mathbb{R}, |\delta_i| < 1 : F(x_1 + \delta_1, \dots, x_n + \delta_n) < G(x_1, \dots, x_n)$$

3.7 Lemma Zu jeder Funktion $F \in R$ gibt es eine Funktion $G \in \mathcal{P}$, die F dominiert.

3.8 Lemma Für alle
$$P \in \mathcal{P}$$
 existiert $F \in R$ mit: Es gibt $(a_1, \ldots, a_n) \in \mathbb{N}^n$ mit $P(a_1, \ldots, a_n) = 0$ gdw es gibt $(b_1, \ldots, b_n) \in \mathbb{R}^n_{\geq 0}$ mit $F(b_1, \ldots, b_n) < 0$.

◆ロト 4個ト 4 恵 ト 4 恵 ト . 重 . 夕久の

Normalformen - Algebraische Darstellungen

Wortproblem - Simplifikation

Wortproblem - 0-Äquivalenz: Positive Ergebnisse

3.9 Satz Richardson

Betrachte Funktionenklasse, die durch Termmenge Λ definiert wird mit

- 1. $\mathbb{Q}, \pi \in \Lambda$
- 2. Var x Identität
- 3. $F, G \in \Lambda$. so (F + G), (F * G), (F/G)
- 4. $F \in \Lambda$, so $\log(|F|)$ und $\exp(F)$ in Λ

Interpretiere $F \in \Lambda$ als $F : \mathbb{R} \to \mathbb{R}$.

Das Prädikat " $F(x) \equiv 0$ " auf Λ ist entscheidbar.



< ロ ト → 付き → → き ト → ま → り Q (~)

Prof. Dr. K. Madlener: Computeralgebra

Normalformen - Algebraische Darstellungen

0000**000000**

Wortproblem - Simplifikation

Wortproblem - 0-Äquivalenz: Positive Ergebnisse

Beweisidee: Komplexität von Ausdrücken + Induktion. Sei etwa v

Unterausdruck mit größter Komplexität etwa $y = \log u$

$$F = a_n y^n + a_{n-1} y^{n-1} + \dots + a_1 y + a_0$$

wende Verfahren an:

$$a_n = 0 \rightsquigarrow F_1 = a_{\underline{n}-1}y^{n-1} + \cdots + a_0, \ F \equiv 0 \ \mathrm{gdw} \ F_1 \equiv 0.$$

$$a_n \neq 0 \rightsquigarrow F_2 := \frac{F}{a_n} = y^n + \frac{a_{n-1}}{a_n}y + \dots$$

$$F_3 = F_2' = ny^{n-1}y' + \dots + \frac{a_n a_0' - a_0 a_n'}{a_n^2}$$

$$F_2 \equiv 0 \rightsquigarrow F_3 \equiv 0$$
 $F_3 \equiv 0 \rightsquigarrow F_2$ konstant.

Klasse ist abgeschlossen gegen Ableitungen und die Ableitungen sind weniger komplex.

$$y = \log u \rightsquigarrow y' = \frac{u'}{u}$$
 u', u weniger komplex.
 $y = e^u$ ist dies nicht der Fall, unterscheide hier

$$v = e^u$$
 ist dies nicht der Fall, unterscheide hie

$$F \equiv a_1 y^n + \dots + a_0 \begin{cases} a_0 \equiv 0 \leadsto F_1 = a_n y^n + \dots + a_1 y = Qy \\ a_0 \not\equiv 0 \leadsto F_2 = F/a_0 \leadsto F_2'/y \equiv 0 \leadsto F \equiv c \end{cases}$$

Normalformen - Algebraische Darstellungen

Formalisierung des Simplifikationsbegriffs

Formalisierung des Simplifikationsbegriffs

Zwei Ziele:

- 1. "Einfachere" äguivalente Objekte zu definieren und sie bei Operationen zu verwenden.
- 2. Wenn möglich kanonische (d. h. eindeutige) Darstellung in (einigen/allen) Äguivalenzklassen festzulegen und wenn möglich effektiv zu bestimmen.

3.10 Definition Sei E Menge syntaktischer Objekte (z. B. Terme über Signatur, Formeln, Wörter, Programme) und sei \sim eine Äquivalenzrelation auf E. Sei weiterhin \prec eine Partialordnung auf E. Eine Simplifikationsfunktion für $[E; \sim]$ bzgl. \prec ist eine rekursive Funktion $f: E \rightarrow E$ mit

i)
$$f(t) \sim t$$
 ii) $f(t) \leq t$

i. Allg. \prec wohlfundierte Partialordnung auf E, d. h. es gibt keine ∞ -Ketten $e_1 \succ e_2 \succ e_3 \succ \cdots$,

z. B. |e| Länge des Ausdrucks $e_1 > e_2$ gdw $|e_1| > |e_2|$.

Prof. Dr. K. Madlener: Computeralgebra

Normalformen - Algebraische Darstellunger

Formalisierung des Simplifikationsbegriffs

Formalisierung des Simplifikationsbegriffs

Eine Normalisierungsfunktion bzgl. \prec ist eine Simplifikationsfunktion f

bzgl. \prec mit f(f(t)) = f(t) für alle t.

D. h. f(t) ist simplifiziert oder in Normalform.

Oft wird verlangt, dass für bestimmte Äquivalenzklassen z. B.

 $[0], [1]: t \sim 0 \text{ so } f(t) = f(0) = 0.$

Eindeutige Normalformen für spezielle Äguivalenzklassen in der Regel solche, die ausgewählte Konstanten der Signatur enthalten.

Eine kanonische Funktion ist eine Simplifikationsfunktion f mit

$$s \sim t$$
 so $f(s) = f(t)$ für alle $s, t \in E$

Sie berechnet eindeutige (kanonische) Repräsentanten für jede Äguivalenzklasse.

Prof. Dr. K. Madlener: Computeralgebra Prof. Dr. K. Madlener: Computeralgebra

Formalisierung des Simplifikationsbegriffs

Beachte: Ist *f* kanonisch, so ist *f* auch Normalisierungsfunktion und

$$s \sim t \text{ gdw } f(s) = f(t)$$

d. h. das Wortproblem für \sim ist entscheidbar.

f ist idempotent (d. h. $f \circ f = f$) und in jeder Äquivalenzklasse gibt es genau ein Element in kanonischer Form.

3.11 Satz Sei E eine entscheidbare Menge syntaktischer Objekte und \sim eine Äquivalenzrelation auf E. Dann gilt \sim entscheidbar (WP-entscheidbar) gdw es gibt eine kanonische Funktion f für $[E;\sim]$.



Prof. Dr. K. Madlener: Computeralgebra

7

Normalformen - Algebraische Darstellungen

Formalisierung des Simplifikationsbegriffs

Berechenbare Quotientenstrukturen

3.12 Satz Sei E entscheidbar, R berechenbare Operation auf E, d. h.

 $R: E^n \to E$ und \sim eine Kongruenz bzgl. R.

Hat E eine kanonische Funktion f bzgl. \sim und ist

 $rep(E) = \{t \in E : f(t) = t\}$ die Menge der kanonischen Repräsentanten, so lässt sich die Quotientenstruktur wie folgt darstellen:

$$R'(s_1,\ldots,s_n):=f(R(s_1,\ldots,s_n)) \text{ für } s_1,\ldots,s_n\in \operatorname{rep}(E)$$

und

$$(\operatorname{rep}(E),R')\cong (E/\sim,R/\sim)$$

rep(E) ist entscheidbar, R' ist berechenbar.

(□) (□) (□) (□) (□)

Normalformen - Algebraische Darstellungen

Formalisierung des Simplifikationsbegriffs

Beispiele: Monoide und Gruppen

3.13 Beispiel

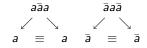
• $(a, b: ba = \lambda)$ Normalformen a^nb^m $n, m \ge 0$ Wortersetzungssystem: $ba \to \lambda$ terminierend (Längenkürz.) konfluent (d. h. eindeutige NF)

$$(a^{n}b^{m})\circ(a^{n'}b^{m'}) = \begin{cases} a^{n+(n'-m)}b^{m'} & n' \geq m \\ a^{n}b^{(m-n')+m'} & m > n' \end{cases}$$

 $(a, b, \bar{a}, \bar{b} : a\bar{a} = \bar{a}a = b\bar{b} = \bar{b}b = \lambda)$

WES: $a\bar{a} \rightarrow \lambda$ $\bar{a}a \rightarrow \lambda$ $b\bar{b} \rightarrow \lambda$ $\bar{b}b \rightarrow \lambda$ terminierend.

Konfluent: Kritische Paare:



◆ロト ◆部ト ◆ミト ◆ミト ・ ミーク **

Prof. Dr. K. Madlener: Computeralgebra

Normalformen - Algebraische Darstellungen

Formalisierung des Simplifikationsbegriffs

Beispiele: Monoide und Gruppen

- Normalformen: Wörter, die keine linke Seite als Teilwort enthalten → reguläre Sprache
- $(a, b : aba = bab = \lambda)$

WES: $aba \rightarrow \lambda \quad bab \rightarrow \lambda \text{ terminierend}.$

Nicht konfluent abab



Hinzunahme von Regeln $b \rightarrow a \rightsquigarrow \mathsf{Knuth}$ Bendix Vervollständigung. Länge-Lexikographische Ordnung $b \succ a$

 $(a, b; b \rightarrow a, a^3 \rightarrow \lambda)$, Repr. λ, a, a^2

Normalformen - Algebraische Darstellungen

Abstraktionsebenen für algebraische Strukturen

Beispiele

a) $E = \mathbb{Z}[x]$

Formebene

Abstraktionsebenen für algebraische Strukturen

- ▶ \mathbb{Z}_m $f(n) = n \mod m$ positive Reste Repr. 0, 1, ..., m-1, Definition von $+, \cdot$ auf \mathbb{Z}_m .
- 1) Objektebene: Menge Operationen = Elemente der Mengen
- II) Form-Ebene

Objekte werden explizit dargestellt "Bezeichner"

mehrere Gleichheiten
$$\equiv$$
 syntaktische $=$ semantische gleiche Bezeichner gleiche Objekte

Typische Bezeichner: Terme
$$12x^2y - 4xy + 9x - 3$$
 $(3x - 1)(4xy + 3)$ $(12y)x^2 + (-4y + 9)x - 3$

Syntaktisch verschieden, aber semantisch gleich.



▶ $p_1, p_2 \in \mathcal{F}$, so auch $(p_1 * p_2) \in \mathcal{F}$

• $p_1, p_2 \in \mathcal{F}$, so auch $(p_1 + p_2) \in \mathcal{F}$

Normalisierungsfunktionen:

$$f_2 \left\{ \begin{array}{l} f_1 \left\{ \begin{array}{l} \text{Multipliziere Produkte aus (Distributivgesetz) } \Sigma \text{ Monom.} \\ \text{Fasse Monome mit gleichem Grad zusammen.} \\ \text{Ordne Monome nach aufsteigendem Grad} \\ \text{(absteigendem)} \end{array} \right.$$

Prof. Dr. K. Madlener: Computeralgebra

Normalformen - Algebraische Darstellungen

Abstraktionsebenen für algebraische Strukturen

Abstraktionsebenen für algebraische Strukturen

III) Datenstrukturebene

Darstellung der Objekte aus Ebenen I), II) im Rechner:

Speicherorganisation

Listen, Felder, Verbunde usw.

Simplifikation definiert auf Ebene II).

Realisiert in Ebene III).

Wichtige Entscheidungen: Welche Darstellungen erlaubt man in Ebene II), wie werden diese in III) dargestellt.

Oft Unterscheidung nötig: Eingabe, Intern, Ausgabe.

Beispiele (2)

Prof. Dr. K. Madlener: Computeralgebra

Normalformen - Algebraische Darstellunger

Abstraktionsebenen für algebraische Strukturer

 f_1 ist Normalisierungsfunktion, f_2 ist kanonische Funktion.

Normalform bzgl. f_1 :

$$a_1x^{e_1} + a_2x^{e_2} + \cdots + a_mx^{e_m}$$
 $e_i \neq e_j$ für $i \neq j$

Kanonische Form bzgl. f2:

$$a_1 x^{e_1} + a_2 x^{e_2} + \dots + a_m x^{e_m}$$
 $e_i < e_j$ für $i < j$

Oft gilt $s \sim t \ \mathrm{gdw} \ M(s,t) \sim 0$, \exists Normalisierungsfunktion \leadsto kanonische Funktion.

Beispiele (3)

b) Abelsche Halbgruppen Varietät

Erzeugende Relationen

 $\Sigma :: a, b, c, f, s$ $E :: as = c^2 s, bs = cs, s = f$

+ Kommutativität

Faktorhalbgruppe des freien komm. Monoids in a, b, c, f, s

Formebene: $\{a^{n_1}b^{n_2}c^{n_3}f^{n_4}s^{n_5} \mid n_i \in \mathbb{N}\}$

 $\circ: M \times M \to M$ Addition der Exponenten.

Kongruenz, die von E erzeugt wird: Ersetzungsregeln

 $\underline{s} \to f$ $\underline{c}f \to bf$ $\underline{b^2}f \to af$ "Modulo Kommutativität"

Definiere kanonische Funktion $\stackrel{*}{\longrightarrow}$ mit kanonischen Formen $\subset a^{n_1}b^{n_2}c^{n_3}f^{n_4}$



Prof. Dr. K. Madlener: Computeralgebra

Normalformen - Algebraische Darstellungen

Abstraktionsebenen für algebraische Strukturen

Beispiele (4)

c) $E = \mathbb{Q}[x, y] : x^3 - x^2, x^2y - x^2$

 $i = \langle x^3 - x^2, x^2y - x^2 \rangle$ E/i

Regeln: $x^3 \rightarrow x^2$ $x^2y \rightarrow x^2$ Reduktionsfunktion



Prof. Dr. K. Madlener: Computeralgebra

$$x^3 - x^2y \stackrel{*}{\longrightarrow}$$

 \rightarrow definiert Simplifikationsfunktion $p \stackrel{*}{\longrightarrow} NF(p)$, sie ist kanonisch (\rightsquigarrow Gröbner Basen).

Normalformen für Polynomringe und Quotientenkörper, d. h.

Normalformen für Polynome und rationale Ausdrücke.

4□ > 4□ > 4□ > 4□ > 4□ > 4□

Normalformen - Algebraische Darstellungen

Normalformen für Polynomringe, Quotientenkörper und Potenzreihen

Beispiele (5)

Ringe: Axiome kommutative Ringe mit 1.

Signatur: 0, 1, -, +, *

Axiome: + Komm., Ass., 0 neutr. El., Gruppe inv. -

* Komm., Ass., Einh. + Distributivgesetz

Gleichheitsaxiome → Varietät.

Univariate Polynome: Formebene.

 $R[x]: a_n x^n + \cdots + a_1 x + a_0, \ n \ge 0, a_i \in R, a_n \ne 0 \cup \{0\}$

System kanonischer Formen für R[x] (dicht) oder dünn alle

Koeffizienten $\neq 0$.

Multivariate Polynome: Formebene.

Rekursive Darstellung: $R[x_1 ... x_n] = R[x_1 ... x_{n-1}][x_n]$

 $\operatorname{grad}(a(\bar{x}))$

$$a(x_1,...,x_n) = \sum_{i=0}^n a_i(x_1...x_{n-1})x_n^i$$

dicht/dünn

◆ロト ◆部 ト ◆ 恵 ト ◆ 恵 ・ 夕 Q ()

Prof. Dr. K. Madlener: Computeralgebra

Normalformen - Algebraische Darstellungen

Normalformen für Polynomringe, Quotientenkörper und Potenzreiher

Beispiele (6)

3.14 Beispiel

$$a(x, y, z) = (3y^2 + (-2z^3)y + 5z^2)x^2 + 4x + ((-6z + 1)y^3 + 3y^2 + (z^4 + 1))$$

Distributive Darstellung $a(\bar{x}) \in D[x]$

$$a(\bar{x}) = \sum_{e \in \mathbb{N}^n} a_e x^e \text{ mit } a_e \in D$$
 dicht /dünn $a_e \neq 0$

 $x^e \quad e \in \mathbb{N}^n$ werden oft Terme genannt.

$$a(x, y, z) = 3x^2y^2 - 2x^2yz^3 + 5x^2z^2 + 4x - 6y^3z + y^3 + 3y^2 + z^4 + 1$$

Reihenfolge der Terme? Ordnungen auf Termmengen, die kompatibel mit Termmultiplikation sind, z. B.

Lex x > y > z $x^2y^2 > x^2yz^3 > x^2z^2 > x > y^3z \cdots$

oder

Grad-Lex

 $x^{2}vz^{2} > x^{2}v^{2} > x^{2}z^{2} > v^{3}z > z^{4} > v^{3} > v^{2} > x$

Beispiele (7)

3.15 Beispiel

$$a(x,y) = ((x^2 - xy + x) + (x^2 + 3)(x - y + 1)) \cdot ((y^3 - 3y^2 - 9y - 5) + x^4(y^2 + 2y + 1))$$
Distributive Darstellung:

$$f_1(a(x,y)) = 5x^2y^3 + 3x^2y^2 - 13x^2y - 10x^2 + 3x^6y + 2x^6 - xy^4 + 7xy^3 \cdots$$

Kanonische distributive Darstellung:

$$f_2(a(x,y)) = x^7y^2 + 2x^7y + x^7 - x^6y^3 + 3x^6y + 2x^6 - x^5y^3 + 2x^5y^2 + \cdots$$

Faktorisierte Normalform:

$$f_3(a(x,y)) = (x^3 - x^2y + 2x^2 - xy + 4x - 3y + 3)(x^4y^2 + 2x^4y + x^4 + y^3 - 3y^2 - 9y - 5)$$

Faktorisierte kanonische Form:

$$f_4(a(x,y)) = (x-y+1)(x^2+x+3)(x^4+y-5)(y+1)^2$$



Prof. Dr. K. Madlener: Computeralgebra

Normalformen - Algebraische Darstellungen

Normalformen für Polynomringe, Quotientenkörper und Potenzreihen

Beispiele (8)

- Rekursive Darstellung
- Distributive Darstellung
 f₁
- Kanonische distributive Darstellung
 (Ordnung auf Termen) f₂
- Faktorisierte Normalform

$$\prod_{\substack{i=1\\f_3}}^k p_i o \prod_{i=1}^k f_2(p_i)$$

- Faktorisierte kanonische Form (D ZPE)

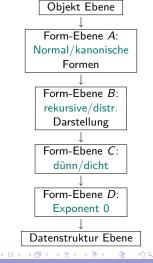
$$\prod_{i=1}^k p_i \to \prod_{i=1}^k f_2(p_i)$$

Faktorisiere die $f_2(p_i)$, fasse gleiche

Faktoren zusammen.

Einheitsnormale Faktorisierung +

Ordnung der Faktoren



Beispiele (9)

- $ightharpoonup f_1, f_2, f_3$ sind "einfach" zu berechnen.
- ▶ f₄ kostspielig!
- $ightharpoonup f_2, f_3$ werden am häufigsten verwendet.
- ► $(x + y)^{1000} y^{1000}$ von f_2 und f_3 expandiert!
- ▶ Weitere Transformationen erwünscht!

→□▶→同▶→□▶ →□ ● めの(

Prof. Dr. K. Madlener: Computeralgebra

Normalformen - Algebraische Darstellungen

Normalformen für Polynomringe, Quotientenkörper und Potenzreiher

Normalformen für rationale Ausdrücke

D Integritätbereich, Quotientenkörper (D) F_D

Annahme: D ZPE-Ring, d.h. GGT existiert.

 $D[x_1,\ldots,x_n]$ $D(x_1\ldots x_n)$

Formebene: $\frac{a}{b}$:: GGT (a, b) = 1, b EN, a, b kanon.

 $(\exp * \exp)$ $(\exp + \exp)$ $\frac{p}{q}$

Normalisierungsfunktion für rationale Ausdrücke

$f_5 ::$

- 1. Bringe in Gestalt $\frac{a}{b}$ mit $a, b \in D[x_1, \dots, x_n]$ (Gemeinsamer Nenner, Ausmultiplizieren) $\frac{a}{b} + \frac{a'}{b'} \rightarrow$
- 2. GGT(a, b) = 1
- $\frac{a}{b} \rightarrow \frac{a'}{b'}$ $a = a' \cdot g, b = b' \cdot g$
- 3. *b* Einheitsnormal: $\frac{a'}{b'} \to \frac{a''}{b''}$, $a'' = a' \cdot (u(b'))^{-1}$, $b'' = b' \cdot (u(b'))^{-1}$
- 4. $\frac{a''}{b''} \to \frac{f_2(a'')}{f_2(b'')}$

Andere Formen: a/b

Fakt/Fakt Fakt / erweitert mit GGT(a, b) = 1, b einheitsnormal.

erweitert / Faktor

Prof. Dr. K. Madlener: Computeralgebra

Normalformen - Algebraische Darstellungen

Normalformen für Polynomringe, Quotientenkörper und Potenzreihen

Normalformen Potenzreihen

Potenzreihen Truncated Power Series: Abbruchgrad t

$$a(x) = \sum_{k=0}^{t} a_k x^k + 0(x^{t+1})$$

→ Problem Normalformen

Explizite Darstellung unendlicher Reihen:

$$e^{x} = \sum_{k=0}^{\infty} \frac{1}{k!} x^{k}, d. h.$$

$$a(x) = \sum_{k=0}^{\infty} f_a(k) x^k$$

Koeffizientenfunktion $f_a: \mathbb{N} \to \mathbb{Q}$ rekursiv.

Datenstrukturebene

Normalformen - Algebraische Darstellungen

Darstellung der Objekte der Formebene im Rechner.

Entscheidung:

Alle nur Normalformen nur kanonische Formen

Ziel: Effiziente Unterstützung (Realisierung) der grundlegenden Operationen.

 $1. \mathbb{Z}, \mathbb{Q}$

Single-Precision Multi-Precision

Wortlänge z. B. 64 Bits

Langzahlen

SP-Zahl mit Vorzeichen: $-2^{63} + 1 \le SP-Zahl \le 2^{63} - 1$

Langzahlen als Listen von SP-Zahlen.

$$(d_0,\ldots,d_{l-1})\longleftrightarrow \sum_{i=0}^{l-1}d_i\beta^i$$

Wahl von β

 $1 \le \beta - 1 \le \mathsf{SP} ext{-}\mathsf{Zahl}$ oder als Feld var. Länge (wie gehabt!)

Prof. Dr. K. Madlener: Computeralgebra

Normalformen - Algebraische Darstellungen

Datenstrukturebene

Datenstrukturebene (Forts.)

Wahl von β :

i) $\beta - 1$ größte SPZ

ii) $\beta = 10^p$ p so groß wie möglich.

Länge / der Liste: Dynamisch oder statisch.

Implementierung: Zeiger oder Felder.

Referenzierte / sequentielle Zuweisung (-Vorzeichen, -Länge)

$$d o \boxed{d_0} \longrightarrow \boxed{d_1} \longrightarrow \cdots \longrightarrow \boxed{d_{l-1}} \longrightarrow \mathsf{n}$$

$$\beta = 10^3$$
 $N = 1/234/567/890$

$$N \rightarrow \boxed{890} \longrightarrow \boxed{567} \longrightarrow \boxed{234} \longrightarrow \boxed{1} \longrightarrow \boxed{1}$$

Feld

Probleme

- Feste Länge (Überlauf), auffüllen mit 0 (Platz Verschwendung).
- Listen, Pointer Kosten Platz, Kosten für nächste Stelle.

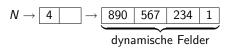
Descriptor Allocation

Mischung

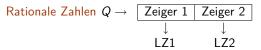
Beschreibungsblock



Länge, Vorzeichen Feld mit $I \beta$ -Zahlen



Problem: Speicherverwaltung kostspielig Garbagecollection



ロト 4回ト 4 重ト 4 重ト 重 めのの

Prof. Dr. K. Madlener: Computeralgebra

0.2

Normalformen - Algebraische Darstellungen

Datenstrukturebene

Datenstrukturen für Polynome

Datenstrukturen für Polynome / rationale Funktionen. Hängen von Entscheidungen auf Formebenen $B \mid C \mid D$ ab.

 $B: Rekursive, dünn \longrightarrow Listen$

C: Distributive, dünn \longrightarrow Felder.

.

 $a_i \in D[x_2, \dots, x_n]$

Normalformen - Algebraische Darstellungen

Datenstrukturebene

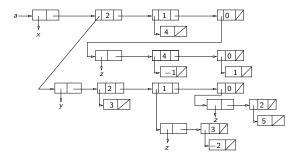
Beispiel

3.16 Beispiel $a(x, y, z) \in \mathbb{Z}[x, y, z]$ mit

$$a(x,y,z) = 3x^2y^2 - 2x^2yz^3 + 5x^2z^2 + 4x - z^4 + 1$$

= $(3y^2 + (-2z^3)y + 5z^2)x^2 + 4x + (-z^4 + 1)$

DS-Darstellung für rekursive dünn



Beachte hier Koeffizienten aus \mathbb{Z} als SPZ (können auch als LZ oder rationale Zahlen dargestellt sein!) Erlaubt Unterscheidung (Polynom, Zahl in \mathbb{Z}, \mathbb{Q}).

イロト イラト イラト ラ りの

Prof. Dr. K. Madlener: Computeralgebra

Normalformen - Algebraische Darstellungen

Datenstrukturebene

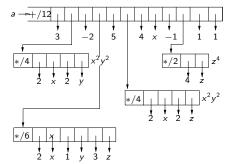
Distributive Form Darstellung Dynamische Felder als DS für Pol. (Maple)

 Type/Length
 Coeff
 Term
 ···
 Coeff
 Term

 Sum Product in Maple

 Type/Length
 Exponent
 Fact
 ···
 Exponent
 Fact

 3.17 Beispiel

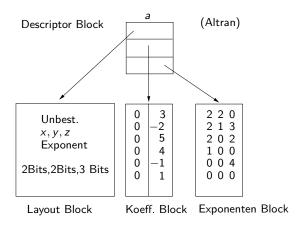


in Stufe

Arithmetik in Polynomringen - Modulare Arithmetik

Arithmetik in Polynomringen

Beispiel (Forts.)



◆ロト ◆園 ト ◆ 恵 ト ◆ 恵 ・ 夕 Q ②

Prof. Dr. K. Madlener: Computeralgebra

97

Arithmetik in Polynomringen - Modulare Arithmetik

Inhalt Kapitel 4

Arithmetik in Polynomringen - Modulare Arithmetik

Arithmetik in Polynomringen

Pseudo-Division mit Rest-Primitiver EA

Modulare Arithmetik

Schnelle Arithmetik

Die schnelle Fourier Transformation (FFT) Anwendung auf

Polynommultiplikation

Anwendung FFT auf Langzahlmultiplikation

Modulare Methoden

Chinesische Reste Algorithmen

Garner & Newton Interpolationsalgorithmen

Arithmetik in Polynomringen

Euklidischer Algorithmus für $D[\bar{x}]$?

Problem: $D[\bar{x}]$ i. Allg. nicht euklidisch, aber ZPE Ring, falls D ZPE-Ring: wie berechnet man GGT?.

4.1 Beispiel $\mathbb{Z}[x]$ seien

$$a(x) = 48x^3 - 84x^2 + 42x - 36$$

$$b(x) = -4x^3 - 10x^2 + 44x - 30$$

Eindeutige EN-Faktorisierungen in $\mathbb{Z}[x]$

$$a(x) = (2) \cdot (3)(2x - 3)(4x^2 - x + 2)$$

$$b(x) = (-1)(2)(2x - 3)(x - 1)(x + 5)$$

$$\rightsquigarrow \mathsf{GGT}(a,b) = 2 \cdot (2x-3)$$

10) 4 A) 4 B) 4 B) 4 A)

Prof. Dr. K. Madlener: Computeralgebra

Arithmetik in Polynomringen - Modulare Arithmetik

Arithmetik in Polynomringer

Beispiel (Forts.)

Berechnung in $\mathbb{Q}[x]$ (euklid. Ring) Zwei Möglichkeiten:

Eindeutige EN-Faktorisierungen in $\mathbb{Q}[x]$

$$a(x) = (48) \left(x - \frac{3}{2}\right) \left(x^2 - \frac{1}{4}x + \frac{1}{2}\right)$$

$$\rightsquigarrow \mathsf{GGT}(a,b) = x - \frac{3}{2}$$

$$b(x) = (-4)\left(x - \frac{3}{2}\right)(x - 1)(x + 5)$$

Euklidischer Algorithmus in $\mathbb{Q}[x]$

$$\rightsquigarrow \mathsf{GGT}(a,b) = x - \frac{3}{2}$$

◆ロト 4個ト 4 恵 ト 4 恵 ト . 重 . 夕久の

Wie hängen die beiden berechneten GGT's voneinander ab? Übung

Pseudo-Division mit Rest-Primitiver EA

Beispiel

4.2 Beispiel

$$a(x) = 3x^3 + x^2 + x + 5$$

 $b(x) = 5x^2 - 3x + 1$ in $\mathbb{Q}[x]$

$$\rightarrow$$
 $a(x) = b(x) \cdot q(x) + r(x)$ mit

$$q(x) = \frac{3}{5}x + \frac{14}{25}$$
 $r(x) = \frac{52}{25}x + \frac{111}{25}$

Hauptkoeffizient von b(x) ist 5. Nenner sind Potenzen von 5.

In $\mathbb{Z}[x]$ ist obige Division nicht möglich: $3=5q_1,q_1\in\mathbb{Z}$ $\mbox{\em 4}$

Wählt man
$$\bar{a}(x) = 5^2 \cdot a(x)$$
, so gilt $5^2 \cdot (3x^3 + x^2 + x + 5) = (5x^2 - 3x + 1) \cdot (15x + 14) + (52x + 111)$ in $\mathbb{Z}[x]$, wobei $\operatorname{grad}(52x + 111) < \operatorname{grad}(5x^2 - 3x + 1)$

→ Pseudo-Divisions-Eigenschaft



Prof. Dr. K. Madlener: Computeralgebra

101

Arithmetik in Polynomringen - Modulare Arithmetik

Pseudo-Division mit Rest-Primitiver EA

PD-Eigenschaft: D ZPE Ring, $a(x), b(x) \in D[x]$ mit $b(x) \neq 0$, $\operatorname{grad}(a(x)) \geq \operatorname{grad}(b(x))$. Dann gibt es Polynome $q(x), r(x) \in D[x]$ mit $\beta^I a(x) = b(x) \cdot q(x) + r(x)$ $\operatorname{grad}(r(x)) < \operatorname{grad}(b(x))$ wobei

- $\qquad \qquad \beta = \mathsf{Haupt}_\mathsf{Koeff}(b(x)),$
- $I = \operatorname{grad}(a(x)) \operatorname{grad}(b(x)) + 1$
- ► q(x):: Pseudo_Quotient und
- ► r(x):: Pseudo_Rest.

Beachte: q(x) und r(x) sind eindeutig und können durch die "übliche" Division bestimmt werden.

Pseudo-Division mit Rest

4.3 Definition

D ZPE Ring $0 \neq a(x) \in D[x]$ heißt primitiv, falls a EN und die Koeffizienten teilerfremd sind (d. h. $\mathsf{GGT}(a_0,\ldots,a_n)=1$ für $\mathsf{grad}(a)=n$).

z. B. $a_n x^n$ ist primitiv in $\mathbb{Z}[x]$ gdw $a_n = 1$.

Der Inhalt (Content) von a(x) (Bez. cont(a(x))) ist der eindeutige EN GGT der Koeffizienten von a(x),

d.h. jedes Polynom hat eine eindeutige Darstellung

$$a(x) = u(a(x)) \cdot cont(a(x)) \cdot PP(a(x))$$

wobei PP(a(x)) primitiv ist: Primitiver Anteil von a(x).

Für 0 setze cont(0) = 0 und PP(0) = 0.

(□) (□) (□) (□) (□) (□) (□)

4 D > 4 D > 4 D > 4 D > 3 D 9 O

Prof. Dr. K. Madlener: Computeralgebra

103

Arithmetik in Polynomringen - Modulare Arithmetik

Pseudo-Division mit Rest-Primitiver EA

Pseudo-Division mit Rest-Rrimitiver EA (Forts.)

Gauss Lemma: Produkt von primitiven Polynomen ist primitiv.

Man erhält:

GGT(a(x), b(x)) =

 $\mathsf{GGT}(\mathsf{cont}(a(x)),\mathsf{cont}(b(x))) \mathsf{GGT}(PP(a(x)),PP(b(x)))$

Berechnung im Koeff. Bereich

Annahme: Berechnung vom GGT im Koeff. Bereich bekannt!

4.4 Satz Sei D ZPE Ring. Sind $a(x), b(x) \in D[x]$ primitiv mit $b(x) \neq 0$ und grad $(a(x)) \geq \operatorname{grad}(b(x))$.

Seien q(x), r(x) Pseudo_Quotient und Pseudo_rest mit

 $\beta^l(a(x)) = b(x) \cdot q(x) + r(x), b(x), r(x) \in D[x], \operatorname{grad}(r(x)) < \operatorname{grad}(b(x)).$ Dann gilt

 $\mathsf{GGT}(\mathsf{a}(\mathsf{x}),\mathsf{b}(\mathsf{x})) = \mathsf{GGT}(\mathsf{b}(\mathsf{x}),\mathsf{PP}(\mathsf{r}(\mathsf{x})))$

Pseudo-Division mit Rest-Primitiver EA (Forts.)

```
Beweis: GGT(\beta^{I}a(x), b(x)) = GGT(b(x), r(x))
GGT(\beta^I a(x), b(x)) = GGT(\beta^I, 1)GGT(a(x), b(x)) = GGT(a(x), b(x))
da a, b primitiv. Somit
     GGT(b(x), r(x)) = GGT(1, cont(r(x)))GGT(b(x), PP(r(x)))
                             = \mathsf{GGT}(b(x), PP(r(x)))
→ Pseudo polynomiale Restefolge zur Berechnung des GGT
→ Primitiver EA
Lässt sich verallgemeinern: PP Restefolge für F_1, F_2:
F_1, F_2, \dots, F_{k-1}, F_k \text{ mit } F_i(x) = \alpha_i F_{i-2}(x) - q_i(x) F_{i-1}(x) \text{ mit}
\operatorname{grad}(F_i) < \operatorname{grad}(F_{i-1})(i > 2) \ \alpha_i \in D, F_i \in D[x].
(Eine Wahl für \alpha_o ist f_{i-1}^{n_i-n_{i-1}+1} f_i HK von F_{i-1}, n_i = \operatorname{grad}(F_i))
```

< ロ ト → 付 ト → 三 ト → 三 → り Q (~)

Prof. Dr. K. Madlener: Computeralgebra

Arithmetik in Polynomringen - Modulare Arithmetik

Pseudo-Division mit Rest-Primitiver EA

Pseudo-Division mit rest-primitiver EA (Forts.)

```
Primitiver EA D[x] D mit GGT-Berechnung
  procedure PEA (a(x),b(x))
  begin
  c(x) := PP(a(x)); D(x) := PP(b(x));
  while D(x) \neq 0 do
      begin
      r(x) := \text{Prem}(c(x), D(x));
      c(x) := D(x);
      D(x) := PP(r(x));
      end
  \gamma := \mathsf{GGT}(\mathsf{cont}(a(x)), \mathsf{cont}(b(x));
  g(x) := \gamma \cdot c(x); return (g(x));
  end.
```

Pseudo-Division mit Rest-Primitiver EA

Beispiel

4.5 Beispiel in $\mathbb{Z}[x]$

$$a(x) = 48x^3 - 84x^2 + 42x - 36$$

$$b(x) = -4x^3 - 10x^2 + 44x - 30$$

Iter.	r(x)	c(x)	d(x)
0	_	· ·	$2x^3 + 5x^2 - 22x + 15$
1	$-68x^2 + 190x = 132$	$2x^3 + 5x^2 - 22x + 15$	$34x^2 - 95x + 66$
2	4280 <i>x</i> - 6420	$34x^2 - 95x + 66$	2x - 3
3	0	2x - 3	0

$$\gamma = \mathsf{GGT}(6,2) = 2$$
 $g(x) = 2 \cdot (2x - 3) = 4x - 6$.

Vorteil: Anwendbar auf $D[\bar{x}]$ für D ZPE mit GGT.

Problem: Wachstum der Koeffizienten bei PD mit Rest!



Prof. Dr. K. Madlener: Computeralgebra

Arithmetik in Polynomringen - Modulare Arithmetik

Pseudo-Division mit Rest-Primitiver EA

Beispiel

4.6 Beispiel

$$a(x) = x^{8} + x^{6} - 3x^{4} - 3x^{3} + 8x^{2} + 2x - 5$$

$$b(x) = 3x^{6} + 5x^{4} - 4x^{2} - 9x + 21$$

Koeffizienten der Polynome in der PP-Restefolge

d. h. teilerfremd GGT 1

Vergleich mit EEA (oder EA) für F[x]!

Prof. Dr. K. Madlener: Computeralgebra Prof. Dr. K. Madlener: Computeralgebra

Modulare Arithmetik

Wie prüft man für große Zahlen a, b, ob $a \cdot b = c$?

"Fingerprinting" Technik:

Wähle SP-Primzahl p und teste, ob $a \cdot b \equiv c \mod p$, d. h. $a \cdot b - c$ ist teilbar durch p oder $a \cdot b$ und c haben gleichen Rest nach Teilung durch p.

 $a^* = a \mod p$, $b^* = b \mod p$, $c^* = c \mod p$ teste, ob $a^* \cdot b^* \equiv c^* \mod p$ (Beachte mehr als $2 \cdot 10^{17}$ 64 Bit PZ).

Testen von Polynomgleichungen $f \cdot g = h$ oder Matrizen $A \cdot B = C$ durch Auswertung an einer Stelle.

Spezialfall von Berechnungen via Homomorphismen.

Hier: $\mathbb{Z} \to \mathbb{Z}_p$ oder \mathbb{Z}_n .



Prof. Dr. K. Madlener: Computeralgebra

10

Arithmetik in Polynomringen - Modulare Arithmetik

Modulare Arithmetik

Darstellungen von \mathbb{Z}_n

- ▶ **Positive Darstellung**: Repräsentanten $\{0, 1, ..., n-1\}$
- ► Symmetrische Darstellung: Repräsentanten

$$\left\{-\left\lfloor \frac{n}{2} \right\rfloor, \ldots, -1, 0, 1, \ldots, \left\lfloor \frac{n}{2} \right\rfloor\right\}$$
 bzw. $\left\{-\frac{n}{2} + 1, \ldots, -1, 0, 1, \ldots, \frac{n}{2}\right\}$

 $\mathsf{Z.B.::} \ \mathbb{Z}_7 \quad \{0,1,2,3,4,5,6\} \ \mathsf{bzw.} \ \{-3,-2,-1,0,1,2,3\}$

 $19 \bmod 7$ ist 5 in positiver Darstellung -2 in symmetrischer Darstellung.

- $-8 \bmod 7$ ist 6 in positiver Darstellung -1 in symmetrischer Darstellung
- ► Addition, Multiplikation, Substraktion \rightsquigarrow über Repräsentanten.
- ▶ n = p Primzahl, so \mathbb{Z}_p Körper: d. h. Inversen EEA, $\mathsf{GGT}(m,p) = 1$, d. h. $s \cdot m + tp = 1$, $s \cdot m \equiv 1 \mod p$.
- ▶ e arith. Ausdruck Berechnung von e mod n. $a \equiv b \mod n \rightsquigarrow a * c \equiv b * c \mod n, * \in \{+, -, \cdot\}$

Modulare Arithmetik in R[x]

Arithmetik in Polynomringen - Modulare Arithmetik

Einfachste Moduli: x - u mit $u \in R$.

Ist $f \in R[x]$, so hat f(x) - f(u) u als Nullstelle, ist also durch x - u teilbar.

Setzt man q = (f(x) - f(u))/(x - u), so f = q(x - u) + f(u) und da f(u) konstant, ist sein Grad $< 1 = \operatorname{grad}(x - u)$ oder $f \equiv f(u) \mod (x - u)$. \leadsto Berechnung modulo x - u ist Auswertung in u.

Auswertungsmorphismus $R[x] \rightarrow R[x]/(x-u)$.

- Restklassenring nach Ideal.
- ightharpoonup Repräsentanten f rem m.



Prof. Dr. K. Madlener: Computeralgebra

1

Arithmetik in Polynomringen - Modulare Arithmetik

Modulare Arithmetik

Modulare Arithmetik in R[x] (Forts.)

4.7 Lemma Ist R euklidischer Bereich, a, $m \in R$, S = R/mR, $a \mod m \in S$ Einheit $\operatorname{gdw} \operatorname{GGT}(a, m) = 1$.

Modulare Inverse kann mit EEA berechnet werden.

4.8 Beispiel $R = \mathbb{Z}$, m = 29, a = 12, GGT(a, m) = 1.

EEA: $5 \cdot 29 + (-12) \cdot 12 = 1$, d. h. $(-12) \cdot 12 \equiv 17 \cdot 12 = 1$, d. h. 17 ist Inverse von 12 mod 29.

 $R = \mathbb{Q}[x], \ m = x^3 - x + 2, \ a = x^2.$ **EEA**: $\left(\frac{1}{4}x + \frac{1}{2}\right)(x^3 - x + 2) + \left(-\frac{1}{4}x^2 - \frac{1}{2}x + \frac{1}{4}\right)x^2 = 1$ d. h. $(-x^2 - 2x + 1)/4$ ist Inverse von $x^2 \mod x^3 - x + 2$.

Beachte: $S = \mathbb{Z}_p \ p$ Primzahl oder S = F[x]/(f), f irreduzibles Polynom $\rightsquigarrow S$ ist Körper.

Endliche Körper $\mathbb{F}_p[x]/(f)$ mit $q=p^n$ Element $(\operatorname{grad}(f)=n)$.



Prof. Dr. K. Madlener: Computeralgebra 110 Prof. Dr. K. Madlener: Computeralgebra 1

Modulare Arithmetik

Modulo irreduzible Polynome f

4.9 Lemma Sei F Körper, $f \in F[x]$ monisch, irreduzibel nicht konstant und K = F[x]/(f). Dann ist K eine Körpererweiterung von F und $f(\alpha) = 0$ für $\alpha = (x \mod f) \in K$.

Beweis: K ist Körper, da f irreduzibel, $F \subset K$, f nicht konstant.

$$f(\alpha) = f(x \bmod f) = (f(x) \bmod f) = 0$$

Prof. Dr. K. Madlener: Computeralgebra

Prof. Dr. K. Madlener: Computeralgebra

Arithmetik in Polynomringen - Modulare Arithmetik

Modulare Arithmetik

Modulo irreduzible Polynome f (Forts.)

4.10 Beispiel $R = \mathbb{F}_5[x], f = x^3 - x + 2, a = x^2.$

f hat keine Nullstellen in $R = \mathbb{F}_5$, ist somit irreduzibel, da vom Grad 3. $\rightsquigarrow F_{125} = F_5[x]/(f)$ ist Körper.

EEA
$$f$$
, a: $(-x-2)(x^3-x+2)+(x^2+2x-1)x^2=1$

d. h. $x^2 + 2x - 1$ ist Inverse von $x^2 \mod x^3 - x + 1$. Setzt man $\alpha = x \mod f$, so gilt

$$\alpha^2 + 2\alpha - 1 = (\alpha^2)^{-1}$$
 in F_{125}

▶ Kosten der Operationen in F[x]/(f): Addition, Multiplikation, Division. $O(n^2)$ Operationen in F.

Die Euler Funktion

Erinnerung:

Eulersche Funktion: $\varphi : \mathbb{N}^+ \to \mathbb{N}^+$

$$\begin{array}{lcl} \varphi(m) & = & |\{0 \leq a \leq m : \mathsf{GGT}(a,m) = 1\}| \\ & = & |Z_m^{\times}| \end{array}$$

wobei Z_m^{\times} die Einheitengruppe von \mathbb{Z}_m ist.

$$\varphi(1) = 1, \ \varphi(p) = p - 1, \ \varphi(p^e) = p^e - p^{e-1} = (p-1)p^{e-1}.$$

4 D > 4 B > 4 B > 4 B > B 990

Prof. Dr. K. Madlener: Computeralgebra

Arithmetik in Polynomringen - Modulare Arithmetik

Schnelle Arithmetik

Schnelle Arithmetik

Wie schnell kann multipliziert/dividiert werden? Sind die Schranken für die Multiplikation bzw. Division von Langzahlen und Polynome, die wir abgeleitet haben, gut?

Multiplikation nach Karatsuba (1962 b = 2)

▶ Multiplikation von Langzahlen (oder Polynome).

Idee: Divide&Conquer Ansatz

$$u = (u_{2n-1} \cdots u_0)_b \quad v = (v_{2n-1} \cdots v_0)_b \quad \text{Basis } b$$

$$\boxed{\bar{u}_1 \quad \bar{u}_0} \quad \boxed{\bar{v}_1 \quad \bar{v}_0} \quad \text{d. h.}$$

$$u = \bar{u}_1 b^n + \bar{u}_0 \quad v = \bar{v}_1 b^n + \bar{v}_0 \quad \text{mit}$$

$$\bar{u}_1 = (u_{2n-1} \cdots u_n)_b \quad \bar{u}_0 = (u_{n-1} \cdots u_0)_b$$

Reduktion:

Multiplikation Zahlen Länge $2n \rightarrow$ Multiplikation Zahlen Länge n

Prof. Dr. K. Madlener: Computeralgebra

Schnelle Arithmetik (Forts.)

$$uv = (\bar{u}_{1}b^{n} + \bar{u}_{0})(\bar{v}_{1}b^{n} + \bar{v}_{0})$$

$$=^{*} \bar{u}_{1}\bar{v}_{1}b^{2n} + (\bar{u}_{1}\bar{v}_{0} + \bar{u}_{0}\bar{v}_{1})b^{n} + \bar{u}_{0}\bar{v}_{0}$$

$$= \bar{u}_{1}\bar{v}_{1}b^{2n} + [(\bar{u}_{1} - \bar{u}_{0})(\bar{v}_{0} - \bar{v}_{1}) + \bar{u}_{1}\bar{v}_{1} + \bar{u}_{0}\bar{v}_{0}]b^{n} + \bar{u}_{0}\bar{v}_{0}$$

$$=^{**} \bar{u}_{1}\bar{v}_{1}(b^{2n} + b^{n}) + (\bar{u}_{1} - \bar{u}_{0})(\bar{v}_{0} - \bar{v}_{1})b^{n} + \bar{u}_{0}\bar{v}_{0}(b^{n} + 1)$$

▶ Aufwand nach (*): T(1) = 1 $T(2n) = 4T(n) + c \cdot n$ für $n = 2^m : T(n) = T(2^m) = \hat{c}(2^m)^2 = \hat{c}n^2$

4□ → 4回 → 4 = → 4 = → 9 < (○)</p>

Prof. Dr. K. Madlener: Computeralgebra

117

Arithmetik in Polynomringen - Modulare Arithmetik

Schnelle Arithmetik

Schnelle Arithmetik (Forts.)

► Aufwand nach (**):

$$T(1) = 1 \quad T(2n) = 3T(n) + c \cdot n$$
für $n = 2^m$:

$$T(n) = 3(3T(2^{m-2}) + c \cdot 2^{m-2}) + c \cdot 2^{m-1}$$

$$= 3^m T(1) + c \cdot 2^{m-1} (1 + 3/2 + \dots + (3/2)^{m-1})$$

$$\approx \hat{c} \cdot 3^m = \hat{c} \ 2^{m \log_2 3} = \hat{c} \ n^{\log_2 3}$$

$$\approx \hat{c} \ n^{1.585}$$

Problem: Konstante größer als bei der Schulmethode, lohnt nur ab Zahlen (b=2) der Länge > 500.

Zusatzplatz für Zwischenergebnis. Implementierung "in place" Multiplikation.

Modulare Darstellung großer Zahlen

Grundlage: Chinesicher Reste Satz (CRT)

Sei R euklidischer Bereich $m_0,\ldots,m_n\in R$ paarweise teilerfremd (d. h. $\mathsf{GGT}(m_i,m_j)=1$ $i\neq j$) und sei $m=m_0\ldots m$

$$m=m_0\cdots m_n$$

$$\Phi_i$$
 : $R \to R/\langle m_i \rangle$ kan.Ring — Homomorphismus $a \to a \mod m_i$

$$\Phi = \Phi_0 \times \cdots \times \Phi_n : R \to R/\langle m_0 \rangle \times \cdots \times R/\langle m_n \rangle$$
$$a \to (a \bmod m_0, \dots, a \bmod m_n)$$

 Φ ist surjektiv mit Kern $\langle m \rangle$.

4 D > 4 B > 4 B > 4 B > 9 C

Prof. Dr. K. Madlener: Computeralgebra

Arithmetik in Polynomringen - Modulare Arithmetik

Schnelle Arithmetik

Modulare Darstellung großer Zahlen (Forts.)

Zeige:: Jede Zahl a < m kann eindeutig durch Liste $a = (a_0, \ldots, a_n)$ mit $a_i = a \mod m_i$ dargestellt werden.

Surjektivität: Behauptung:

Es gibt I_i mit $\Phi(I_i) = (0, ..., 0, 1, 0 ... 0)$

 $i = 0 : m_1 ... m_n = m/m_0 \text{ GGT}(m/m_0, m_0) = 1 \text{ EEA liefert } s, t \in R \text{ mit } s \cdot m/m_0 + tm_0 = 1 = \text{GGT}(m/m_0, m_0).$

Setze $l_0 = s \cdot m/m_0 \rightsquigarrow l_0 \equiv 0 \mod m_i$ $1 \le j \le n$

$$l_0 = s \frac{m}{m_0} \equiv s \frac{m}{m_0} + t m_0 = 1 \mod m_0$$

d. h.
$$\Phi(I_0) = (1, 0 \cdots 0)$$

Prof. Dr. K. Madlener: Computeralgebra

 \rightsquigarrow Algorithmus zur Berechnung von a bei Vorgabe a_0, \ldots, a_n (Lagrange, Garner).

Operationen via mod. Darstellungen

$$ightharpoonup a+b < m \leadsto a+b = (c_0,\ldots,c_n) \text{ mit } c_i = a_i+b_i \text{ mod } m_i$$

$$ightharpoonup a \cdot b < m \leadsto a \cdot b = (c_0, \dots, c_n) \text{ mit } c_i = a_i b_i \mod m_i.$$

 $a = (a_0, \dots, a_n) \qquad b = (b_0, \dots, a_n)$

Aufwand O(n) Operationen (Hier $a_i + b_i \mod m_i$, bzw. $a_i \cdot b_i \mod m_i$) als Elementaroperation.



a, b < m

Prof. Dr. K. Madlener: Computeralgebra

Prof. Dr. K. Madlener: Computeralgebra

12

Arithmetik in Polynomringen - Modulare Arithmetik

Schnelle Arithmetik

Operationen via mod. Darstellungen (Forts.)

Für K[x] K Körper geht dies genauso!

$$a_0,\ldots,a_n\in K$$
 verschiedene Elemente aus K .
$$m_i(x)=(x-a_i)\quad m(x)=m_0(x)\cdots m_n(x)$$

$$\Phi:K[x]/\langle m(x)\rangle\stackrel{\cong}{\to} K[x]/\langle m_0(x)\rangle\times\cdots\times K[x]/\langle m_n(x)\rangle$$

- ▶ Ist a(x) Polynom von Grad höchstens n, so gilt $a(x) \mod m(x) = a(x)$, $a(x) \mod m_i(x) = a(a_i)$
- ▶ $grad(a(x)) \le n$ $\Phi(a(x)) = (a(a_0), \dots, a(a_n))$

Operationen via mod. Darstellungen (Forts.)

Darstellung von Polynom a in Koeffizientendarstellung:

$$a(x) \leftarrow (a_0, \dots, a_n) \text{ mit } a(x) = \sum_{i=0}^n a_i x^i$$

Auswertungsdarstellung:

$$a(x) \leftarrow (\hat{a}_0, \dots, \hat{a}_n) \text{ mit } \hat{a}_i = a(a_i)$$

Multiplikation und Addition, Polynom O(n) Operationen Polynomgrad $\leq (n+1)/2$

$$a(x) \cdot b(x) \leftarrow (\hat{a}_0 \cdot \hat{b}_0, \dots, \hat{a}_n \cdot \hat{b}_n)$$

Wie sieht es mit der Division aus?

◆ロト ◆個ト ◆見ト ◆見ト ■ りゅう

Prof. Dr. K. Madlener: Computeralgebra

Arithmetik in Polynomringen - Modulare Arithmetik

Schnelle Arithmetik

4.11 Beispiel Lagrange Methode

a) $R=\mathbb{Z},\ m_i=p_i^{e_i},\ 0\leq i\leq n,\ p_i\in\mathbb{N}$ verschiedene Primzahlen, $e_i\in\mathbb{N}^+$ $m=\prod_{0\leq i\leq n}=p_i^{e_i}$ ist Primfaktorzerlegung von $m\in\mathbb{Z}.$

(CRT)
$$\mathbb{Z}/\langle m \rangle \cong \mathbb{Z}/\langle p_0^{e_0} \rangle \times \cdots \times \mathbb{Z}/\langle p_n^{e_n} \rangle$$

Für $a_0, \ldots, a_n \in \mathbb{Z}$ beliebig berechnet ein (CRA) eine Lösung $a \in \mathbb{Z}$ der Kongruenzen $a \equiv a_i \mod p_i^{e_i}$, $0 \le i \le n$

z. B.
$$n=1$$
, $m_0=11$, $m_1=13$, $m=11\cdot 13=143$, finde $a\in\mathbb{Z}$ mit $0\le a<143$ und $a\equiv 2\bmod 11$, $a\equiv 7\bmod 13$. Langrange Interpolanden.

EEA für
$$11, 13 : s_0 \cdot 13 + s_1 \cdot 11 = 6 \cdot 13 + (-7) \cdot 11 = 1$$
, d. h.

$$I_0, I_1: I_0 = 6 \cdot 13 = 78, I_1 = (-7) \cdot 11 = -77.$$

$$l_0 \equiv 1 \mod 11$$
, $l_0 \equiv 0 \mod 13$, $l_1 \equiv 0 \mod 11$, $l_1 \equiv 1 \mod 13$.

$$c_0 = a_0 s_0 \mod 11 = 2 \cdot 6 \mod 11 = 1, \ c_1 \equiv 7 \cdot (-7) \mod 13 = 3 \text{ und somit } a = c_0 \frac{m}{m_0} + c_1 \frac{m}{m_1} = 1 \cdot 13 + 3 \cdot 11 = 46 = 4 \cdot 11 + 2 = 3 \cdot 13 + 7.$$

Prof. Dr. K. Madlener: Computeralgebra

Beispiel (Forts.)

b) $R = F[x], m_i = x - a_i, 0 \le i \le n, a_0, ..., a_n \in F \quad (pv).$ $f \equiv f(a_i) \mod (x - a_i), 0 \le i \le n, f \rightarrow (f(a_0), ..., f(a_n))$

d. h. Auswertungshom. in a_0, \ldots, a_n , F^{n+1} koordinatenweise Operationen.

 $I_i \equiv 1 \bmod (x-a_i) \qquad I_i \equiv 0 \bmod (x-a_j), j \neq i$

 $grad(I_i) \le n$ sind die Lagrange Interpolanden

$$I_i = \prod_{\substack{0 \le j \le n \\ j \ne i}} \frac{x - a_j}{a_i - a_j}$$

Für $b_0, \ldots, b_r \in F$, so $f = \sum_{0 \le i \le n} b_i l_i$ Lagrange Interpolationspolynom mit $f(a_i) = b_i$ für 0 < i < n.

d. h. Chinesische Reste Algorithmus für n + 1 Lin Polynome ist Interpolation in n + 1 Werte. Polynom ist eindeutig: Grad n = n



Prof. Dr. K. Madlener: Computeralgebra

Prof. Dr. K. Madlener: Computeralgebra

125

Arithmetik in Polynomringen - Modulare Arithmetik

Die schnelle Fourier Transformation (FFT) Anwendung auf Polynommultiplikation

Die schnelle Fourier Transformation (FFT) Anwendung auf Polynommultiplikation

Koeffizientendarstellung $\overset{\text{Auswertung}}{\longleftrightarrow}$ Wertedarstellung.

Cooley, Tukey: An algorithm for machine calculation of complex fourier series, Math. Comp. 19 (1965) 297-301.

Idee: Fourier Transformierte: Reduktion auf einfachere Operationen

$$\mathsf{trans}(f * g) = \mathsf{trans}(f) \oplus \mathsf{trans}(g)$$

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

Um $a \cdot b$ zu berechnen: $\log(a)$, $\log(b) \leadsto \log(a) + \log(b)$ $\leadsto \operatorname{trans}^{-1}(\quad) = a \cdot b$.

Die allgemeine Fourier Transformation

Die schnelle Fourier Transformation (FFT) Anwendung auf Polynommultiplikation

Die Variablen t unf f stehen für Zeit und Frequenz

$$\mathcal{F}(\mathsf{a}) :: A(\mathsf{f}) = \int_{-\infty}^{\infty} \mathsf{a}(t) e^{2\pi \mathsf{i} \mathsf{f} t} \mathsf{d} t$$

$$\mathcal{F}^{-1}(A) :: a(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} A(f) e^{-2\pi i f t} df$$

◆ロト ◆母ト ◆皇ト ◆皇ト · 皇 · 夕९(

Prof. Dr. K. Madlener: Computeralgebra

12

Arithmetik in Polynomringen - Modulare Arithmetik

Die schnelle Fourier Transformation (FFT) Anwendung auf Polynommultiplikation

Diskrete Fourier Transformation

 a_0, \ldots, a_{n-1} reelle Zahlen, i komplexe Zahl mit $i^2 = -1$, seien

$$A_j := \sum_{k=0}^{n-1} a_k e^{2\pi i j k/n}$$
 $0 \le j < n$

$$a_k = \frac{1}{n} \sum_{j=0}^{n-1} A_j e^{-2\pi i j k/n}$$
 $0 \le k < n$

Interpretation: Auswertung eines Polynoms a(x) an n-Stellen.

D.h.::
$$a(x)=a_{n-1}x^{n-1}+a_{n-2}x^{n-2}+\cdots+a_1x+a_0$$
 $A_j=a(\omega^j)$, wobei $\omega=e^{2\pi i/n}$, $\omega^n=1$ n -te Einheitswurzel d.h.

$$\omega^{j} = e^{2\pi i j/n}, (\omega^{j})^{k} = e^{2\pi i j k/n}$$

Diskrete Fourier Transformation

Koeffizienten Darstellung zur modularen Darstellung (d. h. Wertedarstellung an speziellen Stellen)

$$x_0, \ldots, x_{n-1}$$
 (hier $x_j = \omega^j \omega$ nte-Einheitswurzel.)

$$T_{(x_0,...,x_{n-1})}(a_0,...,a_{n-1})=(\hat{a}_0,...,\hat{a}_{n-1}),$$
 wobei

$$\hat{a}_i = a_0 + a_1 x_i + \dots + a_{n-1} x_i^{n-1}.$$

Setzt man
$$a(x) = a_0 + a_1x + \cdots + a_{n-1}x^{n-1}$$

 \rightsquigarrow Auswertung von Polynomen vom Grad höchstens n-1 an den Stellen $\{x_0,\ldots,x_{n-1}\}.$

Auswertung eines Polynoms vom Grad n-1 (Horner's Regel) an einer Stelle kostet O(n) Operationen (in R). Übliche Kosten an n Stellen $\rightsquigarrow O(n^2)$.

Ziel: Reduktion dieser Kosten auf $O(n \log n)$ durch geeignete Wahl der Auswertungsstellen x_i : prim. E.W.



Prof. Dr. K. Madlener: Computeralgebra

Arithmetik in Polynomringen - Modulare Arithmetik

Die schnelle Fourier Transformation (FFT) Anwendung auf Polynommultiplikation

Diskrete Fourier Transformation: Die Auswertung

Angenommen n gerade, dann $a(x) = b(x^2) + x \cdot c(x^2)$, wobei $b(y) = a_0 + a_2 y + \dots + a_{n-2} y^{n/2-1}, c(y) = a_1 + a_3 y + \dots + a_{n-1} y^{n/2-1}.$

Hierbei haben b(y) und d(y) grad $\leq \operatorname{grad}(a(x))/2$

4.12 Lemma Sei $\{x_0, \dots, x_{n-1}\}$ Punktmenge in R, die die Symmetriebedingung

(*)
$$x_{(n/2)+i} = -x_i$$
, $i \in \{0, 1, \dots, n/2 - 1\}$ erfüllt.

Es gibt ein Auswertungsverfahren, so das für die Kosten T(n) für die Auswertung eines Polynoms vom Grad n-1 an dieser Punktmenge, gilt

$$T(1) = 0$$
 und $T(n) = 2 \cdot T\left(\frac{n}{2}\right) + c \cdot \frac{n}{2}$

für geeignete Konstante c.

Prof. Dr. K. Madlener: Computeralgebra

Die schnelle Fourier Transformation (FFT) Anwendung auf Polynommultiplikation

Diskrete Fourier Transformation: Die Auswertung

Beweis: Wegen (*) gilt $x_0^2 = x_{n/2}^2$, $x_1^2 = x_{n/2+1}^2 \cdots x_{n/2-1}^2 = x_{n-1}^2$, d. h. es gibt nur n/2 verschiedene Quadrate, d. h.

- ▶ a vom Grad höchstens n-1 kann an den Stellen $\{x_0, \ldots, x_{n-1}\}$ ausgewertet werden, durch Auswertung der Polynome b und c an den Stellen $\{x_0^2, \dots, x_{n/2-1}^2\}$, diese sind vom Grad höchstens $\frac{n}{2} - 1$.
- ▶ Hinzukommen n/2 Multiplikationen (Berechnung von x_i^2) und $\frac{n}{2}$ Multiplikationen, Additionen und Substraktionen, um die Werte zu kombinieren. → Behauptung.

Die schnelle Fourier Transformation verwendet dieses Lemma rekursiv. d. h. Symmetrie-Eigenschaft muss für die n/2 Punkte gelten usw.

4 D > 4 B > 4 B > 4 B > B = 900

Arithmetik in Polynomringen - Modulare Arithmetik

Prof. Dr. K. Madlener: Computeralgebra

Die schnelle Fourier Transformation (FFT) Anwendung auf Polynommultiplikation

Symmetriebedingung: Primitive Einheitswurzeln

4.13 **Definition** Primitive Einheitswurzeln Sei R kommutativer Ring, $\omega \in R$ ist prim. n-te EW gdw

- 1. $\omega^{n} = 1$
- 2. $\omega^i \neq 1$ für 0 < i < n (insb. $\omega \neq 1$)

3.
$$\sum_{j=0}^{n-1} \omega^{jp} = 0$$
 für $1 \le p < n$

Die Menge $\{1, \omega, \omega^2, \dots, \omega^{n-1}\}$ ist die Menge der Fourier Punkte zur n-ten EW ω .

Prof. Dr. K. Madlener: Computeralgebra

Diskrete Fourier Transformation bzgl. Primitiven

Einheitswurzeln

Voraussetzung:

n besitze eine multiplikative Inverse in R (z. B. wenn R Körper)

Seien

$$A = (A_{ij})_{n \times n}$$
 mit $A_{ij} = \omega^{ij}$, $0 \le i, j \le n$, $\mathfrak{a} = [a_0, \dots, a_{n-1}]^T$, dann

$$F(\mathfrak{a}) := A\mathfrak{a}$$
, wobei $F(\mathfrak{a})_i = \sum_{k=0}^{n-1} a_k \omega^{ik}$

heißt diskrete Fourier Transformierte von \mathfrak{a} (bzgl. ω).

◆□▶ ◆昼▶ ◆星▶ ◆星 ◆ 夕久で

Prof. Dr. K. Madlener: Computeralgebra

133

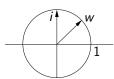
Arithmetik in Polynomringen - Modulare Arithmetik

Beispiel

4.14 Beispiel

a) Sei $R=\mathbb{C}$ komplexe Zahlen, n=8 $\omega=\mathrm{e}^{2\pi i/8}=\mathrm{e}^{\pi i/4}=\frac{(1+i)}{\sqrt{2}} \text{ ist primitive 8-EW}.$

$$\omega^2=e^{\pi i/2}=i$$
 erfüllt auch $(\omega^2)^8=1$, aber $(\omega^2)^4=1$, d. h. ω^2 ist 8-Wurzel von 1, aber nicht primitiv.



$$\sum_{i=0}^{8-1} e^{\pi i j/4} = 1 + e^{\pi i/4} + e^{\pi i/2} + e^{3\pi i/4} + \dots + e^{7\pi i/4} = 0$$

(heben sich paarweise auf!)

Beispiel (Forts.)

b) $R = \mathbb{Z}_{17}$, n = 4. 4 ist eine 4 EW, da $4^4 = 256 \equiv 1 \mod 17$. Sie ist auch primitiv, da $4^2 = 16$ und $4^3 = 13$ und $\sum_{j=0}^3 4^j = 0$ (\mathbb{Z}_{17}). Fourier Punkte: $\{1, 4, 4^2, 4^3\} = \{1, 4, 16, 13\}$

Diskrete Fourier Transformation:

Die schnelle Fourier Transformation (FFT) Anwendung auf Polynommultiplikation

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 4 & 16 & 13 \\ 1 & 16 & 1 & 16 \\ 1 & 13 & 16 & 4 \end{bmatrix} : (\mathbb{Z}_{17})^4 \to (\mathbb{Z}_{17})^4$$

◆ロト ◆部ト ◆注ト ◆注 → りへ()

Prof. Dr. K. Madlener: Computeralgebra

125

Arithmetik in Polynomringen - Modulare Arithmetik

Die schnelle Fourier Transformation (FFT) Anwendung auf Polynommultiplikation

Symmetriebedingung für PEW

4.15 Lemma

Ist ω eine primitive n-te EW, so erfüllen die n Fourier Punkte die Symmetriebedingung *

Beweis:

Da ω primitive n-te EW ist, gilt

$$(\omega^{n/2+j})^2 = \omega^n(\omega^j)^2 = (\omega^j)^2$$
, d. h.

$$(\omega^{n/2+j} - \omega^j)(\omega^{n/2+j} + \omega^j) = ((\omega^{n/2+j})^2 - (\omega^j)^2) = 0$$

Ist
$$\omega^{n/2+j} - \omega^j = 0$$
 so $\omega^{n/2} = 1$ 4.

Also
$$\omega^{n/2+j} + \omega^j = 0$$
, d. h. $\omega^{n/2+j} = -\omega^j$.

(R muss wohl Integerbereich sein?).

Symmetriebedingung: Rekursiv

4.16 Lemma

Sei ω primitive n-te EW, n gerade. Dann

- a) ω^2 ist primitive n/2-EW.
- b) Die n/2 Quadrate $\{1, \omega^2, \omega^4, \dots, \omega^n\}$ erfüllen die Symmetriebedingung *.

Beweis:

Wegen $(\omega^2)^{n/2} = \omega^n = 1$ ist $\omega^2 n/2$ -EW. Sie ist auch primitiv, da für k < n/2 mit $(\omega^2)^k = 1$ folgt $\omega^{2k} = 1$ mit 2k < n.

Die zweite Behauptung folgt aus vorherigem Lemma.

Grundlage für die rekursive Auswertung der Fouriertransformation ist für primitive 2^m -EW gegeben.



Prof. Dr. K. Madlener: Computeralgebra

137

Arithmetik in Polynomringen - Modulare Arithmetik

Die schnelle Fourier Transformation (FFT) Anwendung auf Polynommultiolikation

Beispiel: PEW

4.17 Beispiel \mathbb{Z}_{41} , n=8 symmetrische Darstellung von \mathbb{Z}_{41}

14 primitive 8-EW mit Fourier Punkte $\{1,14,-9,-3,-1,-14,9,3\}$

 $14^2 = -9$ ist primitive 4-EW mit Fourier Punkte $\{1, -9, -1, 9\}$.

 $(-9)^2 = -1$ ist primitive 2-EW mit Fourier Punkte $\{1, -1\}$.

Sei
$$a(x) = 5x^6 + x^5 + 3x^3 + x^2 - 4x + 1 \in \mathbb{Z}_{41}[x].$$

$$a(x) = b(y) + xc(y)$$
 für $y = x^2$ und $b(y) = 5y^3 + y + 1$,

 $c(y) = y^2 + 3y - 4.$

Auswertung von a(x) an den 8 Punkten $\{1, 14, -9, -3, -1, -14, 9, 3\}$:

Werte b(y) und c(y) an $\{1, -9, -1, -9\}$

b(y) = d(z) + ye(z), wobei $z = y^2$, d(z) = 1, e(z) = 5z + 1.

Werte d(z) und e(z) an $\{1, -1\}$.

 $d(1) = 1, e(1) = 6 \Rightarrow b(1) = 7, b(-1) = -5.$

 $d(-1) = 1, e(-1) = -4 \rightsquigarrow b(-9) = -4, b(9) = 6.$

Analog c(1) = 0, c(-1) = -2, c(-9) = 9, c(9) = -19 und

a(3) = b(9) + 3c(9) = -10 und a(-3) = b(9) - 3c(9) = -19. Ergebnis: $A \leftarrow \mathsf{FFT}(8, 14, a(x)) = (7, -1, 8, -19, 7, -7, -18, -10)$

Prof. Dr. K. Madlener: Computeralgebra 138 Prof. Dr. K. Madlener: Computeralgebra

Occosion Company Compa

Schnelle Fourier Transformation (FFT)

```
procedure FFT(N, \omega, a(x))
begin
                          {N Potenz von 2, \omega primitive n-te EW, a(x) Polynom}
                {mit Grad (a(x)) < N - 1. Ausgabe N Komponenten der FFT}
if N=1 then
    A_0 := a_0
else
    begin
    b(x) := \sum_{i=0}^{N/2-1} a_{2i} \cdot x^{i}; c(x) := \sum_{i=0}^{N/2-1} a_{2i+1} \cdot x^{i};
    B := FFT(N/2, \omega^2, b(x)); C := FFT(N/2, \omega^2, c(x));
    for i from 0 to N/2 - 1 do
         begin
         A_i := B_i + \omega^i C_i; A_{N/2+i} := B_i - \omega^i C_i;
    end
return ((A_0, A_1, \dots, A_{N-1})):
end
```

Prof. Dr. K. Madlener: Computeralgebra

4 D > 4 D > 4 D > 4 D > 3 D 9 O

Arithmetik in Polynomringen - Modulare Arithmetik

Die schnelle Fourier Transformation (FFT) Anwendung auf Polynommultiplikation

Schnelle Fourier Transformation (FFT): Analyse

Aufwand: $O(n \log n)$ Operationen: Sei $n = 2^m$ dann T(1) = 0. $T(2^k) = 2T(2^{k-1}) + c2^{k-1}$ k > 1

$$T(n) = T(2^m) = 2T(2^{m-1}) + c2^{m-1} = 2^2T(2^{m-2}) + c2^{m-1}2$$

= $2^3T(2^{m-3}) + c2^{m-1}3 \cdots = 2^mT(1) + c2^{m-1}m$

$$= c2^{m-1}m = c\frac{n}{2}\log n$$

Beachte: Eignet sich gut für Parallelisierung: Rekurrenzgleichung:

$$T^{p}(2^{k}) = T^{p}(2^{k-1}) + c2^{k-1}$$

(FFT):: Ergebniss

Modulare Darstellung eines Polynoms vom Grad N-1 an N-Fourierpunkte kann mit $O(N \log N)$ Grundoperationen in R(K)berechnet werden.

$$R[x] \atop {\sf grad} \leq N-1 \rightarrow R[x]/(x-\omega^0) \times \cdots \times R[x]/(x-\omega^{N-1})$$

Wie sieht es mit der Umkehrung aus

$$T_{(x_0,...,x_{N-1})} \leftrightarrow V(x_0,...,x_{N-1}) = \begin{vmatrix} 1 & x_0 \cdots (x_0)^{N-1} \\ 1 & x_1 \cdots (x_1)^{N-1} \\ \vdots & \vdots & \vdots \\ 1 & x_{N-1} \cdots (x_{N-1})^{N-1} \end{vmatrix}$$

Vandermonde Matrix

Prof. Dr. K. Madlener: Computeralgebra

Arithmetik in Polynomringen - Modulare Arithmetik

Die schnelle Fourier Transformation (FFT) Anwendung auf Polynommultiplikation

Inverse für Vandermonde Matrix

Finde Inverse der Vandermonde Matrix: Gauss Elimination $O(N^3)$

Polynominterpolation:

Gegeben N Punkte $(q_0, \ldots, q_{N-1}) \in R$, finde Polynom vom Grad höchstens N-1 mit

$$\hat{a}_i = a(x_i) = q_i \text{ für } i = 0, 1, \dots, N-1$$

Lagrange Interpolation oder Newton Interpolation Kosten $O(N^2)$ Operationen.

4 D > 4 B > 4 B > 4 B > 900

Die schnelle Fourier Transformation (FFT) Anwendung auf Polynommultiplikation

Inverse Fourier Transformation

4.18 Definition

Die Inverse diskrete Fourier Transformation (IDFT) für eine Menge Fourier Punkte ist definiert durch

$$S_{(1,\omega\cdots\omega^{N-1})}(q_0,\ldots,q_{N-1})=(\bar{q}_0,\ldots,\bar{q}_{N-1})$$

wobei

$$\bar{q}_j = N^{-1} \sum_{k=0}^{N-1} q_k (\omega^{-j})^k$$

Hierbei ist ω primitive n-te EW.

4.19 Satz DFT und IDFT sind inverse Transformationen. d. h.

$$T_{(1,\omega,...,\omega^{N-1})}$$
 $S_{(1,\omega,...,\omega^{N-1})} = ID$, $S_{(1,\omega,...,\omega^{N-1})}$ $T_{(1,\omega,...,\omega^{N-1})} = ID$

Prof. Dr. K. Madlener: Computeralgebra

Arithmetik in Polynomringen - Modulare Arithmetik

Die schnelle Fourier Transformation (FFT) Anwendung auf Polynommultiplikation

Inverse Fourier Transformation

Beweis: Sei 0 . Dann

 $(\omega^p)^N = (\omega^N)^p = 1$ und $(\omega^p) \neq 1$. Da ω PEW

 $(x^{N}-1) = (x-1)(x^{N-1} + x^{N-2} + \dots + x + 1)$, d. h. ω^{p} ist Nullstelle von $x^{N-1} + ... + x + 1 \rightarrow 0 = (\omega^p)^{N-1} + (\omega^p)^{N-2} + \cdots + (\omega^p) + 1$.

Für $0 und <math>-N (Mult. <math>\omega^{-p(N-1)}$).

Für p = 0 ist der Ausdruck N.

Sei $T_{(1,\omega,\ldots,\omega^{N-1})}(a_0,\ldots,a_{N-1})=(\hat{a}_0,\ldots,\hat{a}_{N-1})$ mit

 $\hat{a}_i = \sum_{i=0} a_j(\omega^i)^j$, $i = 0, \dots, N-1$

 $N^{-1} \sum_{i=0}^{N-1} \hat{a}_i \omega^{-ki} = N^{-1} \sum_{i=0}^{N-1} \sum_{i=0}^{N-1} a_j \omega^{ij} \omega^{-ki}$

 $= N^{-1} \sum_{i=0}^{N-1} a_j \sum_{i=0}^{N-1} \omega^{ij} \omega^{-ki} = N^{-1} \sum_{i=0}^{N-1} a_j \left(\sum_{i=0}^{N-1} \omega^{(j-k)i} \right) = a_k$

Prof. Dr. K. Madlener: Computeralgebra Prof. Dr. K. Madlener: Computeralgebra

Inverse Fourier Transformation

$$V(1,\omega,\ldots,\omega^{N-1})^{-1} = N^{-1} \begin{vmatrix} 1 & 1 \cdots 1 \\ 1 & \omega^{-1} \cdots \omega^{-(N-1)} \\ \vdots \\ 1 & \omega^{-(N-1)} \cdots \omega^{-(N-1)^2} \end{vmatrix}$$
$$= N^{-1}V(1,\omega^{-1},\ldots,\omega^{-(N-1)})$$

<ロ > → □ > → □ > → □ > → □ → ○ へ ○

Prof. Dr. K. Madlener: Computeralgebra

145

Arithmetik in Polynomringen - Modulare Arithmetik

Die schnelle Fourier Transformation (FFT) Anwendung auf Polynommultiplikation

Beispiel

4.20 Beispiel In \mathbb{Z}_{17} , $\omega=4$ primitive 4-te EW. Inverse $S_{(1,4,16,13)}:(\mathbb{Z}_{17})^4\to(\mathbb{Z}_{17})^4$

$$(4^{-1}) \begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & 13 & 16 & 4 \\ 1 & 16 & 1 & 16 \\ 1 & 4 & 16 & 4 \end{vmatrix} = \begin{vmatrix} 13 & 13 & 13 & 13 \\ 13 & 16 & 4 & 1 \\ 13 & 4 & 13 & 4 \\ 13 & 1 & 4 & 16 \end{vmatrix}$$

Sowohl T als auch S sind Fourier Transformationen.

- $T_{(1,\omega,\ldots,\omega^{N-1})}\vec{p} = \vec{q}$
- $ightharpoonup N^{-1} T_{(1,\omega^{-1},...,\omega^{-(N-1)})} \vec{q} = \vec{p}$. Da ω^{-1} primitive n-te EW.

Die inverse Fourier Transformation kann mit $O(N \log N)$ Operationen berechnet werden.

(□) (□) (□) (□) (□)

Die schnelle Fourier Transformation (FFT) Anwendung auf Polynommultiplikation

Schnelle Polynommultiplikation

$$a(x)$$
 grad m $a(x) \cdot b(x)$ grad $m + n$ $b(x)$ grad n

- Sei $N = 2^k > m + n$ ω prim. n-te EW
- $T_{(1,\omega,\ldots,\omega^{N-1})}(a_0,\ldots,a_m,0,\ldots,0) = (\hat{a}_0,\ldots,\hat{a}_m,\ldots,\hat{a}_{N-1})$
- $T_{(1,\omega,\ldots,\omega^{N-1})}(b_0,\ldots,b_n,0,\ldots,0) = (\hat{b}_0,\ldots,\hat{b}_n,\ldots,\hat{b}_{N-1})$
- ► $a(x) \cdot b(x) = (c_0, c_1, \dots, c_{m+n})$
- $T_{(1,\omega,\ldots,\omega^{N-1})}(c_0,\ldots,c_{m+n},0,\ldots,0) = (\hat{a}_0\hat{b}_0,\ldots,\hat{a}_{N-1}\hat{b}_{N-1})$

∢ロト∢御と∢選と∢選と 選 め

147

Arithmetik in Polynomringen - Modulare Arithmetik

Prof. Dr. K. Madlener: Computeralgebra

Die schnelle Fourier Transformation (FFT) Anwendung auf Polynommultiplikation

Schnelle Fourier Polynommultiplikation

Prof. Dr. K. Madlener: Computeralgebra 146 Prof. Dr. K. Madlener: Computeralgebra 148

Beispiel

4.21 Beispiel

$$a(x) = 3x^3 + x^2 - 4x + 1$$

 $b(x) = x^3 + 2x^2 + 5x - 3$ (\mathbb{Z}_{41})
14 primitive 8-te EW (wie eben).

$$A = FFT(8, 14, a(x)) = (1, 9, -19, -18, 3, 16, 19, -3)$$

$$B = FFT(8, 14, b(x)) = (5, 5, 0, 14, -7, -6, -10, 16)$$

$$C = (5,4,0,-6,20,-14,15,-7)$$
$$= FFT(8,14,a(x)b(x))$$

$$c = 8^{-1}FFT(8,3,-7x^7+15x^6-14x^5+20x^4-6x^3+4x+5)$$

$$= (-3, 17, 20, -11, 13, 7, 3, 0)$$

$$c(x) = 3x^6 + 7x^5 + 13x^4 - 11x^3 + 20x^2 + 17x - 3$$

$$(\lambda) = 3\lambda + 1\lambda + 13\lambda - 11\lambda + 20\lambda + 11\lambda - 3$$

Arithmetik in Polynomringen - Modulare Arithmetik

Prof. Dr. K. Madlener: Computeralgebra

Die schnelle Fourier Transformation (FFT) Anwendung auf Polynommultiplikation

Berechnung primitiver n-ter EW

- ► $F = \mathbb{C}$ einfach $\omega = e^{2\pi i/n}$ z. B. $e^{\pi i/6} = (\sqrt{3} + i)/2$ ist primitive 12-te EW in \mathbb{C}
- $ightharpoonup \mathbb{Q}, \mathbb{R}, \mathbb{Z}_p$
 - **4.22** Satz \mathbb{Z}_p hat primitive n-te EW $\operatorname{gdw} n \mid p-1$.

Beweis: Ist w primitive n-te EW in \mathbb{Z}_p , so bildet die Menge der Fourier Punkte $\{1, \omega, \dots, \omega^{n-1}\}$ eine zyklische Untergruppe der multiplikativen Gruppe von \mathbb{Z}_p . Diese hat p-1 Elemente $\leadsto n \mid p-1$ (Lagrange).

Die multiplikative Gruppe endlicher Körper ist zyklisch. Sei α erzeugendes Element der multiplikativen Gruppe von $\mathbb{Z}_p: \mathbb{Z}_p^{\mathsf{x}} = \{1, \alpha, \alpha^2, \dots, \alpha^{p-2}\}$ mit $\alpha^{p-1} = 1$

Sei
$$n \mid p-1$$
. Setzt man $\omega = \alpha^{p-1/n}$, so gilt $\omega^n = \alpha^{p-1} = 1$, d. h. ω ist n-te EW. Für $0 < k < n$, gilt $(p-1) \cdot k/n < (p-1)$, $\omega^k = \alpha^{(p-1)k/n} \neq 1$, also ist ω primitive n-te EW.

Berechnung primitiver n-ter EW

Die schnelle Fourier Transformation (FFT) Anwendung auf Polynommultiplikation

4.23 Beispiel In \mathbb{Z}_{41} gilt $8 \mid (41-1)$, d. h. es gibt primitive 8-te EW in \mathbb{Z}_{41} , z. B. 14 ist primitive 8-te EW in \mathbb{Z}_{41} .

Wie bestimmt man eine primitive n-te EW, wenn $n \mid p-1$ testen ! oder finde erzeugende für \mathbb{Z}_p^{\times} .

Anwendung $\mathbf{n} = \mathbf{2}^r$ für Fourier-Transformation $2^r \mid p - 1$ oder $p = 2^r k + 1$ für ein k ungerade.

Solche Primzahlen heißen Fourier Primzahlen zu 2^r.

Vorteil: Es gibt viele primitive Elemente.

Hilfsatz: Seien $a, b \in \mathbb{Z}$ mit GGT(a, b) = 1. Die Anzahl der Primzahlen $\leq x$ der Form ak + b, k = 1, 2, ... ist in etwa

$$\frac{x}{\log x \cdot \Phi(a)} \ (\Phi \ \text{Euler Funktion}).$$

Da alle ungeraden Zahlen $< 2^r$ teilerfremd zu 2^r sind und dies die Hälfte der ganzen Zahlen ist, gilt $\Phi(2^r) = 2^{r-1}$, d. h. es gibt etwa $\frac{X}{\log X \cdot 2^{r-1}}$

Fourier Primzahlen $\leq x$.

Prof. Dr. K. Madlener: Computeralgebra

Arithmetik in Polynomringen - Modulare Arithmetik

Die schnelle Fourier Transformation (FFT) Anwendung auf Polynommultiplikation

Berechnung primitiver n-ter EW

4.24 Beispiel Sei $x=2^{31}$ SP-Zahlen 32 Bit Wörter. Für r=20 $\Rightarrow \frac{2^{31}}{\log(2^{31})\cdot\Phi(2^{19})}\approx 130$ Primzahlen der Form $2^e\cdot k+1$, $e\geq 20$ im Intervall $2^{20}<<2^{31}$.

Jede solche Fourier Primzahl kann zur Berechnung von FFT's der Größe 2^{20} verwendet werden.

4.25 Satz a ist erzeugendes Element für \mathbb{Z}_p^{\times} $\operatorname{gdw} a^{(p-1)/q} \neq 1 \mod p$ für jeden Primfaktor von p-1.

Beweis folgt aus Lagrange $(H \leq G \rightsquigarrow |H| \mid |G|)$

 \leadsto Probabilistischer Algorithmus um erzeugendes Element für \mathbb{Z}_p^{\times} : Faktorisiere p-1 (möglich für $p\approx 2^{31}, 2^{63}$?) vorprozess. Wähle zufällig $a\in\{2,\ldots,p-1\}$, berechne $a^{p-1/q}$ für alle Teiler q von p-1.

◆□▶ ◆□▶ ◆□▶ ◆□▶ □ ◆○○○

Beispiel

4.26 Beispiel

Wegen $41 - 1 = 40 = 2^3 \cdot 5$, Primfaktoren 2,5, d. h. ein Element a erzeugt \mathbb{Z}_{41}^* , falls $a^8 \neq 1 \neq a^{20}$, z. B.

15:
$$15^8 = 18 \mod 41$$
 $15^{20} \equiv -1 \mod 41$
 $\neq 1$ $\neq 1$

Also ist 15 ein erzeugendes Element für \mathbb{Z}_{41}^{\times} , ist insbesondere eine primitive 40 EW in \mathbb{Z}_{41} , da $15^{40}=1 \mod 41$ und $\alpha^P \neq 1 \mod 41$ für 0 .

Die Anzahl der Erzeugenden für \mathbb{Z}_p^{\times} ist $\Phi(p-1)$, d. h. Anteil $\Phi(p-1)/(p-1) \approx 3/\pi^2$, 0.3 Wahrscheinlichkeit.



Prof. Dr. K. Madlener: Computeralgebra

153

Arithmetik in Polynomringen - Modulare Arithmetik

Occosion Occ

Beispiel (Forts.)

Prof. Dr. K. Madlener: Computeralgebra

 \mathbb{Z}_m N gegeben, bestimme m und ω , $N = 2^k$

▶ *N* invertierbar in $\mathbb{Z}_m \rightsquigarrow \mathsf{GGT}(N,m) = 1$.

$$a \in R, N = 2^k \leadsto \sum_{i=0}^{N-1} a^i = \prod_{i=0}^{k-1} \left(1 + a^{2^i}\right)$$
 Ind. nach k

$$= \left(1 + a\right) \sum_{i=0}^{N/2-1} \left(a^2\right)^i$$

▶ Sei $m = \omega^{N/2} + 1$ mit $\omega \in R$, $\omega \neq 0$. Dann

$$\sum_{i=0}^{N-1} \omega^{ip} \equiv 0 \mod m \text{ für } 1 \leq p < N$$

4□ > 4□ > 4□ > 4□ > 4□ > 4□

Beispiel (Forts.)

Beweis: Zeige $1+\omega^{2^jp}\equiv 0 \mod m$ für ein $j\ 0\leq j< k$. Sei $p=2^sp'$ mit p' ungerade, dann $0\leq s< k$. Wähle j mit $j+s=k-1 \leadsto 1+\omega^{2^jp}=1+\omega^{2^{k-1}p'}=1+(m-1)^{p'}$ wegen $(m-1)\equiv -1 \mod m,\ p'$ ungerade $\leadsto (m-1)^{p'}\equiv -1 \mod m,\ \leadsto$ Behauptung.

4.27 Satz

Seien n, ω positive Potenzen von 2 und $m = \omega^{n/2} + 1$, dann besitzt n Inverse in \mathbb{Z}_m und ω ist in \mathbb{Z}_m primitive n-te EW.

Beweis: $\omega \neq 1$ $\omega^n = \omega^{n/2} \cdot \omega^{n/2} \equiv (-1)(-1) \mod (\omega^{n/2} + 1)$.

Problem: primitive EW $R[x]/\langle x^n + 1 \rangle$

$$x^n \equiv -1 \mod (x^n + 1)$$
 $x^{2n} = (x^n)^2 \equiv 1 \mod (x^n + 1)$

 $\omega = (x \mod (x^n + 1))$ ist 2n-te EW.

(ロ) (部) (注) (注) (型)

Prof. Dr. K. Madlener: Computeralgebra

155

Arithmetik in Polynomringen - Modulare Arithmetik

Anwendung FFT auf Langzahlmultiplikation

Anwendung FFT auf Langzahlmultiplikation

Multiplikation nach Schönhage-Strassen:: div & conq + FFTA

Idee: Partitionierung der Zahlen in b-Blöcke der Länge l, d. h. $n=b\cdot l$, falls n Länge der Eingabezahlen.



Die b-Blöcke werden als Koeffizienten eines Polynoms (vom Grad b-1) mit Koeffizienten $<2^l$ aufgefasst.

Wertet man diese Polynome an geeigneten Stellen aus, multipliziert diese Werte und interpoliert, so lässt sich das Produkt bestimmen.

FFT + Faltungssätze.

Aufwand: $O(n \log n \log \log n)$ für die Multiplikation von Langzahlen der Länge n. Siehe von zur Gathen/Gerhard pp.225.

Prof. Dr. K. Madlener: Computeralgebra 156

Modulare Methoden

Modulare Algorithmen: Allgemeine Methoden

Anwendungsfälle

- ▶ FFT-Anwendung zur Multiplikation von Polynomen.
- ▶ GGT-Berechnung in $\mathbb{Z}_m \cong \mathbb{Z}_{m_0} \times \cdots \times \mathbb{Z}_{m_k}$ um Koeffizientenwachstum zu vermeiden.
- $\begin{array}{ccc} \blacktriangleright & \mathbb{Z}[x] & \to & \mathbb{Z}_p[x] \\ \text{nicht euklid} & \text{euklidisch} \end{array}$
- ► Faktorisierung, Wurzelberechnung,...

3 Varianten:

Big-Prime, Small-Primes, Prime-Power



4□ > 4問 > 4 = > 4 = > 9 Q ○

Prof. Dr. K. Madlener: Computeralgebra

157

Arithmetik in Polynomringen - Modulare Arithmetik

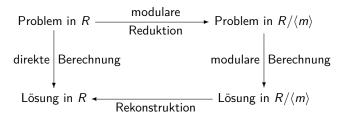
Modulare Methoden

Modulare Methoden: Big-Prime

Big-Prime:

R euklidisch

m = p



Benötigt werden:

- ► Schranke für die Lösung in *R*.
- ► Finde geeignete Moduli.

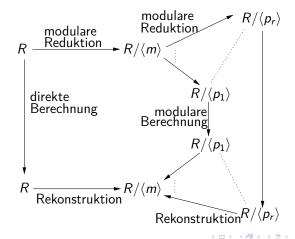
ulare Methoden

Modulare Methoden: Small-Primes

► Small Primes::

$$m = p_1 \dots p_r$$

$$p_i \neq p_i (i \neq j)$$



Prof. Dr. K. Madlener: Computeralgebra

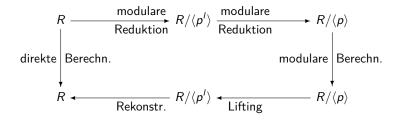
400

Arithmetik in Polynomringen - Modulare Arithmetik

Modulare Methode

Modulare Methoden: Prime Power

▶ **Prime-Power**:: $m = p^l$ p Primzahl



- ▶ Wahl der *p_i* steht frei, z. B. Fourier Primzahlen (schnelle Polynomarithmetik)
- ► Verteilung (Parallelisierung)

Prof. Dr. K. Madlener: Computeralgebra 158

Prof. Dr. K. Madlener: Computeralgebra

160

Chinesische Reste Algorithmen

Chinesische Reste Algorithmen

Die Algorithmen von Garner und Newton:

Umkehrung modularer- und Auswertungshomomorphismen.

4.28 Beispiel Wachstum der Zwischenergebnisse.

Systeme linearer Gleichungen. Gauss Methode::

n-Gleichungen, n unb, Koeffizientenlänge w.

 \sim Reduziertes System mit Koeffizienten $\approx 2^{n-1}w$ Länge



Prof. Dr. K. Madlener: Computeralgebra

Arithmetik in Polynomringen - Modulare Arithmetik

Chinesische Reste Algorithmen

Beispiel (Fort.)

Cramers Regel::

Prof. Dr. K. Madlener: Computeralgebra

$$x = \frac{\text{Det}[\]}{\text{Det}[\]}$$
 $y = \dots$ $z = \dots$

wobei Länge Det $] \approx n \cdot w$, d. h. Ergebnis (Länge) ist nicht Ursache der Komplexität.

Beachte Methode ist anwendbar auf lineare Gleichungssysteme mit Koeffizienten in Polynomringen. Dann tritt exponentielles Wachstum der Grade der Polynome (als Koeffizienten) auf.

Normierung durch Rechnung im Quotientenkörper. Kosten!

Ringmorphismen

Chinesische Reste Algorithmen

 $\Phi: R \to R'$ Homomorphismus, falls

i)
$$\Phi(a+b) = \Phi(a) + \Phi(b)$$
 $(a, b \in R)$

ii)
$$\Phi(ab) = \Phi(a)\Phi(b)$$
 $(a, b \in R)$

iii)
$$\Phi(1) = 1$$

iv)
$$(\Phi(0) = 0$$
 $\Phi(-a) = -\Phi(a)$

Prof. Dr. K. Madlener: Computeralgebra

Arithmetik in Polynomringen - Modulare Arithmetik

Chinesische Reste Algorithmer

Ringmorphismen: Beispiele

4.29 Beispiel

a) Modulare Homomorphismen: $m \in \mathbb{Z}$

$$\Phi_m \mathbb{Z}[x_1,\ldots,x_n] \to \mathbb{Z}_m[x_1,\ldots,x_n]$$

$$mit \ \Phi_m(x_i) = x_i \qquad \Phi_m(a) = (a \bmod m) \quad a \in \mathbb{Z}.$$

b) Auswertungshomomorphismen: $\alpha \in D$

$$\Phi_{x_i-\alpha}: D[x_1,\ldots,x_n] \to D[x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n]$$

$$\Phi_{\mathbf{x}_i-\alpha}(\mathbf{a}(\mathbf{x}_1,\ldots,\mathbf{x}_n))=\mathbf{a}(\mathbf{x}_1,\ldots,\mathbf{x}_{i-1},\alpha,\mathbf{x}_{i+1},\ldots,\mathbf{x}_n)$$

c) Komposition Homomorphismen:

$$\mathbb{Z}[x_1,\ldots,x_n] \xrightarrow{\Phi_p} \mathbb{Z}_p[x_1,\ldots,x_n] \xrightarrow[x_n-\alpha_n\in\mathbb{Z}_p]{} \cdots \xrightarrow[x_2-\alpha_2\in\mathbb{Z}_p]{} \mathbb{Z}_p[x_1]$$

Chinesische Reste Algorithmen

Ringmorphismen: Beispiel

$$a(x) = 7x + 5$$
 $b(x) = 2x - 3$ $c(x) = a(x)b(x)$

$$\varphi_1 := \Phi_{\mathsf{x}-\mathsf{0}}\Phi_\mathsf{5} : \mathbb{Z}[\mathsf{x}] \to \mathbb{Z}_\mathsf{5} \quad \varphi_1(\mathsf{a}) = \mathsf{0}$$

$$\varphi_1(b) = 2 \quad \varphi_1(c) = 0$$

$$\varphi_2 := \Phi_{\mathsf{x}-1}\Phi_{\mathsf{5}} : \mathbb{Z}[\mathsf{x}] \to \mathbb{Z}_{\mathsf{5}} \quad \varphi_2(\mathsf{a}) = 2$$

$$\varphi_2(b) = 4$$
 $\varphi_2(c) = 3$

$$c(0) \equiv 0 \mod 5$$
, $c(1) \equiv 3 \mod 5$, $c(2) \equiv 4 \mod 5$

Analog mit m = 7

$$c(0) \equiv 6 \mod 7$$
, $c(1) \equiv 2 \mod 7$, $c(2) \equiv 5 \mod 7$.

Rekonstruktion von c(x) in $\mathbb{Z}[x]$?



Prof. Dr. K. Madlener: Computeralgebra

165

Arithmetik in Polynomringen - Modulare Arithmetik

Chinesische Reste Algorithmen

Ideale und Homomorphismen (Erinnerung)

R kommutativer Ring $\emptyset \neq I \subset R$ ist Ideal, falls

- i) $a b \in I$ für $a, b \in I$.
- ii) $ar \in I$ für $a \in I, r \in R$.

Prof. Dr. K. Madlener: Computeralgebra

Insbesondere $0 \in I : \{0\}$ ist Ideal, R ist Ideal, I ist abgeschlossen bzgl. $+,\cdot,-$.

- ▶ $\Phi: R \to R'$ Homomorphismus Kern $(\Phi) = \{r \in R: \Phi(r) = 0\}$ ist Ideal.
- ▶ $\Phi_1: R \to R', \ \Phi_2: R \to R'', \ \operatorname{Kern}(\Phi_1) = \operatorname{Kern}(\Phi_2), \ \operatorname{so} \ \operatorname{ist} \Phi_1(R) \cong \Phi_2(R) \cong R/\operatorname{Kern}(\Phi_1)$

Ideale und Homomorphismen (Erinnerung)

- ▶ $I = \langle a_1, \dots, a_n \rangle = \{a_1r_1 + \dots + a_nr_n : r_i \in R\}$. Das von $\{a_1, \dots, a_n\}$ erzeugte Ideal.
- ▶ I heißt endlich erzeugt (ee), falls $I = \langle a_1, \dots, a_n \rangle$ Hauptideal, falls $I = \langle a \rangle$ $a \in R$.
- ▶ R noethersch, falls jedes Ideal ee. R Hauptidealring, falls jedes Ideal HI ist. (z. B. R euklidisch ~> R HIR). D noethersch, so ist D[x] noethersch.
- ► Summen, Produkte, Potenzen von Idealen

$$\langle a \rangle \cdot \langle b \rangle = \langle ab \rangle \quad \langle a \rangle^i = \langle a^i \rangle \quad i \ge 1$$

 $\langle a \rangle + \langle b \rangle \neq \langle a + b \rangle$
 $= \langle \mathsf{GCD}(a, b) \rangle \text{ in HIR.}$

Prof. Dr. K. Madlener: Computeralgebra

167

Arithmetik in Polynomringen - Modulare Arithmetik

Chinesische Reste Algorithme

Chinesischer Restesatz Algorithmen

R euklidisch, $m_0, \ldots, m_n \in R$ paarweise teilerfremd. $m = m_0, \ldots, m_n$ $\langle m \rangle = \langle m_0 \rangle \cdots \langle m_n \rangle$

$$\Phi_i := \Phi_{m_i} : R \to R/\langle m_i \rangle :: a \to a \mod m_i$$

 $\Phi: \Phi_0 \times \cdots \times \Phi_n: R \to R/\langle m_0 \rangle \times \cdots \times R/\langle m_n \rangle$ ist surjektiv mit Kern $\langle m \rangle$.

$$R/\langle m \rangle \cong R/\langle m_0 \rangle \times \cdots \times R/\langle m_n \rangle$$

Problem: Gegeben $(a_0, \dots, a_n) \in R/\langle m_0 \rangle \times \dots \times R/\langle m_n \rangle$. Berechne $a \in R/\langle m \rangle$ mit $\Phi_i(a) = a_i$.

Interpolationsproblem

1. Lösung Lagrange: I_i mit $\Phi_i(I_i) = 1$ sonst 0 $(i \neq i)$ über EEA.

Interpolationsproblem: Anwendungsfälle

2 Anwendungsfälle:

Rekonstruktion von a aus $\Phi_{m_i}(a), i = 0, \ldots, n$

- $ightharpoonup \mathbb{Z}$: Finde $\mathfrak{a} \in \mathbb{Z}$ mit $u < \mathfrak{a} < u + m$, $\mathfrak{a} \equiv \mathfrak{a}_i \mod m_i$ für festes u (z. B. u = 0 oder u = -m/2).
- $\blacktriangleright \mathbb{Z}[x_1,\ldots,x_{\nu}]:\Phi_{\rho}:\mathbb{Z}[x_1,\ldots,x_{\nu}]\to\mathbb{Z}_{\rho}[x_1,\ldots,x_{\nu}]$ $\Phi_{I}: \mathbb{Z}_{p}[x_{1}, \ldots, x_{\nu}] \to \mathbb{Z}_{p}[x_{1}] \text{ (oder } \mathbb{Z}_{p})$

I Komposition von Auswertungshomomorphismen $x_i - \alpha_i$, $\alpha_i \in \mathbb{Z}_p$.

 $\Phi_{x-\alpha_i}: D[x] \to D$ D Polynomring über \mathbb{Z}_p , $\alpha_i \in \mathbb{Z}_p$, $i = 0, \ldots, n$.

Es gibt ein eindeutiges Polynom $\mathfrak{a}(x) \in F_D[x]$ mit

$$\operatorname{grad}_{x}(\mathfrak{a}(x)) \leq n$$
 $\mathfrak{a}(x_{i}) = \mathfrak{a}_{i} \in D$ $0 \leq i \leq n$.



Prof. Dr. K. Madlener: Computeralgebra

Prof. Dr. K. Madlener: Computeralgebra

Arithmetik in Polynomringen - Modulare Arithmetik

Garner & Newton Interpolationsalgorithme

Garner & Newton Interpolationsalgorithmen

Mixed Radix Darstellung: $a \in R/\langle m \rangle$ lässt sich darstellen als

(*)
$$\mathfrak{a} = \nu_0 + \nu_1(m_0) + \nu_2(m_0m_1) + \cdots + \nu_n \left(\prod_{i=0}^{n-1} m_i\right)$$

wobei $\nu_k \in R/\langle m_k \rangle$ $k = 0, 1, \ldots, n$. **Beachte**: Ausdruck (*) muss richtig interpretiert werden.

- ▶ Summe und Produkte sind in $R/\langle m \rangle$ zu rechnen.
- ▶ Die $\nu_k \in R/\langle m_k \rangle \hookrightarrow R/\langle m \rangle$. Diese Einbettung ist möglich, da $\langle m \rangle \subset \langle m_k \rangle$ für $k = 0, 1, \ldots, n$, d. h. die Repräsentanten mod m_k können als Repräsentanten $\mod m$ gewählt werden.
- ▶ Existenz und Eindeutigkeit als Verallgemeinerung der Standarddarstellung einer Zahl zur basis β :: $a = \sum_{i=0}^{n} \nu_i \beta^i$ $0 \le a < \beta^{n+1}, 0 \le \nu_i < \beta$

Garner & Newton Interpolationsalgorithmen

- ▶ Fall \mathbb{Z} : \mathbb{Z}_{m_k} , \mathbb{Z}_m in positiver Darstellung oder \mathbb{Z}_{m_k} , \mathbb{Z}_m in symmetrischer Darstellung. Nicht gemischt!
- ▶ Fall D[x] D Polynomring über \mathbb{Z}_p , $\alpha_i \in \mathbb{Z}_p$ (D), $\nu_k \in F_D$, $0 \le k \le n$.

$$a(x) = \nu_0 + \nu_1(x - \alpha_0) + \nu_2(x - \alpha_0)(x - \alpha_1) + \dots + \nu_n \prod_{i=0}^{n-1} (x - \alpha_i)$$

Darstellung ist eindeutig unter diesen Bedingungen Existenz!

Jede Menge von Polynomen $m_k(x) \in F_D[x], k = 0, ..., n$ mit $grad(m_k(x)) = k$ ist Basis für Polynome mit grad < n über F_D .

Prof. Dr. K. Madlener: Computeralgebra

Arithmetik in Polynomringen - Modulare Arithmetik

Garner & Newton Interpolationsalgorithm

Newton Koeffizienten

Berechnung der ν_i , i = 0, ..., n, $\nu_i \in R/\langle m_k \rangle$

$$a = \nu_0 + \nu_1(m_0) + \nu_2(m_0m_1) + \cdots + \nu_n \left(\prod_{i=0}^{n-1} m_i\right)$$

$$\Phi_i(a) = a_i \qquad i = 0, \ldots, n \qquad a_i \in R/\langle m_i \rangle$$

- $a \equiv \nu_0 \mod m_0$ (oder $a(\alpha_0) = \nu_0 = a_0$), d. h. $\nu_0 = a_0$.
- ▶ Sind die Koeffizienten ν_0, \ldots, ν_{k-1} bestimmt, so folgt

$$a \equiv \nu_0 + \nu_1(m_0) + \cdots + \nu_k \left(\prod_{i=0}^{k-1} m_i \right) \pmod{m_k}$$

Prof. Dr. K. Madlener: Computeralgebra

Wähle ν_k , so dass

$$\nu_0 + \nu_1(m_0) + \dots + \nu_k \left(\prod_{i=0}^{k-1} m_i \right) \equiv a_k \pmod{m_k}$$

Da GGT
$$\left(\prod_{i=0}^{k-1} m_i, m_k\right) = 1$$
, ist $\prod_{i=0}^{k-1} m_i$ invertierbar $\mod m_k$ (Beachte im Polynomfall

$$a(\alpha_k) = \nu_0 + \nu_1(\alpha_k - \alpha_0) + \dots + \nu_k \prod_{i=0}^{k-1} (\alpha_k - \alpha_i) = a_k \in D$$

Da die
$$\alpha_i \in \mathbb{Z}_p, \alpha_i \neq \alpha_j, i \neq j$$
, folgt $\prod_{i=0}^{k-1} (\alpha_k - \alpha_i) \in \mathbb{Z}_p$ invertierbar).

Prof. Dr. K. Madlener: Computeralgebra

Arithmetik in Polynomringen - Modulare Arithmetik

Garner & Newton Interpolationsalgorithmen

Newton Koeffizienten (Forts.)

Also gilt

$$\nu_k \equiv \left[a_k - \left[\nu_0 + \nu_1(m_0) + \dots + \nu_{k-1} \left(\prod_{i=0}^{k-2} m_i \right) \right] \right] \cdot \left(\prod_{i=0}^{k-1} m_i \right)^{-1} \mod m_k$$

oder

$$\nu_k = \left[a_k - \left[\nu_0 + \nu_1(\alpha_k - \alpha_0) + \dots + \nu_{k-1} \left(\prod_{i=0}^{k-2} (\alpha_k - \alpha_i) \right) \right] \right] \cdot \left(\prod_{i=0}^{k-1} (\alpha_k - \alpha_i) \right)^{-1}$$

d. h. $\nu_k \in \mathbb{Z}/\langle m_i \rangle = \mathbb{Z}_{m_i}$ bzw. $\nu_k \in D$.

◆□ ト ◆□ ト ◆ ■ ト ◆ ■ ・ 夕 ○ ○

Garner's Algorithmus/Newton Interpol. Algorithmus Gemischte Basisdarstellung

```
procedure INTEGERCRA ((m_0,\ldots,m_n),(a_0,\ldots,a_n)) \{m_i\in\mathbb{Z},\operatorname{GGT}(m_i,m_j)=1\ (i\neq j),a_i\in\mathbb{Z}_{m_i}\} \{\operatorname{Ausgabe}\ a\in\mathbb{Z}_m\ \operatorname{mit}\ m=\Pi m_i\ a\equiv a_i\ \operatorname{mod}\ m_i,i=1,\ldots,n\} \{\operatorname{Schritt}\ 1:\operatorname{Berechne}\ \operatorname{die}\ \operatorname{ben\"{o}tigten}\ \operatorname{Inversen}\ \} \{\operatorname{Inverse}(a,q)=a^{-1}\ \operatorname{mod}\ q\} for k from 1 to n do begin \operatorname{product}:=\Phi_{m_k}(m_0); for i from 1 to k-1 do \operatorname{product}:=\Phi_{m_k}(\operatorname{product}\cdot m_i); \gamma_k:=\operatorname{inverse}(\operatorname{product},m_k); end
```

Prof. Dr. K. Madlener: Computeralgebra 175

Arithmetik in Polynomringen - Modulare Arithmetik

Compa & New York Designation of Company Compan

Garner's Algorithmus (Forts.)

```
  \left\{ \begin{array}{l} \text{Schritt 2: Berechne die } \left\{ \nu_k \right\} \right\} \\ \nu_0 := a_0; \\ \text{for } k \text{ from 1 to } n \text{ do} \\ \text{begin} \\ \text{temp} := \nu_{k-1}; \\ \text{for } j \text{ from } k-2 \text{ to 0 do} \\ \text{temp} := \Phi_{m_k}(\text{temp} \cdot m_j + \nu_j); \\ \nu_k := \Phi_{m_k}((a_k - \text{temp})\gamma_k); \\ \text{end} \\ \left\{ \begin{array}{l} \text{Schritt 3: Transformiere gemischte Radixdarstellung in Standard Darstellung} \\ a := \nu_n; \\ \text{for } k \text{ from } n-1 \text{ to 0 do} \\ a := am_k + \nu_k; \\ \text{return } (a) \end{array} \right.
```



Bemerkungen zu Garner's Algorithmus

Üblicherweise m_i SP-Zahlen oder $\alpha_i \in \mathbb{Z}_p$ mit p SP-Zahl.

Die Reste a_i sind im Fall \mathbb{Z} auch SPZ. a ist dann Langzahl (die Liste (a_0, \ldots, a_n) kann als Langzahldarstellung interpretiert werden). Bis auf Schritt 3 nur Operationen mit SPZ.

Beachte Schreibweise:

 $\Phi_{m_k}(Ausdruck) \equiv Werte-Ausdruck in <math>\mathbb{Z}_{m_k}(R/\langle m_k \rangle)$.

Alle Operationen werden mit SP-Zahlen bzw. \mathbb{Z}_p Zahlen gemacht.

Inverse mit EEA (in \mathbb{Z}).

$$\mathsf{GGT}(a,q) = 1 \leadsto \mathsf{sa} + \mathsf{tq} = 1, \ \Phi_q(\mathsf{s}) = (\mathsf{s} \bmod q) = \mathsf{inverse}(a,q).$$

Schritt 3: Operationen in \mathbb{Z} . Warum kommt Element aus \mathbb{Z}_m als Ergebnis heraus?



Prof. Dr. K. Madlener: Computeralgebra

Arithmetik in Polynomringen - Modulare Arithmetik

Garner & Newton Interpolationsalgorithme

Bemerkungen zu Garner's Algorithmus

Symmetrische Darstellung: $|\nu_k| \leq (m_k - 1)/2$

 $k = 0, \ldots, n$

$$|a| \leq \frac{m_0 - 1}{2} + \frac{m_1 - 1}{2}m_0 + \dots + \frac{m_n - 1}{2}\left(\prod_{i=0}^{n-1}m_i\right)$$

$$\leq rac{1}{2} \left[\left(\prod_{i=0}^n m_i
ight) - 1
ight]$$

Auch für $0 < \nu_k < m_k - 1$ $k = 0, \ldots, n$

$$a \leq \left(\prod_{i=0}^{n-1} m_i
ight) - 1$$

Berechnet wird

$$a = \nu_0 + m_0(\nu_1 + m_1(\nu_2 + \cdots + m_{n-2}(\nu_{n-1} + m_{n-1}(\nu_n))\cdots))$$

4□ > 4問 > 4 = > 4 = > 9 Q ○

Newton Interpolationsalgorithmus

- Im Fall D[x] sind die Homomorphismen Auswertungshomomorphismen an Stellen α_i d.h $\Phi_{\mathbf{x}-\alpha_i}:D[\mathbf{x}]\to D$ *D* Polynomring über \mathbb{Z}_p , $\alpha_i \in \mathbb{Z}_p$, $i = 0, \ldots, n$. Zu bestimmen ist eind. Polynom $a(x) \in F_D[x]$ mit $grad(a(x)) \le n$ mit $a(\alpha_i) = a_i \in D \ (0 < i < n).$
- ▶ Man beachte, dass in den Anwendungen die a; und somit die berechneten ν_i polynome mit Koeffizienten in \mathbb{Z}_p sind und bei der Bestimmung von ν_i nur Koeffizientenoperationen durchzuführen sind.
- \triangleright Beide Algorithmen sehen identisch aus. Im NIA steht an Stelle der m_i stets $(\alpha_k - \alpha_i)$ und für Φ_{m_k} steht stets Φ_p und die Inverse ist in \mathbb{Z}_p zu berechnen.
- ▶ In beiden Algorithmen haben die Objekte stets drei Darstellungen.

Prof. Dr. K. Madlener: Computeralgebra

Arithmetik in Polynomringen - Modulare Arithmetik

Garner & Newton Interpolationsalgorithm

Beispiel Garner's Algorithmus

4.30 Beispiel

Angenommen SP-Zahlen beschränkt -100 < a < 100 (2 Bit). Modulii: $m_0 = 99, m_1 = 97, m_2 = 95, m = m_0 m_1 m_2 = 919985$ Symmetrische konsistente Darstellung: -456142 < a < 456142

▶ Bestimme $a \in \mathbb{Z}_m$ mit $a \equiv 49 \mod 99 \equiv -21 \mod 97 \equiv -30 \mod 95$ $a_0 = 49, a_1 = -21, a_2 = -30.$

Garner:

Schritt 1:

$$\begin{split} \gamma_1 &= m_0^{-1} \mod m_1 = 99^{-1} \mod 97 = 2^{-1} \mod 97 = -48 \\ \gamma_2 &= (m_0 m_1)^{-1} \mod m_2 = 8^{-1} \mod 95 = 12 \end{split}$$

Schritt 2: Gemischte Basiskoeffizienten für a

$$\nu_0 = 49 \text{, } \nu_1 = -35 \text{, } \nu_2 = -28$$

$$a = 49 - 35(99) - 28(99)(97) = -272300$$

4 D > 4 D > 4 D > 4 D > 3 D 9 O

Prof. Dr. K. Madlener: Computeralgebra Prof. Dr. K. Madlener: Computeralgebra

Beispiel (Forts.)

4.31 Beispiel Eingangsproblem : System linearer Gleichungen.

Schwierigkeit: Muss keine Lösung in Z haben!

$$x_1 = \det \begin{vmatrix} 1 & 44 & 74 \\ -2 & 14 & -10 \\ 34 & -28 & 20 \end{vmatrix}$$
 $y_1 = \det \begin{vmatrix} 22 & 1 & 74 \\ 15 & -2 & 10 \\ -25 & 34 & 20 \end{vmatrix}$

$$z_1 = \det \begin{vmatrix} 22 & 44 & 1 \\ 15 & 14 & -2 \\ -25 & -28 & 34 \end{vmatrix} d = \det \begin{vmatrix} 22 & 44 & 74 \\ 15 & 14 & -10 \\ -25 & -28 & 20 \end{vmatrix}$$

$$x = x_1/d \quad y = y_1/d \quad z = z_1/d \in \mathbb{Q}$$

▶ In \mathbb{Z}_p berechne $x \mod p$, $y \mod p$, $z \mod p$, $d \mod p$ via Gauss \leadsto aus $x_1 \equiv xd \mod p$, $y_1 \equiv yd \mod p$, $z_1 \equiv zd \mod p$ $\underset{CRA}{\leadsto} x_1, y_1, z_1, d$ aus $\mathbb{Z} \leadsto \mathbb{Q}$ Lösung.



Garner & Newton Interpolationsalgorithmer

Modulare Darstellungen für x₁ und d

$$x_1 = (2, -5, -2, 5, 9), d = (-2, 1, 4, -2, -8)$$

 $m_0 = 7, m_1 = 11, m_2 = 13, m_3 = 17, m_4 = 19$

Garner
$$\rightsquigarrow x_1 = -44280,..., d = -7380$$

Vergleiche diese mit den Zahlen die über Gauss Elimination in $\ensuremath{\mathbb{Z}}$ auftreten!

$$\Rightarrow x = \frac{-44280}{-7380} = 6$$
 $y = \frac{40590}{-7380} = -\frac{11}{2} \ z = \frac{-11070}{-7380} = \frac{3}{2}$

Problem hier: Lösung ist nicht ganzzahlig, sondern in \mathbb{Q} . Rekonstruktion rationaler Lösungen bei Koeffizienten in \mathbb{Z} .

< ロ ト → 付 ト → 三 ト → 三 → り Q (~)

Prof. Dr. K. Madlener: Computeralgebra

181

Arithmetik in Polynomringen - Modulare Arithmetik

Garner & Newton Interpolationsalgorithme

Beispiel (Forts.)

In \mathbb{Z}_7 :

$$x + 2y - 3z = 1$$
 Gauss $x \equiv -1 \mod 7$
 $x - 3z = -2$ \Rightarrow $y \equiv -2 \mod 7$
 $3x - z = -1$ $z \equiv -2 \mod 7$
 $d \equiv -2 \mod 7$

In \mathbb{Z}_{11} , \mathbb{Z}_{13} , \mathbb{Z}_{17} , \mathbb{Z}_{19} liefert

$$x_1 \equiv -5 \mod 11$$
 $y_1 \equiv 0 \mod 11$ $z_1 \equiv -4 \mod 11$ $d \equiv 1 \mod 11$
 -2 4 6 $4 \mod 13$
 5 -6 -3 $-2 \mod 17$
 9 6 7 $-8 \mod 19$



◆ロト 4個ト 4 恵 ト 4 恵 ト . 重 . 夕久の

Arithmetik in Polynomringen - Modulare Arithmetik

Garner & Newton Interpolationsalgorithme

Newton Interpolationsalgorithmus

4.32 Beispiel Polynombeispiel

Bestimme Polynom $a(x,y) \in \mathbb{Z}_{97}[x,y]$ vom Max. Grad 2 in x und Max.

Grad 1 in y mit

$$a(0,0) = -21$$
 $a(0,1) = -30$
 $a(1,0) = 20$ $a(1,1) = 17$

$$a(2,0) = -36$$
 $a(2,1) = -31$

▶ Rekonstruktion von a(x, y) in $\mathbb{Z}_{97}[x, y]/\langle x - 0 \rangle$

$$D = \mathbb{Z}_{97}$$
, $\alpha_0 = 0$, $\alpha_1 = 1$, $a_0 = -21$, $a_1 = -30$

Berechne Polynom $a(0, y) \in \mathbb{Z}_{97}[y]$:

Schritt 1:

$$\gamma_1 = (\alpha_1 - \alpha_0)^{-1} \mod 97 = 1^{-1} \mod 97 = 1.$$

Schritt 2:

Newton Koeff:
$$\nu_0 = -21, \nu_1 = -9$$

Prof. Dr. K. Madlener: Computeralgebra 182 Prof. Dr. K. Madlener: Computeralgebra 184

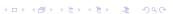
Garner & Newton Interpolationsalgorithmer

Polynombeispiel (Forts.)

Schritt 3:

$$a(0, y) = -21 - 9(y - 0) = -9y - 21$$

- ▶ Analog: $\mathbb{Z}_{97}[x,y]/\langle x-1\rangle$ und $\mathbb{Z}_{97}[x,y]/\langle x-2\rangle$ liefert a(1,y)=-3y+20 a(2,y)=5y-36
- Multivariater Schritt: Garner mit $D = \mathbb{Z}_{97}[y]$ $\alpha_0 = 0$, $\alpha_1 = 1$, $\alpha_2 = 2$, $a_0 = a(0, y)$, $a_1 = a(1, y)$, $a_3 = a(2, y)$
- Gesucht $a(x,y) \in D[x] = \mathbb{Z}_{97}[y][x]$.



Prof. Dr. K. Madlener: Computeralgebra

185

Arithmetik in Polynomringen - Modulare Arithmetik

Garner & Newton Interpolationsalgorithmen

Polynombeispiel: Berechnung von a(x, y)

▶ Schritt 1:

$$\gamma_1 = 1, \gamma_2 = [(\alpha_2 - \alpha_0)(\alpha_2 - \alpha_1)]^{-1} \mod 97 = -48$$

► Schritt 2:

$$\nu_0 = -9v - 21$$
, $\nu_1 = 6v + 41$, $\nu_2 = v$

► Schritt 3:

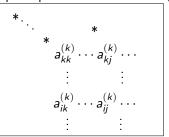
$$a(x,y) = (-9y - 21) + (6y + 41)(x - 0) + y(x - 0)(x - 1)$$

Beispiel: Modulare Determinantenberechnung

4.33 Beispiel Modulare Determinantenberechnung (vzGG S101)

Sei $A = (a_{ij})_{1 \le i,j \le n} \in \mathbb{Z}^{n \times n}$. Berechne det A

Gauss Elimination über \mathbb{Q} , $2n^3$ Operationen in \mathbb{Q} . Ist dies "pol Zeit"? Die Anzahl der Wortoperationen hängt von den Zähler und Nenner der Zwischenergebnissen ab. Wie ist ihr Wachstum? Betrachte Stufe k bei der Elimination. A nichtsingulär und keine Zeilen oder Spaltenpermutationen notwendig.



 $\begin{array}{l} a_{kk}^{(k)} \neq 0 \text{ für} \\ k < i \leq n, \ k \leq j \leq n \\ a_{ij}^{(k+1)} = a_{ij}^{(k)} - \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} a_{kj}^{(k)}. \\ \text{Sei } b_k \text{ obere Schranke für Zähler} \\ \text{und Nenner der } a_{ij}^{(k)} \ (1 \leq i,j \leq n). \end{array}$

k-1 Pivoting-Schritte

Prof. Dr. K. Madlener: Computeralgebra

10

Arithmetik in Polynomringen - Modulare Arithmetik

Garner & Newton Interpolationsalgorithme

Beispiel: Modulare Determinantenberechnung (Forts.)

Insbesondere: $|a_{ij}| \le b_0$ für $1 \le i, j \le n$. Es folgt

$$b_k \leq 2b_{k-1}^4 \leq 4b_{k-2}^{4^2} \leq \cdots \leq 2^k b_0^{4^k}$$

d. h. exponentiell in der Länge der Eingabe $n^2\lambda(b_0)\approx n^2\log b_0$

Ist Gauss Elimination überhaupt polynomial in Eingabelänge? Ja, aber nichttrivialer Beweis.

2 Alternativen: Big-Prime, Small-Primes

Garner & Newton Interpolationsalgorithme

Modulare Determinantenberechnung (Forts.)

Sei $d = \det A$. Wähle Primzahl p > 2|d|. Wende Gauss Elimination auf $A \bmod p \in \mathbb{Z}_p^{n \times n}$ an. Sei r Ergebnis in symmetrischer Darstellung von \mathbb{Z}_p , d. h. $r \equiv d \mod p - \frac{p}{2} < r < \frac{p}{2}$. Da $p \mid d - r$ und $|d - r| \le |d| + |r| \le \frac{p}{2} + \frac{p}{2} = p$ folgt d = r. Schranken für det A: Hadamard Ungleichung

$$|\det A| \le n^{n/2}B^n \text{ mit } B = \max_{1 \le i,j \le n} |a_{ij}|$$

Wortlänge $\lambda(C) = \lambda(n^{n/2}B^n)$ ist $\frac{1}{64}\log_2 C = \frac{1}{64}n\left(\frac{1}{2}\log_2 n\right) + \log_2 B$ Polynomial in Eingabelänge $n^2 \lambda(B)$.

- ▶ Primzahl p zwischen 2C und 4C. Finden (prob. Algorithmus). Arithmetik modulo $p = O(\log^2 C)$ Wortoperationen.
- $ightharpoonup O(n^3n^2(\log n + \log B)^2)$ Wortoperationen.



Prof. Dr. K. Madlener: Computeralgebra

Garner & Newton Interpolationsalgorithme

Arithmetik in Polynomringen - Modulare Arithmetik

Modulare Determinantenberechnung (Forts.)

Small Primes:

$$C = n^{n/2}B^n$$
, $r = \lceil \log_2(2C+1) \rceil$.

- ▶ Wähle *r* verschiedene Primzahlen $m_0, \ldots, m_{r-1} \in \mathbb{N}$.
- ▶ Für 0 < i < r berechne $d_i \equiv \det A \mod m_i$ (Gauss) in \mathbb{Z}_{m} .
- ▶ Chinesischer R.A $d \equiv d_i \mod m_i$ 0 < i < r.

Dann det $A \equiv d \mod m_i$ und somit det $A \equiv d \mod m$ für $m = m_0 \dots m_{r-1}$.

Wegen $m > 2^r > 2^{n/2} nB^n > 2|d|$ gilt $d = \det A$.

Modulare Determinantenberechnung (Forts.)

Kosten: Berechnung der r Primzahlen (ersten r PZ),

- ▶ $O(r \log^2 r \log \log r)$ Wort-Operationen, $\log m_i \in O(\log r)$. $\log m = \sum_{0 \le i \le r} \log m_i \in O(r \log r).$
- ▶ Operationen $\operatorname{mod} m_i \leftrightarrow O(\log^2 m_i)$, d. h. $O(\log^2 r)$ Operationen. $O(n^3 r \log^2 r)$ Wortoperationen, $A \mod m_i \to O(n^2 r \log^2 r)$.
- ightharpoonup r Werte $O(n^2r^2\log^2r)$.

$$O(n^4 \log^2(nB)(\log^2 n + (\log \log B)^2))$$

Praxis: Vorberechnung von Primzahlen mit Wortlänge.

Prof. Dr. K. Madlener: Computeralgebra

Newton's Iteration und Hensel's Konstruktion

Inhalt Kapitel 5

Newton's Iteration und Hensel's Konstruktion

Motivation

P-adische und ideal-adische Darstellungen

Ideal-adische Darstellung und Approximation

Iteration nach Newton für F(u) = 0

Ideal-adische Newton Iteration

Hensel's Lemma

Hensel Lifting

Multifaktor Hensel Lifting. Algorithmus nach Zassenhaus

Multivariate Verallgemeinerung von Hensel's Lemma

Lösung diophantischer Polynomgleichungen in $\mathbb{Z}_{n'}[x_1]$

Newton's Iteration und Hensel's Konstruktion

- ▶ Umkehrung modularer & Auswertungs-Homomorphismen.
- Anwendung von Newton's Iterationsmethoden zur Lösung von Polynomgleichungen.

$$\mathbb{Z}[x_1,\ldots,x_{\nu}]\to\mathbb{Z}_{p}[x_1,\ldots,x_{\nu}]\to\mathbb{Z}_{p}[x_1]$$

- ▶ Im Gegensatz zur Interpolation, benötigt man nur einen Bildwert in $\mathbb{Z}_p[x_1]$.
- ▶ Problem bei Small-Primes Methode: Die Anzahl der Bildprobleme die gelöst werden müssen kann exponentiell in der Größe der Lösung wachsen.

<ロ > < 個 > < 国 > < 重 > く 重 > く 重 > り へ ②

Prof. Dr. K. Madlener: Computeralgebra

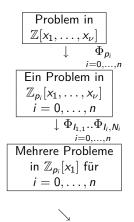
193

Motivation

löse Problem

in $\mathbb{Z}_{p_i}[x_1]$ $i=0,\ldots,n$

Problem bei Small-Primes Methode



Lösung in $\mathbb{Z}[x_1,\ldots,x_{\nu}]$ \uparrow CRA

Eine Bildlösung in $\mathbb{Z}_{p_i}[x_1,\ldots,x_{\nu}]$ $i=0,\ldots,n$ \uparrow Interpolation

mehrere Lösungen in $\mathbb{Z}_{p_i}[x_1]$ für $i=0,\ldots,n$

<□ > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□** > <**□**

P-adische und ideal-adische Darstellungen

Problem: Inversion von $\Phi_p : \mathbb{Z}[x] \to \mathbb{Z}_p[x]$. Startpunkt: Andere Darstellung für ganze Zahlen und Polynomen.

ightharpoonup p-ungerade Primzahl, $u \in \mathbb{Z}$, p-adische Darstellung

$$u = u_0 + u_1 p + u_2 p^2 + \cdots + u_n p^n$$

mit
$$p^{n+1} > 2|u|$$
 $u_i \in \mathbb{Z}_p$ $(0 \le i \le n)$.

Hierbei kann man entweder die positive oder die symmetrische Darstellung von \mathbb{Z}_p verwenden.

- ▶ Verfahren zur Bestimmung der *p*-adischen Darstellung:
 - $egin{aligned} \mathbf{v} & u \equiv u_0 \bmod p, \ \mathrm{d. h.} & (*) & u_0 = \Phi_p(u). \ u_1 : u u_0 & \mathrm{muss} \ \mathrm{durch} \ p \ \mathrm{teilbar} \ \mathrm{sein, \ d. h.} \ & \frac{u u_0}{p} = u_1 + u_2 p + \cdots + u_n p^{n-1} &
 ightharpoonup u_1 = \Phi_p\left(rac{u u_0}{p}
 ight) \end{aligned}$
 - Allgemein

$$(**) \quad u_i = \Phi_p\left(\frac{u - (u_0 + u_1p + u_2p^2 + \dots + u_{i-1}p^{i-1})}{p^i}\right) \quad i = 1, 2, \dots, n$$

Prof. Dr. K. Madlener: Computeralgebra

19

Newton's Iteration und Hensel's Konstruktion

P-adische und ideal-adische Darstellungen

Beispiel: p-adische Zahlendarstellung

5.1 Beispiel

$$u = -272300$$
 $p = 97$

$$u_0 = \Phi_p(u) = -21$$

$$u_1 = \Phi_p\left(\frac{u - u_0}{p}\right) = \Phi_p\left(\frac{-272279}{97}\right) = \Phi_p(-2807) = 6$$

$$u_2 = \Phi_p\left(\frac{u - [u_0 + u_1 p]}{p^2}\right) = -29$$

$$u_3 = 0$$

d.h.
$$-272300 = -21 + 6(97) - 29(97)^2$$

Verallgemeinerung für Polynomen

$$u(x) = \sum_{e} u_{e} x^{e} \in \mathbb{Z}[x]$$

Seien p und n so gewählt, dass $p^{n+1} > 2u_{\text{max}} = 2\max|u_e|$.

▶ Werden die u_e in ihrer p-adischen Darstellung $u_e = \sum_{i=0}^n u_{e,i} p^i$ mit $u_{e,i} \in \mathbb{Z}_p$ ausgedrückt, so

$$u(x) = \sum_{e} \left(\sum_{i=0}^{n} u_{e,i} p^{i} \right) x^{e} = \sum_{i=0}^{n} \underbrace{\left(\sum_{e} u_{e,i} x^{e} \right)}_{\in \mathbb{Z}_{p}[x]} p^{i}$$

p-adische Polynomdarstellung von u.

$$u(x) = u_0(x) + u_1(x)p + \cdots + u_n(x)p^n$$

mit $u_i(x) \in \mathbb{Z}_p[x]$ i = 0, ..., n. Verfahren: (*) (**) bleiben gültig.

Prof. Dr. K. Madlener: Computeralgebra

Newton's Iteration und Hensel's Konstruktion

P-adische und ideal-adische Darstellunge

Beispiel: p-adische Polynomdarstellung

5.2 Beispiel
$$u(x) = 14x^2 - 11x - 15 \in \mathbb{Z}[x]$$
 $p = 5$

$$u_0(x) = \Phi_p(u(x)) = -x^2 - x$$

•
$$u_1(x) = \Phi_p\left(\frac{u(x) - u_0(x)}{p}\right) = -2x^2 - 2x + 2$$

•
$$u_2(x) = \Phi_p\left(\frac{u(x) - [u_0(x) + u_1(x)p]}{p^2}\right) = x^2 - 1$$

Da
$$u_3(x) = 0$$

$$u(x) = (-x^2 - x) + (-2x^2 - 2x + 2)5 + (x^2 - 1)5^2$$

Approximation und p-adische Darstellung

$$a(x) \equiv b(x) \mod q(x) \text{ gdw } a(x) - b(x) \in \langle q(x) \rangle$$

$$u(x) \equiv u_0(x) \mod p$$
 und

$$u(x) \equiv u_0(x) + u_1(x)p + \cdots + u_{k-1}(x)p^{k-1} \mod p^k$$

$$ightharpoonup (p) > (p^2) > \cdots > (p^k) > \cdots$$
 Idealkette.

5.3 Definition Sei $a(x) \in \mathbb{Z}[x]$.

Ein Polynom $b(x) \in \mathbb{Z}[x]$ heißt eine p-adische Approximation n-ter Ordnung von a(x), falls

$$a(x) \equiv b(x) \bmod p^n$$

Der Fehler bei der Approximation von a(x) durch b(x) ist $a(x) - b(x) \in \mathbb{Z}[x].$

Prof. Dr. K. Madlener: Computeralgebra

Newton's Iteration und Hensel's Konstruktion 000**00000000000**000000000000

P-adische und ideal-adische Darstellunger

Multivariate Taylor-Reihendarstellung

Verallgemeinerung der p-adischen Darstellung

Sei

$$\Phi_I: \mathbb{Z}_p[x_1,\ldots,x_{\nu}] \to \mathbb{Z}_p[x_1]$$

mit Kern
$$I = \langle x_2 - \alpha_2, \dots, x_{\nu} - \alpha_{\nu} \rangle$$
, $\alpha_i \in \mathbb{Z}_p(2 \le i \le \nu)$.

Gesucht: Urbilder von Φ_L

Lösung: Geeignete Wahl der Darstellung einer Lösung

$$\tilde{u}=u(x_1,\ldots,x_{\nu})\in\mathbb{Z}_p[x_1,\ldots,x_{\nu}]$$
 mit vorgegebenen "ersten Term" $u^{(1)}\in\mathbb{Z}_p[x_1]$, wobei

$$u^{(1)} = \Phi_I(\tilde{u})$$

Beachte $u^{(1)} = u(x_1, \alpha_2, \dots, \alpha_{\nu})$

Multivariate Taylor-Reihendarstellung (Forts.)

 \blacktriangleright Korrespondierend zur p-adischen Darstellung, wähle für die Lösung \tilde{u} eine Darstellung

$$\tilde{u} = u^{(1)} + \Delta u^{(1)} + \Delta u^{(2)} + \Delta u^{(3)} + \cdots$$

um die restlichen Terme zu bestimmen, betrachte den Fehler

$$e^{(1)} = \tilde{u} - u^{(1)}$$

Es gilt $\Phi_I(e^{(1)}) = 0$, d.h. $e^{(1)} \in I$. Da I von $(x_i - \alpha_i)$ erzeugt wird gilt

(x)
$$e^{(1)} = \sum_{i=2}^{\nu} c_i(x_i - \alpha_i) \text{ mit } c_i \in \mathbb{Z}_p[x_1, \dots, x_{\nu}]$$

Für $\Delta u^{(1)}$ wähle man die linearen Terme aus (x). d. h.

$$\Delta u^{(1)} = \sum_{i=2}^{\nu} u_i(x_1)(x_i - \alpha_i) \text{ mit } u_i(x_1) = \Phi_I(c_i) \ \ 2 \le i \le \nu.$$

Prof. Dr. K. Madlener: Computeralgebra

Prof. Dr. K. Madlener: Computeralgebra

P-adische und ideal-adische Darstellunger

Prof. Dr. K. Madlener: Computeralgebra

Newton's Iteration und Hensel's Konstruktion

P-adische und ideal-adische Darstellunge

Multivariate Taylor-Reihendarstellung (Forts.)

 \triangleright Wir erhalten als Approximation von \tilde{u}

$$u^{(2)} = u^{(1)} + \Delta u^{(1)} \text{ mit } \Delta u^{(1)} \in I.$$

Neuer Fehlerterm

$$e^{(2)} = \tilde{u} - u^{(2)} = e^{(1)} - \Delta u^{(1)}$$

Wiederum

$$e^{(2)} = \sum_{i=2}^{\nu} (c_i - u_i(x_1))(x_i - \alpha_i)$$

wobei

$$c_i - u_i(x_1) \in I \quad 2 < i < \nu$$

d. h.
$$e^{(2)} \in I^2$$

4□ > 4問 > 4 = > 4 = > 9 Q ○

Multivariate Taylor-Reihendarstellung (Forts.)

- ▶ Potenzen von $I = \langle x_2 \alpha_2, x_3 \alpha_3, \dots \rangle$ $I^2 = \langle (x_2 - \alpha_2)^2, (x_2 - \alpha_2)(x_3 - \alpha_3), (x_3 - \alpha_3)^2, \dots \rangle$ $I^3 = \langle (x_2 - \alpha_2)^3, (x_2 - \alpha_2)^2, (x_3 - \alpha_3), (x_2 - \alpha_2), (x_3 - \alpha_3)^2, (x_3 - \alpha_3)^3 \dots \rangle$
- ▶ Drückt man $e^{(2)}$ in Basis von I^2 aus. so

$$e^{(2)} = \sum_{i=2}^{\nu} \sum_{j=i}^{\nu} c_{ij}(x_i - \alpha_i)(x_j - \alpha_j) \text{ mit } c_{ij} \in \mathbb{Z}_p[x_1, \dots, x_{\nu}]$$

Dann

$$\Delta u^{(2)} = \sum_{i=2}^{\nu} \sum_{j=i}^{\nu} u_{ij}(x_1)(x_i - \alpha_i)(x_j - \alpha_j)$$

wobei

 $u_{ii}(x_1) = \Phi_I(c_{ii}), 2 < i < j < \nu$

Newton's Iteration und Hensel's Konstruktion

Multivariate Taylor-Reihendarstellung (Forts.)

ightharpoonup Damit hat man die Approximation für \tilde{u}

$$u^{(3)} = u^{(2)} + \Delta u^{(2)} = u^{(1)} + \Delta u^{(1)} + \Delta u^{(2)}$$

▶ Mit $e^{(3)} = \tilde{u} - u^{(3)}$ folgt $e^{(3)} \in I^3$ und als Korrekturterm $\Delta u^{(3)} \in I^3$ der Form

$$\Delta u^{(3)} = \sum_{i=2}^{\nu} \sum_{j=i}^{\nu} \sum_{k=j}^{\nu} u_{ijk}(x_1)(x_i - \alpha_i)(x_j - \alpha_j)(x_k - \alpha_k)$$
mit $u_{ijk}(x_1) \in \mathbb{Z}_p[x_1]$

 \triangleright Prozess terminiert, da \tilde{u} Polynom, der letzte Term enthält d (=Totalgrad von \tilde{u}) geschachtelte Summationen.

Newton's Iteration und Hensel's Konstruktion

Ideal-adische Darstellung und Approximation

Multivariate Taylor-Reihendarstellung (Forts.)

Multivariate Taylor-Reihendarstellung bzgl. Ideal

$$I = \langle x_2 - \alpha_2, \dots, x_{\nu} - \alpha_{\nu} \rangle$$
, $\alpha_i \in \mathbb{Z}_p$ für Polynom $\tilde{u} = u(x_1, \dots, x_{\nu}) \in \mathbb{Z}_p[x_1, \dots, x_{\nu}]$ ist

$$u(x_{1},...,x_{\nu}) = u(x_{1},\alpha_{2},...,\alpha_{\nu}) + \sum_{i=2}^{\nu} u_{i}(x_{1})(x_{i} - \alpha_{i})$$

$$+ \sum_{i=2}^{\nu} \sum_{j=i}^{\nu} u_{ij}(x_{1})(x_{i} - \alpha_{i})(x_{j} - \alpha_{j})$$

$$+ \sum_{i=2}^{\nu} \sum_{j=i}^{\nu} \sum_{k=j}^{\nu} u_{ijk}(x_{1})(x_{i} - \alpha_{i})(x_{j} - \alpha_{j})(x_{k} - \alpha_{k})$$

$$+ \cdots$$

▶ d totaler Grad von $u(x_1,...,x_{\nu})$ bzgl. $x_2,...,x_{\nu}$, gibt Anzahl der geschachtelten Summationen.



Prof. Dr. K. Madlener: Computeralgebra

Newton's Iteration and Hensel's Konstruktion

Ideal-adische Darstellung und Approximation

Ideal-adische Darstellung und Approximation

► Analogie zur P-adischen Darstellung:

$$\tilde{u} = u^{(1)} + \Delta u^{(1)} + \Delta u^{(2)} + \dots + \Delta u^{(n)}$$

Mit $u^{(1)} = u_0(\vec{x}) \in \mathbb{Z}[\vec{x}]/\langle p \rangle$.

$$\Delta u^{(k)} = u_k(\vec{x})p^k \in \langle p \rangle^k$$
 für $k = 1, 2, \dots, n$

▶ Beachte $\mathbb{Z}[\vec{x}]/\langle p \rangle = \mathbb{Z}_p[\vec{x}]$ und $\langle p \rangle^k = \langle p^k \rangle$.

Der Koeffizient $u_k(\vec{x})$ im Ausdruck für $\Delta u^{(k)}$ erfüllt als ein Vielfaches des Basiselementes des Ideals in dem es liegt:

$$u_k(\vec{x}) \in \mathbb{Z}[\vec{x}]/\langle p \rangle$$
 $1 \le k \le n$

▶ Im p-adischen Falle kann man eine Folge p-adischer Approximationen der Ordnung k + 1 definieren::

$$u^{(k+1)} \in \mathbb{Z}[\vec{x}]/\langle p \rangle^{k+1}$$
 für $k = 1, 2, \dots, n$, wobei $u^{(k+1)} = u^{(1)} + \Delta u^{(1)} + \Delta u^{(2)} + \dots + \Delta u^{(k)}$.

Ideal-adische Darstellung und Approximation (Forts.)

Beim definieren des k-ten Elements der Folge aus der Approximation $u^{(k)} \in \mathbb{Z}[x]/\langle p \rangle^k$, erhält die neue Approximation $u^{(k+1)} \in \mathbb{Z}[x]/\langle p \rangle^{k+1}$ durch addieren des Terms $\Delta u^{(k)} \in \langle p \rangle^k$. Diese Addition

$$u^{(k+1)} = u^{(k)} + \Delta u^{(k)}$$

ist eine Addition in dem größeren Ring $\mathbb{Z}[\vec{x}]/\langle p \rangle^{k+1}$. Sie ist erlaubt durch sie natürliche Einbettung des Ringes $\mathbb{Z}[\vec{x}]/\langle p \rangle^k$ in den größeren Ring $\mathbb{Z}[\vec{x}]/\langle p \rangle^{k+1}$

▶ Daher liegen die sukzessiven p-adischen Approximationen $u^{(1)}, u^{(2)}, u^{(3)}, \dots$ von $\tilde{u} \in \mathbb{Z}[\vec{x}]$ in einer Folge von Unterringen wachsender Größe von $\mathbb{Z}[\vec{x}]$

$$\mathbb{Z}[\vec{x}]/\langle p \rangle \subset \mathbb{Z}[\vec{x}]/\langle p \rangle^2 \subseteq \mathbb{Z}[\vec{x}]/\langle p \rangle^3 \subseteq \cdots \subseteq \mathbb{Z}[\vec{x}]$$

Prof. Dr. K. Madlener: Computeralgebra

Newton's Iteration und Hensel's Konstruktion

Ideal-adische Darstellung und Approximatio

Ideal-adische Darstellung und Approximation (Forts.)

- ▶ Da $u \in \mathbb{Z}[\vec{x}]$ gibt es ein n mit:: $\mathbb{Z}[\vec{x}]/\langle p \rangle^{n+1}$ groß genug um u zu enthalten.
- ▶ Die multivariate Taylor-Reihendarstellung kann als analoge Darstellung aufgefasst werden, wobei I an Stelle von $\langle p \rangle$ tritt. Es gilt

$$\mathbb{Z}_p[\vec{x}]/I \subset \mathbb{Z}_p[\vec{x}]/I^2 \subseteq \cdots \subseteq \mathbb{Z}_p[\vec{x}]$$

5.4 Definition Sei D noetherscher Integritätsbereich und I ideal in D. Zu $a \in D$ heißt $b \in D$ eine ideal-adische Approximation n-ter Ordnung für a bezüglich des Ideales I, falls

$$a = b \mod I^n$$

Der Fehler bei der Approximation von a durch b ist das Element $a-b\in D$. Mit $\Phi_{I^n}:D\to D/I^n$ gilt $\Phi_{I^n}(a-b)=0$.

Prof. Dr. K. Madlener: Computeralgebra

Iteration nach Newton für F(u) = 0

Iteration nach Newton für F(u) = 0

Lineare p-adische Iteration

▶ Inversion von $\Phi_p : \mathbb{Z}[x] \to \mathbb{Z}_p[x]$. Nur eine Primzahl p, als bekannt $u_0(x) \in \mathbb{Z}_p[x]$: Bild der gesuchten Lösung $u(x) \in \mathbb{Z}[x]$, d.h. $u_0(x)$ ist p-adische Approximation erster Ordnung.

Gesucht: Methode zur Berechnung der Approximation der Ordnung k, d.h.

$$u_0(x) + u_1(x)p + \dots + u_{k-1}(x)p^{k-1} \in \mathbb{Z}_{p^k}[x] \quad k = 1, \dots, n+1$$

Lifting von Bild $u_0(x) \in \mathbb{Z}_p[x]$.

- \triangleright Offenbar benötigt man zusätzliche Information, um u(x) festzulegen. Üblicherweise ist diese Information in Form von Gleichungen gegeben, die u(x) erfüllen muss
 - z. B. F(u) = 0 wobei $F(u) \in \mathbb{Z}[x][u]$. Z.B. $u^2 a(x) = 0$.



4□ > 4問 > 4 = > 4 = > 9 Q ○

Prof. Dr. K. Madlener: Computeralgebra

Newton's Iteration und Hensel's Konstruktion Iteration nach Newton für F(u) = 0

Klassisches Newton-Verfahren zur Lösung einer nichtlinearen Gleichung F(u) = 0

▶ Entwicklung von F als Taylor-Reihe um Punkt $u^{(k)}$:

$$F(u) = F(u^{(k)}) + F'(u^{(k)})(u - u^{(k)}) + \frac{1}{2}F''(u^{(k)})(u - u^{(k)})^2 + \cdots$$

▶ Setzt man $u = \tilde{u}$ und betrachtet man nur lineare Terme, so $0 = F(u^{(k)}) + F'(u^{(k)})(\tilde{u} - u^{(k)}), d.h.$

$$u^{(k+1)} = u^{(k)} - \frac{F(u^{(k)})}{F'(u^{(k)})} \quad (F'(u^{(k)}) \neq 0)$$

▶ Startpunkt $u^{(1)} \rightsquigarrow Quadratische Konvergenz$.

Newton-Verfahren zur Lösung einer nichtlinearen Gleichung F(u) = 0

Annahme: Es gibt eine Lösung $\tilde{u} = u(x) \in \mathbb{Z}[x]$. Gegeben: Approximation erster Ordnung $u_0(x) \in \mathbb{Z}_p[x]$ von \tilde{u}

▶ Schreibt man die Lösung in ihrer p-adischen Pol-Darstellung

$$\tilde{u} = u_0(x) + u_1(x)p + \cdots + u_n(x)p^n$$

- Aufgabe: Bestimme die $u_i(x) \in \mathbb{Z}_p[x]$ i = 1, 2, ..., n wobei $u^{(k)} = u_0(x) + u_1(x)p + \cdots + u_{k-1}(x)p^{k-1}$ 1 < k < n + 1 die p-adische Approximation k-ter Ordnung von \tilde{u} ist.
- Gesucht Iterationsformel, die im Schritt k aus $u^{(k)}$ den p-adischen Polynomkoeffizienten $u_k(x) \in \mathbb{Z}_p[x]$ berechnet und somit zu $u^{(k+1)} = u^{(k)} + u_k(x)p^k$ 1 < k < n führt.

Prof. Dr. K. Madlener: Computeralgebra

Newton's Iteration und Hensel's Konstruktion

Iteration nach Newton für F(u) = 0

Newton-Verfahren:: Gleichung F(u) = 0 (Forts.)

5.5 Lemma Sei $a(x) \in D[x]$. Dann gilt in D[x][y]

$$a(x + y) = a(x) + a'(x)y + b(x, y)y^{2}$$

für ein geeignetes Polynom $b(x, y) \in D[x, y]$.

5.6 Lemma Sei $a(x, y) \in D[x, y]$ bivariates Polynom. Dann gilt im Polynomring $D[x, y][\xi, \eta]$.

$$\begin{array}{ll} a(x+\xi,y+\eta) & = & a(x,y) + a_x(x,y)\xi + a_y(x,y)\eta \\ & & + b_1(x,y,\xi,\eta)\xi^2 + b_2(x,y,\xi,\eta)\xi\eta \\ & & + b_3(x,y,\xi,\eta)\eta^2 \end{array}$$

4 D > 4 B > 4 B > 4 B > 3 9 9 9

Für geeignete Polynome $b_i(x, y, \xi, \eta) \in D[x, y, \xi, \eta]$

Prof. Dr. K. Madlener: Computeralgebra Prof. Dr. K. Madlener: Computeralgebra Newton's Iteration und Hensel's Konstruktion

Newton-Verfahren:: Gleichung F(u) = 0 (Forts.)

▶ Das Polynom $F(u) \in \mathbb{Z}[x][u]$ hat dann folgende Darstellung

$$F(u^{(k)} + u_k(x)p^k) = F(u^{(k)}) + F'(u^{(k)})u_k(x)p^k + g(u^{(k)}, u_k(x)p^k)[u_k(x)]^2p^{2k}$$

für ein $g(u, w) \in D[u, w]$.

- ▶ Wegen $u^{(k)} \equiv \tilde{u} \mod p^k$ und $F(\tilde{u}) = 0$ folgt $F(u^{(k)}) \equiv 0 \mod p^k$.
- ► Analog gilt $F(u^{(k)} + u_{\nu}(x)p^k) \equiv 0 \mod p^{k+1}$ falls $u^{(k+1)} = u^{(k)} + u_{\nu}(x)p^{(k)}$.

Prof. Dr. K. Madlener: Computeralgebra

Newton's Iteration und Hensel's Konstruktion

Iteration nach Newton für F(u) = 0

Newton-Verfahren:: Gleichung F(u) = 0 (Forts.)

▶ D. h.

$$\frac{F(u^{(k)} + u_k(x)p^k)}{p^k} = \frac{F(u^{(k)})}{p^k} + F'(u^{(k)})u_k(x) + g(u^{(k)}, u_k(x)p^k)[u_k(x)]^2p^k$$

▶ Wende Φ_{D} an, so erfüllt $u_{k}(x) \in \mathbb{Z}_{p}[x]$ die Gleichung

$$0 = \Phi_p\left(\frac{F(u^{(k)})}{p^k}\right) + \Phi_p(F'(u^{(k)}))u_k(x) \in \mathbb{Z}_p[x]$$

Wegen $u^{(k)} \equiv u^{(1)} \mod p$ für k > 1 gilt

$$F'(u^{(k)}) \equiv F'(u^{(1)}) \mod p$$

4□ > 4問 > 4 = > 4 = > 9 Q ○

Newton-Verfahren:: Gleichung F(u) = 0 (Forts.)

Genügt die gegebene Approximation erster Ordnung $u^{(1)}$ der Bedingung

$$F'(u^{(1)}) \not\equiv 0 \bmod p$$

so ist der gesuchte p-adische Polynomkoeffizient

(#)
$$u_k(x) = -\frac{\Phi_p\left(\frac{F(u^{(k)})}{p^k}\right)}{\Phi_p(F'(u^{(1)}))} \in \mathbb{Z}_p[x]$$

Diese Division ist exakt im Polynomring $\mathbb{Z}_p[x]$, falls eine Polynomlösung existiert.

Die Gleichung (#) ist die lineare Aktualisierungsformel zusammen mit $u^{(k+1)} = u^{(k)} + u_k(x)p^k \implies \text{lineare p-adische Newtonverfahren.}$ Beachte: $F(u^{(k)})$ wird in $\mathbb{Z}[x]$ durchgeführt, sowie auch Division durch p^k erst dann Φ_p anwenden!

Prof. Dr. K. Madlener: Computeralgebra

Newton's Iteration und Hensel's Konstruktion

Iteration nach Newton für F(u) = 0

Beispiel: Iteration nach Newton für Quadratwurzel

5.7 Beispiel Bestimme Polynom $u(x) \in \mathbb{Z}[x]$, das die Quadratwurzel des **Polynoms**

$$a(x) = 36x^4 - 180x^3 + 93x^2 + 330x + 121 \in \mathbb{Z}[x]$$

(unter der Annahme, dass a(x) quadratisch ist)

$$u(x)$$
 als Lösung von $F(u) = a(x) - u^2 = 0$

Wähle
$$p = 5$$
 $\Phi_5(a(x)) = x^4 - 2x^2 + 1 \in \mathbb{Z}_5[x].$

Approximation 1 Ordnung muss Quadratwurzel von $\Phi_5(a(x))$ sein, d. h. $u^{(1)} = u_0(x) = x^2 - 1 \in \mathbb{Z}_5[x].$

$$F'(u) = -2u$$
 und $\Phi_5(F'(u^{(1)})) = \Phi_5(-2u^{(1)}) = -2x^2 + 2$

Iteration nach Newton für F(u) = 0

Newton's Iteration und Hensel's Konstruktion

Beispiel: Quadratwurzel

Dann

$$u_1(x) = -\frac{\Phi_5\left(\frac{F(u^{(1)})}{5}\right)}{(-2x^2+2)} = -\frac{\Phi_5\left(\frac{35x^4 - 180x^3 + 95x^2 + 330x + 120}{5}\right)}{(-2x^2+2)}$$
$$= -\frac{(2x^4 - x^3 - x^2 + x - 1)}{(-2x^2+2)} = x^2 + 2x - 2 \in \mathbb{Z}_5[x]$$

und $u^{(2)} = (x^2 - 1) + (x^2 + 2x - 2)5 \in \mathbb{Z}_{25}[x]$ Analog

$$u_2(x) = -\frac{(-2x^3 + 2x)}{(-2x^2 + 2)} = -x \in \mathbb{Z}_5[x]$$

d. h. $u^{(3)} = (x^2 - 1) + (x^2 + 2x - 2)5 + (-x)5^2 \in \mathbb{Z}_{125}[x]$ $F(u^{(3)}) = 0 \leadsto \text{Terminierung, d. h.}$ Quadratwurzel ist $u(x) = u^{(3)} = 6x^2 - 15x - 11 \in \mathbb{Z}[x].$



Prof. Dr. K. Madlener: Computeralgebra

217

Newton's Iteration und Hensel's Konstruktion

Beispiel: Division mit Rest

5.8 Beispiel Division mit Rest über Newton Iteration

 \mathbb{Z} , F[x] sind Euklidische Bereiche \leadsto Division mit Rest. Komplexität $O(n^2)$ (Wort- oder Körperoperationen)

- ► Kann verbessert werden auf O(M(n)) wobei M die Multiplikationsschranke ist.
- Polynomfall: Sei D Ring $a, b \in D[x]$ Grade n, m mit $m \le n$, b monisch. Finde $q, r \in D[x]$ mit a = qb + r Grad(r) < Grad(b). [Da b monisch ist, ist die Existenz sicher].
- ► Es gilt:

$$(*) x^n a\left(\frac{1}{x}\right) = \left(x^{n-m} q\left(\frac{1}{x}\right)\right) \cdot \left(x^m b\left(\frac{1}{x}\right)\right) + x^{n-m+1} \left(x^{m-1} r\left(\frac{1}{x}\right)\right)$$

Beispiel: Division mit Rest

Setze $rev_k(a):=x^ka(1/x)$. Für k=n erhält man $a=a_nx^n+a_{n-1}x^{n-1}+\cdots+a_1x+a_0$ $rev_n(a)=a_0x^n+a_1x^{n-1}+\cdots+a_{n-1}x+a_n$ $(*') rev_n(a)=rev_{n-m}(q)\cdot rev_m(b)+x^{n-m+1}\cdot rev_{m-1}(r)$

d.h.

$$rev_n(a) \equiv rev_{n-m}(q) \cdot rev_m(b) \mod x^{n-m+1}$$

▶ Da $rev_m(b)$ 1 als Konstanten Koeffizienten hat, ist es invertierbar $\mod x^{n-m+1}$ also

$$rev_{n-m}(q) \equiv rev_n(a)rev_m(b)^{-1} \mod x^{n-m+1}$$

hieraus lassen sich q und r berechnen:

$$q = rev_{n-m}(rev_{n-m}(q))$$
 und $r = a - qb$

7 D N 7 D N 7 E N 7 E N

Prof. Dr. K. Madlener: Computeralgebra

Beispiele (Forts.)

z.B. $a = 5x^5 + 4x^4 + 3x^3 + 2x^2 + x$ $b = x^2 + 2x + 3$ $\mathbb{F}_7[x]$

 $rev_5(a) = x^4 + 2x^3 + 3x^2 + 4x + 5$ $rev_2(b) = 3x^2 + 2x + 1$

Wie berechnet man $rev_2(b)^{-1} \mod x^4$?

 $rev_2(b)^{-1} \equiv 4x^3 + x^2 + 5x + 1 \mod x^4$ in $\mathbb{F}_7[x]$ $\rightsquigarrow rev_3(q) \equiv 6x^3 + x + 5 \mod x^4$

 $\Rightarrow q = 5x^3 + x^2 + 6 \text{ und } r = a - qb = 3x + 3$

Iteration nach Newton für F(u) = 0

Inversion modulo x^I in D[x]

Problem:

Gegeben $f \in \mathbb{D}[x]$, $I \in \mathbb{N}$ mit f(0) = 1Finde $g \in \mathbb{D}[x]$ mit $fg \equiv 1 \mod x^I$

Newton Iteration Lösungen von $\Phi(g) = 0$ aus Anfangsnäherung g_0 :

$$g_{i+1} = g_i - \frac{\Phi(g_i)}{\Phi'(g_i)}$$

• $\Phi(g) = \frac{1}{g} - f = 0$

$$ightharpoonup g_{i+1} = g_i - \frac{1/g_i - f}{-1/g_i^2} = 2g_i - fg_i^2$$

4日 → 4周 → 4 目 → 4 目 → 9 Q (*)

4□ > 4問 > 4 = > 4 = > 9 Q ○

Prof. Dr. K. Madlener: Computeralgebra

Newton's Iteration und Hensel's Konstruktion

Iteration nach Newton für F(u)=0

Inversion modulo x^I in D[x]

Seien f(0) = 1, $g_0 = 1$, $g_{i+1} \equiv 2g_i - fg_i^2 \mod x^{2^{i+1}}$.

Dann $fg_i \equiv 1 \mod x^{2^i}$ für $i \geq 0$

Beweis: Ind. i = 0 $f \cdot g_0 \equiv f(0)g_0 \equiv 1 \cdot 1 \equiv 1 \mod x^{2^0}$

Ind. Schritt:

$$1 - fg_{i+1} \equiv 1 - f(2g_i - fg_i^2)$$

$$\equiv 1 - 2fg_i + f^2g_i^2$$

$$\equiv (1 - fg_i)^2$$

$$\equiv 0 \mod x^{2^{i+1}}$$

▶ Beachte: Ist f(0) Einheit ungleich 1, so verwende für g_o $f(0)^{-1}$. Ist f(0) keine Einheit, so gibt es keine Inverse von $f \mod x^I$ da aus $fg \equiv 1 \mod x^I \rightsquigarrow f(0)g(0) = 1$

Inversion modulo x^I in D[x]: Beispiel

5.9 Beispiel $f = 3x^2 + 2x + 1$ in $\mathbb{F}_7[x]$, I = 4

Alg. berechnet mit $g_0 = 1$ $r = 2 = \lceil \log(I) \rceil$

$$g_1 \equiv 2g_0 - fg_0^2 = 2 - (3x^2 + 2x + 1) \equiv 5x + 1 \mod x^2$$

$$g = g_2 \equiv 2g_1 - fg_1^2 = 2x^4 + 4x^3 + x^2 + 5x + 1 \equiv 4x^3 + x^2 + 5x + 1 \mod x^4$$

- ▶ Aufwand: $I = 2^r$ $3M(I) + I \in O(M(I))$ Arithm. Operationen (siehe auch von zur Gathen/Gerhard s.246)
- ▶ Division mit Rest nach diesem Verfahren kostet

$$4M(n) + M(n) + O(n)$$

Ringoperationen

 $M(n) \in O(n \log n \log \log n)$

◆ロト ◆部ト ◆見ト ◆見ト ・ 見・り

4 D > 4 B > 4 B > 4 B > 3 9 9 9

Prof. Dr. K. Madlener: Computeralgebra

Newton's Iteration und Hensel's Konstruktion

Iteration nach Newton für F(u)=0

P-adische Inversion mit Newton Iteration

5.10 Beispiel Sei R beliebiger Ring $0 \neq p \in R$. p-adische Darstellung ist auch hier sinnvoll.

Problem: Berechnung eines Inversen von $a \mod p^l$ l > 1, aus Inverse von $a \mod p$.

Gegeben: b_0 mit $ab_0 \equiv 1 \mod p$

Gesucht: $b \text{ mit } ab \equiv 1 \text{ mod } p^l :: \text{Liften von Inversen.}$

$$\begin{array}{ll} \textbf{procedure} \ \mathsf{InvLift} \ (a,b_0,I) & \{ab_0 \equiv 1 \ \mathsf{mod} \ p & I \in \mathbb{N}\} \\ r := \lceil \log I \rceil & \\ \textbf{for} \ i = 1 \ \textbf{to} \ r \ \textbf{do} & \\ \text{berechne} \ b_i := (2b_{i-1} - ab_{i-1}^2) \ \mathsf{mod} \ p^{2^i} & \\ \textbf{return} \quad b_r & \end{array}$$

Behauptung:
$$ab_i \equiv 1 \mod p^{2^i}$$
 Induktion: $i = 0$ $1 - ab_{i+1} \equiv 1 - a(2b_i - ab_i^2) \equiv 1 - 2ab_i + a^2b_i^2 \equiv (1 - ab_i)^2 \equiv 0 \mod p^{2^{i+1}}$ Bsp:: $R = \mathbb{Z}, p > 1$ oder $R = D[x]$ p monisch, grad $b < l$ grad p etwa $p = x$.

Prof. Dr. K. Madlener: Computeralgebra 224

P-adische Inversion mit Newton Iteration (Forts.)

5.11 Folgerung Sei *R* Ring, $p \in R$, $l \in \mathbb{N}^+$::

a ist invertierbar mod p' gdw a invertierbar mod p.

Aufwand: $O(M(l \log p))$ Wortoperationen, M multipl. Kosten bzw. O(M(I grad p)) Operationen in D.

Newton Iteration mit quadratischer Konvergenz

5.12 Lemma Sei $F \in R[u]$, $a, b \in R$ mit $F(a) \equiv 0 \mod p^k$ für ein $k \in \mathbb{N}^+$, F'(a) invertierbar modulo p. Weiterhin gelte

$$(*) b \equiv a - F(a)F'(a)^{-1} \bmod p^{2k}$$

Dann gilt $F(b) \equiv 0 \mod p^{2k}$, $b \equiv a \mod p^k$ und F'(b) ist invertierbar mod p.

"Ist a eine gute Approximation einer Nullstelle von F, so ist b eine bessere Approximation, mindestens doppelt so gut".



Prof. Dr. K. Madlener: Computeralgebra

Prof. Dr. K. Madlener: Computeralgebra

Iteration nach Newton für F(u) = 0

Newton's Iteration und Hensel's Konstruktion

Iteration nach Newton für F(u) = 0

Quadratische Konvergenz der Newton Iteration

Beweis: F'(a) ist invertierbar mod p^k , d. h. rechte Seite von (*) ist wohldefiniert. $F'(a)^{-1} \mod p^{2k}$ lässt sich aus $F'(a)^{-1} \mod p$ berechnen.

Da $p^k \mid p^{2k}$ gilt (*) auch mod $p^k \rightsquigarrow b \equiv a - F(a)F'(a)^{-1} \equiv a \mod p^k$.

$$F(b) \equiv F(a) + F'(a)(b-a) + \leadsto (b-a)^{2}$$

$$\equiv F(a) + F'(a)(b-a) \equiv F(a) + F'(a)(-F(a)F'(a)^{-1})$$

$$\equiv 0 \mod p^{2k}$$

Da $p^{2k} \mid (a-b)^2$ und $F(a) \equiv 0 \mod p^k$.

Wegen $a \equiv b \mod p^k$ gilt $a \equiv b \mod p$. $F(a) \equiv F(b) \mod p$ für alle $F \in R[u]$. Insbesondere für F'.

Für p Primelement im euklidischem Bereich ist die Bedingung F'(a)invertierbar mod $p \operatorname{gdw} F'(a) \not\equiv 0 \operatorname{mod} p$.

Algorithmus p-adische Newton Iteration

```
begin
                        {Eingabe : F \in R[u], R \text{ Ring}, p \in R, l \in \mathbb{N}^+, a_0 \in R,}
           {Startlösung mit F(a_0) \equiv 0 \mod p, F'(a_0) invertierbar mod p, f'(a_0)
                                          \{s_0 \text{ modulare Inverse für } F'(a_0) \text{ mod } p\}
                {Ausgabe : a \in R mit F(a) \equiv 0 \mod p^l und a \equiv a_0 \mod p}
r := \lceil \log I \rceil
for i := 1 to r - 1 do
     begin
    berechne a_i, s_i \in R mit
    a_i \equiv a_{i-1} - F(a_{i-1})s_{i-1} \mod p^{2'}
    s_i \equiv 2s_{i-1} - F'(a_i)s_{i-1}^2 \mod p^{2^i}
    end
Berechne a \in R mit a \equiv a_{r-1} - F(a_{r-1})s_{r-1} \mod p^l
return a
end
```

Newton's Iteration und Hensel's Konstruktion

Algorithmus p-adische Newton Iteration

Korrektheit: Sei $a_r \equiv a_{r-1} - F(a_{r-1})s_{r-1} \mod p^{2^r}$.

▶ Dann $a \equiv a_r \mod p^l$ und es genügt die Invarianten.

$$a_i \equiv a_0 \mod p$$
, $F(a_i) \equiv 0 \mod p^{2^i}$ $s_i \equiv F'(a_i)^{-1} \mod p^{2^i}$

Für 0 < i < r. Per Induktion zu zeigen (Anwendung Lemma+Inversionsalg.).

Ist $R = \mathbb{Z}$ oder R = F[x], F Körper, und $p \in R$ prim oder irreduzibel, so ist der Startwert als Lösung für Polynom in $K = R/\langle p \rangle$.

Aufwand:

$$R = D[x], F \in R[u], p = x, I = 2^k, \operatorname{grad}_u F = n, \operatorname{grad}_x F < I \leadsto O(nM(I)) + O(nI)$$
 Operationen in D .
 $R = \mathbb{Z}, 0 \le a_0 \le p$, F grad n , mit Koeffizienten $\le p^I$

 $\rightsquigarrow O(nM(l \log p))$ Wortoperationen.

4□ > 4問 > 4 = > 4 = > 9 Q ○

4 D > 4 D > 4 D > 4 D > 3 D 9 O

Beispiel

5.13 Beispiel

i) $R = \mathbb{Z}$, p = 5 bestimme nicht-triviale Lösung von $u^4 \equiv 1 \mod 625$, d. h. $F(u) = u^4 - 1$.

Startlösung $a_0 = 2$, da $F(2) \equiv 0 \mod 5$.

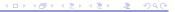
$$F'(2) = 4 \cdot 2^3 \equiv 2 \not\equiv 0 \mod 5$$
, d. h. $s_0 \equiv 2^{-1} \equiv 3 \mod 5$

$$a_1 \equiv a_0 - F(a_0)s_0 = 2 - 15 \cdot 3 \equiv 7 \mod 25$$

$$s_1 \equiv 2s_0 - F'(a_1)s_0^2 = 2 \cdot 3 - 1372 \cdot 3^2 \equiv 8 \mod 25$$

$$a \equiv a_1 - F(a_1)s_1 = 7 - 2400 \cdot 8 \equiv 182 \mod 625$$

In der Tat gilt $182^4 = 1 + 1755519 \cdot 625$.



Prof. Dr. K. Madlener: Computeralgebra

229

Newton's Iteration und Hensel's Konstruktion

Iteration nach Newton für F(u)=0

Beispiel

ii) $R = \mathbb{F}_3[x]$ p = x. Bestimme Quadratwurzel a von f = x + 1 modulo x^4 mit a(0) = -1. $F = u^2 - f \in \mathbb{F}_3[x][u]$ $a_0 = -1$ als Startlösung, da $a_0(0) = -1$, $F(a_0) = -x \equiv 0$ mod x sowie $F'(a_0) = 2a_0 \equiv 1 \not\equiv 0$ mod x, d.h. $s_0 = 1$.

$$a_1 \equiv a_0 - F(a_0)s_0 = -1 - (-x)1 = x - 1 \mod x^2$$

$$s_1 \equiv 2s_0 - F'(a_1)s_0^2 = 2 \cdot 1 - 2(x - 1) \cdot 1^2$$

$$= x + 1 \mod x^2$$

$$a \equiv a_1 - F(a_1)s_1 = x - 1 - x^2(x + 1)$$

Offenbar

$$(-x^3 - x^2 + x - 1)^2 = (x + 1) + x^4(x^2 - x - 1)$$

Ideal-adische Newton Iteration

▶ Inversion eines multivariaten Auswertungshomomorphismus

$$\Phi_I: \mathbb{Z}_p[x_1,\ldots,x_{\nu}] \to \mathbb{Z}_p[x_1]$$

mit Kern
$$I = \langle x_2 - \alpha_2, \dots, x_{\nu} - \alpha_{\nu} \rangle$$
 $\alpha_i \in \mathbb{Z}_p$ $2 \le i \le \nu$.

▶ Startpunkt: Approximation erster Ordnung zur gesuchten Lösung $\tilde{u} \in \mathbb{Z}_p[\vec{x}]$.

$$u^{(1)} = \Phi_I(\tilde{u}) \in \mathbb{Z}_p[x_1] = \mathbb{Z}_p[\vec{x}]/I$$

- ▶ Zusatzinformation: \tilde{u} Lösung der Polynomgleichung F(u) = 0, wobei $F(u) \in \mathbb{Z}_p[\vec{x}][u]$.
- ▶ Ziel: Definition einer Iterationsformel $u^{(k+1)} = u^{(k)} + \Delta u^{(k)}$, wobei $u^{(i)}$ ideal-adische Approximation *i*-ter Ordnung und $\Delta u^{(k)} \in I^k$.

◆ロト ◆問 ▶ ◆ 恵 ▶ ◆ 恵 ◆ 夕 ♀

Prof. Dr. K. Madlener: Computeralgebra

Prof. Dr. K. Madlener: Computeralgebra

Newton's Iteration und Hensel's Konstruktion

Ideal-adische Newton Iteratio

Ideal-adische Newton Iteration (Forts.)

Durch Anwendung von Hilfssatz erhält man als Taylorentwicklung $F(\underbrace{u^{(k)} + \Delta u^{(k)}}) = F(u^{(k)}) + F'(u^{(k)}) \Delta u^{(k)} + G(u^{(k)}, \Delta u^{(k)}) [\Delta u^{(k)}]^2$

Ideal-adische Approximation der Ordnung k+1, d. h. $u^{(k+1)}$ so $F(u^{(k)}+\Delta u^{(k)})\in I^{k+1}$ und wegen $\Delta u^{(k)}\in I^k$ folgt $[\Delta u^{(k)}]^2\in I^{2k}$, d. h. wendet man $\Phi_{I^{k+1}}$ an, so gilt

$$(*) \ \ 0 = \Phi_{J^{k+1}}(F(u^{(k)})) + \Phi_{J^{k+1}}(F'(u^{(k)}))\Delta u^{(k)} \in \mathbb{Z}_p[\vec{x}]/J^{k+1}$$

und $\Delta u^{(k)}$ muss diese Gleichung erfüllen für k=1 so $\Delta u^{(1)} \in I$ und $\Delta u^{(1)} = \sum_{i=1}^{\nu} u_i(x_1)(x_i - \alpha_i)$ mit $u_i(x_1) \in \mathbb{Z}_p[x_1]$



Ideal-adische Newton Iteration-Erster Schritt

Da $u^{(1)} \equiv \tilde{u} \mod I$, gilt auch $F(u^{(1)}) \in I$, d. h.

$$(**) \quad F(u^{(1)}) = \sum_{i=2}^{\nu} c_i(x_i - \alpha_i) \text{ für } c_i \in \mathbb{Z}_p[\vec{x}] \quad 2 \le i \le \nu$$

Nun wird der Homomorphismus Φ_{l^2} für k=1 angewandt und da alle ideal-adischen Terme vom Totalgrad ≥ 2 unterdrückt werden, folgt

$$\Phi_{I^2}(F(u^{(1)})) = \sum_{i=2}^{\nu} c_i(x_1)(x_i - \alpha_i)$$

wobei die Koeffizienten $c_i(x_1) \in \mathbb{Z}_p[x_1]$ definiert sind, durch die Koeffizienten $c_i \in \mathbb{Z}_p[\vec{x}]$ in (**) d. h.

$$c_i(x_1) = \Phi_I(c_i) \quad 2 \le i \le \nu$$

<ロ > → □ > → □ > → □ > → □ → ○ へ ○

Prof. Dr. K. Madlener: Computeralgebra

Newton's Iteration und Hensel's Konstruktion

Ideal-adische Newton Iteration

$$0 = \sum_{i=2}^{\nu} c_i(x_1)(x_i - \alpha_i) + \Phi_{I^2}(F'(u^{(1)})) \left[\sum_{i=2}^{\nu} u_i(x_1)(x_i - \alpha_i) \right] \in \mathbb{Z}_p[\vec{x}]/I^2$$

gilt nach Gleichung (*). Die ideal-adische Darstellung von $\Phi_{l^2}(F'(u^{(1)}))$ hat für geeignete Koeffizienten $d_i(x_1) \in \mathbb{Z}_p[x_1]$ $2 \le i \le \nu$ die Form:

$$\Phi_{I^{2}}(F'(u^{(1)})) = \Phi_{I}(F'(u^{(1)})) + \sum_{i=2}^{\nu} d_{i}(x_{1})(x_{i} - \alpha_{i})$$

Setzt man dies ein, so bleibt

$$0 = \sum_{i=2}^{\nu} c_i(x_1)(x_i - \alpha_i) + \Phi_I(F'(u^{(1)})) \left[\sum_{i=2}^{\nu} u_i(x_1)(x_i - \alpha_i) \right] \in \mathbb{Z}_{\rho}[\vec{x}]/I^2$$

$$u_i(x_1) = -\frac{c_i(x_1)}{\Phi_I(F'(u^{(1)}))} \in \mathbb{Z}_p[x_1] \quad 2 \le i \le \nu$$

Aktualisierungsformel, die den Korrekturterm $\Delta u^{(1)}$ festlegt.

Newton's Iteration und Hensel's Konstruktion

Ideal-adische Newton Iteration-Folgeschritte

Im allgemeinen Iterationsschritt ist der k-te Korrekturterm $\Delta u^{(k)} \in I^k$ der Term mit Totalgrad k in der ideal-adischen Darstellung der Lösung $\tilde{u} = u(x_1, \dots, x_{\nu})$.

Analog zum ersten Schritt erhält man, falls $F'(u^{(1)}) \not\equiv 0 \mod I$, die lineare ideal-adische Newton Iteration:

$$u^{(n+1)} = u^{(k)} + \Delta u^{(k)}$$
, wobei

$$\Delta u^{(k)} = \sum_{i_1=2}^{\nu} \cdots \sum_{i_k=i_{k-1}}^{\nu} u_{\mathbf{i}}(x_1) \prod_{j=1}^{k} (x_{i_j} - \alpha_{i_j})$$

wobei $\mathbf{i} = (i_1, \dots, i_k)$ bezeichnet und

Prof. Dr. K. Madlener: Computeralgebra

233

Newton's Iteration und Hensel's Konstruktion

Ideal-adische Newton Iteratio

Ideal-adische Newton Iteration-Folgeschritte (Forts.)

$$(\#) \quad u_{\mathbf{i}}(x_1) = -\frac{c_{\mathbf{i}}(x_1)}{\Phi_I(F'(u^{(1)}))} \in \mathbb{Z}_p[x_1]$$

Hierbei gilt

$$\Phi_{I^{k+1}}(F(u^{(k)})) = \sum_{i_1=2}^{\nu} \cdots \sum_{i_k=i_{k-1}}^{\nu} c_i(x_1) \prod_{j=1}^{k} (x_{i_j} - x_{i_j})$$

Die Division in (#) muss exakt sein (in $\mathbb{Z}_p[x_1]$), wenn die Gleichung F(u)=0 eine Lösung besitzt.

Die $c_i(x_1)$ sind die Koeffizienten der Terme mit Totalgrad k in der ideal-adischen Darstellung von $F(u^{(k)})$ und weiterhin hat $F(u^{(k)})$ keine Terme vom Totalgrad kleiner k (bezüglich l).

Es werden in Iterationsschritten k alle ideal-adischen Terme der Lösung \tilde{u} mit Totalgrad k (bezüglich l) bestimmt.

4□ > 4ⓓ > 4 분 > 4 분 > . 분 . ∽ Q

Newton's Iteration und Hensel's Konstruktion

Ideal-adische Newton Iteration

Beispiel

5.14 Beispiel Bestimme Polynom $u(x, y, z) \in \mathbb{Z}_5[x, y, z]$, das die Quadratwurzel des folgenden Polynoms ist:

$$a(x, y, z) = x^4 + x^3y^2 - x^2y^4 + x^2yz + 2x^2z - 2x^2 - 2xy^3z + xy^2z$$
$$-xy^2 - y^2 + z^2 + yz^2 - yz + z^2 - 2z + 1 \in \mathbb{Z}_5[x, y, z]$$

Dann ist u Lösung der Polynomgleichung

$$F(u) = a(x, y, z) - u^2 = 0$$

Wähle Auswertungspunkte y=0, z=0, d. h. $I=\langle y,z\rangle$. Die ideal-adische Approximation erster Ordnung $u^{(1)}=u(x,0,0)\in\mathbb{Z}_5[x]$ muss eine Quadratwurzel von a(x,0,0) in $\mathbb{Z}_5[x]$ sein. Es ist $a(x,0,0)=x^4-2x^2+1$, d. h. $u^{(1)}=u(x,0,0)=x^2-1\in\mathbb{Z}_5[x]$. Um die lineare ideal-adische Newton Iteration anzuwenden, beachte dass

$$\Phi_I(F'(u^{(1)})) = \Phi_I(-2u^{(1)}) = -2x^2 + 2$$



Prof. Dr. K. Madlener: Computeralgebra

23

Newton's Iteration und Hensel's Konstruktion

Ideal-adische Newton Iteration

Beispiel (Forts.)

Es ist nützlich a(x,y,z) in seiner ideal-adischen Darstellung bezüglich I auszudrücken: d. h.

$$a(x, y, z) = [(x^4 - 2x^2 + 1)] + [(2x^2 - 2)z] + [(x^3 - x)y^2 + (x^2 - 1)yz + z^2] + [(x^3 - x)y^3z - y^2z^2]$$

$$+ [(-x^2)y^4 + (-2x)y^3z - y^2z^2]$$

Nun ist

$$\Phi_{I^2}(F(u^{(1)})) = \Phi_{I^2}(a(x, y, z) - (x^2 - 1)^2) = (2x^2 - 2)z \in \mathbb{Z}_5[x, y, z]/I^2$$

Der erste Korrekturterm ist $\Delta u^{(1)} = u_2(x)y + u_3(x)z$, wobei $u_2(x) = 0$, da in $\Phi_{I^2}(F(u^{(1)}))$ auch 0 und

$$u_3(x) = -\frac{c_3(x)}{(-2x^2+2)} = -\frac{(2x^2-2)}{(-2x^2+2)} = 1 \in \mathbb{Z}_5[x]$$

d. h.

$$u^{(2)} = u^{(1)} + \Delta u^{(1)} = (x^2 - 1) + z \in \mathbb{Z}_5[x, y, z]/I^2$$

Beispiel (Forts.)

Für die nächste Iteration gilt

$$\Phi_{I^3}(F(u^{(2)})) = \Phi_{I^3}(a(x, y, z) - [(x^2 - 1) + z]^2) =$$

$$= (x^3 - x)y^2 + (x^2 - 1)yz \in \mathbb{Z}_5[x, y, z]/I^3$$

Neuer Korrekturterm ist $\Delta u^{(2)}=u_{22}(x)y^2+u_{23}(x)yz+u_{33}(x)z^2$, wobei $u_{33}(x)=0$, da in $\Phi_{I^3}(F(u^{(2)}))$ auch null und

$$u_{22}(x) = -\frac{c_{22}(x)}{(-2x^2+2)} = -\frac{(x^3-x)}{(-2x^2+2)} = -2x \in \mathbb{Z}_5[x]$$

$$u_{23}(x) = -\frac{c_{23}(x)}{(-2x^2+2)} = -\frac{(x^2-1)}{(-2x^2+2)} = -2 \in \mathbb{Z}_5[x]$$

Prof. Dr. K. Madlener: Computeralgebra

Newton's Iteration und Hensel's Konstruktion

Ideal-adische Newton Iteratio

Beispiel (Forts.)

Also

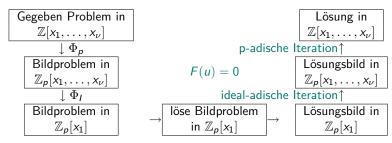
$$u^{(3)} = u^{(2)} + \Delta u^{(2)} = (x^2 - 1) + z + (-2x)y^2 + (-2)yz \in \mathbb{Z}_5[x, y, z]/I^3$$

Als nächstes stellt man fest: $F(u^{(3)}) = 0$, d. h. die gesuchte Quadratwurzel von a(x, y, z) ist

$$u(x, y, z) = u^{(3)} = x^2 - 2xy^2 - 2yz + z - 1 \in \mathbb{Z}_5[x, y, z]$$

Homomorphismus Diagramm

zur Lösung multivariater Polynomprobleme mit der p-adischen und der ideal-adischen Newton-Iterationen



Inversion von Homomorphismen der Form $\Phi_{I,p}: \mathbb{Z}[x_1,\ldots,x_{\nu}] \to \mathbb{Z}_p[x_1]$ unter der Annahme, dass u sich als Lösung von F(u)=0 darstellen lässt.

 $F(u) \in \mathbb{Z}[x_1, \dots, x_{\nu}][u]$: eine Variable wird "geliftet", reicht oft nicht aus!

<ロ > < 個 > < 種 > < 種 > を重 > ・ 重 ・ 夕 < で

4□ > 4問 > 4 = > 4 = > 9 Q ○

Prof. Dr. K. Madlener: Computeralgebra

Newton's Iteration und Hensel's Konstruktion

Prof. Dr. K. Madlener: Computeralgebra

Hensel's Lemma

Einführung zu Hensel's Lemma

- ▶ Liften von Lösungen für F(u, w) = 0 mit $F(u, w) \in \mathbb{Z}[x_1, \dots, x_{\nu}][u, w]$.
- ▶ z.B. Faktorisierung von $a(x_1, \ldots, x_{\nu})$: $F(u, w) = a(x_1, \ldots, x_{\nu}) uw = 0$ $\Phi_{I,p} : \mathbb{Z}[x_1, \ldots, x_{\nu}] \to \mathbb{Z}_p[x_1]$, d. h. Problem der Faktorisierung in $\mathbb{Z}_p[x_1]$ (relative einfach).

 $w(x_1,\ldots,x_{\nu})\equiv w_0(x_1) \bmod \langle I,p\rangle$

- ► Sei etwa $u_0(x_1)$ Faktor von $\Phi_{I,p}(a) = a_0(x_1)$, dann $a_0(x_1) = u_0(x_1)w_0(x_1)$, wobei $w_0(x_1) = \frac{a_0(x_1)}{u_0(x_1)} \in \mathbb{Z}_p[x_1]$
- ▶ Liften von u_0 und w_o m pol. $u, w \in \mathbb{Z}[x_1, \dots, x_{\nu}]$ mit $F(u, w) = a(x_1, \dots, x_{\nu}) uw = 0 \text{ und}$ $u(x_1, \dots, x_{\nu}) \equiv u_0(x_1) \bmod \langle I, p \rangle$

Einführung zu Hensel's Lemma (Forts.)

Newton Iteration für nichtlineare Gleichung in 2 Variablen

- ▶ Berechnung von GCD für $a(x_1, \ldots, x_{\nu}), b(x_1, \ldots, x_{\nu}) \in \mathbb{Z}[x_1, \ldots, x_{\nu}]$ \leadsto berechne GCD $(a_0(x_1), b_0(x_1))$ in $\mathbb{Z}_{\rho}[x_1]$ (Euklidisch).
- Ist $u_0(x_1) = GGT(a_0(x_1), b_0(x_1))$, so sei Cofaktor $w_0(x_1) = \frac{a_0(x_1)}{u_0(x_1)}$. Lifte $u_0(x_1)$ und $w_0(x_1)$ zu $u(x_1, \dots, x_{\nu})$ $w(x_1, \dots, x_{\nu})$ mit F(u, w) = 0. (Auch b kann verwendet werden). $w = u(x_1, \dots, x_{\nu})$ ist Teiler von $a(x_1, \dots, x_{\nu})$ auch von $b(x_1, \dots, x_{\nu})$?

Newton's Iteration und Hensel's Konstruktion

Prof. Dr. K. Madlener: Computeralgebra

Hensel's Lemr

Allgemeine Form der Newton Iteration für F(u, w) = 0

lacktriangle Korrekturterme $\Delta u^{(k)}, \Delta w^{(k)}$ mit Hilfe von

$$F(u^{(k)} + \Delta u^{(k)}, w^{(k)} + \Delta w^{(k)}) = F(u^{(k)}, w^{(k)}) + F_u(u^{(k)}, w^{(k)}) \Delta u^{(k)} + F_w(u^{(k)}, w^{(k)}) \Delta w^{(k)} + E$$

wobei E Ausdruck mit nicht-linearen $\Delta u^{(k)}, \Delta w^{(k)}$

► Linearisierung, linke Seite auf 0 ↔

(*)
$$F_u(u^{(k)}, w^{(k)}) \Delta u^{(k)} + F_w(u^{(k)}, w^{(k)}) \Delta w^{(k)} = -F(u^{(k)}, w^{(k)})$$

4 D F 4 D F 4 D F 9 9 9 9

▶ ~→ Problem del Lösung Diophantischer Polynomgleichungen.

242 Prof. Dr. K. Madlener: Computeralgebra

Hensel's Lemma

Allgemeine Form der Newton Iteration für F(u, w) = 0 (Forts.)

▶ Lösung diophantischer Polynomgleichung der Form

$$A^{(k)}\Delta u^{(k)} + B^{(k)}\Delta w^{(k)} = C^{(k)}$$

mit Polynomen $A^{(k)}, B^{(k)}, C^{(k)}$ und umb. Polynomen $\Delta u^{(k)}, \Delta w^{(k)}$

- ▶ Möglichkeiten: Keine Lösung, viele Lösungen. Lösbar, falls $C^{(k)}$ vielfaches vom $GGT(A^{(k)}, B^{(k)})$.
- Wir beschränken uns nun auf die Gleichung $F(u, w) = a(x_1, ..., x_{\nu}) uw = 0.$
- ▶ Lösbarkeit hängt wesentlich von F(u, w) ab.



Prof. Dr. K. Madlener: Computeralgebra

245

Newton's Iteration und Hensel's Konstruktion

Hensel's Lemma

Hensel's Lemma

Seien $a(x) \in \mathbb{Z}[x]$ und $u_0(x), w_0(x) \in \mathbb{Z}_p[x]$ mit

$$a(x) \equiv u_0(x)w_0(x) \bmod p$$

"Lifte" nach $\mathbb{Z}[x]$, d.h. Inversion von $\Phi_p: \mathbb{Z}[x] \to \mathbb{Z}_p[x]$. Berechne $\bar{u} = u(x), \bar{w} = w(x)$ in $\mathbb{Z}[x]$ mit

$$F(\bar{u}, \bar{w}) = a(x) - u(x)w(x) = 0$$

so dass

Prof. Dr. K. Madlener: Computeralgebra

$$u(x) \equiv u_0(x) \bmod p$$
 $w(x) \equiv w_0(x) \bmod p$

Betrachte \bar{u}, \bar{w} in ihren p-adischen Darstellungen.

$$\bar{u} = u_0(x) + u_1(x)p + \cdots + u_n(x)p^n$$

$$\bar{w} = w_0(x) + w_1(x)p + \cdots + w_n(x)p^n$$

Hensel's Lemma (1)

Newton's Iteration und Hensel's Konstruktion

- ▶ Hierbei ist n groß genug, d. h. ist $\frac{1}{2}p^{n+1}$ beschränkt alle Beträge der ganzzahligen Koeffizienten in a(x) und \bar{u}, \bar{w} .
- ▶ Wie bestimmt man die pol. p-adischen Koeffizienten $u_i(x), w_i(x) \in \mathbb{Z}_p[x]$ für i = 1, 2, ..., n?
- ▶ Seien $u^{(k)}, w^{(k)}$ die p-adische Approximation der Ordnung k für $\bar{u}, \bar{w},$ d. h. bis p^{k-1} und sei

$$\Delta u^{(k)} = u_k(x)p^k$$
 $\Delta w^{(k)} = w_k(x)p^k$

Beachte dabei $u^{(1)} = u_0(x)$ und $w^{(1)} = w_0(x)$.

▶ Korrekturterme müssen die diophantische Polynomgleichung * modulo p^{k+1} erfüllen $F_u = -w$, $F_w = -u$, d. h.

$$-w^{(k)}\Delta u^{(k)} - u^{(k)}\Delta w^{(k)} \equiv -[a(x) - u^{(k)}w^{(k)}] \mod p^{k+1}$$

4 D L 4 D L 4 E L 4 E L E L

Prof. Dr. K. Madlener: Computeralgebra

Newton's Iteration und Hensel's Konstruktion

Hensel's Lemma

Hensel's Lemma (2)

▶ Da $u^{(k)}w^{(k)}$ eine p-adische Approximation von a(x) der Ordnung k sein muss, kann man durch p^k dividieren, d. h.

$$w^{(k)}u_k(x) + u^{(k)}w_k(x) \equiv \frac{a(x) - u^{(k)}w^{(k)}}{p^k} \mod p$$

▶ Wendet man nun Φ_{ρ} an, unter Verwendung von $\Phi_{\rho}(w^{(k)}) = w_0(x)$ und $\Phi_{\rho}(u^{(k)}) = u_0(x)$, so erhält man

$$w_0(x)u_k(x) + u_0(x)w_k(x) = \Phi_p\left[\frac{a(x) - u^{(k)}w^{(k)}}{p^k}\right]$$

die in $\mathbb{Z}_p[x]$ zu lösen ist.

Hensel's Lemma

Hensel's Lemma (3)

▶ p prim, so $\mathbb{Z}_p[x]$ euklidisch. d.h. Sind $u_0(x)$, $w_0(x)$ teilerfremd \leadsto eindeutige Lösung $\sigma(x), \tau(x) \in \mathbb{Z}_p[x]$ mit

$$\sigma(x)u_0(x) + \tau(x)w_0(x) = \Phi_p \left[\frac{a(x) - u^{(k)}w^{(k)}}{p^k} \right]$$

wobei $grad(\sigma(x)) < grad(w_0(x))$.

► Man setze $u^{(k+1)} = u^{(k)} + \tau(x)p^k$ $w^{(k+1)} = w^{(k)} + \sigma(x)p^k$

Behauptung Dies sind die richtigen p-adischen Approximationen der Ordnung k+1 für \bar{u} bzw. \bar{w} .

Definiere

Newton's Iteration und Hensel's Konstruktion

Hensel's Lemma (5)

 $c^{(k)}(x) = \Phi_p \left[\underbrace{\frac{a(x) - u^{(k)}(x)w^{(k)}(x)}{p^k}}_{\text{allo Oper in } \mathbb{Z}_p - [x]} \right]$ (1)

Es gibt Polynome $\sigma^{(k)}(x), \tau^{(k)}(x) \in \mathbb{Z}_p[x]$ mit

$$\sigma^{(k)}(x)u^{(1)}(x) + \tau^{(k)}(x)w^{(1)}(x) \equiv c^{(k)}(x) \bmod p$$
 (2)

und $\operatorname{grad}(\sigma^{(k)}(x)) < \operatorname{grad}(w^{(1)}(x))$. Setze

$$u^{(k+1)}(x) = u^{(k)}(x) + \tau^{(k)}(x)p^k \quad w^{(k+1)}(x) = w^{(k)}(x) + \sigma^{(k)}(x)p^k$$

920 E 4E 4E 4 G 4 4 G 1

Prof. Dr. K. Madlener: Computeralgebra

249

Newton's Iteration und Hensel's Konstruktion

Hensel's Lemma

Newton's Iteration und Hensel's Konstruktion

Prof. Dr. K. Madlener: Computeralgebra

Hensel's Lem

Hensel's Lemma (4)

5.15 Satz Hensel's Lemma 1900

Sei p prim in \mathbb{Z} und sei $a(x) \in \mathbb{Z}[x]$. Seien $u^{(1)}(x), w^{(1)}(x) \in \mathbb{Z}_p[x]$ teilerfremd mit $a(x) \equiv u^{(1)}(x)w^{(1)}(x)$ mod p.

Dann gibt es für $k \ge 1$ Polynome $u^{(k)}(x), w^{(k)}(x) \in \mathbb{Z}_{p^k}[x]$, so dass

$$a(x) \equiv u^{(k)}(x)w^{(k)}(x) \bmod p^k$$

und

$$u^{(k)}(x) \equiv u^{(1)}(x) \bmod p \quad w^{(k)}(x) \equiv w^{(1)}(x) \bmod p$$

Beweis: Induktion nach k: k=1 Voraussetzung. Angenommen richtig für $k\geq 1$, d. h. $u^{(k)}(x), w^{(k)}(x)\in \mathbb{Z}_{p^k}[x]$ mit Behauptung.

Hensel's Lemma (6)

Dann Multiplikation $\mod p^{k+1}$ ergibt

$$u^{(k+1)}(x)w^{(k+1)}(x) \equiv u^{(k)}(x)w^{(k)}(x) + (\sigma^{(k)}(x)u^{(1)}(x) + \tau^{(k)}(x)w^{(1)}(x))p^k \mod p^{k+1}$$

$$\equiv u^{(k)}(x)w^{(k)}(x) + c^{(k)}(x)p^k \mod p^{k+1}$$

$$\equiv a(x) \mod p^{k+1}$$
 (1)

→ Behauptung.

5.16 Folgerung : Eindeutigkeit der Hensel Konstruktion Im Satz (Hensel's Lemma): Ist $a(x) \in \mathbb{Z}[x]$ monisch und dementsprechend wenn die teilerfremden Faktoren $u^{(1)}(x), w^{(1)}(x) \in \mathbb{Z}_p[x]$ monisch gewählt werden, so sind für alle $k \geq 1$ die monischen Polynomfaktoren $u^{(k)}(x), w^{(k)}(x) \in \mathbb{Z}_{p^k}[x]$ eindeutig bestimmt.

Hensel's Lemma

Hensel's Lemma (7)

Beweis: Induktion nach k: k=1 klar. Angenommen richtig für ein $k \ge 1$

$$\begin{array}{l} u^{(k+1)}(x), w^{(k+1)}(x) \in \mathbb{Z}_{p^{k+1}}[x] \text{ monisch mit} \\ a(x) \equiv u^{(k+1)}(x)w^{(k+1)}(x) \bmod p^{k+1} \text{ und} \\ u^{(k+1)}(x) \equiv u^{(1)}(x) \bmod p, w^{(k+1)}(x) \equiv w^{(1)}(x) \bmod p. \\ \text{Insbesondere} \\ a(x) \equiv u^{(k+1)}(x)w^{(k+1)}(x) \bmod p^k \text{ nach Induktion Voraussetzung} \\ u^{(k+1)}(x) \equiv u^{(k)}(x) \bmod p^k, \ w^{(k+1)}(x) \equiv w^{(k)}(x) \bmod p^k. \\ \text{D. h.} \end{array}$$

$$u^{(k+1)}(x) = u^{(k)}(x) + \tau(x)p^k$$
 $w^{(k+1)}(x) = w^{(k)}(x) + \sigma(x)p^k$

für Polynome $\sigma(x), \tau(x) \in \mathbb{Z}_p[x]$.



Prof. Dr. K. Madlener: Computeralgebra

253

Newton's Iteration und Hensel's Konstruktion

Hensel's Lemma

Hensel's Lemma (7)

Behauptung: Diese sind eindeutig: Da $a(x), u^{(1)}(x), w^{(1)}(x)$ monisch, muss

$$\gcd(\sigma(x)) < \gcd(w^{(1)}(x)) \quad \gcd(\tau(x)) < \gcd(u^{(1)}(x))$$
 gelten.

(Da $u^{(k+1)}(x)$ und $w^{(k+1)}(x)$ immer gleiche Leitterme wie $u^{(1)}(x)$ bzw. $w^{(1)}(x)$ haben müssen!).

$$a(x) \equiv u^{(k)}(x)w^{(k)}(x) + (\sigma(x)u^{(1)}(x) + \tau(x)w^{(1)}(x))p^k \mod p^{k+1}$$

$$\sigma(x)u^{(1)}(x) + \tau(x)w^{(1)}(x)) \equiv \frac{a(x) - u^{(k)}(x)w^{(k)}(x)}{p^k} \bmod p$$

 $\sigma(x), \tau(x) \in \mathbb{Z}_p[x]$ sind eindeutig.

(Da Lösungen diophantischer Gleichungen unter diesen Voraussetzungen eindeutig sind. Siehe Geddes et al. Theorem 2.6 s.44)

Quadratisches Hensel Lifting

Die Idee von Hensel's Lifting lässt sich etwas allgemeiner formulieren: Quadratisches Hensel Lifting

- ▶ Sei R kommutativer Ring mit 1 (z. B. $R = \mathbb{Z}, R = F[y]$). $a, u, w \in R[x], m \in R$ mit $a \equiv uw \mod m$. Diese Faktorisierung soll nun geliftet werden zu Faktorisierung $a \equiv \hat{u}\hat{w} \mod m^2$.
- ▶ Annahme es gibt $s, t \in R[x]$ mit $su + tw \equiv 1 \mod m$ (z. B. wenn u, w teilerfremd mod m sind: Ist $R/\langle m \rangle$ Körper, so EEA in $R/\langle m \rangle[x]$ möglich).
- Setze

$$e = a - uw$$
, $\hat{u} = u + te$, $\hat{w} = w + se$: wie eben.
 $a - \hat{u}\hat{w} = a - uw - use - wte - ste^2$

$$= a - uw - (su + tw)e - ste2$$

= $(1 - su - tw)e - ste2 \equiv 0 \mod m2$

(Da $e \equiv 0 \mod m \pmod{1 - su - tw} \equiv 0 \mod m$).

Prof. Dr. K. Madlener: Computeralgebra

055

Newton's Iteration und Hensel's Konstruktion

Hensel Lifting

Quadratisches Hensel Lifting (Forts.)

- Startet man mit Primelement p für m, so kann man diesen Prozess induktiv (durch Mitliften der Kongruenz $su + tw \equiv 1$) fortsetzen, um Faktorisierungen bzgl. beliebiger Potenzen von p zu erhalten.
 - **5.17 Beispiel** Wir hatten Newton's Iteration zur Berechnung einer nicht-trivialen Lösung für $x^4 1 \equiv 0 \mod 625$ verwendet mit Startwert $x = 2 \mod 5$.

Dies kann als Lifting einer Faktorisierung gesehen werden: nämlich von $x^4 - 1 \equiv (x - 2)(x^3 + 2x^2 - x - 2) \mod 5$ zu Faktorisierung mod 625. Im obigen Kontext: $a = x^4 - 1$ p = 5

$$u = x^3 + 2x^2 - x - 2$$
 $w = x - 2$. Die Polynome u, w sind teilerfremd mod 5: EEA liefert $s = -2, t = 2x^2 - 2x - 1$, so dass $su + tw \equiv 1 \mod 5$.

Quadratisches Hensel Lifting (Forts.)

$$e = a - uw = x^{4} - 1 - x^{4} - 2x^{3} + x^{2} + 2x + 2x^{3} + 4x^{2} - 2x - 4 = 5x^{2} - 5$$

$$\hat{u} = u + te = 10x^{4} - 9x^{3} - 13x^{2} + 9x + 3$$

$$\hat{w} = w + se = -10x^{2} + x + 8$$

Und somit

$$a - \hat{u}\hat{w} = 25(4x^6 - 4x^5 - 8x^4 + 7x^3 + 5x^2 - 3x - 1) \equiv 0 \mod 25$$
, d. h. $a \equiv \hat{u}\hat{w} \mod 25$.

Problem: Grade von \hat{u} , \hat{w} sind größer als die von u bzw. w insbesondere ist ihre Summe > grad a.

▶ Dies geschieht z. B. wenn Vielfache von m Nullteiler mod m^2 sind und somit ist Produkt der Hauptkoeffizienten $\equiv 0 \mod m$.



Prof. Dr. K. Madlener: Computeralgebra

257

Newton's Iteration und Hensel's Konstruktion

Quadratisches Hensel Lifting (Forts.)

- ▶ Dies kann vermieden werden durch Verwendung der Division mit Rest in $\mathbb{R}[x]$. Sie ist möglich wenn der Divisor monisch ist:
- ► Altlemma:
 - i) $a,b\in R[x],\ b\neq 0$ monisch, dann gibt es eindeutige Polynome $q,r\in R[x]$ mit a=qb+r und grad r< grad b.
 - ii) Sind a, b, q, r wie in i) und gilt $a \equiv 0 \mod m$ für ein $m \in R$, so $q \equiv r \equiv 0 \mod m$.

(Beweis ii): Sei grad $b = n \ge 0 \implies$ aus $a \equiv 0 \mod m \ b \mod m$, d. h. $q \equiv 0 \mod m$ und somit auch $r \equiv 0 \mod m$.

▶ Überlegung führt zu: Algorithmus Hensel Schritt

Newton's Iteration und Hensel's Konstruktion

Hensel Lifting

```
//Eingabe: m \in R  a, u, w, s, t \in R[x] mit a \equiv uw \mod m, su + tw \equiv 1 \mod m, wobei w monisch, grad a = n = \operatorname{grad} u + \operatorname{grad} w, grad s < \operatorname{grad} w und grad t < \operatorname{grad} u. Ausgabe: Polynome u^*, w^*, s^*, t^* \in R[x] mit a \equiv u^*w^* \mod m^2 \text{ und } s^*u^* + t^*w^* \equiv 1 \mod m^2 wobei w^* monisch, u^* \equiv u \mod m, w^* \equiv w \mod m, s^* \equiv s \mod m, t^* \equiv t \mod m, grad u^* = \operatorname{grad} u, grad u^* = \operatorname{grad} u
```

- 1. Berechne $e, q, r, u^*, w^* \in R[x]$ mit grad r < grad w und $e \equiv a uw \mod m^2$, $se \equiv qw + r \mod m^2$ $u^* \equiv u + te + qu \mod m^2$. $w^* \equiv w + r \mod m^2$
- 2. Berechne $b, c, d, s^*, t^* \in R[x]$ mit grad $d < \text{grad } w^*$ und $b \equiv su^* + tw^* 1 \mod m^2, \quad sb \equiv cw^* + d \mod m^2$ $s^* \equiv s d \mod m^2, \quad t^* \equiv t tb cu^* \mod m^2$
- 3. Return u^*, w^*, s^*, t^* .

◆□▶ ◆□▶ ◆□▶ ◆□▶ ◆□ ◆○

Prof. Dr. K. Madlener: Computeralgebra

Newton's Iteration und Hensel's Konstruktion

Hensel Lifting

Algorithmus Hensel-Schritt: Korrektheit

5.18 Satz Der Algorithmus ist korrekt. Aufwand für:

- ▶ $R = \mathbb{Z} : O(M(n)M(\log m))$ Wortoperationen, falls Eingaben in max-Norm $< m^2$.
- ▶ $R = F[y] : O(M(n)M(\text{grad}_y m))$ Operationen in F Falls grad in y der Eingaben $< 2\text{grad}_y m$.

Beweis: Nachrechnen. \mathbb{Z} : Grad der Polynome $\leq n$. Koeffizienten $\leq m^4$ Länge $O(\log m)$. Division mit Rest O(M(n)).

Beispiel

5.19 Beispiel
$$a = x^4 - 1 = 0$$
 fortgesetzt. $u = x^3 + 2x^2 - x - 2$, $w = x - 2$, $s = -2$, $t = 2x^2 - 2x - 1$ $su + tw \equiv 1 \mod 5$, $e = a - uw = 5x^2 - 5$

►
$$se = -10x^2 + 10 : x - 2 = -10x + 5$$
 mit Rest $r = -5$
d. h. $q = -10x + 5$, $r = -5$ (mod 25)
 $u^* \equiv u + te + qu \equiv x^3 + 2x^2 - x - 2 + (2x^2 - 2x - 1)(5x^2 - 5) + (-10x + 5)(x^3 + 2x^2 - x - 2)$
 $\equiv x^3 + 7x^2 - x - 7 \mod 25$
 $w^* \equiv w + r \equiv x - 2 - 5 \equiv x - 7 \mod 25$.

- ▶ Dann $a \equiv u^*w^* \mod 25$. Die Grade von u^*, w^* sind die von u bzw. w und die Polynome sind einfacher als zuvor.
- ▶ 7 ist somit Lösung von $x^4 1 \equiv 0 \mod 25$ und $7 \equiv 2 \mod 5$ (Startlösung).



Prof. Dr. K. Madlener: Computeralgebra

26

Newton's Iteration und Hensel's Konstruktion

Beispiel (Forts.)

ightharpoonup Zur Berechnung von s^*, t^* die zur nächsten Iteration benötigt werden. Berechne

$$b \equiv su^* + tw^* - 1 \equiv -2x^3 + 11x^2 + 2x - 11 + (2x^2 - 2x - 1)(x - 7) - 1$$

$$\equiv -5x^2 - 10x - 5 \mod 25$$

$$sb = 10x^2 - 5x + 10 : x - 7 = 10x - 10$$
 Rest -10 mod 25
= = = d

- $ightharpoonup s^* \equiv s d \equiv 8 \mod 25$
- $t^* \equiv t tb cu^* \equiv -8x^2 12x 1 \mod 25$

Es gilt

 $s^*u^*+t^*w^*\equiv 1$ mod 25 und die Grade von s^*,t^* stimmen mit denen von s bzw. t überein.

Newton's Iteration und Hensel's Konstruktion

Hensel Lifting

Beispiel (Forts.)

5.20 Satz Hensel's Lemma Quadratisches Liften

Sei $I \in \mathbb{N}^+$ und es gelten die Eingangsbedingungen für Hensel-Schritt Algorithmus, dann lassen sich Polynome, die die Ausgabebedingungen erfüllen mit m^2 ersetzt durch m^I .

Beweis: Hensel-Schritt induktiv: m ersetzt durch m, m^2, m^4, \dots

5.21 Beispiel
$$x^4 - 1 \equiv 0$$
 Fortsetzung. Sei $m = 5$ $a, u_1 = u^*$ $w_1 = w^*$ $s_1 = s^*$ $t_1 = t^*$ in $\mathbb{Z}[x]$ aus Beispiel.

► Es gilt $a \equiv u_1 w_1 \mod 25$, $s_1 u_1 + t_1 w_1 \equiv 1 \mod 25$.



Prof. Dr. K. Madlener: Computeralgebra

Newton's Iteration und Hensel's Konstruktion

Hensel Lifting

Beispiel (Forts.)

Anwendung von Hensel-Schritt Algorithmus liefert.

- $e_2 \equiv a u_1 w_1 \equiv 50x^2 50 \mod 625.$
 - $q_2 \equiv -225x + 300 \mod 625$ und $r_2 \equiv -175 \mod 625$
 - $u_2 \equiv x^3 + 182x^2 x 182 \mod 625$ und $w_2 \equiv x 182 \mod 625$
- $b_2 \equiv s_1 u_2 + t_1 w_2 1 \equiv -225x^2 + 300x 25 \mod 625$
 - ▶ $c_2 \equiv 75x 200 \mod{625}$ und $d_2 \equiv 275 \mod{625}$
- ► $s_2 \equiv -267 \mod 625$
 - $t_2 \equiv 267x^2 312x 176 \mod 625$
- ▶ Dann $s_2u_2 + t_2w_2 \equiv 1 \mod 625$.
- ▶ $a \equiv u_2w_2 \mod 625$, d. h. 182 ist 4-Wurzel von 1 mod 625 kongruent zur Startlösung 2 mod 5.

Beispiel: Hensel Lifting

5.22 Beispiel Sei m = 3 $a = x^4 - 2x^3 - 11x^2 + 4x + 3 \in \mathbb{Z}[x]$. Dann gilt

- $a \equiv x(x+1)(x^2+1) \mod 3$, $u_0 = x^2 + xw_0 = x^2 + 1$ teilerfremd mod 3.
- $ightharpoonup s_0 = x + 1$ $t_0 = -x + 1$ $s_0 u_0 + t_0 w_0 \equiv 1 \mod 3$

Zwei Hensel Schritte liefern:

- $e_1 \equiv a u_0 w_0 \equiv -3x^3 3x^2 + 3x + 3 \mod 9$ $a_1 \equiv -3x^2 + 3x + 3 \mod 9$ $r_1 \equiv 3x \mod 9$
- $\mu_1 = x^2 + 4x + 3 \mod 9$
- $w_1 \equiv x^2 + 3x + 1$, $b_1 \equiv 3x^2 + 3$, $c_1 \equiv 3x + 3$, $d_1 \equiv 0 \mod 9$
- $ightharpoonup s_1 \equiv x + 1 \mod 9$
- $t_1 \equiv -x 2 \mod 9$ $e_2 \equiv a u_1 w_1 \equiv -9x^3 27x^2 9x \mod 81$ $a_2 \equiv -9x^2 - 9x \mod 81$ $r_2 \equiv 0 \mod 81$



Prof. Dr. K. Madlener: Computeralgebra

Newton's Iteration und Hensel's Konstruktion

Hensel Lifting

Beispiel (Forts.)

- $u_2 \equiv x^2 5x + 3 \mod 81$
- $w_2 \equiv x^2 + 3x + 1 \mod 81$ $b_2 \equiv -9x^2 9x \mod 81$ $c_2 \equiv -9x + 9 \mod 81$ $d_2 \equiv -27x - 9 \mod 81$
- $ightharpoonup s_2 \equiv 28x + 10 \mod 81$
- $t_2 \equiv -28x 29 \mod 81$

$$e_{3} = a - u_{2}w_{2} = x^{4} - 2x^{3} - 11x^{2} + 4x + 3$$

$$-\underbrace{(x^{2} - 5x + 3)(x^{2} + 3x + 1)}_{= -(x^{4} + 3x^{3} + x^{2} - 5x^{3} - 15x^{2} - 5x + 3x^{2} + 9x + 3)}_{= x^{4} - 2x^{3} - 11x^{2} + 4x + 3}$$

$$= 0$$

d.h. Wir erhalten sogar die Faktorisierung in $\mathbb{Z}[x]$, da u_2, w_2 irreduzibel in $\mathbb{Z}[x]$. 4 D > 4 B > 4 E > 4 E > E 990 Newton's Iteration und Hensel's Konstruktion

Hensel Lifting

Eindeutigkeit des Hensel-Liftings

5.23 Satz

Sei R Ring, $m \in R$ nicht Nullteiler, $l \in \mathbb{N}^+$. $u, w, u^*, w^*, s, t \in R[x]$ nicht Null mit $su + tw \equiv 1 \mod m$. Die Hauptkoeffizienten von u und w seien keine Nullteiler mod m, u und u^* (bzw. w und w^*) haben gleiche Hauptkoeffizienten, gleichen Grad und $u \equiv u^* \mod m$ bzw. $w \equiv w^* \mod m$. Gilt $uw \equiv u^*w^* \mod m^l$, so $u \equiv u^* \mod m^l$ und $w \equiv w^* \mod m^l$.

Beweis: Angenommen $u \not\equiv u^* \mod m^l$ oder $w \not\equiv w^* \mod m^l$. Wähle $1 \le i \le I$ maximal, so dass $m^i \mid u^* - u$ und $m^i \mid w^* - w$. D. h. $u^* - u = gm^i, w^* - w = hm^i$ $g, h \in R[x]$ und $m \nmid g$ oder $m \nmid h$. O.b.d.A. $m \nmid g$

$$0 \equiv u^*w^* - uw = u^*(w^* - w) + w(u^* - u)$$

= $(u^*h + wg)m^i \mod m^l$

Prof. Dr. K. Madlener: Computeralgebra

Newton's Iteration und Hensel's Konstruktion

Hensel Lifting

Eindeutigkeit des Hensel-Liftings (Forts.)

- ▶ Da *m* kein Nullteiler ist, gilt $m \mid m^{l-i} \mid (u^*h + wg)$.
- ▶ Bezeichne mit Reduktion mod m: Dann $\overline{su} + \overline{tw} = 1$, $\overline{u^*} = \overline{u}$, $\overline{u^*} \overline{h} + \overline{w} \overline{g} = 0$ also $0 = \overline{t}(\overline{u^*} \ \overline{h} + \overline{w} \ \overline{g}) = \overline{t} \ \overline{uh} + (1 - \overline{su})\overline{g}$ $= (\overline{t} \ \overline{h} - \overline{s}\overline{g}) \ \overline{u} + \overline{g}$, d. h. $\overline{u} \mid \overline{g}$
- ▶ Wegen $HK(u) = HK(u^*)$ und grad $u = \text{grad } u^*$ gilt grad $\overline{g} < \text{grad } \overline{u}$. Da $HK(\bar{u}) = \overline{HK(u)}$ kein Nullteiler ist auch \overline{u} kein Nullteiler und \overline{g} muss 0 Polynom sein. Widersprch zu $m \nmid g$.

Folgerung

5.24 Folgerung Sei R euklidisch, $p \in R$ Primelement. $l \in \mathbb{N}^+$, $f,g,u \in R[x]$ nicht Null mit $p \nmid \mathsf{HK}(f)$, $f \bmod p$ quadratfrei, $g \mid f$ in R[x], u monisch, nicht konstant mit $u \mid f \bmod p^l$, $u \mid g \bmod p$. Dann gilt $u \mid g \bmod p^l$

Beweis: Seien $h, v, w \in R[x]$ mit

 $f \equiv gh \equiv uw \mod p'$ und $g \equiv uv \mod p$. Da $f \mod p$ quadratfrei, ist auch $g \mod p$ quadratfrei und $\mathsf{GGT}(u \mod p, v \mod p) = 1$ in $\mathbb{F}_p[x]$.

- ▶ Hensel's Lemma liefert $u^*, v^* \in R[x]$, so dass $u^* \equiv u \mod p$, $v^* \equiv v \mod p$ und $g \equiv u^*v^* \mod p^l$.
- ▶ Wegen $uvh \equiv gh \equiv uw \mod p$ gilt $vh \equiv w \mod p$. Also $v^*h \equiv vh \equiv w \mod p$ und $u^*(v^*h) \equiv gh = f \equiv uw \mod p^l$.
- ▶ Da u, v teilerfremd mod p sind, liefert die Eindeutigkeit $u \equiv u^* \mod p^l$ und somit $g \equiv uv^* \mod p^l$, d. h. $u \mid g \mod p^l$.



Prof. Dr. K. Madlener: Computeralgebra

Prof. Dr. K. Madlener: Computeralgebra

269

Newton's Iteration und Hensel's Konstruktion

Hensel Lifting

Folgerung

Es gibt auch eine ∞-Version von Hensel's-Lemma.

P-adische Vervollständigung von R: für $p \in R$ prim (irreduzibel).

 $R_{(p)}$ Elemente $\sum_{i\geq 0} a_i p^i \quad 0 \leq a_i < p$, $(R = \mathbb{Z})$, sonst F[[y]] falls $p = y \quad R = F[y]$.

5.25 Satz ∞-Version Hensel's Lemma

Kongruenzen mod m^2 werden durch = in $R_{(p)}$ ersetzt.

► Lineare vs quadratische Iteration (Lifting)

$$p, p^2, p^3, \dots$$
 $p, p^2, p^4, p^8 \dots$

- Quadratische Iteration muss nicht effizienter als die lineare Iteration sein.
 Der Vorteil weniger Iterationen machen zu müssen wird durch Kosten für die Finzeliteration relativiert
- ► Vergleich durch Miola und Yun: Quadratische Iteration.

 Teuerer: Hauptsächlich wegen Berechnung von b, sb, s* und t*.

Beispiele

Frage: Lässt sich jede teilerfremde Faktorisierung von $a(x) = u_0(x)w_0(x)$ in $\mathbb{Z}_p[x]$ zu einer Faktorisierung in $\mathbb{Z}[x]$ liften? **Drei Beispiele**:

5.26 Beispiel 1

$$\begin{array}{ll} a(x) = x^3 + 10x^2 - 432x + 5040 \in \mathbb{Z}[x] & p = 5 \\ \Phi_5(a(x)) & = & x^3 - 2x \in \mathbb{Z}_5[x] \\ & = & x(x^2 - 2) \\ u_1(x) = x & w_1(x) = x^2 - 2 \text{ teilerfremd Hensel} \\ s(x) = -2x & t(x) = 2 & (-2x)x + 2(x^2 - 2) \equiv 1 \mod 5 \end{array}$$

► Lineare Iteration:

Iteration k am Ende $5^{k+1} \mid e(x)$.

∢ロト∢御と∢選と∢選と 選 め

Prof. Dr. K. Madlener: Computeralgebra

27

Newton's Iteration und Hensel's Konstruktion

Hensel Lifting

Beispiele (Forts.)

5.27 Beispiel 2 Divergenz der Hensel-Iteration $a(x) = x^4 + 1 \in \mathbb{Z}[x]$ ist irreduzibel über $\mathbb{Z}[x]$. Sei $p = 5 \leadsto \Phi_5(a(x)) = x^4 + 1$ Faktorisierung in $\mathbb{Z}_5[x]$.

▶ $x^4+1=(x^2+2)(x^2-2)\in \mathbb{Z}_5[x].$ $u_1(x)=x^2+2$ $w_1(x)=x^2-2$ sind teilerfremd in $\mathbb{Z}_5[x]$, d.h. Hensel Konstruktion kann angewendet werden. Konstruktion einer Folge von Faktoren mit

$$a(x) \equiv u^{(k)}(x)w^{(k)}(x) \mod p^k \text{ (bzw. } p^{2^k})k = 1, 2, 3, .$$

►
$$s(x) = -1$$
 $t(x) = 1$ $-1(x^2 + 2) + 1(x^2 - 2) \equiv 1 \mod 5$
 $a(x) - (x^2 + 2)(x^2 - 2) = x^4 + 1 - x^4 + 4 = 5 = e(x)$

Beispiele (Forts.)

Ende	$\sigma(x)$	$\tau(x)$	u(x)	w(x)	e(x)
lter.					

Nr.

▶ ∞-Folge Faktoren in $\mathbb{Z}_{5^k}[x]$ Ende von Iteration k gilt stets

$$u(x)w(x) \equiv x^4 + 1 \bmod 5^{k+1}$$

▶ Dies gilt sogar für jede Primzahl p.



4□ > 4問 > 4 = > 4 = > 9 Q ○

Prof. Dr. K. Madlener: Computeralgebra

Prof. Dr. K. Madlener: Computeralgebra

Newton's Iteration und Hensel's Konstruktion

Hensel Lifting

menser Lirting

Schranke für die Anzahl der Iterationen

► Apriori Schranke für die Anzahl der Iterationen:

 $B \ge \max\{|b|: b \text{ Koeffizienten in Polynom } a \text{ oder in jeden möglichen Faktor von } a \text{ mit Grad} \le \max\{\gcd(u^{(1)}), \gcd(w^{(1)})\}$

 $p^{l} > 2B$ Schranke für die Anzahl der Iterationen.

Newton's Iteration und Hensel's Konstruktion

Hensel Lifting

Beispiele (Forts.)

5.28 Beispiel 3 Das Leitkoeffizienten Problem

Nicht-monischer Fall:
$$c(x) = \frac{a(x) - u(x)w(x)}{p}$$

▶ $\sigma(x)u^{(1)}(x) + \tau(x)w^{(1)} \equiv c(x) \mod p$ Eindeutigkeit wird mit grad $\sigma(x) < \text{grad } (w^{(1)}(x))$ erreicht.

Updates:

$$u(x) := u(x) + \tau(x)p$$
 $w(x) := w(x) + \sigma(x)p$

→ Hauptkoeffizienten von w wird niemals verändert.

Im monischen Fall gilt auch grad $(\tau(x)) < \text{grad } (u^{(1)}(x))$

 \leadsto Hauptkoeffizienten von u wird ebenfalls niemals verändert.

i.A. grad
$$(c(x)) \le \text{grad } (a(x)) = \text{grad } (u^{(1)}(x)) + \text{grad } (w^{(1)}(x)),$$

d. h. grad
$$(\tau(x)) \leq \operatorname{grad}(u^{(1)}(x))$$
.

◆□▶◆圖▶◆圖▶◆圖▶ ■ め

Prof. Dr. K. Madlener: Computeralgebra

4 D > 4 B > 4 E > 4 E > E 990

Newton's Iteration und Hensel's Konstruktion

Hensel Lifting

Beispiele (Forts.)

► Alle Veränderungen vom Hauptkoeffizienten müssen in u realisiert werden.

$$a(x) = 12x^3 + 10x^2 - 36x + 35 \in \mathbb{Z}[x]$$

= $u(x)w(x) = (2x + 5)(6x^2 - 10x + 7) \in \mathbb{Z}[x]$

2 ist Einheit in $\mathbb{Z}_5[x]$.

Wahl der Anfangsfaktoren: 2 zum Faktor x oder 2 zum Faktor $x^2 + 2$

d.h.
$$\Phi_5(a(x)) = (2x)(x^2 + 2) = (x)(2x^2 - 1) \in \mathbb{Z}_5[x]$$

► Die richtigen Faktoren sind

$$u^{(1)}(x) = 2x$$
 und $w^{(1)}(x) = x^2 + 2$

▶ Hensel's Konstruktion: s(x) = x t(x) = -2

$$u(x) = 2x$$
 $w(x) = x^2 + 2$ $e(x) = 10x^3 + 10x^2 - 40x + 35 \mod 5$

Beispiele (Forts.)

$\sigma(x)$	$\tau(x)$	u(x)	w(x)	e(x)
		2	2 . 0	10 3 . 10 2 . 40 25
_	_	2 <i>x</i>	$x^{2} + 2$	$10x^3 + 10x^2 - 40x + 35$
-2x - 1	2x + 1	12x + 5	$x^2 - 10x - 3$	$125x^2 + 50x + 50$
2x + 1	1	12x + 30	$x^2 + 40x + 22$	$-500x^2 - 1500x - 625$
-2x - 2	0	12x + 30	$x^2 - 210x - 103$	$2500x^2 + 7500x + 3125$
2x + 1	0	12x + 30	$x^2 + 1040x + 522$	$-12500x^2 - 37500x - 15625$

- ► Aufsplittung von 12 in 2 · 6? d. h.
- $u^{(3)}(x) = 12x + 30$ $w^{(3)}(x) = x^2 + 40x + 22$



Prof. Dr. K. Madlener: Computeralgebra

277

Newton's Iteration und Hensel's Konstruktion

Hensel Lifting

Beispiele (Forts.)

Prof. Dr. K. Madlener: Computeralgebra

► Sind bis auf Einheiten die richtigen Faktoren

$$u^{(3)}(x)w^{(3)}(x) = 12x^3 + 510x^2 + 1464x + 660 \in \mathbb{Z}[x]$$

- ▶ $6^{-1}u^{(3)}(x) \mod \mathbb{Z}_{125}[x]$ $6w^{(3)}(x) \mod \mathbb{Z}_{125}[x]$ $6^{-1} = 21 \rightsquigarrow u(x) = 21u^{(3)}(x) = 2x + 5 \in \mathbb{Z}_{125}[x]$ $u(x) = 6w^{(3)}(x) = 6x^2 - 10x + 7 \in \mathbb{Z}_{125}[x]$
- ▶ Verwendet wird: 6 ist der richtige LK von w.
- ▶ Beachte: $a \in \mathbb{Z}_{p^k}$ ist Einheit $gdw p \nmid a$ in \mathbb{Z} .

Liften von Faktorisierungen

5.29 Satz Sei $a(x) \in \mathbb{Z}[x]$, p Primelement in \mathbb{Z} , $p \nmid \mathsf{HKoeff}(a(x))$.

- Seien $u^{(1)}(x), w^{(1)}(x) \in \mathbb{Z}_p[x]$ teilerfremd über \mathbb{Z}_p mit $a(x) \equiv u^{(1)}(x)w^{(1)}(x) \bmod p$.
- Seien $u^{(k)}(x), w^{(k)}(x)$, die von der Hensel Konstruktion bestimmten Faktoren mit $a(x) \equiv u^{(k)}(x)w^{(k)}(x) \mod p^k$, $u^{(k)}(x) \equiv u^{(1)}(x) \mod p$, $w^{(k)}(x) \equiv w^{(1)}(x) \mod p$.
- ▶ Gibt es Polynome $u(x), w(x) \in \mathbb{Z}[x]$ mit a(x) = u(x)w(x) in $\mathbb{Z}[x]$ und $n(u(x)) \equiv n(u^{(1)}(x))$ mod p bzw. $n(w(x)) \equiv n(w^{(1)}(x))$ mod p, wobei n die Normalisierung "mache Polynom monisch als Element von $\mathbb{Z}_p[x]$ ".
- ▶ Dann sind für alle $k \ge 1$ die Polynome $\Phi_{p^k}(u(x))$ und $u^{(k)}(x)$, sowie $\Phi_{p^k}(w(x))$ und $w^{(k)}(x)$ assoziiert im Ring $\mathbb{Z}_{p^k}[x]$.

Prof. Dr. K. Madlener: Computeralgebra

Prof. Dr. K. Madlener: Computeralgebra

< E ▶ -E *)

Newton's Iteration und Hensel's Konstruktion

Hensel Lifting

Liften von Faktorisierungen (Beweis)

Beweis: Sei $k \ge 1$. Nach Vor $p \nmid HKoeff(a(x))$, d. h. HKoeff(a(x)) ist Einheit in $\mathbb{Z}_{p^k}[x]$.

 $\bar{a}(x) = \mathsf{HKoeff}(a(x))^{-1}a(x) \in \mathbb{Z}_{p^k}[x]$ ist monisch. Wegen

$$\mathsf{HKoeff}(a(x)) \equiv \mathsf{HKoeff}(u^{(k)}(x))\mathsf{HKoeff}(w^{(k)}(x)) \bmod p^k$$

ist p kein Teiler der HKoeff, d. h. sie sind Einheiten $\text{mod } p^k$.

- $\bar{u}^{(k)} := \mathsf{HKoeff}(u^{(k)}(x))^{-1}u^{(k)}(x) \in \mathbb{Z}_{p^k}[x] \text{ und}$
- $\bar{w}^{(k)} := \mathsf{HKoeff}(w^{(k)}(x))^{-1}w^{(k)}(x) \in \mathbb{Z}_{p^k}[x] \text{ sind monisch.}$
- ▶ Die Voraussetzungen sind auch für $\bar{a}, \bar{u}^{(k)}, \bar{w}^{(k)}$ erfüllt und somit sind die $\bar{u}^{(k)}, \bar{w}^{(k)}$ eindeutig bestimmt.

<ロ > < 個 > < 量 > < 達 > り へ き > ・ 達 ・ り へ で

Newton's Iteration und Hensel's Konstruktion

Liften von Faktorisierungen (Beweis)

- Angenommen es gibt u(x), w(x) mit a(x) = u(x)w(x) in $\mathbb{Z}[x]$. $n(u(x)) \equiv n(u^{(1)}(x)) \mod p, \ n(w(x)) \equiv n(w^{(1)}(x)) \mod p.$
- ▶ Betrachte $\Phi_{n^k}(u(x))$ und $\Phi_{n^k}(w(x)) \in \mathbb{Z}_{n^k}[x]$. Wie eben kann man diese in $\mathbb{Z}_{n^k}[x]$ normalisieren zu $\bar{u}(x), \bar{w}(x)$.
- ▶ Wegen der Eindeutigkeit gilt dann: $\bar{u}^{(k)}(x) = \bar{u}(x)$ $\bar{w}^{(k)}(x) = \bar{w}(x)$ in $\mathbb{Z}_{n^k}[x]$ d. h.
- $u^{(k)}(x)$ und $\Phi_{p^k}(u(x))$ sind assoziiert im Ring $\mathbb{Z}_{p^k}[x]$,
- ▶ Analog für $w^{(k)}(x)$ und $\Phi_{n^k}(w(x))$.

Prof. Dr. K. Madlener: Computeralgebra

Newton's Iteration und Hensel's Konstruktion

Hensel Lifting

Beispiel (Forts.)

- **5.30 Beispiel** $a(x) = 12x^3 + 10x^2 36x + 35 \in \mathbb{Z}[x]$ $= u(x)w(x) = (2x+5)(6x^2-10x+7) \in \mathbb{Z}[x]$
- $\Phi_{\rm F}(a(x)) = (2x) \cdot (x^2 + 2) = u^{(1)}(x) \cdot w^{(1)}(x)$
- $\Phi_5(u(x)) = \Phi_5(2x+5) = 2x$
- $\Phi_5(w(x)) = \Phi_5(6x^2 10x + 7) = x^2 + 2$
- $u^{(2)}(x) = 12x + 5$ $w^{(2)}(x) = x^2 10x 3$
- $\Phi_{25}(u(x)) = 2x + 5$ $\Phi_{25}(w(x)) = 6x^2 10x + 7$ Unterscheiden sich um Einheiten, z. B. $-4(6x^2 - 10x + 7)$ $\equiv x^2 - 10x - 3$ -4(12x + 5) = 2x + 5
- $w^{(3)}(x) = x^2 + 40x + 22$ $u^{(3)}(x) = 12x + 30$
- $\Phi_{125}(w(x)) = 6x^2 10x + 7$ $\Phi_{125}(u(x)) = 2x + 5$ $6^{-1}u^{(3)}(x) = 21u^{(3)}(x) = 2x + 5$, $6w^{(3)}(x) = 6x^2 - 10x + 7$

4□ > 4個 > 4 = > 4 = > = 90

Modifizierte Hensel Konstruktion: nicht monischer Fall

▶ Sei $a(x) \in \mathbb{Z}[x]$ primitiv, $a(x) \equiv u^{(1)}(x)w^{(1)}(x)$ mod p und es gebe $u(x), w(x) \text{ mit } a(x) = u(x)w(x) \in \mathbb{Z}[x].$

Es gelte $n(u^{(1)}(x)) \equiv n(u(x)) \mod p$, $n(w^{(1)}(x)) \equiv n(w(x)) \mod p$.

- $ightharpoonup \alpha = \mathsf{HKoeff}(a(x)), \ \mu = \mathsf{HKoeff}(u(x)), \ \nu = \mathsf{HKoeff}(w(x)),$
- $ho = \mu \nu$. Sei $\bar{a}(x) = \alpha a(x)$. Suche Faktorisierung von $\bar{a}(x) = \mu \nu u(x) w(x) = [\nu u(x)] [\mu w(x)] = \bar{u}(x) \bar{w}(x).$
- ▶ Die Hauptkoeffizienten der Faktoren sind bekannt, beide = α . Modifiziere die Hensel Konstruktion so. dass

$$\bar{u}^{(k)}(x) = \Phi_{p^k}(\bar{u}(x)) \quad \bar{w}^{(k)} = \Phi_{p^k}(\bar{w}(x))$$

$$\begin{split} \bar{u}^{(k)}(x) &= \Phi_{p^k}(\alpha \mathsf{HKoeff}(u^{(k)})^{-1}u^{(k)}(x)), \\ \bar{w}^{(k)}(x) &= \Phi_{p^k}(\alpha \mathsf{HKoeff}(w^{(k)})^{-1}w^{(k)}(x)) \\ & \leadsto u(x) = pp(\bar{u}(x)) \text{ und } w(x) = pp(\bar{w}(x)) \end{split}$$

Prof. Dr. K. Madlener: Computeralgebra

Newton's Iteration und Hensel's Konstruktion

Hensel Lifting

Modifizierte Hensel Konstruktion: nicht monischer Fall

5.31 Beispiel

$$a(x) = 12x^3 + 10x^2 - 36x + 35 \in \mathbb{Z}[x]$$
 primitiv.
 $\equiv u^{(1)}(x)w^{(1)}(x) \mod 5$

- Mit $u^{(1)}(x) = 2x$, $w^{(1)}(x) = x^2 + 2$.
- $\bar{a}(x) = 12a(x) = 144x^3 + 120x^2 432x + 420$ $\bar{a}(x) \equiv \bar{u}^{(1)}(x)\bar{w}^{(1)}(x) \mod 5 \text{ mit}$ $\bar{u}^{(1)}(x) = \Phi_5(12\ 2^{-1}(2x)) = 2x$ $\bar{w}^{(1)}(x) = \Phi_5(12 \ 1^{-1}(x^2 + 2)) = 2x^2 - 1$
- ▶ Hensel Schritt auf $\bar{a}(x)$. $u^{(2)}(x) = \bar{u}^{(1)}(x) + (-x+1)5 = -3x+5$ $w^{(2)}(x) = \bar{w}^{(1)}(x) + (x-1)5 = 2x^2 + 5 - 6$
- $\bar{u}^{(2)}(x) = \Phi_{25}(12 (-3)^{-1}(-3x+5)) = 12x+5$ $\bar{w}^{(2)}(x) = \Phi_{25}(12\ 2^{-1}(2x^2 + 5x - 6)) = 12x^2 + 5x - 11$

 $w^{(3)}(x) = \bar{w}^{(2)}(x) + (-x+1)5^2 = 12x^2 - 20x + 14$

$$\bar{u}^{(3)}(x) = u^{(3)}(x) = 12x + 30$$
$$\bar{w}^{(3)}(x) = w^{(3)}(x) = 12x^2 - 20x + 14$$

▶ Dann $\bar{a}(x) - \bar{u}^{(3)}(x)\bar{w}^{(3)}(x) = 0$ also Faktorisierung von $\bar{a}(x)$.

$$u(x) = pp(\bar{u}^{(3)}(x)) = 2x + 5$$

$$w(x) = pp(\bar{w}^{(3)}(x)) = 6x^2 - 10x + 7$$

- ▶ Weitere Methoden zur Vermeidung bzw. Lösung des HKoeff-Problems mit kleineren Multiplikatoren.
- ▶ Siehe *G*, *C*, *L* Kap. 6. Insbesondere die Replace_LC Operation (Yun).

285

Newton's Iteration und Hensel's Konstruktion

Prof. Dr. K. Madlener: Computeralgebra

Multifaktor Hensel Lifting. Algorithmus nach Zassenhaus

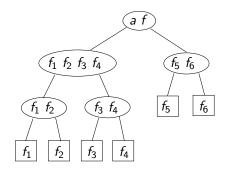
Multifaktor Hensel Lifting

- ▶ Lifte Faktorisierung in mehr als zwei Faktoren.
- ▶ Sei R Ring, $m \in R$, $f, f_1, \ldots, f_r \in R[x]$, so dass $\mathsf{HKoeff}(f)$ Einheit $\mathsf{mod}\, m$ ist, f_1, \ldots, f_r monisch und $f \equiv \mathsf{HKoeff}(f) f_1 \ldots f_r \bmod m$. Dann gibt es $a \in R$ mit $a \cdot \mathsf{HKoeff}(f) \equiv 1 \bmod m$
- ▶ Ordne die monischen Faktoren ν von f modulo m als Binärbaum τ der Tiefe $d = \lceil \log_2 r \rceil$, mit Blättern f_1, \ldots, f_r , Wurzel af, so dass jeder innere Knoten Produkt seiner beiden Söhne modulo m ist.
- ▶ Offenbar gibt es mehrere Möglichkeiten ein Polynom in Zweierfaktoren zu zerlegen. Problem der kombinatorischen Explosion.

Multifaktor Hensel Lifting. Algorithmus nach Zassenhaus

Multifaktor Hensel Lifting (Forts.)

▶ Mehrere Anordnungen sind möglich. z. B. r = 6.



Prof. Dr. K. Madlener: Computeralgebra

Newton's Iteration und Hensel's Konstruktion

Multifaktor Hensel Lifting. Algorithmus nach Zassenhaus

Multifaktor Hensel Lifting (Forts.)

- ▶ Gibt es jeden inneren Knoten $\nu \in R[x]$ mit Söhnen $g_{\nu}, h_{\nu} \in R[x]$, Polynome $s_{\nu}, t_{\nu} \in R[x]$ mit grad $s_{\nu} <$ grad h_{ν} , grad $t_{\nu} <$ grad g_{ν} und
- ► $s_{\nu}g_{\nu} + t_{\nu}h_{\nu} \equiv 1 \mod m$, so heißt τ ein Faktorbaum von f modulo m (existiert stets falls f quadratfrei ist).
- ▶ Man erhält einen Faktorbaum τ_2 von f modulo m^2 durch Anwendung des Hensel Schritts von der Wurzel hin zu den Blättern. Ist $R/\langle m \rangle$ Körper so O.K

4 D > 4 B > 4 E > 4 E > 9 9 0

Prof. Dr. K. Madlener: Computeralgebra 286 Prof. Dr. K. Madlener: Computeralgebra

Multifaktor Hensel Lifting. Algorithmus nach Zassenhaus

Algorithmus Liften eines Faktorisierungsbaumes mod m

Eingabe: $m \in R$, $f \in R[x]$ mit Grad n, $a_0 \in R$ mit $a_0HKoeff(f) \equiv 1 \mod m$, $I \in \mathbb{N}$, Faktorbaum τ für $f \mod m$ mit Wurzel a_0f und f Blätter

Ausgabe: Eine Inverse $a^* \in R$ von $HKoeff(f) \mod m^l$ und ein Faktorbaum τ^* von f modulo m^l mit Wurzel a^*f , so dass jeder Knoten $\nu^* \in R[x]$ von τ^* kongruent modulo m zum entsprechenden Knoten $\nu \in R[x]$ von τ

Beachte: Baumstruktur bleibt unverändert.

Prof. Dr. K. Madlener: Computeralgebra

289

Newton's Iteration und Hensel's Konstruktion

Multifaktor Hensel Lifting. Algorithmus nach Zassenhaus

Algorithmus Multifaktor Hensel Lifting (MFHL)

 $1 d := \lceil \log_2 I \rceil, \tau_0 := \tau$ 2 for $j = 1 \dots d$ do

3 {Lifte Inverse von HKoeff(f)}

 $a_j :\equiv 2a_{j-1} - HKoeff(f)a_{j-1}^2 \mod m^{2^j};$ $\tau_i := \tau_{i-1};$

Ersetze Wurzel von τ_i durch $a_i f$

4 {Lifte Baum}

for jeden inneren Knoten $\nu \in R[x]$ von

 au_{j} von der Wurzel abwärts ${f do}$

5 Call Hensel_Schritt_Alg mit $m^{2^{j-1}}$ um die Kongruenzen $\nu \equiv g_{\nu}h_{\nu}$ und $s_{\nu}g_{\nu}+t_{\nu}h_{\nu}\equiv 1 \bmod m^{2^{j-1}}$ zu Kongruenzen modulo $m^{2^{j}}$ zu Liften

6 **return** a_d und τ_d

4□ > 4□ > 4□ > 4□ > 4□ > 4□

Multifaktor Hensel Lifting. Algorithmus nach Zassenhaus

Algorithmus Multifaktor Hensel Lifting (MFHL)

5.32 Satz Der Algorithmus MFHL ist korrekt bzgl. seiner spec. Er benötigt $0(M(n)\log rM(I\log m))$ Wortoperationen, falls $R=\mathbb{Z},\ m>1$ alle Eingaben mit max_Norm $< m^I$ und

 $O(M(n)\log rM(I\operatorname{grad}_y m))$ Operationen in F, falls R=F[y], Körper F und y-Grad aller Eingaben kleiner als $I\operatorname{grad}_y m$. Beweis:

Schritt 3 ist korrekt nach inversen Berechnung, nach Newton-Quadrat-Iteration; d. h. die Wurzel von τ_j ist der monische Vielfache von $f \mod m^{2^j}$.

- ▶ Durch Induktion nach j zeige τ_j ist Faktorbaum von f mod m^{2^j} und jeder Knoten von τ_i ist kongruent mod m zum entsprechenden Knoten von τ .
 - i = 0 klar.
 - ▶ $j \ge 1$. Wurzel- $\tau_i \equiv \text{Wurzel-}\tau_{i-1} \mod m$.

291

Newton's Iteration und Hensel's Konstruktion

Prof. Dr. K. Madlener: Computeralgebra

Multifaktor Hensel Lifting. Algorithmus nach Zassenhaus

Algorithmus MFHL Korrektheit(Forts.)

- ▶ Behauptung über τ_j folgt nun durch Induktion über Baum und Korrektheit von Hensel Schritt. Wegen $l \leq 2^d$ ist τ_d auch Faktorbaum für $f \mod m^l$.
- ▶ $R = \mathbb{Z}$: Reduktion der Koeffizienten von f modulo m, m^2 , m^4 , . . . kann mit $O(nM(I \log m))$ Wortoperationen durchgeführt werden.
- ▶ Schritt 5 kostet $0(M(\text{grad } v)M(2^j \log m))$ Wortoperationen
- ▶ In einer Stufe von τ_j ist die Summe der Grade aller Knoten höchstens n, d h. die Kosten für diese Stufe ist höchstens $0(M(n)M(2^j\log m))$ Wortoperationen.
- ▶ Es gibt $d \in O(\log r)$ Stufen und die Kosten von 4 und 5 für festes j ist $O(M(n)\log rM(2^j\log m))$ Wortoperationen. Dies dominiert Schritt 3. Behauptung folgt aus $\sum_{1\leq i\leq d}2^j\leq 4I$.
- ▶ Durch balancieren des Faktorbaums bzgl. Grad lässt sich der Faktor log r durch die Entropie $H(n_1/n, ..., n_r/n)(n_i = \text{grad } f_i)$ ersetzen.

Prof. Dr. K. Madlener: Computeralgebra 292

Multifaktor Hensel Lifting. Algorithmus nach Zassenhaus

Beispiel $\mod 5$ Liften 1. Stufe $x^4 - 1$ -2x - 1 2x - 1-2x + 9 2x + 9 $x^2 + 2x + 2$ $x^{2} + 8x +$ -2x + 2 $x^{2} - 8x +$ -11 1 $\sqrt{x+2}$ Liften 2. Stufe mod 25 Liften 1. Stufe $x^4 - 1$ -2x + 9 2x + 9-202x - 91 | 202x - 91 $x^{2} - 8x +$ $x^{2} + 8x +$ -183x+182+183x+182-4 -4 4 Liften 2. Stufe (x + 7)(x+1)

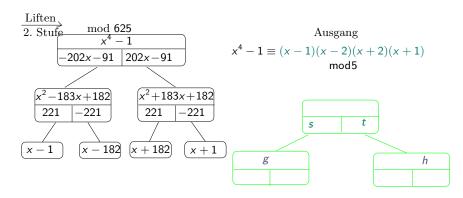
Prof. Dr. K. Madlener: Computeralgebra

293

Newton's Iteration und Hensel's Konstruktion

Multifaktor Hensel Lifting. Algorithmus nach Zassenhaus

Beispiel(Forts.)



$sg + th \equiv 1 \text{ Mod}$

Faktorisierung in $\mathbb{Z}[x]$ mit quadratischem Hensel Lifting. Der Algorithmus Faktorisierung in $\mathbb{Z}[x]$ nach Zassenhaus

```
//Eingabe: f \in \mathbb{Z}[x] quadratfrei, primitiv, grad n \ge 1 mit HKoeff(f) > 0, max_norm f = A. Ausgabe: Irreduzible Faktoren \{f_1, \ldots, f_k\} \subseteq \mathbb{Z}[x] von f//

1 if n = 1 then return\{f\}
b := HKoeff(f); B := (n+1)^{1/2}2^nAb;
c := (n+1)^{2n}A^{2n-1}; \gamma := \lceil 2\log_2 c \rceil;

2 repeat wähle Primzahl p \le 2\gamma \ln \gamma, \bar{f} := f \mod p
until p \nmid b and \bar{f} quadratfrei in \mathbb{F}_p[x]

I := \lceil \log_p(2B+1) \rceil

3 \{ \text{Modulare Faktorisierung} \}
Berechne h_1, \ldots, h_r \in \mathbb{Z}[x] mit max_norm höchstens p/2 die nicht konstant, monisch und irreduzibel modulo p mit f \equiv bh_1 \cdots h_r \mod p
```

Prof. Dr. K. Madlener: Computeralgebra

Newton's Iteration und Hensel's Konstruktion

Multifaktor Hensel Lifting. Algorithmus nach Zassenhaus

Algorithmus (Forts.)

4 {Hensel Lifting}

Prof. Dr. K. Madlener: Computeralgebra

 $a := b^{-1} \mod p$

Verwende EEA in $\mathbb{F}_p[x]$ um Faktorbaum für f modulo p mit Blätter $h_1\cdots h_r$ zu bestimmen

Call MFHL um Faktorisierung $f \equiv bg_1 \cdots g_r \mod p^l$ mit monischen Polynome $g_1, \ldots, g_r \in \mathbb{Z}[x]$ mit max_norm höchstens $p^l/2$ so dass $g_i \equiv h_i \mod p$ $(1 \leq i \leq r)$ zu berechnen

5 {Initialisiere die Indexmenge T der modularen Faktoren, die noch behandelt werden müssen, die Menge G der gefundenen Faktoren, sowie Restpolynom das noch faktorisiert werden muss f^* }

$$T := \{1, \ldots, r\}; s := 1; G := \emptyset; f^* := f;$$

Multifaktor Hensel Lifting. Algorithmus nach Zassenhaus

Algorithmus (Forts.)

4 D > 4 B > 4 E > 4 E > 9 Q Q

Prof. Dr. K. Madlener: Computeralgebra

297

Newton's Iteration und Hensel's Konstruktion

11 return $G \cup \{f^*\}$

Multifaktor Hensel Lifting. Algorithmus nach Zassenhaus

Algorithmus (Forts.)

- Hierbei ist $\|f\|_1 = \sum_{0 \le 1 \le \mathsf{grad}\ f} |f_i|, \ \|f_1\|_\infty \le \|f\|_1 \le (n+1)\|f\|_\infty$
- ▶ $\|g^*\|_1 \|h^*\|_1 \le B$ gdw. $g^*h^* = bf^*$ " \sim " Mignotes Schranke (vzG. S. 156). " \sim " wegen $g^*h^* \equiv bf^* \mod p^l$. $\|g^*h^*\|_{\infty} \le \|g^*h^*\|_1 \le \|g^*\|_1 \|h^*\|_1 \le B < p^{l/2}$, d. h. | alle Koeff. | $< p^l/2 \sim$ gleich.
 - 5.33 Satz (Beweis später). Der Algorithmus ist korrekt, Kosten später.

Algorithmus Zassenhaus: Beispiel

5.34 Beispiel $f = 6x^4 + 5x^3 + 15x^2 + 5x + 4 \in \mathbb{Z}[x]$.

Wähle p = 5, $\bar{f} = x^4 - 1$ mit $f \equiv \bar{f} \mod 5$.

 \bar{f} ist quadratfrei in $\mathbb{Z}_5[x]$. $B := \sqrt{5} \cdot 2^4 \cdot 15 \cdot 6 \approx 3220$,

▶ $I = \lceil \log_5(2B+1) \rceil = 6.$

Newton's Iteration und Hensel's Konstruktion

Schritt 3: modulare Faktorisierung

$$f \equiv bh_1h_2h_3h_4 = 1(x-1)(x+1)(x-2)(x+2) \mod 5$$

Schritt 4: Liften eines Faktorbaumes für f mod 5 zu Faktorbaum für f modulo 5⁶ (mod 5⁶ aus Schranke I).

4□ > 4률 > 4혈 > 4혈 > 4 혈 > 4 혈

Prof. Dr. K. Madlener: Computeralgebra

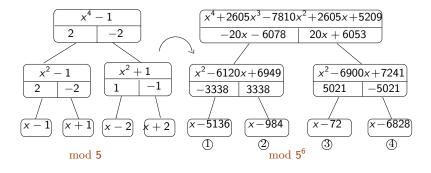
299

Newton's Iteration und Hensel's Konstruktion

Multifaktor Hensel Lifting. Algorithmus nach Zassenhaus

Algorithmus Zassenhaus: Beispiel (Forts.)

Schritt 4:



Newton's Iteration und Hensel's Konstruktion Multivariate Verallgemeinerung von Hensel's Lemma

Algorithmus Zassenhaus: Beispiel (Forts.)

▶ Teilmengen von $S \subseteq \{1, 2, 3, 4\}$ mit S = 1: Keine Faktorisierung.

$$S = \{1,3\} : g^* \equiv bg_1g_3 = 6(x - 5136)(x - 72)$$

$$\equiv 6x^2 + 2x + 2 \mod 5^6$$

$$h^* \equiv bg_2g_4 = 6(x - 984)(x - 6828) \equiv 6x^2 + 3x + 12 \mod 5^6.$$

$$\|g^*\|_1\|h\|_1 \le B$$
, d.h. $g^*h^* = bf^*$.

 $pp(g^*) = 3x^2 + x + 1$ $pp(h^*) = 2x^2 + x + 4$ sind die irreduziblen Faktoren von f.



Prof. Dr. K. Madlener: Computeralgebra

Newton's Iteration und Hensel's Konstruktion

Multivariate Verallgemeinerung von Hensel's Lemma

Multivariate Verallgemeinerung von Hensel's Lemma

Problemstellung: Finde $u, w \in \mathbb{Z}[x_1, \dots, x_{\nu}]$ mit

$$F(u,w)=a(x_1,\ldots,x_{\nu})-uw=0$$

und

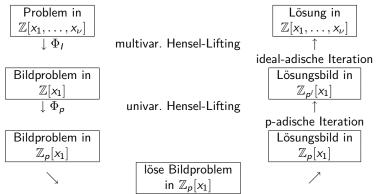
$$u(x_1, \ldots, x_{\nu}) \equiv u_0(x_1) \mod \langle I, p \rangle$$

 $w(x_1, \ldots, x_{\nu}) \equiv w_0(x_1) \mod \langle I, p \rangle$

Bei Geg. u_0, w_0 mit $a \equiv u_0 w_0 \mod \langle I, p \rangle$

$$I = \langle x_2 - \alpha_2, \dots, x_{\nu} - \alpha_{\nu} \rangle, p$$
 Primzahl.

Homomorphismus Diagramm



Beachte den Unterschied zum Lifting von F(u) = 0 via Newton Iteration: Φ_I und Φ_p vertauscht. Bei p-adischer Iteration steht $\mathbb{Z}_{p^I}[x_1]$ und nicht $\mathbb{Z}[x_1]$: Hierbei muss / groß genug gewählt werden. Trennung vom multivariaten und univariaten Fall.

Prof. Dr. K. Madlener: Computeralgebra

Newton's Iteration und Hensel's Konstruktion

Multivariate Verallgemeinerung von Hensel's Lemma

Multivariate Verallgemeinerung von Hensel's Lemma

Die Grundoperation bei der Hensel Iteration zur Lösung von $F(u, w) = a(x_1, \dots, x_{\nu}) - uw = 0$ ist die Bestimmung der Lösungen einer polynomialen diophantischen Gleichung der Form

(*)
$$A^{(k)} \Delta u^{(k)} + B^{(k)} \Delta w^{(k)} = C^{(k)}$$

Für Korrekturterme $\Delta u^{(k)}$, $\Delta w^{(k)}$ mit $A^{(k)}$, $B^{(k)}$ und $C^{(k)}$ gegebene Polynome.

- ▶ Bei Vertauschung von Φ_I , Φ_D müssten für das I-adische Lifting die Gleichungen in $\mathbb{Z}_p[x_1]$ gelöst werden und für das p-adische Lifting in $\mathbb{Z}_p[x_1,\ldots,x_{\nu}].$
- ▶ Die Lösung von (*) im euklidischen Bereich $\mathbb{Z}_p[x]$ ist einfach, aber in $\mathbb{Z}_p[x_1,\ldots,x_{\nu}]$ schwierig (kein euklidischer Bereich).

Prof. Dr. K. Madlener: Computeralgebra

Newton's Iteration und Hensel's Konstruktion

Lösung diophantischer Polynomgleichungen in $\mathbb{Z}_{-1}[x_1]$

Multivariate Verallgemeinerung von Hensel's Lemma

- ▶ Bei der vorgeschlagenen Anordnung erst p-adische Iteration, dann ideal-adische Iteration wird die Lösung "einfacher". Die Gleichung (*) muss dann im Ring $\mathbb{Z}_{p'}[x_1]$ gelöst werden. Dieser Ring ist zwar kein euklidischer Ring aber "fast", da $\mathbb{Z}_{p'}$ fast ein Körper ist (die Nullteiler sind bekannt und auch die invertierbaren Elemente).
- ▶ D. h. man kann EEA verwenden. Wählt man p richtig, so lässt sich die Gleichung (*) lösen und die Lösung aus $\mathbb{Z}_{p'}[x_1]$ zu Lösung in $\mathbb{Z}[x_1,\ldots,x_{\nu}]$ liften.
- ▶ Problem: richtige Wahl der α_i , d. h. Nullstellen um keine wichtigen Informationen zu verlieren.

<ロ > < 個 > < 国 > < 重 > く 重 > く 重 > り へ ②

Prof. Dr. K. Madlener: Computeralgebra

Newton's Iteration und Hensel's Konstruktion

305

Prof. Dr. K. Madlener: Computeralgebra

Lösung diophantischer Polynomgleichungen in Z , [x1]

Consider the Control of the Control

Lösung diophantischer Polynomgleichungen in $\mathbb{Z}_{p'}[x_1]$

Wende Newton's Iteration um Lösung in $\mathbb{Z}_p[x_1]$ zu Lösung in $\mathbb{Z}_{p'}[x_1]$ zu liften

Problem: Finde Polynome $s^{(l)}(x_1), t^{(l)}(x_1) \in \mathbb{Z}_{p^l}[x_1]$, die die Gleichung

(*)
$$s^{(I)}(x_1)u(x_1) + t^{(I)}(x_1)w(x_1) \equiv 1 \mod p^I$$

mit $u(x_1), w(x_1) \in \mathbb{Z}_{p'}[x_1]$ Polynome, so dass $\Phi_p(u(x_1)), \Phi_p(w(x_1))$ teilerfremd in $\mathbb{Z}_p[x_1]$, d. h. Newton's Iteration wird auf

$$G(s,t) = s \cdot u(x_1) + t \cdot w(x_1) - 1 = 0$$
 angewendet.

Wie beim Hensel's Einzelschrittverfahren kann die Lösung der diophantischen Gleichung von $\mathbb{Z}_p[x_1]$ nach $\mathbb{Z}_{p'}[x_1]$ geliftet werden.

Lineare oder quadratische Iteration.

Lösung diophantischer Polynomgleichungen in $\mathbb{Z}_{p'}[x_1]$

Lineare Iteration::

$$s^{(k+1)} = s^{(k)} + \Delta s^{(k)}, t^{(k+1)} = t^{(k)} + \Delta t^{(k)}$$

(#)
$$u(x_1)s_k(x_1) + w(x_1)t_k(x_1) \equiv \frac{1 - s^{(k)}u(x_1) - t^{(k)}w(x_1)}{p^k} \mod p$$

wobei $\Delta s^{(k)} = s_k(x_1)p^k$ $\Delta t^{(k)} = t_k(x_1)p^k$.

 $s^{(1)},t^{(1)}$ werden aus (*) in $\mathbb{Z}_p[x_1]$ bestimmt mit EEA. Für $k=1,2,\ldots,I-1$ wird (#) in $\mathbb{Z}_p[x]$ gelöst unter Verwendung von $s^{(1)},t^{(1)}$.

Wir erhalten somit folgenden Satz.

Newton's Iteration und Hensel's Konstruktion

Lösung diophantischer Polynomgleichungen in $\mathbb{Z}_{p'}[x_1]$

5.35 Satz Sei p Primzahl, $l \in \mathbb{N}^+$ und $u(x_1), w(x_1) \in \mathbb{Z}_{p^l}[x_1]$ mit

- i) $p \nmid \mathsf{HKoeff}(u(x_1)), p \nmid \mathsf{HKoeff}(w(x_1)).$
- ii) $\Phi_{\rho}(u(x_1))$ und $\Phi_{\rho}(w(x_1))$ teilerfremd in $\mathbb{Z}_{\rho}[x_1]$

Dann gibt es für jeden Polynom $c(x_1) \in \mathbb{Z}_{p^l}[x_1]$ eindeutig bestimmte Polynome $\sigma(x_1), \tau(x_1) \in \mathbb{Z}_{p^l}[x_1]$ mit

$$\sigma(x_1)u(x_1) + \tau(x_1)w(x_1) \equiv c(x_1) \bmod p^{l}$$

und

$$grad(\sigma(x_1)) < grad(w(x_1))$$

Ist $grad(c(x_1)) < grad(u(x_1)) + grad(w(x_1))$ erfüllt, so gilt auch

$$\mathsf{grad}(\tau(x_1)) < \mathsf{grad}(\mathit{u}(x_1))$$

Beweis: Existenz klar. Eindeutigkeit: Siehe Beweis der Eindeutigkeit bei Hensel's quadratischem Lifting.

マロトマラトマミトマミトマミトマミトマミトマミトマランスで Prof. Dr. K. Madlener: Computeralgebra

Prof. Dr. K. Madlener: Computeralgebra 33

Lösung diophantischer Polynomgleichungen in $\mathbb{Z}_{-I}[x_1]$

Multivariate Hensel Konstruktion

Finde multivariate Polynome

$$u(x_1,...,x_{\nu}), w(x_1,...,x_{\nu}) \in \mathbb{Z}_{p'}[x_1,...,x_{\nu}]$$
 mit

$$a(x_1,\ldots,x_{\nu})-uw\equiv 0 \bmod p^I$$
,

so dass

$$u(x_1,\ldots,x_n) \equiv u^{(1)}(x_1) \bmod \langle I,p' \rangle$$

$$w(x_1,\ldots,x_n) \equiv w^{(1)}(x_1) \bmod \langle I,p' \rangle$$

wobei $u^{(1)}(x_1), w^{(1)}(x_1) \in \mathbb{Z}_{p'}[x_1]$ gegeben mit

$$a(x_1, ..., x_{\nu}) - u^{(1)}(x_1)w^{(1)}(x_1) \equiv 0 \mod \langle I, p' \rangle$$

$$a(x_1,\ldots,x_{\nu})\in \mathbb{Z}_{p^l}[x_1,\ldots,x_{\nu}], l\in\mathbb{N},$$

 $l=\langle x_2-\alpha_2,\ldots,x_{\nu}-\alpha_{\nu}\rangle.$



Prof. Dr. K. Madlener: Computeralgebra

Newton's Iteration and Hensel's Konstruktion

Lösung diophantischer Polynomgleichungen in $\mathbb{Z}_{I}[x_1]$

Multivariate Hensel Konstruktion

Bezeichnet man die gesuchten Lösungen mit \bar{u}, \bar{w} und betrachtet man ihre I-adischen Entwicklungen, so

$$ar{u} = u^{(1)} + \Delta u^{(1)} + \Delta u^{(2)} + \dots + \Delta u^{(d)}$$
 bzw.
 $ar{w} = w^{(1)} + \Delta w^{(1)} + \Delta w^{(2)} + \dots + \Delta w^{(d)}$

wobei d maximaler totaler Grad von Termen in \bar{u} oder \bar{w} , $u^{(1)} = \Phi_I(\bar{u}), \ w^{(1)} = \Phi_I(\bar{w}) \ \text{und} \ \Delta u^{(k)}, \Delta w^{(k)} \in I^{(k)} \ (k = 1, 2, \dots, d).$ Multivariate Taylor Darstellung

$$\Delta u^{(k)} = \sum_{i_1=2}^{\nu} \sum_{i_2=i_1}^{\nu} \cdots \sum_{i_k=i_{k-1}}^{\nu} u_i(x_1) \prod_{j=1}^{k} (x_{i_j} - \alpha_{i_j})$$

$$\mathbf{i} = (i_1, \dots, i_k) \quad u_{\mathbf{i}}(x_1) \in \mathbb{Z}_{p'}[x_1].$$

Analog mit $\Delta w^{(k)}$.

4□ > 4問 > 4 = > 4 = > 9 Q ○

Lösung diophantischer Polynomgleichungen in $\mathbb{Z}_{-1}[x_1]$

Multivariate Taylor Darstellung

Zu lösen ist

(*)
$$w^{(k)} \Delta u^{(k)} + u^{(k)} \Delta w^{(k)} \equiv a(x_1 \dots x_{\nu}) - u^{(k)} w^{(k)} \mod \langle I^{k+1}, p' \rangle$$

wobei $u^{(k)}w^{(k)}$ die ideal-adische Approximation der Ordnung k sind, d. h.

$$a(x_1,...,x_{\nu}) - u^{(k)}w^{(k)} \in I^k$$

Rechte Seite von (*) hat die Gestalt

$$\sum_{i_2=2}^{\nu} \sum_{i_2=i_1}^{\nu} \cdots \sum_{i_k=i_{k-1}}^{\nu} c_i(x_1) \prod_{j=1}^{k} (x_{i_j} - \alpha_{i_j})$$

für geeignete $c_i(x_1) \in \mathbb{Z}_{p'}[x_1]$. Ersetzen und Koeffizientenvergleich liefert

$$(**) \quad w^{(k)}u_{i}(x_{1}) + u^{(k)}w_{i}(x_{1}) \equiv c_{i}(x_{1}) \bmod \langle I, p' \rangle$$

Hieraus lassen sich die I-adischen Koeffizienten $u_i(x_1), w_i(x_1) \in \mathbb{Z}_{p'}[x_1]$ bestimmen.

Prof. Dr. K. Madlener: Computeralgebra

Newton's Iteration und Hensel's Konstruktion

Multivariate Hensel Konstruktion

Da dies eine Kongruenz mod I ist, kann man Φ_I auf die linke Seite anwenden, d. h. zu lösen ist.

$$w^{(1)}(x_1)u_i(x_1) + u^{(1)}(x_1)w_i(x_1) \equiv c_i(x_1) \bmod p^l$$

wobei $u^{(1)}(x_1), w^{(1)}(x_1) \in \mathbb{Z}_{p'}[x_1]$ die Ausgangspolynome der Lösung sind. Unter den vorgegebenen Bedingungen gilt sogar Eindeutigkeit der Lösungen.

5.36 Satz Multivariate Hensel Konstruktion

Sei
$$p$$
-Primzahl, $I \in \mathbb{N}^+$, $a(x_1,\ldots,x_{\nu}) \in \mathbb{Z}_{p^l}[x_1,\ldots,x_{\nu}]$, $I = \langle x_2 - \alpha_2,\ldots,x_{\nu} - \alpha_{\nu} \rangle$, $\alpha_2,\ldots,\alpha_{\nu} \in \mathbb{Z}_p$, $p \nmid \mathsf{HKoeff}(\Phi_I(a(x_1,\ldots,x_{\nu})))$ und seien $u^{(1)}(x_1), w^{(1)}(x_1) \in \mathbb{Z}_{p^l}[x_1]$ mit

- i) $a(x_1, \ldots, x_{\nu}) \equiv u^{(1)}(x_1)w^{(1)}(x_1) \mod \langle I, p' \rangle$
- ii) $\Phi_{n}(u^{(1)}(x_{1})), \Phi_{n}(w^{(1)}(x_{1}))$ teilerfremd in $\mathbb{Z}_{n'}[x_{1}]$

Lösung diophantischer Polynomgleichungen in $\mathbb{Z}_{-1}[x_1]$

Satz: Multivariate Hensel Konstruktion

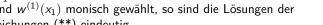
▶ Dann gibt es für $k \ge 1$ multivariate Polynome $u^{(k)}, w^{(k)} \in \mathbb{Z}_{n'}[x_1, \dots, x_{n'}]/I^k$, so dass

$$a(x_1,\ldots,x_{\nu})\equiv u^{(k)}w^{(k)} \bmod \langle I^k,p^l\rangle$$

und $u^{(k)} \equiv u^{(1)}(x_1) \mod \langle I, p^I \rangle$ $w^{(k)} \equiv w^{(1)}(x_1) \mod \langle I, p^I \rangle$

- \blacktriangleright Eindeutigkeit: Falls $a(x_1, \dots, x_{\nu})$ monisch bzgl. x_1 , d. h. der Koeffizient in $a(x_1,\ldots,x_{\nu})$ von $x_1^{d_1}$ ist 1, wobei d_1 der Grad von a in x_1 ist. Werden $u^{(1)}(x_1)$ und $w^{(1)}(x_1)$ monisch gewählt, so sind die Lösungen der diophantischen Gleichungen (**) eindeutig.
- Problem.

 exponentielles Wachstum für Zwischenergebnisse



▶ Probleme bei der Anwendung: Leading Coeff. Problem und Bad Zero



4□ > 4問 > 4 = > 4 = > 9 Q ○

Prof. Dr. K. Madlener: Computeralgebra

Newton's Iteration and Hensel's Konstruktion

Lösung diophantischer Polynomgleichungen in $\mathbb{Z}_{I}[x_1]$

Beispiel

5.37 Beispiel Sei p = 5 l = 1

$$a(x, y, z) = x^2y^4z - xy^9z^2 + xyz^3 + 2x - y^6z^4 - 2y^5z$$

$$I = \langle y - 1, z - 1 \rangle \max x$$
-Grad 2.

$$a(x, y, z) \equiv x^2 + 2x + 2 \mod \langle I, 5 \rangle$$

Es gilt

$$a(x, y, z) \equiv (x - 2)(x - 1) \mod \langle I, 5 \rangle$$
.

Wählt man $u^{(1)}(x) = x - 2$, $w^{(1)}(x) = x - 1$, so sind die Bedingungen vom Satz erfüllt.

a(x, y, z) ist nicht monisch aber w(x, y, z) ist monisch und somit liefert Hensel Lifting die richtige Antwort

Beispiel (Forts.)

Betrachte die I-adische Darstellung von a(x, y, z):

$$a(x, y, z) \equiv (x^2 + 2x + 2) - (x^2 + 1)(y - 1) + (x^2 + x - 1)(z - 1)$$

$$+ (x^2 - x)(y - 1)^2 - (x^2 - 1)(y - 1)(z - 1) + (2x - 1)(z - 1)^2 \\ - (x^2 - x)(y - 1)^3 + (x^2 - 2x)(y - 1)^2(z - 1) - \\ - (x + 1)(y - 1)(z - 1)^2 + (x - 1)(z - 1)^3 + (x^2 - x)(y - 1)^4 \\ + (-x^2 + 2x)(y - 1)^3(z - 1) - x(y - 1)^2(z - 1)^2 \\ + (x - 1)(y - 1)(z - 1)^3 - (z - 1)^2 - (x - 2)(y - 1)^5 \\ + (x^2 - 2x)(y - 1)^4(z - 1) + x(y - 1)^3(z - 1)^2 \\ - (y - 1)(z - 1)^4 + (x - 1)(y - 1)^6 - (2x + 1)(y - 1)^5(z - 1) \\ - x(y - 1)^4(z - 1)^2 - x(y - 1)^7 + (2x + 1)(y - 1)^6(z - 1) \\ - \cdots \\ - \cdots \\ - (y - 1)^6(z - 1)^4 - x(y - 1)^9(z - 1)^2 \bmod 5$$

4 D > 4 B > 4 B > 4 B > B 990

Prof. Dr. K. Madlener: Computeralgebra

Newton's Iteration und Hensel's Konstruktion

Lösung diophantischer Polynomgleichungen in $\mathbb{Z}_{I}[x_{1}]$

Hensel Konstruktion für das Beispiel

I-adische Darstellung enthält 38 Terme im Vergleich zu 6 Terme in der I-adischen Darstellung bzgl. $I = \langle y, z \rangle$.

Problem: Anzahl der zu lösenden polynomialen diophantischen Gleichungen ist proportional zur Anzahl der Terme in der I-adischen Darstellung von a(x, y, z)

Die Hensel Konstruktion für dieses Beispiel liefert

$$u^{(7)} = (x-2) + (-x+1)(y-1) + (x-2)(z-1) + x(y-1)^{2}$$

$$+(-x-2)(y-1)(z-1) + (-2)(z-1)^{2} + (-x)(y-1)^{3} +$$

$$+x(y-1)^{2}(z-1) + (-2)(y-1)(z-1)^{2} + (z-1)^{3}$$

$$+(x)(y-1)^{4} + (-x)(y-1)^{3}(z-1) + (1)(y-1)(z-1)^{3}$$

$$+(x)(y-1)^{4}(z-1)$$

$$w^{(7)} = (x-1) + (-1)(z-1) + (-1)(y-1)^{5} + (-1)(y-1)^{5}(x-1)$$

Lösung diophantischer Polynomgleichungen in $\mathbb{Z}_{-I}[x_1]$

Hensel Konstruktion für das Beispiel

Ausmultiplizieren mod5 liefert

$$u^{(7)} \equiv xy^4z + yz^3 + 2 \mod 5$$
 $w^{(7)} \equiv x - y^5z \mod 5$

Die Iteration hält hier, da

$$e^{(7)} = a(x, y, z) - u^{(7)}w^{(7)} = 0.$$

Problem: Auswertungspunkt $\neq 0$. Leider kann man nicht immer Auswertungspunkte = 0 wählen, da $p \nmid \mathsf{HKoeff}(\Phi_I(a(x_1, \ldots, x_{\nu})))$.

Möglichkeit: Variablentransformation

$$x_j \leftarrow x_j + \alpha_j$$
 $2 \le j \le \nu$, falls $I = \langle x_2 - \alpha_2, \dots, x_{\nu} - \alpha_{\nu} \rangle$

Problem der Zwischenergebnisse bleibt erhalten.



Prof. Dr. K. Madlener: Computeralgebra

31

Newton's Iteration und Hensel's Konstruktion

Lösung diophantischer Polynomgleichungen in $\mathbb{Z}_{n^I}[x_1]$

Möglichkeit (Forts.)

▶ Möglichkeiten zur einfacheren Berechnung siehe G.C.L 262 \rightarrow dünn besetzte MV Polynome.

$$c_i(x_1) = \frac{1}{n_1! \dots n_m!} \Phi_I \left(\left(\frac{\partial}{\partial x_{j-1}} \right)^{n_1} \cdots \left(\frac{\partial}{\partial x_{j_m}} \right)^{n_m} e^{(k)} \right)$$

Wang EEZ-GCD Algorithmus: Variablenweise

Prof. Dr. K. Madlener: Computeralgebra

$$\mathbb{Z}_{p'}[x_1] \to \mathbb{Z}_{p'}[x_1, x_2] \to \mathbb{Z}_{p'}[x_1, x_2, x_3] \cdots$$

GCD Berechnung

Inhalt Kapitel 6

Anwendungen modularer und p-adischer Methoden

GCD Berechnungen

Faktorisierung

Quadratfreie Faktorisierung

 ${\sf Getrennte}\ {\sf Grad}\ {\sf Faktorisierung-Distinct}\ {\sf Degree}\ {\sf Factorization}$

Equal-Degree Factorization (Gleiche-Grad-Faktorisierung)-

Algorithmus von Cantor und Zassenhaus

Anwendung: Nullstellen-Bestimmung

Faktorisierungsalgorithmen, die auf linearer Algebra basieren

Anwendung: Irreduzible Polynome: Test und Konstruktion

Faktorisierung in $R[x_1, \ldots, x_n]$, R ZPE Ring

Faktorisierung in K[x] für K algebraischer Zahlenkörper

Prof. Dr. K. Madlener: Computeralgebra

, = ,, ,, ,

Anwendungen modularer und p-adischer Methoden

GCD Berechnunge

GCD Berechnung - Faktorisierung

GCD (GGT)-Berechnungen

- \blacktriangleright klassisch EEA (euklid. Ringe) Z.B. F[x] O(M(n)logn) Körperoperationen.
- ▶ (Pseudo-) Polynomiale Restefolgen, reduzierte PRS (primitiver EA $\mathbb{Z}[x_1,\ldots,x_{\nu}]$) (kleiner Grad ≤ 2) Problem Koeffizientenwachstum
- Sylvester Matrix und Subresultanten
- ▶ Modularer Algorithmus (Brown) Big-Prime, Small-Primes
- p-adisch EZGCD (Moses u. Yun)
- ► EEZ-GCD (Wang).
- GCD-Heuristic

Prof. Dr. K. Madlener: Computeralgebra

320

GCD Berechnungen

GCD Berechnung: Beispiel

Seien

$$a(x) = x^8 + x^6 - 3x^4 - 3x^3 + 8x^2 + 2x - 5, b(x) = 3x^6 + 5x^4 - 4x^2 - 9x + 21$$

EEA in $\mathbb{Q}[x]$ liefert PRF mit $r_5(x) = -\frac{1288744821}{543880225}$, d.h. a(x), b(x) sind teilerfremd in $\mathbb{Z}[x]$.

Problem: Koeffizientenwachstum + Berechnung im Quotientenkörper (GGT-Berechnungen).

Modular: $\phi_{23}: \mathbb{Z}[x] \to \mathbb{Z}_{23}[x]$. EAA in $\mathbb{Z}_{23}[x]$ liefert 1 als GGT.

 $\phi_2: \mathbb{Z}[x] \to \mathbb{Z}_2[x]$. EAA in $\mathbb{Z}_2[x]$ liefert x+1 als GGT \rightsquigarrow

Unglückliche Homomorphismen.

p-adisch: EZGCD, EEZ-GCD (Wang)

Messungen: Siehe vz Gathen, Gerhard (S 183 Fälle $\mathbb{Z}[x], F[x, y]$)

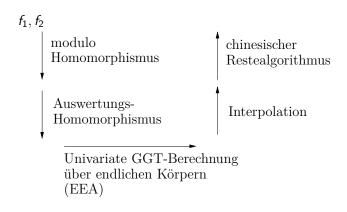
Prof. Dr. K. Madlener: Computeralgebra

Anwendungen modularer und p-adischer Methoden

GCD Berechnunger

GCD Berechnungen - Schemata

Modularer Algorithmus: f_1 , f_2 Multivariate Polynome



Faktorisierung

Anwendungen: Simplifikation, symbolische Integration

Lösung von Polynomgleichungen Kodierungstheorie, Zahlentheorie

Kryptographie

Lösungsweg

- ▶ Reduktion auf Problem der Faktorisierung in $\mathbb{Z}_n[x]$
- Quadratfreie-Faktorisierung
- Distinct-Degree Factorization (Getrennte Grad Faktorisierung)
- Equal-Degree Factorization (Gleicher-Grad Faktorisierung)
- ► Berlekamp's Algorithmen

Prof. Dr. K. Madlener: Computeralgebra

Anwendungen modularer und p-adischer Methoder

Faktorisierung

Welche Ringe sind interessant für die Faktorisierung

$$\mathbb{Z}[x], \mathbb{Q}[x], R[x]$$
 R ZPE Ring, $F[x]$ F endlicher Körper, $\mathbb{Z}[x_1, \ldots, x_{\nu}], \mathbb{Q}[x_1, \ldots, x_{\nu}], \mathbb{Q}(\sqrt{2}), \mathbb{Q}(\sqrt{2}, \sqrt{3})$

R ZPE-Ring, $a \in R[x]$

cont(a) = GGT(Koeff von a) (einheitsnormal)

pp(a) = a/cont(a) d.h. a primitiv gdw cont(a) = 1

Es gilt cont(ab) = cont(a)cont(b) und pp(ab) = pp(a)pp(b).

Sind a, b primitiv, so auch ab, d. h. prim-Elemente von R[x] sind die prim-Elemente von R plus primitive Polynome in R[x], die irreduzibel in K[x] sind, wobei K Quotientenkörper von R ist. (Beweis!)

4□ > 4問 > 4 = > 4 = > 9 Q ○

4 D F 4 B F 4 B F 9 9 9 9

Faktorisierung (Forts.)

Insbesondere für Faktorisierung in $\mathbb{Q}[x]$ bzw. $\mathbb{Z}[x]$:

Ist $a \in \mathbb{Z}[x]$ primitiv $\leadsto a = f_1 \dots f_k$ in irreduziblem $f_i \in \mathbb{Q}[x]$ liefert Faktorisierung $a = f_1^* \dots f_k^*$ mit $f_i^* \in \mathbb{Z}[x]$ irreduzibel. (Durch Multiplikation mit Nennern und Entfernung von Inhalt).

 $a \in \mathbb{Z}[x]$ beliebig. Faktorisierung von a ist Faktorisierung vom Inhalt von a (als Element von \mathbb{Z}) + Faktorisierung von pp(a), d. h.

Faktorisierung in $\mathbb{Z}[x] \rightsquigarrow$ Faktorisierung in $\mathbb{Q}[x]$ plus Faktorisierung in $\mathbb{Z}[x]$



Prof. Dr. K. Madlener: Computeralgebra

endungen modularer und p-adischer Methoder

Quadratfreie Faktorisierung

Quadratfreie Faktorisierung

Reduktion des Faktorisierungsproblems auf Faktorisierung von Polynomen ohne Mehrfachfaktoren.

6.1 Definition Sei R ZPE-Ring, $a(x) \in R[x]$ primitives Polynom. a(x) heißt quadratfrei, falls a(x) ohne Mehrfachfaktoren, d. h. es gibt kein b(x), grad $(b) \ge 1$, $b(x)^2 |a(x)|$.

Die quadratfreie Faktorisierung von a(x) ist

$$a(x) = \prod_{i=1}^{k} a_i(x)^i$$
 (genauer:: die Folge der $a_i(x)$)

wobei für jedes i $a_i(x)$ quadratfreies Polynom und

$$\mathsf{GGT}(a_i(x), a_i(x)) = 1 \text{ für } i \neq j$$

Beachte: Einige der a; in der QFF von a können 1 sein.

4□ > 4問 > 4 = > 4 = > 9 Q ○

Anwendungen modularer und p-adischer Methoden

Quadratfreie Faktorisierung

6.2 Beispiel

Sei
$$a(x) = (x^2 + 1)(x^2 - 1)^4(x^3 + 3x)^5$$
. Hierbei sind $a_2(x) = a_3(x) = 1$.

Beachte ebenfalls, dass die a; nicht faktorisiert sein müssen.

QFF(
$$a(x)$$
) = $(x^2 + 1, 1, 1, x^2 - 1, x^3 + 3x)$. (Folge der Quadratfreienfaktoren von $a(x)$).

Quadratfreiheit wird über die Ableitung bestimmt.

Sei
$$a = \sum_{i=0}^{n} a_i x^i$$
, dann ist die Ableitung von a ::

$$a'(x) = a_1 + 2a_2x + \cdots + na_nx^{n-1}$$

hierbei ist
$$n = \underbrace{1 + 1 + \dots + 1}_{n-mal}$$
 n -mal.

Es gelten die üblichen Ableitungsregeln!

Anwendungen modularer und p-adischer Methoden

Prof. Dr. K. Madlener: Computeralgebra

Quadratfreie Faktorisierun

Charakterisierung der Quadratfreiheit

6.3 Lemma Sei a(x) primitiv in R[x]. R ZPE-Ring mit Charakteristik 0 (d. h. $\underbrace{1+\cdots+1}_{}\neq 0$ für alle $n\geq 1$).

Sei
$$c(x) = GGT(a(x), a'(x))$$
, dann hat a mehrfache Faktoren $\operatorname{gdw} c(x) \neq 1$. Also $a(x)$ quadratfrei $\operatorname{gdw} c(x) = 1$.

Beweis:
$$\curvearrowright$$
: Sei $a(x) = b(x)^2 w(x)$, grad $b \ge 1$ $\rightsquigarrow a'(x) = 2b(x)b'(x)w(x) + b(x)^2 w'(x) = b(x)\hat{w}(x) \rightsquigarrow c(x) \ne 1$.

$$f$$
: Angenommen $c(x) \neq 1$, aber $a(x)$ quadratfrei.

$$a(x) = p_1(x)p_2(x)\cdots p_k(x), \ p_i(x) \ \text{irreduzibel}.$$

 $grad(p_i(x)) \geq 1, \ \mathsf{GGT}(p_i(x), p_i(x)) = 1 \ \mathsf{für} \ i \neq j. \ \mathsf{Dann}$

$$a'(x) = p'_1(x)p_2(x)\cdots p_k(x) + \cdots + p_1(x)\cdots p_{k-1}(x)p'_k(x)$$

Angenommen
$$p_i(x)|c(x)$$
 (Es gibt mindestens ein solches i .)
O.B.d.A. $i=1$ $p_1(x)|a'(x) \leadsto p_1(x)|p_1'(x)p_2(x) \cdots p_k(x) \leadsto p_1(x)|p_1'(x)$ geht nur, wenn $p_1'(x)=0$. Char $0 \leadsto p_1(x)$ konstant. $\mbox{$\frac{i}{2}$}$

Aus
$$a(x) = \prod_{i=1}^k a_i(x)^i \rightsquigarrow$$

$$a'(x) = \sum_{i=1}^k a_1(x) \cdots i a_i(x)^{i-1} a_i'(x) \cdots a_k(x)^k \text{ also}$$

$$c(x) = \text{GCD}(a(x), a'(x)) = \prod_{i=2}^k a_i(x)^{i-1}. \text{ (Beweis!)}$$
Setzt man $w(x) := a(x)/c(x) = a_1(x)a_2(x) \cdots a_k(x).$

$$\rightsquigarrow w(x) \text{ ist Produkt der quadratfreien Faktoren ohne Multiplizitäten.}$$
Abspaltung von $a_1(x)$:: Sei $y(x) = \text{GCD}(c(x), w(x))$, so gilt $a_1(x) = w(x)/y(x)$.

Dann weiter mit c(x). \rightsquigarrow Berechnung der $a_i(x)$.

ロト 4個ト 4厘ト 4厘ト 連 め900

Prof. Dr. K. Madlener: Computeralgebra

nwendungen modularer und p-adischer Methoden

Ouddratfreie Faktorisierung

Quadratirele Faktorisierung

Prof. Dr. K. Madlener: Computeralgebra

Algorithmus quadratfreie Faktorisierung

```
procedure Square_Free(a(x))
\{a(x) \in R[x], \text{ primitiv, char}(R) = 0, R \text{ ZPE}\}
\{\text{Ausgabe quadratfreie Faktorisierung von } a(x) \}
i := 1; outp := 1; b(x) := a'(x);
c(x) := \text{GGT}(a(x), b(x)); w(x) := a(x)/c(x);
while c(x) \neq 1 do
begin
y(x) := \text{GGT}(w(x), c(x)); z(x) := w(x)/y(x);
outp := outp \sqcup z(x); i := i + 1;
w(x) := y(x); c(x) := c(y)/y(x);
end
outp := outp \sqcup w(x);
return outp, i.
```

Korrektheit ok. Komplexität Übung. 0(kKosten GGT(a, a'))

Anwendungen modularer und p-adischer Methoden

Quadratfreie Faktorisierur

Algorithmus quadratfreie Faktorisierung

6.4 Beispiel Sei
$$a(x) = x^8 - 2x^6 + 2x^2 - 1 \in \mathbb{Z}[x]$$

$$b(x) = a'(x) = 8x^7 - 12x^5 + 4x,$$

 $c(x) = x^4 - 2x^2 + 1, w(x) = x^4 - 1.$

Da $c(x) \neq 1$ nach einem Schleifendurchgang

$$y(x) = x^2 - 1$$
, $z(x) = outp = x^2 + 1$,
 $i = 2$, $w(x) = c(x) = x^2 - 1$

2 Schleifendurchgang

$$y(x) = x^2 - 1$$
, $z(x) = 1$, out $p = (x^2 + 1) \sqcup 1$
 $i = 3$, $w(x) = x^2 - 1$, $c(x) = 1$

$$outp := outp \sqcup w(x) = (x^2 + 1) \sqcup 1 \sqcup (x^2 - 1).$$

Dies ist die quadratfreie Faktorisierung.

Prof. Dr. K. Madlener: Computeralgebra

331

Anwendungen modularer und p-adischer Methoden

Quadratfreie Faktorisierung

Effizientere Methoden: Yun's QFF-Algorithmus

Sei
$$a(x) = a_1(x)a_2(x)^2 \cdots a_k(x)^k$$
, QFF von $a(x)$. Dann $a'(x) = a'_1(x)a_2(x)^2 \cdots a_k(x)^k + \cdots + ka_1(x)a_2(x)^2 \cdots a_k(x)^{k-1}a'_k(x)$ $= \sum_{1 \leq i \leq k} i \frac{a(x)}{a_i(x)} a'_i(x)$ $= \operatorname{GGT}(a_i(x), a_j(x)) = 1$ für $i \neq j$. Also $c(x) = \operatorname{GGT}(a(x), a'(x)) = \prod_{i=2}^k a_i(x)^{i-1}$. Sei $w(x) = a(x)/c(x) = \prod_{i=1}^k a_i(x)$ Produkt der QFF von $a(x)$. Dann $y(x) = a'_1(x)a_2(x) \cdots a_k(x) + \cdots + ka_1(x) \cdots a_{k-1}(x)a'_k(x)$

330 Prof. Dr. K. Madlener: Computeralgebra 332

Quadratfreie Faktorisierung

Yun's QFF-Algorithmus (Forts.)

Setzt man

$$z(x) = y(x) - w'(x) = y(x) - \sum_{i=1}^{k} a_1(x) \cdots a'_i(x) \cdots a_k(x)$$

$$= a_1(x)a'_2(x) \cdots a_k(x) + \cdots + (k-1)a_1(x) \cdots a_{k-1}(x)a'_k(x)$$

$$= a_1(x)[a'_2(x) \cdots a_k(x) + \cdots + (k-1)a_2(x) \cdots a_{k-1}(x)a'_k(x)]$$
So erhält man den ersten QF-Term durch Berechnen von

$$a_1(x) = \mathsf{GGT}(w(x), z(x))$$

Der Unterschied bisher ist die zusätzliche Berechnung der Ableitung.

Der nächste Schritt ist es die QFF von c(x) zu bestimmen. Hierfür sind die entsprechenden w(x), y(x) und z(x) bestimmt durch

$$\begin{aligned} & w(x) = w(x)/a_1(x) = a_2(x) \cdots a_k(x) \\ & y(x) = z(x)/a_1(x) = a_2'(x) \cdots a_k(x) + \cdots + (k-1)a_2(x) \cdots a_{k-1}(x)a_k'(x) \\ & z(x) = y(x) - w'(x) = a_2(x)[a_3'(x) \cdots a_k(x) + \cdots + (k-2)a_3(x) \cdots a_k'(x)] \\ & \text{und somit } a_2(x) = \mathsf{GGT}(w(x), z(x)). \ \mathsf{Usw}. \end{aligned}$$

(ロ) (個) (E) (E) (E) のQ(

Prof. Dr. K. Madlener: Computeralgebra

333

Anwendungen modularer und p-adischer Methoden

Yun's quadratfreier Faktorisierungsalgorithmus

```
procedure Square_Free_Yun(a(x))
{Eingabe:a(x) \in R[x], primitiv char(R) = 0, R ZPE, grad(a) = n}
{Ausgabe: quadratfreie Faktorisierung von a(x)}
```

begin

(1)
$$i := 1$$
; **output** $:= 1$;

$$b(x) := a'(x); c(x) := GGT(a(x), b(x));$$

(2)
$$w(x) := a(x)/c(x); y(x) := b(x)/c(x); z(x) := y(x) - w'(x)$$

while $z(x) \neq 0$ do

begin

$$g(x) := \mathsf{GGT}(w(x), z(x));$$

 $outp := outp \sqcup g(x); i := i + 1;$

$$w(x) := w(x)/g(x); y(x) := z(x)/g(x); z(x) := y(x) - w'(x)$$

end

Prof. Dr. K. Madlener: Computeralgebra

(3)
$$outp := outp \sqcup w(x);$$

return $(outp, i);$
end.

Quadratfreie Faktorisierung

Yun's quadratfreier Faktorisierungsalgorithmus

6.5 Satz Yun's QFFA ist korrekt und benötigt $0(M(n) \log n)$ Operationen in R. (Zweimal Kosten für die GGT-Berechnung von a(x), a'(x)).

Korrektheit folgt aus der Vorüberlegung.

Für die Kosten: Sei
$$(g_1, \ldots, g_m)$$
 QFF von a und $d_i = \operatorname{grad} g_i$ $1 \le j \le m$.

Schritt (1) kostet $0(M(n) \log n)$.

Seien w_i, y_i, z_i Werte beim Eingang Durchgang i.

$$\operatorname{\mathsf{grad}}(w_i) = \sum_{i < j < m} d_j, \operatorname{\mathsf{grad}}(y_i) = \operatorname{\mathsf{grad}}(w_i) - 1, \operatorname{\mathsf{grad}}(z_i) = \operatorname{\mathsf{grad}}(y_i).$$

Die GGT Berechnungen im *i*-ten Durchgang kostet $0(M(\operatorname{grad}(w_i))\log n)$ und die zwei Divisionen $0(M(\operatorname{grad}(w_i)))$ Operationen in F (M Kosten der Multiplikation) wegen der Subaditivität von M gilt

$$\sum_{1 \leq i \leq m} M(\operatorname{grad}(w_i)) \leq M(\sum_{1 \leq i \leq m} \operatorname{grad}(w_i)) = M(\sum_{1 \leq i \leq j \leq m} d_j)$$

$$= M(\sum_{1 \leq i \leq m} id_i) = M(n) \rightsquigarrow \mathsf{Behauptung}$$

Prof. Dr. K. Madlener: Computeralgebra

Anwendungen modularer und p-adischer Methoder

Quadratfreie Faktorisierun

Beispiel

6.6 Beispiel

Sei $f = abc^2d^4$ für verschiedene monische irreduziblen Polynome $a, b, c, d \in R[x], c(x) = GGT(f, f') = cd^3$.

$$w_1 = f/c(x) = abcd, y_1 = f'/c(x) = a'bcd + ab'cd + 2abc'd + 4abcd'$$

$$z_1 = y_1 - w_1' = abc'd + 3abcd'$$

$$g_1 = \mathsf{GGT}(\mathit{abcd}, \mathit{abc'd} + 3\mathit{abcd'}) = \mathit{ab}$$

$$w_2 = abcd/ab = cd, y_2 = (abc'd + 3abcd')/ab = c'd + 3cd'$$

$$z_2 = 2cd'$$

$$g_2 = \mathsf{GGT}(cd, 2cd') = c$$

$$w_3 = cd/c = d$$
, $y_3 = 2cd'/c = 2d'$ $z_3 = d'$

$$g_3 = \mathsf{GGT}(d, d') = 1$$

$$w_4 = d/1 - d$$
, $w_4 = d'/1 = d'$ $z_4 = 0$

$$g_4 = d$$

$$(ab, c, 1, d)$$
 Länge 4.

Prof. Dr. K. Madlener: Computeralgebra 336

Quadratfreie Faktorisierung

char $R \neq 0$ R =endlicher Körper char p

$$F=R=GF(q)=\mathbb{F}_q$$
 mit $q=p^m$, p Primzahl, $\mathrm{char}(R)=p$ $a=\sum_{0\leq i\leq n}a_ix^i
ot\in F\wedge f'=0$ gdw

für jedes i mit $a_i \neq 0$ gilt p|i, d. h. $ia_i x^{i-1} = 0$ in F[x]

$$a = \sum_{0 \le i \le n/p} a_{ip} x^{ip} = \left(\sum_{0 \le i \le n/p} a_{ip} x^i\right)^p$$
, falls $F = \mathbb{F}_p$.

Da $(g+h)^p = g^p + h^p$ für alle $g, h \in \mathbb{F}_p[x]$ und $a_{ip}^p = a_{ip}$ für alle $a_{ip} \in \mathbb{F}_p$. Z.B.: $a(x) = x^{13} + 1$ in \mathbb{F}_{13} , so $a'(x) = 13x^{12} = 0$

$$(x+1)^{13} = x^{13} + {13 \choose 1}x^{12} + \dots + {13 \choose 12}x + 1 = x^{13} + 1 = a(x)$$

< ロ > → □ > → □ > → □ > → □ = → つへで

Prof. Dr. K. Madlener: Computeralgebra

Anwendungen modularer und p-adischer Methoden

char $R \neq 0$ R =endlicher Körper char p

6.7 Lemma \mathbb{F}_q mit $q = p^m$, p Primzahl, dann gilt für alle $r, s \in \mathbb{F}_q$

(1)
$$r^q = r$$

kleiner Fermat Satz

(2)
$$r^{1/p} = r^{q/p} = r^{p^{m-1}}$$

ist *p*-te Wurzel von *r*

(3)
$$(r+s)^{p^j} = r^{p^j} + s^{p^j}$$

 $j=0,1,\ldots,m$

Beweis: $r \in \mathbb{F}_q \leadsto \{1, r, r^2, \dots\}$ ist zyklisch und endliche Untergruppe der mult. Gruppe von \mathbb{F}_q . Diese hat die Ordnung $q-1 \leadsto$ Ordnung von r teilt q-1 (Lagrange),

$$r^{q-1}=1 \rightsquigarrow (1)$$

Prof. Dr. K. Madlener: Computeralgebra

$$(r^{p^{m-1}})^{p} = r^{p^{m'}} = r^{q} = r \rightsquigarrow (2)$$

$$(r+s)^{p^{j}} = r^{p^{j}} + {\binom{p^{j}}{1}} r^{p^{j-1}} s + \dots + {\binom{p^{j}}{p^{j-1}}} r s^{p^{j-1}} + s^{p^{j}}$$

$$= r^{p^{j}} + s^{p^{j}} \rightsquigarrow (3)$$

char $R \neq 0$ R =endlicher Körper char p (Forts.)

6.8 Lemma $a \in \mathbb{F}_q[x]$ a' = 0 gdw a ist eine p-te Potenz in $\mathbb{F}_q[x]$.

Beweis: \(\scale \) klar

$$b(x)^{p} = b_{0}^{p} + b_{1}^{p}x^{p} + \dots + b_{k}^{p}x^{kp} = a_{0} + a_{p}x^{p} + \dots + a_{kp}x^{kp} = a(x)$$

Prof. Dr. K. Madlener: Computeralgebra

Anwendungen modularer und p-adischer Methoden

Quadratfreie Faktorisierun

char $R \neq 0$ R =endlicher Körper char p (Forts.)

Sei $a=f_1^{e_1}\cdots f_r^{e_r}$ die irreduzible Faktorisierung von a. Angenommen für ein i $1\leq i\leq r,$ $f_i'=0 \leadsto f_i$ ist eine p-te Potenz, d. h. f_i ist nicht irreduzible. (d. h. Ableitungen irreduzibler Polynome sind ungleich null), d. h. $f_i'\neq 0$ und wegen grad $f_i'<$ grad f_i folgt $\mathsf{GGT}(f_i',f_i)$ ist nicht f_i und somit 1, da f_i irreduziblel. Es kann jedoch $e_if_i'=0$ gelten, wenn $p|e_i$. Wegen $a'=\sum_{1\leq i< r}e_i\frac{a}{f_i}f_i'$ gilt aber $f_i^{e_i}|a'$.

Somit gilt

6.9 Lemma Sei F endlicher Körper, $a \in F[x]$ nicht konstant. Dann gilt

a ist quadratfrei gdw GGT(a, a') = 1

→ Algorithmus zur QFF in endlichen Körpern

```
procedure Square Free FF(a(x), q)
{Eingabe: monisches Polynom a(x) \in \mathbb{F}_q[x], q = p^mBerechne QFF von a(x)}
i = 1; output := 1; b(x) := a'(x);
if b(x) \neq 0 then
   begin
    c(x) := GGT(a(x), b(x)); w(x) := a(x)/c(x);
    while w(x) \neq 1 do
        begin
        y(x) := GGT(w(x), c(x)); z(x) := w(x)/y(x);
        outp := outp \sqcup z(x)^i : i := i + 1:
        w(x) := y(x); c(x) := c(x)/y(x);
        end
   if c(x) \neq 1 then
        c(x) := c(x)^{1/p}; outp := outp \sqcup (Square_Free FF(c(x), q))<sup>p</sup>
   end
else
    a(x) := a(x)^{1/p}; outp := (Square_Free FF(a(x), q))<sup>p</sup>
return (outp)
end.
```

Prof. Dr. K. Madlener: Computeralgebra

Anwendungen modularer und p-adischer Methoder

Quadratfreie Faktorisierung

Beispiel

Beachte: Hier QF Faktoren mit Exponenten, um a rekonstruieren zu können. $(f_1 \sqcup \cdots \sqcup f_k)^p \equiv f_1^p \sqcup \cdots \sqcup f_k^p$. **6.10 Beispiel** Sei $\mathbb{F}_3[x] = \mathbb{Z}_3[x]$ und $a(x) = x^{11} + 2x^9 + 2x^8 + x^6 + x^5 + 2x^3 + 2x^2 + 1$

Dann

$$a'(x) = 2x^{10} + x^7 + 2x^4 + x$$
 und
 $c(x) = \mathsf{GGT}(a(x), a'(x)) = x^9 + 2x^6 + x^3 + 2$
 $w(x) = a(x)/c(x) = x^2 + 1 \neq 1$

Eingang while loop

$$y(x) = x + 2$$
 $z(x) = x + 1$ out $p = x + 1$

Updates

$$i = 2$$
 $w(x) = x + 2$ $c(x) = x^8 + x^7 + x^6 + x^2 + 1$

Zweiter Durchgang liefert

$$y(x) = x + 2$$
 $z(x) = 1$ $outp = (x + 1) \sqcup 1^2$

Updates

$$i = 3$$
 $w(x) = x + 2$ $c(x) = x^7 + 2x^6 + x + 2$

Beispiel (Forts.)

Quadratfreie Faktorisierung

Dritter Durchgang

$$y(x) = x + 2$$
 $z(x) = 1$ $outp = (x + 1) \sqcup 1^2 \sqcup 1^3$

Updates

$$i = 4$$
 $w(x) = x + 2$ $c(x) = x^6 + 1$

Vierter Durchgang

$$y(x) = 1$$
 $z(x) = x + 2$ out $p = (x + 1) \sqcup 1^2 \sqcup 1^3 \sqcup (x + 2)^4$

Updates

$$i = 5$$
 $w(x) = 1$ $c(x) = x^6 + 1$

Da w(x) = 1 Ausgang der While Schleife, da $c(x) \neq 1$ muss c dritte Potenz sein. Dritte Wurzel ist $x^2 + 1$. Rekursiver Aufruf mit $x^2 + 1$ Ableitung $2x \rightsquigarrow quadratfrei$.

outp =
$$(x + 1) \sqcup (x + 2)^4 \sqcup (x^2 + 1)^3$$
, d. h. $((x + 1), 1, (x^2 + 1), (x + 2))$ QFF von a.

Prof. Dr. K. Madlener: Computeralgebra

Anwendungen modularer und p-adischer Methoden

Quadratfreie Faktorisierung

Symbolisches Beispiel

6.11 Beispiel Symb-Beispiel: Seien A, B, C, D guadratfreie teilerfremde Polynome in $\mathbb{Z}_3[x]$ und $a = A \cdot B^3 \cdot C^5 \cdot D^9$.

Initialisierung: $b = a' = A'B^3C^5D^9 + 2AB^3C^4C'D^9$, d. h.

 $GGT(a, b) = B^3C^4D^9 = c.$

Die QF-Terme (ohne die 3 | Exponent) sind in $w = a/c = A \cdot C$

while loop v = GGT(c, w) = C z = w/C = A. outp = A

i = 2 w = C $c = B^3C^3D^9$

while loop y = C z = 1, $outp = A \sqcup 1$

i = 3 w = C $c = B^3C^2D^9$

4- while loop Ausgang

i = 5w = C $c = B^3 D^9$ $outp = A \sqcup 1 \sqcup 1 \sqcup 1$ 5-Durchgang v = 1 z = C $outp = A \sqcup 1 \sqcup 1 \sqcup 1 \sqcup C^5$

i = 6 w = 1 $c = B^3D^9$

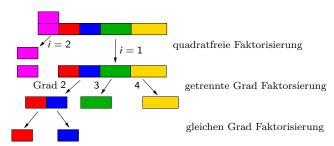
Da $c \neq 1 \rightsquigarrow QFF(c)^{1/3} = BD^3 \rightsquigarrow (B \sqcup 1 \sqcup D^3)^3 \rightsquigarrow A \sqcup C^5 \sqcup B^3 \sqcup D^3$

Man kann auch Yun's Variante verwenden mit gleicher Komplexitätsschranke.

Distinct Degree Factorization

Spaltung der irreduziblen Faktoren nach Grad::

Getrennte Grad Faktorisierung



 $a(x) \in \mathbb{F}_q[x], \quad q = p^m$, quadratfrei.

Gesucht Faktorisierung von a(x) der Form $a(x) = \prod a_i(x)$ wobei a_i Produkt der irreduziblen Faktoren von a(x) mit Grad i, d. h.

 $grad(a_i) = k \cdot i :: k$ Faktoren mit Grad i.

Prof. Dr. K. Madlener: Computeralgebra

Anwendungen modularer und p-adischer Methoden

Getrennte Grad Faktorisierung-Distinct Degree Factorization

Satz von Fermat: Folgerungen

Erinnerung: Kleiner Fermatscher Satz:: $0 \neq a \in \mathbb{F}_q$, so $a^{q-1} = 1$ und $a^q = a$ alle $a \in F_q$, d. h. $x^q - x = \prod_{a \in F_q} (x - a)$ in $F_q[x]$.

Allgemeiner

Prof. Dr. K. Madlener: Computeralgebra

6.12 Lemma Für $d \ge 1$ ist $x^{q^d} - x \in \mathbb{F}_q[x]$ Produkt aller monischen irreduziblen Polynome in $\mathbb{F}_q[x]$, deren Grad d teilt.

Kleiner Fermat angewendet auf \mathbb{F}_{q^d} zeigt $h = x^{q^d} - x$ ist Produkt aller x - a mit $a \in \mathbb{F}_{q^d}$.

Falls $g^2 \mid h$ (in \mathbb{F}_q) mit $g \in \mathbb{F}_q[x] \setminus \mathbb{F}_q$, so teilt ein x - a auch g und somit $(x - a)^2 \mid h$.

Dies geht nicht, d. h. $x^{q^d} - x$ ist quadratfrei (einfacher GGT(h, h') = 1).

Satz von Fermat: Folgerungen

Es genügt zu zeigen: Für $f \in \mathbb{F}_q[x]$, monisch, irreduzibel mit $\operatorname{Grad}(f) = n$:

$$f \mid x^{q^d} - x$$
 gdw $n \mid d$

Sei f irreduzibel, monisch, $n \mid d$, $d = n \cdot s$.

Betrachte $F = \mathbb{F}_q[x]/\langle f \rangle$ ist Körper mit q^n Elementen.

Kleiner Fermat liefert für $a \in F$

$$a^{q^n} = a$$
 und somit $a^{q^d} = (\underbrace{(a^{q^n})^{q^n} \dots)}^{q^n} = a$

Betrachte a = [x] Repräsentant von x in F.

$$[h] = [x^{q^d} - x] = [x]^{q^d} - [x] = a^{q^d} - a = 0$$
 in F , d.h. $h \equiv 0 \mod f$, und somit $f \mid h$.

Prof. Dr. K. Madlener: Computeralgebra

7 - 740

34

Anwendungen modularer und p-adischer Methoden

Getrennte Grad Faktorisierung-Distinct Degree Factorization

Satz von Fermat: Folgerungen

Umgekehrt sei f monisch, irreduzibel grad(f) = n, $f \mid x^{q^d} - x$.

Betrachte die Körpererweiterung $\mathbb{F}_q \subseteq \mathbb{F}_{q^d}$.

Da $f \mid x^{q^d} - x$ folgt aus kleinen Fermat angewendet mit \mathbb{F}_{q^d} , dass es $A \subseteq \mathbb{F}_{q^d}$ gibt mit $f = \prod_{a \in A} (x - a)$.

Wähle $a \in A$ und sei $\mathbb{F}_q[x]/\langle f \rangle \cong \mathbb{F}_q(a) \subseteq \mathbb{F}_{q^d}$, wobei $\mathbb{F}_q(a)$ kleinster Teilkörper von \mathbb{F}_{q^d} , der a enthält.

Dieser Körper hat q^n Elemente und \mathbb{F}_{q^d} ist eine Erweiterung von $\mathbb{F}_q(a)$, d. h. $q^d = (q^n)^s$ für ein s also $n \mid d$.

Prof. Dr. K. Madlener: Computeralgebra

◆□ > ◆@ > ◆差 > ◆差 > 差 め Q @

348

Getrennte Grad Faktorisierung-Distinct Degree Factorization

Getrennte Grad Faktorisierung (Forts.)

Anwendung: Sei $a(x) = \prod a_i(x)$. Um das Produkt aller linearen irreduziblen Faktoren von a(x) zu bestimmen, genügt es

$$a_1(x) = \mathsf{GGT}(a(x), x^q - x)$$

zu berechnen.

Setzt man $a(x) = a(x)/a_1(x)$, so hat a keine linearen irreduziblen Faktoren, d.h.

$$a_2(x) = \mathsf{GGT}(a(x), x^{q^2} - x)$$

Usw. Hat a(x) Grad n, so muss man nur Faktoren bis zum Grad n/2bestimmen.

6.13 Beispiel 1
$$a = x(x+1)(x^2+1)(x^2+x+2) \in \mathbb{F}_3[x]$$
 getrennte GF $(x^2+x, \quad x^4+x^3+x+2)$ \uparrow \uparrow Grad1 Grad2 $\text{GGT}(a,x^3-x) = x^2+x, \quad \text{GGT}(a/x^2+x,x^9-x) = x^4+x^3+x+2$

Prof. Dr. K. Madlener: Computeralgebra

Anwendungen modularer und p-adischer Methoder

2) $a(x) = x^{63} + 1 \in \mathbb{F}_2[x]$, dann

Getrennte Grad Faktorisierung-Distinct Degree Factorizatio

Beispiel (Forts.)

$$\begin{array}{l} a_1(x) = \mathsf{GGT}(a(x), x^2 - x) = x + 1 & 1\text{-Faktor Grad 1} \\ a(x) = a(x)/a_1(x) = \frac{x^{63} + 1}{x + 1} = x^{62} + x^{61} + \dots + x^2 + x + 1 \\ a_2(x) = \mathsf{GGT}(a(x), x^4 - x) = x^2 + x + 1 & 1\text{-Faktor Grad 2} \\ a(x) = a(x)/a_2(x) = x^{60} + x^{57} + x^{54} + \dots + x^6 + x^3 + 1 \\ a_3(x) = \mathsf{GGT}(a(x), x^8 - x) = x^6 + x^5 + x^4 + x^3 + x^2 + x + 1 \end{array}$$

2-Faktoren Grad 3
$$a(x) = x^{54} + x^{53} + x^{51} + x^{50} + x^{48} + x^{46} + x^{45} + x^{42} + x^{33} + x^{30} + x^{29} + x^{27} + x^{25} + x^{24} + x^{22} + x^{21} + x^{12} + x^{11} + x^9 + x^8 + x^6 + x^4 + x^3 + x + 1$$

GGT(
$$a(x)$$
, $x^{64} - x$) = $a(x)$ = $a_6(x)$
 $x^{63} + 1 = (x+1)(x^2 + x + 1)(x^6 + x^5 + x^4 + x^3 + x^2 + x + 1)$ $a_6(x)$

 $GGT(a(x), x^{16} - x) = 1$. $GGT(a(x), x^{32} - 1) = 1$.

Getrennte Grad Faktorisierung-Distinct Degree Factorization

Algorithmus Getrennte Grad Faktorisierung

procedure PARTIALFACTOR-DD(a(x), a)

{Eingabe: Quadratfreies mon. Polynom $a(x) \in \mathbb{F}_{a}[x], n = \text{grad}(a) > 0$ } {Ausgabe: Getrennte Grad Zerlegung $(a_1, \ldots, a_s), s \leq n/2 \text{ von } a(x)$ }

- $w := x; a_0 := 1; i := 0;$
- repeat (2)

i := i + 1; call wied. quadrat. Algorithm. in $R = \mathbb{F}_{\sigma}(x)/\langle a(x) \rangle$

- um $w = w^q \mod a(x)$ zu berechnen (3)
- $a_i := \mathsf{GGT}(w x, a(x));$ (4)if $a_i(x) \neq 1$ then $a(x) := a(x)/a_i(x); w(x) := w(x) \text{ mod } a(x);$ until a(x) = 1: return (a_1,\ldots,a_i)

Die GGT-Berechnungen GGT($a(x), x^{q'} - x$) werden durch Berechnung von $x^{q^i} - x$ modulo a(x), d. h. Berechnung wird in $\mathbb{F}_a[x]/(a(x))$ durchgeführt

(z. B. wiederholtes Quadrieren um $(x^{q^{i-1}})^q \mod (a(x))$ zu berechnen).

Prof. Dr. K. Madlener: Computeralgebra

Anwendungen modularer und p-adischer Methoden

Getrennte Grad Faktorisierung-Distinct Degree Factorization

Algorithmus Getrennte Grad Faktorisierung (Forts.)

6.14 Satz Algorithmus Getrennte Grad Faktorisierung ist korrekt, d. h. es wird die getrennte Grad-Zerlegung von a berechnet.

Aufwand: $O(sM(n)\log(nq))$ Operationen in \mathbb{F}_q , wobei s der größte Grad eines irreduziblen Faktors von a ist.

z.Z. Für i-ten Durchgang gilt:

 $w_i \equiv x^{q'} \mod f_i, \quad f_i = G_{i+1} \cdots G_t, a_i = G_i \text{ für } i > 1,$ wobei (G_1, \ldots, G_t) die getrennte Grad-Zerlegung von a ist.

Induktion nach i: i = 0 klar. i > 0 wegen

 $w_i \equiv w_{i-1}^q \equiv (x^{q^{i-1}})^q = x^{q^i} \mod f_{i-1}$ d. h. $w_i - x \equiv x^{q^i} - x \mod f_i$ und

$$a_i = GGT(w_i - x, f_{i-1}) = GGT(x^{q^i} - x, f_{i-1})$$

Also ist a_i Produkt aller monisch irreduziblen Polynome in $\mathbb{F}_{\sigma}[x]$ deren Grad i teilt und $f_{i-1} = G_i \cdots G_t$ teilen, d. h. $a_i = G_i$ und somit $f_i = G_i \cdots G_t / G_i = G_{i+1} \cdots G_t$. i = t beim Ausgang.

Getrennte Grad Faktorisierung-Distinct Degree Factorization

Algorithmus Getrennte Grad Faktorisierung (Forts.)

Kosten für die Berechnung von w_i in Schritt (2) $O(\log q)$ Multiplikationen mod a, d. h. $O(M(n)\log q)$ Operationen in \mathbb{F}_q .

Die Kosten in (3) und (4) sind ebenfalls $O(M(n) \log n)$ Operationen in $\mathbb{F}_q[x]$.

Berechnung kann gestoppt werden sobald grad $f_i = \text{grad } a(x) < 2(i+1)$, da alle irreduziblen Faktoren von f_i grad mindestens i+1 haben, d. h. a(x) ist irreduzibel. Mit dieser Überprüfung: early abort

Somit $i = \max\{m_1/2, m_2\} \le n/2$, wobei m_1 und m_2 die Grade des größten und zweitgrößten irreduziblen Faktors von a(x) sind.

Beachte in Schritt 2 w_i wird nur mod f_{i-1} benötigt.



Prof. Dr. K. Madlener: Computeralgebra

Anwendungen modularer und p-adischer Methoder Getrennte Grad Faktorisierung-Distinct Degree Factorization

Beispiel

6.15 Beispiel Sei q = 3 Algorithmenverlauf für

$$a(x) = x^8 + x^7 - x^6 + x^5 - x^3 - x^2 - x \in \mathbb{F}_3[x]$$

 $a'(x) = -x^7 + x^6 - x^4 + x - 1$ GGT $(a, a') = 1$, d. h. QF

$$w_1 = x^3 \mod a = x^3$$

$$a_1 = \mathsf{GGT}(x^3 - x, a) = x \neq 1$$

$$f_1 = a/a_1 = x^7 + x^6 - x^5 + x^4 - x^2 - x - 1$$
 (neues a)

 w_1 unverändert x^3

$$w_2 = w_1^3 \mod a = x^9 \mod a = -x^7 + x^6 + x^5 + x^4 - x$$

$$a_2 = GGT(w_2 - x, f_1) = GGT(-x^7 + x^6 + x^5 + x^4 + x, f_1)$$

$$= x^{4} + x^{3} + x - 1$$

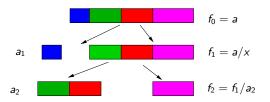
$$f_{2} = f_{1}/a_{2} = a/a_{2} = \frac{x^{7} + x^{6} - x^{5} + x^{4} - x^{2} - x - 1}{x^{4} + x^{3} + x - 1}$$

$$= x^3 - x + 1$$

Prof. Dr. K. Madlener: Computeralgebra

Beispiel (Forts.)

Der Algorithmus würde noch eine Iteration durchführen aber $grad(f_2) < 2(2+1) = 6 \Leftrightarrow nicht notwendig, da f_2 irreduzibel.$ a hat einen Lin-Faktor x, zwei verschiedene irreduziblen guadratische Faktoren, da Grad $a_2 = 4$ und einen irreduziblen kubischen Faktor $x^3 - x + 1$.



Anwendungen modularer und p-adischer Methoden

Prof. Dr. K. Madlener: Computeralgebra

Equal-Degree Factorization (Gleiche-Grad-Faktorisierung)- Algorithmus von Cantor und Zassenhaus

Equal-Degree Factorization (Gleiche-Grad-Faktorisierung)

Der Algorithmus von Cantor und Zassenhaus

Faktorisiere die ai, die aus der Getrennte-Grad-Faktorisierung berechnet

Ungerade Primzahlpotenzen, Char 2 Fall getrennt.

6.16 Beispiel
$$a(x) = x^{15} - 1 \in \mathbb{F}_{11}[x]$$
. DDF liefert

$$a(x) = a_1(x)a_2(x) = (x^5 - 1)(x^{10} + x^5 + 1)$$

a hat 5 lineare Faktoren, 5 irreduzible quadratische Faktoren.

Probabilistische Verfahren um Faktoren zu finden.

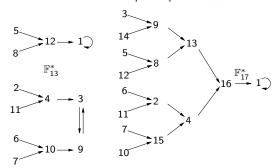
4□ > 4問 > 4 = > 4 = > 9 Q ○

Prof. Dr. K. Madlener: Computeralgebra

4 D F 4 B F 4 B F 9 9 9 9

Gleiche-Grad-Faktorisierung (1)

Betrachte Quadrat-Abbildung $\sigma: \mathbb{F}_q^* \to \mathbb{F}_q^*$ mit $\sigma(a) = a^2$, z. B.



Jedes Element hat entweder zwei oder 0 eingehende Pfeile. Zwei bedeutet ist Quadrat

gleiche Anzahl

0 bedeutet ist kein Quadrat

< □ ▶ 4큠 ▶ 4블 ▶ 4블 ▶ 볼 ∽)Q(

Prof. Dr. K. Madlener: Computeralgebra

Anwendungen modularer und p-adischer Methoder

Equal-Degree Factorization (Gleiche-Grad-Faktorisierung)- Algorithmus von Cantor und Zassenhaus

Gleiche-Grad-Faktorisierung (2)

6.17 Lemma Sei q Primzahlpotenz, $k \mid q-1$. $S = \{b^k : b \in \mathbb{F}_q^*\}$ die Menge der k-ten Potenzen in \mathbb{F}_q^* . Dann gilt

- i) S ist eine Untergruppe der Ordnung (q-1)/k
- ii) $S = \{ a \in \mathbb{F}_q^* : a^{(q-1)/k} = 1 \}$

Beweis: S als Bild eines Homomorphismus $(\sigma_k : a \to a^k)$ ist Untergruppe von \mathbb{F}_a^* .

Der Kern von σ_k ist $\ker \sigma_k = \{a \in \mathbb{F}_q^* : \sigma_k(a) = 1\} = \{a \in \mathbb{F}_q^* : a^k = 1\}$ d.h die Menge der k-ten EW. Da \mathbb{F}_q Körper ist hat $x^k - 1 \in \mathbb{F}_q[x]$ höchstens k Wurzeln in $\mathbb{F}_q[x]$, d. h. $|\ker \sigma_k| \leq k$. Wegen $(b^k)^{(q-1)/k} = b^{q-1} = 1$ für $b \in \mathbb{F}_q^*$ (Fermat), gilt $S \subseteq \ker \sigma_{(q-1)/k}$, d. h. $|S| \leq (q-1)/k$. Also

$$\begin{array}{l} q-1=|\mathbb{F}_q^*|=|\ker\sigma_k||\mathrm{Bild}\ \sigma_k|=|\ker\sigma_k|\cdot|S|\leq k(q-1)/k=q-1 \leadsto \\ |\ker\sigma_k|=k \quad |S|=(q-1)/k \ \mathrm{und}\ S=\ker\sigma_{(q-1)/k} \end{array}$$

Gleiche-Grad-Faktorisierung (3)

Equal-Degree Factorization (Gleiche-Grad-Faktorisierung)- Algorithmus von Cantor und Zassenhaus

Wendet man das Lemma 6.17 mit k = 2 und k = (q - 1)/2 an, so gilt

6.18 Lemma Sei q ungerade Primzahlpotenz und $S = \{a \in \mathbb{F}_q^* : \exists b \in \mathbb{F}_q^* \ a = b^2\}$ Menge der Quadrate. Dann

- i) $S \subseteq \mathbb{F}_q^*$ ist multiplikative Ugr. der Ordnung (q-1)/2
- ii) $S = \{ a \in \mathbb{F}_q^* \mid a^{(q-1)/2} = 1 \}$
- iii) $a^{(q-1)/2} \in \{1, -1\}$ für alle $a \in \mathbb{F}_q^*$

Faktorisierungsaufgabe: Sei $a \in \mathbb{F}_q[x]$, grad a = n, monisch und $d \in \mathbb{N}^+$ mit $d \mid n$ und jeder irreduzible Faktor von a habe den Grad d. Dann gibt es r = n/d solcher Faktoren und $a = f_1 \cdots f_r$, f_i verschiedene monische irreduziblen in $\mathbb{F}_q[x]$ o.B.d.A. $r \geq 2$. Bestimme die f_i .

(ロ) (部) (目) (目) (目) (9)

Prof. Dr. K. Madlener: Computeralgebra

Anwendungen modularer und p-adischer Methoden

Equal-Degree Factorization (Gleiche-Grad-Faktorisierung)- Algorithmus von Cantor und Zassenhaus

Gleiche-Grad-Faktorisierung (4)

Da $\mathsf{GGT}(f_i, f_j) = 1$ für $i \neq j$, gibt es nach chinesischem Restesatz Ring Homomorphismus

$$\chi: R = \mathbb{F}_q[x]/\langle a \rangle \to \mathbb{F}_q[x]/\langle f_1 \rangle \times \cdots \times \mathbb{F}_q[x]/\langle f_r \rangle = R_1 \times \cdots \times R_r$$

Die R_i sind Körper mit q^d Elemente und algebraische Erweiterungen vom Grad d von \mathbb{F}_q , d . h . alle isomorph.

$$\mathbb{F}_{q^d} \cong R_i = \mathbb{F}_q[x]/\langle f_i \rangle \supseteq \mathbb{F}_q$$

Für $f \in \mathbb{F}_q[x]$. Sei $f \mod a \in R$ und $\chi(f \mod a) = (f \mod f_1, \ldots, f \mod f_r) = (\chi_1(f), \ldots, \chi_r(f))$, wobei $\chi_i(f) = f \mod f_i \in R_1$ gilt. Es gilt für $f \in \mathbb{F}_q[x]$, i < r, $f_i \mid f \mod \chi_i(f) = 0$.

Hat man ein $f \in \mathbb{F}_q[x]$ mit einigen $\chi_i(f) = 0$ und anderen nicht null, so ist $\mathsf{GGT}(f,a)$ ein nichttrivialer Teiler von a.

 \leadsto Probabilistisches Verfahren um Spaltungspolynom f von a zu bestimmen.

Gleiche-Grad-Faktorisierung (5)

Sei q ungerade. Setze $e = (q^d - 1)/2$.

Für alle $\beta \in R_i^* = \mathbb{F}_{\sigma^d}^*$ gilt $\beta^e \in \{1, -1\}$ und beide Möglichkeiten treten gleich oft vor (Lemma 6.18 mit q^d an Stelle von q).

Wählt man $f \in \mathbb{F}_q[x]$ mit Grad f < n und GGT(a, f) = 1 zufällig, so sind $\chi_1(f),\ldots,\chi_r(f)$ unabhängige uniform verteilte Elemente aus $\mathbb{F}_{\sigma^d}^*$ und $\varepsilon_i = \chi_i(f^e) \in R_i$ ist 1 oder -1. Jedes mit Wahrscheinlichkeit 1/2.

Somit

$$\chi(f^e-1)=(\varepsilon_1-1,\ldots,\varepsilon_r-1)$$

und $f^e - 1$ ist Spaltungspolynom, es sei denn $\varepsilon_1 = \cdots = \varepsilon_r$. Dieses kann mit Wahrscheinlichkeit $2(1/2)^r = 2^{-r+1} < 1/2$ vorkommen.



4□ > 4個 > 4 = > 4 = > = 90

Prof. Dr. K. Madlener: Computeralgebra

Anwendungen modularer und p-adischer Methoder

Equal-Degree Factorization (Gleiche-Grad-Faktorisierung)- Algorithmus von Cantor und Zassenhau

Beispiel

6.19 Beispiel Fortsetzung:: In $\mathbb{F}_{11}[x]$

$$a(x) = (x^5 - 1)(x^{10} + x^5 + 1) = a_1 a_2$$

5 lineare Faktoren. 5 guadratische Faktoren.

$$n = 5$$
 $d = 1$ $e = (11^{1} - 1)/2 = 5$ $a_1 = x^5 - 1$

Zufallspolvnom: x + 4

$$GGT(a_1, (x + 4)^5 - 1) = x^2 + 5x + 5$$

(x⁵ - 1) = (x² + 5x + 5)(x³ - 5x² - 2x + 2)

Zufallspolynom: x + 8

$$GGT(x^2 + 5x + 5, (x + 8)^5 - 1) = x - 1 \text{ mit}$$

$$x^2 + 5x + 5 = (x - 1)(x - 5)$$
 2 lineare Faktoren.

GGT
$$(x^3 - 5x^2 - 2x + 2, (x + 8)^5 - 1) = x - 4$$
, wobei $x^3 - 5x^2 - 2x + 2 = (x - 4)(x^2 - x + 5)$.

Man erhält
$$a_1(x) = (x-1)(x-3)(x-4)(x-5)(x+2)$$
.

Beispiel (Forts.)

Spaltung von $a_2(x)$ nach Zufallsmuster $e = (11^2 - 1)/2 = 60$

Equal-Degree Factorization (Gleiche-Grad-Faktorisierung)- Algorithmus von Cantor und Zassenhaus

Zufallspolynom: x + 2

$$GGT(a_2(x), (\underline{x+2})^{60} - 1) = x^6 + 3x^5 + 4x^4 - 2x^3 + 5x^2 + 4x - 2$$

$$a_2(x) = (x^6 + 3x^5 + 4x^4 - 2x^3 + 5x^2 + 4x - 2)(x^4 - 3x^3 + 5x^2 - x + 5)$$

Versuche mit x + 7

$$\begin{aligned} \mathsf{GGT}(x^4-3x^3+5x^2-x+5, (\underline{x+7})^{60}-1) &= x^2+3x-2 \text{ und} \\ \mathsf{GGT}(x^6+3x^5+4x^4-2x^3+5x^2+4x-2, (x+7)^{60}-1) &= x^4+2x^3+x^2-5x-2 \end{aligned}$$

3-Faktoren Grad 2. verwende $x^4 + 2x^3 + x^2 - 5x - 2 \Rightarrow$

$$(x^2+3x-2)(x^2+5x+3)(x^2+4x+5)(x^2-2x+4)(x^2+x+1) = a_2(x).$$

Prof. Dr. K. Madlener: Computeralgebra

Anwendungen modularer und p-adischer Methoden

Equal-Degree Factorization (Gleiche-Grad-Faktorisierung)- Algorithmus von Cantor und Zassenhaus

Algorithmus: Gleiche-Grad-Faktorisierung

procedure Equal Degree Splitting $(a(x), d, a = p^m)$

{Eingabe: QF monisches Polynom $a \in \mathbb{F}_q[x]$, grad a = n, $g = p^m$, p ungerade, d < n, $d \mid n$, alle irreduzibeln Faktoren von a mit Grad d

{Ausgabe: Ein echter monischer Faktor $g \in \mathbb{F}_q[x]$ von a oder "Failure"}

- 1 Wähle $f \in \mathbb{F}_q[x]$ mit grad f < n zufällig if $f \in \mathbb{F}_a$ then return "Failure"
- $2 g_1 := GGT(a, f)$

if $g_1 \neq 1$ then return g_1

3 Call repeated squaring algorithm in $\mathbb{F}_a[x]/\langle a(x)\rangle$

um $b = f^{(q^d-1)/2} \mod a(x)$ zu berechnen

4 $g_2 := GGT(b-1, a)$

if $g_2 \neq 1$ and $g_2 \neq a$ then return g_2

else return ...Failure"

end.



Algorithmus (Forts.)

6.20 Satz Der Algorithmus ist korrekt bzgl. seiner Spezifikation.

"Failure" wird mit der Wahrscheinlichkeit $< 2^{1-r} < 1/2$ mit r = n/d > 2ausgegeben.

Die Anzahl der erwarteten Operationen in \mathbb{F}_q ist $O((d \log q + \log n)M(n))$.

Beweis:

- ▶ Für GGT(a, f) = 1, so 2^{-r+1} als Fehlerwahrscheinlichkeit, wegen Schritt (2): $< 2^{-r+1}$.
- ▶ Kosten für die Schritte 2) und 4) $0(M(n) \log n)$.
- ▶ Schritt 3: $2 \log_2(q^d) \in O(d \log q)$ Multiplikationen mod a, d. h. $0(M(n)d\log q)$ Operationen in \mathbb{F}_q .
- ▶ Ruft man den Algorithmus k mal auf, so gilt Failure Wahrscheinlichkeit $< 2^{(1-r)k} < 2^{-k}$.



Prof. Dr. K. Madlener: Computeralgebra

Anwendungen modularer und p-adischer Methoder

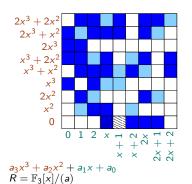
Equal-Degree Factorization (Gleiche-Grad-Faktorisierung)- Algorithmus von Cantor und Zassenhau

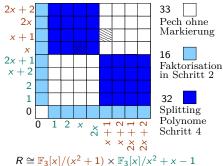
Algorithmus: Beispiel in $\mathbb{F}_3[x]$

- $a(x) = x^8 + x^7 x^6 + x^5 x^3 x^2 x$ hat einen linearen Faktor: x. zwei irreduzible Faktoren Grad 2:: $x^4 + x^3 + x - 1$ d = 2, einen irreduziblen Faktor Grad 3:: $x^3 - x + 1$
- $ightharpoonup a(x) = x^4 + x^3 + x 1$ faktorisiert sich in r = 2 irreduziblen Polynome mit Grad d = 4/r = 2.
- ightharpoonup Angenommen f = x + 1 erste Wahl. Dann ist $g_1 = GGT(f, a) = GGT(x + 1, x^4 + x^3 + x - 1) = 1$ $b = (x+1)^4 \mod a = (x+1)^4 \mod x^4 + x^3 + x - 1 = -1$ $g_2 = \mathsf{GGT}(b-1, a) = \mathsf{GGT}(1, a) = 1$ Pech gehabt!
- ightharpoonup Zweite Wahl: f = x. Dann $g_1 = GGT(f, a) = GGT(x, x^4 + x^3 + x - 1) = 1$ $b = x^4 \mod a = -x^3 - x + 1$ $g_2 = GGT(b-1, a) = GGT(-x^3 - x, x^4 + x^3 + x^2 - 1) = x^2 + 1 \Rightarrow$
- $ightharpoonup x^2 + 1$ ist einer der irr. Faktoren und $a/g_2 = x^2 + x + 1$ der andere.

Algorithmus (Forts.)

Equal-Degree Factorization (Gleiche-Grad-Faktorisierung)- Algorithmus von Cantor und Zassenhaus





Will man alle r-Faktoren bestimmen, so rekursive Anwendung auf die einzelnen Spaltungs-Faktoren.

Prof. Dr. K. Madlener: Computeralgebra

 $a_i \in \mathbb{F}_3$

Anwendungen modularer und p-adischer Methoden

Equal-Degree Factorization (Gleiche-Grad-Faktorisierung)- Algorithmus von Cantor und Zassenhaus

Algorithmus Gleiche Grad Faktorisierung

procedure Equal Degree Fact (a(x), d, q){Eingabe: QF monisches Polynom $a \in \mathbb{F}_q[x]$, p ungerade,} $\{q = p^m, \text{grad } a = n, d \mid n \text{ alle irreduziblen Faktoren grad } d\}$ {Ausgabe: die monischen irreduziblen Faktoren von a in $\mathbb{F}_a[x]$ }

- begin
- if n = d then return a call Equal Degree Splitting (a(x), d, q) bis
- ein echter Faktor $g \in \mathbb{F}_q[x]$ von a gefunden. $FAC \leftarrow Equal_Degree_Fact(g, d, q) \cup Equal_Degree_Fact(a/g, d, q)$ return (FAC) end.

Algorithmus Gleiche Grad Faktorisierung (Forts.)

6.21 Satz Ein QF-Polynom vom Grad $n = r \cdot d$ mit r irreduziblen Faktoren vom Grad d kann vollständig durch diesen Algorithmus faktorisiert werden mit einer erwarteten Anzahl von Operationen in \mathbb{F}_q von $0((d \log q + \log n)M(n) \log r)$.

Die Arbeitsweise der Prozedur kann mit Hilfe eines markierten Baums beschrieben werden. Die Marken der Knoten sind Faktoren von a.

- a Marke der Wurzel.
- ▶ Die Blätter sind markiert mit den irreduziblen Faktoren von a.
- ▶ Falls in Schritt 2 Failure, so ist ein Sohn mit gleicher Marke, sonst sind 2 Söhne mit Marken g bzw. a/g.

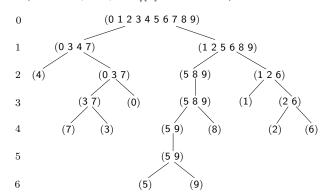
Prof. Dr. K. Madlener: Computeralgebra

Anwendungen modularer und p-adischer Methoder

Equal-Degree Factorization (Gleiche-Grad-Faktorisierung)- Algorithmus von Cantor und Zassenhau

Algorithmus Gleiche Grad Faktorisierung (Forts.)

6.22 Beispiel $a = f_0 \dots f_9 \in \mathbb{F}_q[x]$, f_i mon. irr. paarweise verschieden.



4 0 3 4 40 3 4 5 3 4 5 5 4 5 5

Equal-Degree Factorization (Gleiche-Grad-Faktorisierung)- Algorithmus von Cantor und Zassenhaus

Algorithmus Gleiche Grad Faktorisierung (Forts.)

- ▶ Produkt der Marken in einer Stufe ist Teiler von a, d.h. Grad vom Produkt höchstens n.
- ▶ Kosten für Knoten vom Grad m ist $0((d \log q + \log m)M(n))$ Operationen in \mathbb{F}_a . Subadditivität vom $M \rightsquigarrow \text{Kosten für jede Stufe}$: $O((d \log q + \log n)M(n))$ Operationen.
- ▶ Erwartete Tiefe ist $O(\log r)$ ($r < n \rightsquigarrow$ Behauptung).
- ► Tiefenschranke: Beweis Im Algorithmus Equal Degree Splitting ist die Wahrscheinlichkeit, das $f \mod f_i$ und $f \mod f_i$ weder beide Quadrate, noch beide nicht Quadrate, mindestens 1/2. (Chin-RS)
- \triangleright Die Wahrscheinlichkeit, dass f_i und f_i in Stufe k durch einen Aufruf von EDS getrennt werden (falls sie noch nicht getrennt sind) ist somit mindestens 1/2. Also ist die Wahrscheinlichkeit, dass f_i und f_i in Stufe knoch nicht getrennt sind höchstens $(1/2)^k$ und dies gilt für jedes Paar irreduzibler Faktoren von a.

Prof. Dr. K. Madlener: Computeralgebra

Anwendungen modularer und p-adischer Methoden

Equal-Degree Factorization (Gleiche-Grad-Faktorisierung)- Algorithmus von Cantor und Zassenhaus

Algorithmus Gleiche Grad Faktorisierung (Forts.)

- ▶ Es gibt $(r^2 r)/2 < r^2$ solcher Paare.
- \triangleright Die Wahrscheinlichkeit p_k , dass nicht alle irreduziblen Faktoren in Tiefe kgetrennt sind, ist höchstens r^22^{-k} .
 - Diese ist die Wahrscheinlickeit, dass der Baum die Tiefe > k hat und $p_{k-1} - p_k$ ist die Wahrscheinlichkeit der Baumtiefe genau k.
- ▶ Sei $s = [2 \log_2 r]$, dann ist die erwartete Baumtiefe

$$\sum_{k \ge 1} k(p_{k-1} - p_k) = \sum_{k \ge 0} p_k = \sum_{0 \le k < s} p_k + \sum_{s \le k} p_k \le \sum_{0 \le k < s} 1 + \sum_{s \le k} r^2 2^{-k} = s + r^2 2^{-s} \sum_{k > 0} 2^{-k} \le s + 2 \in 0(\log r).$$

- Beispiel: Tiefe $6 < \lceil 2 \log_2 10 \rceil + 2 = 9$.
- ▶ Für Varianten siehe vzG,G Übung 14.7.

4 D > 4 D > 4 D > 4 D > 3 D 9 O

Fall Charakteristik 2

Für Char= 2 Varianten der Algorithmen: Verwende *m*-tes Spur-Polynom über \mathbb{F}_2

$$T_m = x^{2^{m-1}} + x^{2^{m-2}} + \dots + x^4 + x^2 + x \in \mathbb{F}_2[x]$$

Angenommen $q=2^k$ für ein $k\in\mathbb{N}^+$, $f\in\mathbb{F}_q[x]$ quadratfrei, grad f=n, mit r > 2 irreduziblen Faktoren $f_1, \ldots, f_r \in \mathbb{F}_q[x]$ $R_i = \mathbb{F}_{\sigma}[x]/\langle f_i \rangle$ $\chi_i : R \to R_i$ wie gehabt. $R = \mathbb{F}_{a}[x]/\langle f \rangle$

- i) $x^{2^m} + x = T_m(T_m + 1) \rightsquigarrow T_m(\alpha) \in \mathbb{F}_2$ für $\alpha \in \mathbb{F}_{2^m}$ $(T_m \text{ ist } \mathbb{F}_2 \text{ linear})$ $T_m(\alpha) = 0$ und $T_m(\alpha) = 1$ gleichwahrscheinlich 1/2
- ii) Angenommen alle irreduziblen Faktoren von f haben den grad d, dann ist $\chi_i(T_{kd}(\alpha)) \in \mathbb{F}_2$ für $\alpha \in R$, somit für $\alpha \in R$ zufällig \rightsquigarrow $T_{kd}(\alpha) \in \mathbb{F}_2$ mit Wahrscheinlichkeit $2^{1-r} < 1/2$.
- iii) Berechne $b = T_{kd}(f)$ mod a im Algorithmus Equal Degree Splitting.



Prof. Dr. K. Madlener: Computeralgebra

Anwendungen modularer und p-adischer Methoder

Equal-Degree Factorization (Gleiche-Grad-Faktorisierung)- Algorithmus von Cantor und Zassenhau

Eigenschaften

Die wesentlichen Eigenschaften, die verwendet wurden, sind folgende Faktorisierungen:

► Für *q* ungerade:

*
$$x^q - x = x(x^{(q-1)/2} - 1)(x^{(q-1)/2} + 1)$$

d. h. für $W = \{v(x) \in \mathbb{F}_q[x] : v(x)^q = v(x) \mod a(x)\}$ und $v(x) \in W$ ist $v(x)(v(x)^{(q-1)/2}-1)(v(x)^{(q-1)/2}+1)=v(x)^q-v(x)\equiv 0 \mod a(x)$ und die nichttrivialen gemeinsamen Faktoren von $v(x)^q - v(x)$ verteilen sich auf die drei Polynome.

Für q gerade, d. h. $q = 2^k$, gilt * nicht, aber

**
$$x^{2^k} + x = T_k(x)(T_k(x) + 1)$$

 $\text{ Wahrscheinlichkeit } \mathsf{GGT}(\overset{T_{kd}(f)}{\leadsto},a) \text{ nicht trivial } \geq 1/2 \\ (f^{(q-1)/2}-1,a)$

4日 > 4間 > 4目 > 4目 > 三

Vollständiger Faktorisierungsalgorithmus für endliche Körper

Equal-Degree Factorization (Gleiche-Grad-Faktorisierung)- Algorithmus von Cantor und Zassenhau

Eingabe: Polynom $a(x) \in \mathbb{F}_q[x], a \notin \mathbb{F}_q, q = p^m, p$ Primzahl.

Ausgabe: Die monischen irreduziblen Faktoren von a mit ihren Vielfachheiten.

Monisch → QFF-Faktorisierung → DD-Faktorisierung → ED-Faktorisierung.

Aufwand für grad a = n: Erwartete Anzahl von OP in \mathbb{F}_a $O(nM(n)\log(qn))$, d. h. polynomial in n und $\log q$.

 $n^2 + n \log q$ Operationen in \mathbb{F}_q (mit Frobenius Aut.) siehe vz G/G. Auch für Variante ohne QFF-Faktorisierung zu verwenden (S. 365). Frobenius Automorphismus: $\sigma: \mathbb{F}_{q^n} \to \mathbb{F}_{q^n}$, $a \to a^q$, es gilt $\sigma^n = id$ und kann als Automorphismus von $R = \mathbb{F}_q[x]/\langle f \rangle$ für f quadratfrei betrachtet werden → Iterated Frobenius (Siehe S.374).

Prof. Dr. K. Madlener: Computeralgebra

Anwendungen modularer und p-adischer Methoder

Anwendung: Nullstellen-Bestimmun

Anwendung: Nullstellen-Bestimmung

Problem: Bestimme Nullstellen von $a(x) \in \mathbb{F}_a[x]$. Es genügt die linearen irreduziblen Faktoren von a(x) zu berechnen, d. h. $GGT(x^q - x, a(x)) = g$ und dann ED-Faktorisierung anzuwenden.

procedure Root Finding (a(x), q){Eingabe: nichtkonstantes Polynom $a(x) \in \mathbb{F}_q[x], q = p^m$.} {Ausgabe: Die Nullstellen von a(x) in \mathbb{F}_a .}

- 1 call Repeated Squaring Algorithmus in $R = \mathbb{F}_a[x]/(a(x))$ zur Berechnung von $x^q \mod a(x) =: h$
- 2 g := GGT(h-x,a)if g = 1 then return \varnothing
- 3 call Equal Degree Fact(g, 1, q)// es werden die irreduziblen linearen Faktoren $x - u_1, \dots, x - u_r$ mit r = grad g berechnet / /
- 4 return u_1, \ldots, u_r



Anwendungen modularer und p-adischer Methoden

Anwendung: Nullstellen-Bestimmung

Anwendung: Nullstellen-Bestimmung

Anwendung: Nullstellen-Bestimmung (Forts.)

6.23 Folgerung Für $a(x) \in \mathbb{F}_q[x]$ mit grad a = n berechnet Root_Finding alle Nullstellen von a(x) in \mathbb{F}_q die erwartete Anzahl von Operationen in \mathbb{F}_q ist:

 $0(M(n)\log n\log(nq))$

Nullstellen-Bestimmung in $\mathbb Z$

Problem: $a(x) \in \mathbb{Z}[x]$. Bestimme alle Nullstellen in \mathbb{Z} .

▶ Anwendung modularer Methoden, z.B. der Big-Prime Methode.

<ロ > → □ > → □ > → □ → ○ へ ○

Prof. Dr. K. Madlener: Computeralgebra

Anwendungen modularer und p-adischer Methoden

Anwendung: Nullstellen-Bestimmung

Anwendung: Nullstellen-Bestimmung in $\mathbb Z$

```
procedure Root_Finding_\mathbb{Z} (a(x), p) {Eingabe: nicht-konstantes Polynom a(x) \in \mathbb{Z}[x], grad a = n,} { Max_Norm (a) = \parallel a \parallel_{\infty} = A} {Ausgabe: Nullstellen von a in \mathbb{Z}}
```

begin

- 1 $B := 2n(A^2 + A)$ p Primzahl zwischen B + 1 und 2B
- 2 Call Root_Finding(a(x),p) //bestimmt $\{u_1 \mod p, \ldots, u_r \mod p\}$ Nullstellenmenge in \mathbb{F}_p von $a(x) \mod p$, $u_i \in \mathbb{Z}$, $|u_i| < p/2$ alle i//
- 3 for $1 \le i \le r$ berechne $v_i \in \mathbb{Z}[x]$ mit $\operatorname{grad}(v_i) = n - 1$, $||v_i||_{\infty} < p/2$ und $a(x) \equiv (x - u_i)v_i \mod p$
- 4 return $\{u_i : 1 \le i \le r, |u_i| \le A \land ||v_i||_{\infty} \le nA\}$ end

Anwendung: Nullstellen-Bestimmung in Z (Forts.)

6.24 Satz Korrektheit und Komplexität:

Der Algorithmus berechnet alle ganzzahligen Nullstellen von a(x). Die Kosten für Schritt 2 sind

$$0(M(n)\log n\log(nA)M(\log(nA))\log\log(nA))$$

und von Schritt 3 pro u_i sind $0(nM(\log(nA)))$ Wortoperationen. Falls p größer als Computerwort, so $\log^2 p$ bzw. $M(\log p)$ Wortoperationen.

Beweis

Die Schranke folgt aus Folgerung unter Berücksichtigung von $\log p \in O(\log(nA))$. Die Kosten für jedes u_i in Schritt 3 sind O(n) Additionen und Multiplikationen in \mathbb{F}_p , d. h. $O(nM(\log(nA)))$ Wortoperationen.

Die Kosten für die Bestimmung von p werden hier nicht berücksichtigt.

Prof. Dr. K. Madlener: Computeralgebra

Anwendungen modularer und p-adischer Methoden

Anwendung: Nullstellen-Bestimmung

Anwendung: Nullstellen-Bestimmung in \mathbb{Z} (Forts.)

Korrektheit: o.b.d.A. $a(0) \neq 0$ (Division von Potenzen von x).

- ▶ Gilt a(u) = 0 für ein $u \in \mathbb{Z}$, so $(x u) \mid a(x)$ und somit teilt u den konstanten Koeffizienten von a(x), d. h. $|u| \le A < p/2$. d.h. Alle verschiedenen Nullstellen von a(x) in \mathbb{Z} können eindeutig aus ihren modp Bilder wiedergewonnen werden.
- ▶ Zeige: $a(u_i) = 0$ gdw $|u_i| \le A$ und $||v_i||_{\infty} \le nA$.
- ▶ Angenommen $a(u_i) = 0$ dann $|u_i| \le A$ (siehe oben) und $\|a/(x u_i)\|_{\infty} \le nA < p/2$. Da $a/(x u_i) \equiv v_i \mod p$ und beide Seiten mit |Koeff.| $< p/2 \leadsto$ gleich $\leadsto \|v_i\|_{\infty} \le nA$.
- ▶ Umgekehrt: Sei $|u_i| \le A$ und $||v_i||_{\infty} \le nA$, dann $||(x-u_i)v_i||_{\infty} \le (1+A)nA < p/2$, d. h. die Kongruenz $a(x) \equiv (x-u_i)v_i$ mod p ist eine Identität.

Beachte die Wahl von $p \ge B+1 > 2n(A^2+A)$ garantiert p/2 > A, nA, (1+A)nA.

Faktorisierungsalgorithmen, die auf linearer Algebra basierer

Faktorisierungsalgorithmen, die auf linearer Algebra basieren

Faktorisierungsalgorithmen, die auf linearer Algebra basieren

Die Algorithmen von Berlekamp 1967/1970.

Erste Faktorisierungsalgorithmen für Polynome über endliche Körper, die pol. Laufzeiten hatten.

Anstelle der Getrennte-Grad Faktorisierung werden Methoden der linearen Algebra verwendet um das Polynom zu spalten.

- ▶ Sei $a(x) \in \mathbb{F}_q[x]$ quadratfrei, monisch grad n > 0.
- ▶ $R = \mathbb{F}_q[x]/\langle a \rangle$ ist Vektorraum der Dimension n über \mathbb{F}_q (sogar eine \mathbb{F}_q -Algebra).
- ▶ Die Abbildung $\beta = \sigma id : R \to R$ mit $\beta(f) = f^q f$ ist \mathbb{F}_q -linear.
- ▶ Wie bestimmt man den Kern von β :

□▶→□▶→□▶→□▶ □ ∽9へ@

2

Anwendungen modularer und p-adischer Methoder

Prof. Dr. K. Madlener: Computeralgebra

Faktorisierungsalgorithmen, die auf linearer Algebra basieren

Grundlagen für Berlekamps Algorithmen

▶ Ist $a = f_1 \cdots f_r$ die Faktorisierung von a in verschiedenen monischen irreduziblen Polynome aus $\mathbb{F}_q[x]$, so gilt nach chinesischem Restsatz

$$R \cong \mathbb{F}_q[x]/\langle f_1 \rangle \times \cdots \times \mathbb{F}_q[x]/\langle f_r \rangle$$

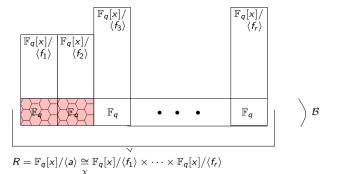
- ▶ Die $\mathbb{F}_q[x]/\langle f_i \rangle$ sind Körper mit $q^{\operatorname{grad} f_i}$ Elementen und enthalten \mathbb{F}_q (Konstanten mod f_i).
- $\begin{array}{ll} \text{F\"{u}r } f \in \mathbb{F}_q[x] \text{ gilt} \\ f \bmod a \in \ker \beta & \iff f^q \equiv f \bmod a \\ & \iff f^q \equiv f \bmod f_i \quad \text{f\"{u}r } 1 \leq i \leq r \\ & \iff f \bmod f_i \in \mathbb{F}_q \quad \text{f\"{u}r } 1 \leq i \leq r \end{array}$

Nach kleinem Fermat (alle Nullstellen von x^q-x liegen in \mathbb{F}_q , da $x^q-x=\prod_{a\in\mathbb{F}_q}(x-a)$ in $\mathbb{F}_q[x]$).

◆ロト 4周ト 4 章 ト 4 章 ト 章 めなべ。

Grundlagen für Berlekamps Algorithmen

Also ist $\mathcal{B} = \text{Kern } \beta$ der Unterraum, der $\mathbb{F}_q \times \cdots \times \mathbb{F}_q = \mathbb{F}_q^r$ entspricht.



Prof. Dr. K. Madlener: Computeralgebra

4 D > 4 B > 4 B > 4 B > B 990

Faktorisierungsalgorithmen, die auf linearer Algebra basierer

Anwendungen modularer und p-adischer Methoder

Grundlagen für Berlekamps Algorithmen

- $ightharpoonup \mathcal{B}$ ist sogar eine \mathbb{F}_q -Unteralgebra von R: Die Berlekamp-Unteralgebra.
- ▶ d.h. $f \mod a \in \mathcal{B} \iff \chi(f \mod a) = (a_1 \mod f_1, \cdots, a_r \mod f_r)$ für Konstanten $a_1, \ldots, a_r \in \mathbb{F}_q$.
- ▶ Die Matrix $Q \in \mathbb{F}_q^{n \times n}$, die den Frobenius-Hom. $\sigma: f \to f^q$ bezüglich der Basis $x^{n-1} \mod a, \cdots, x \mod a, 1 \mod a$ von R darstellt, heißt Petr-Berlekamp-Matrix von a. $x^{qj} \equiv q_{j,0} + q_{j,1}x + \cdots + q_{j,n-1}x^{n-1}$

Der Berlekamp Faktorisierungsalgorithmus basiert nun auf folgenden Berechnungen:

- ▶ Bestimme zunächst eine Basis $b_1 \mod a, \ldots, b_r \mod a$ von \mathcal{B} durch Gauss-Elimination angewendet auf Q I.
- ▶ Beachte: a ist irreduzibel \iff $r = 1 \iff$ Rang (Q I) = n 1. r gibt somit die Anzahl der irreduziblen Faktoren von a(x) an.

Prof. Dr. K. Madlener: Computeralgebra 384

Berlekamp Faktorisierungsalgorithmus

- ▶ Angenommen q sei ungerade und sei $b=c_1b_1+\cdots+c_rb_r$ eine zufällige Linearkombination der Basiselemente mit $c_1,\ldots,c_r\in\mathbb{F}_q$ unabhängig gewählt, d. h. b mod a ist ein zufälliges Element aus \mathcal{B} .
- ▶ Wende (q-1)/2 Trick, wie bei ED-Faktorisierung, an: Die $b \mod f_i$ sind gleichmäßig zufällig verteilte Elemente von \mathbb{F}_q für $1 \le i \le r$. Falls keines der f_i b teilt, so ist $b^{(q-1)/2} \equiv \pm 1 \mod f_i$ und beide Möglichkeiten treten mit Wahrscheinlichkeit 1/2 auf, unabhängig für alle i (Lemma 6.18).
- ▶ Beachte: Falls b nicht konstant ist, so gilt $a(x) = \prod_{s \in \mathbb{F}_q} \mathsf{GGT}(b-s, a(x))$ Da $x^q x = \prod_{s \in \mathbb{F}_q} (x-s)$, d.h. $b^q b = \prod_{s \in \mathbb{F}_q} (b-s)$

und
$$f_i \mid a$$
 für alle $i \rightsquigarrow f_i \mid b^q - b = \prod_{s \in \mathbb{F}_q} (b - s)$

<ロ > → □ > → □ > → □ > → □ = → つへで

Prof. Dr. K. Madlener: Computeralgebra

385

Anwendungen modularer und p-adischer Methoden

Faktorisierungsalgorithmen, die auf linearer Algebra basieren

Berlekamp Faktorisierungsalgorithmus (Forts.)

Für $s \neq t$ gilt aber $\mathsf{GGT}(b-s,b-t) = 1$, d. h. für gegebenes i gilt $f_i \mid b-s_k$ für genau ein s_k , d. h.

$$a = GGT(b^q - b, a) = GGT(\prod_{s \in \mathbb{F}_a} (b - s), a) = \prod_{s \in \mathbb{F}_a} GGT(b - s, a)$$

▶ Berechnung von *Q*:

$$\begin{array}{lll} x^q \bmod a(x), & x^{2q} \bmod a(x), \dots, x^{(n-1)q} \bmod a(x) \\ a(x) & = & a_0 + a_1 x + \dots + a_{n-1} x^{n-1} + x^n \\ x^m & \equiv & r_{m,0} + r_{m,1} x + \dots + r_{m,n-1} x^{n-1} \bmod a(x) \\ x^{m+1} & \equiv & -r_{m,n-1} a_0 + (r_{m,0} - r_{m,n-1} a_1) x + \dots \\ & & + (r_{m,n-2} - r_{m,n-1} a_{n-1}) x^{n-1} \\ & \equiv & r_{m+1,0} + r_{m+1,1} x + \dots + r_{m+1,n-1} x^{n-1} \end{array}$$

 $r_{m+1,0} = -r_{m,n-1}a_0$ $r_{m+1,i} = r_{m,i-1} - r_{m,n-1}a_i$ $i = 1, \dots, n-1 \rightsquigarrow O(qn^2)$ Körperoperationen.

Berlekamp-Algorithmus: Ein echter Faktor

{Eingabe: Monisches QF-Polynom $a \in \mathbb{F}_q[x]$, grad(a) = n, q ungerade PZP} {Ausgabe: Entweder echter faktor g von a oder Failure, d.h. Las-Vegas Typ PA}

- 1 Call repeated squaring Algorithm in $\mathbb{F}_q[x]/\langle a \rangle$ zur Berechnung von $x^q \mod a$;
- 2 **for** $0 \le i < n$ berechne $x^{qi} \mod a = \sum_{0 \le j < n} q_{ij} x^j$ $Q := (q_{ij})_{0 \le i,j < n}$; //siehe auch Bemerkung zur Berechnung von Q//
- 3 Wende Gausselimination auf $Q-I \in \mathbb{F}_q^{n \times n}$ an um die Dimension und eine Basis $b_1 \mod a, \ldots, b_r \mod a$ der Berlekamp Algebra \mathcal{B} zu bestimmen, hierbei sind $b_1, \ldots, b_r \in \mathbb{F}_q[x]$ mit Graden < n if r=1 then return a:
- 4 Wähle unabhängige zufällige $c_1, \ldots, c_r \in \mathbb{F}_q$ $b := c_1 b_1 + \cdots + c_r b_r$;
- 5 $g_1 := \mathsf{GGT}(b, a)$; if $g_1 \neq 1$ then return g_1 ;
- 6 Call repeated squaring algorithm in $R = \mathbb{F}_q[x]/a$ zur Berechnung von $f := b^{(q-1)/2} \mod a$:
- 7 $g_2 := \mathsf{GGT}(f-1,a);$ if $g_2 \neq 1$ and $g_2 \neq a$ then return g_2 else return "Failure"

Prof. Dr. K. Madlener: Computeralgebra

207

Anwendungen modularer und p-adischer Methoden

Berlekamp-Algorithmus: Ein echter Faktor

6.25 Satz Der Algorithmus ist korrekt und Failure kommt mit Wahrscheinlichkeit $\leq 1/2$ vor. Er benötigt $0(n^3 + M(n)\log q)$ Operationen in \mathbb{F}_q .

Beweis: Korrektheit: Falls $g_1 \neq 1$, so echter Teiler. Falls $g_2 \neq 1$ und $g_2 \neq a$, so ebenfalls echter Teiler. Falls $g_1 = 1$ in Schritt 5, so ist g_2 trivial (d. h. 1 oder a). In Schritt 7 $\operatorname{gdw} f^{(q-1)/2} \equiv 1 \operatorname{mod} f_i$ für alle i. Diese Fälle kommen mit der Wahrscheinlichkeit 2^{-1} vor. Die Erfolgswahrscheinlichkeit ist mindestens $1 - 2 \cdot 2^{-r} \geq 1/2$. Da $r \geq 2$ ist. Die Kosten für Schritt 1 sind $0(M(n)\log q)$ Körper Operationen. Schritt 2 verwendet n-2 Multiplikationen $\operatorname{mod} a$, d. h. 0(nM(n)) Operationen in \mathbb{F}_q .

Kosten für Schritt 3 $0(n^3)$ dominiert die Kosten von 2, die $0(n\ r)$ Körper für Schritt 4 und die $0(M(n)\log n)$ Operationen der GGT Berechnungen in den Schritten 5 und 7.

6 benötigt $0(M(n) \log q)$ Körperoperationen.

4□ > 4ⓓ > 4 ≧ > 4 ≧ > □ = ∅ Q €

Faktorisierungsalgorithmus

Für eine vollständige Faktorisierung von a(x) wird die Basis von \mathcal{B} nur einmal berechnet, der Spaltungsprozess der Schritte 4-7 wird rekursiv auf g und a/g angewandt. Alle irreduziblen Faktoren mit erwarteten Operationenzahl von $0(n^3 + M(n) \log r \log q)$.

Viele Varianten + Verbesserungen des Berlekamp Algorithmus in der Literatur. Problem für q groß: Die Kosten für die Erzeugung von Q und die Berechnung der GGT's wird durch $O(qkn^2)$ dominiert. Somit nur brauchbar für kleine q's. Etwa Polynom mit 4 Faktoren mit grad n = 100über $\mathbb{F}_{3^{14}}$ benötigt 191 Milliarden Körperoperationen. Generiere Matrix durch binäres Potenzieren. Variante von Zassenhaus (siehe Geddes et Al S 360).

Kaltofen und Lobo: Minimalpolynom-Berechnung.

$$0(M(n^2)\log n + M(n)\log q).$$

Dies ist wichtig, falls log q klein im Vergleich zu n ist.

Variante, Big Prime Berlekamp Algorithm siehe Geddes et.al.

Prof. Dr. K. Madlener: Computeralgebra

Anwendungen modularer und p-adischer Methoder

Faktorisierungsalgorithmen, die auf linearer Algebra basierer

Beispiel

6.26 Beispiel

$$a(x) = x^6 - 3x^5 + x^4 - 3x^3 - x^2 - 3x + 1 \in \mathbb{Z}_{11}[x] = \mathbb{F}_{11}[x]$$

 $Q \text{ Matrix } 6 \times 6, \ x^{q \cdot j} \equiv q_{j,0} + q_{j,1}x + \dots + q_{j,5}x^5 \text{ mod } a.$
Zeile 0 von $Q (1,0,0,0,0,0), 1 \equiv 1 \text{ mod } a(x)$

Zeile 0 von
$$Q$$
 (1,0,0,0,0,0), 1 \equiv 1 mod $a(x)$
 $x \equiv x \mod a(x)$
 $x^2 \equiv x^2 \mod a(x)$
 $x^3 \equiv x^3 \mod a(x)$
 $x^4 \equiv x^4 \mod a(x)$
 $x^5 \equiv x^5 \mod a(x)$
 $x^6 \equiv 3x^5 - x^4 + 3x^3 + x^2 + 3x - 1 \mod a(x)$
 $x^7 \equiv 3x^6 - x^5 + 3x^4 + x^3 + 3x^2 - x$
 $\equiv -3x^5 - x^3 - 5x^2 - 3x - 3 \mod a(x)$
 \vdots
 $x^{11} \equiv 5x^5 - 5x^4 - 3x^3 - 3x^2 + 5x + 3 \mod a(x)$

Faktorisierungsalgorithmen, die auf linearer Algebra basieren

Beispiel (Forts.)

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 3 & 5 & -3 & -3 & -5 & 5 \\ 3 & -5 & -5 & 1 & -1 & 0 \\ -2 & 4 & -1 & 3 & -4 & -2 \\ -4 & -3 & -1 & 0 & 0 & -3 \\ -3 & -1 & -4 & -3 & -1 & -3 \end{bmatrix}$$

Basis für Q - I, bringe in Δ -Form 0,1 in diagonalen, falls 1 einzige 1 in Zeile dreiecksidempotenter Form.

$$Q - I = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 4 & -3 & -3 & -5 & 5 \\ 3 & -5 & 5 & 1 & -1 & 0 \\ -2 & 4 & -1 & 2 & -4 & -2 \\ -4 & -3 & -1 & 0 & -1 & -3 \\ -3 & -1 & -4 & -3 & -1 & -4 \end{bmatrix}$$

Prof. Dr. K. Madlener: Computeralgebra

Anwendungen modularer und p-adischer Methoden

Faktorisierungsalgorithmen, die auf linearer Algebra basierer

Beispiel (Forts.)

$$L = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & -1 & -1 & 0 & 0 \\ 0 & 0 & 4 & 2 & 0 & 0 \end{bmatrix}$$

idemp-Form,

Basis für Nullraum ablesen, da (I-L)L = 0

◆ロト 4個ト 4 恵 ト 4 恵 ト . 重 . 夕久の

Faktorisierungsalgorithmen, die auf linearer Algebra basieren

Beispiel (Forts.)

Basis für \mathcal{B} :

$$v^{(1)} = (1, 0, 0, 0, 0, 0), v^{(2)} = (0, 1, 1, 1, 1, 0), v^{(3)} = (0, 0, -4, -2, 0, 1)$$

d. h. als Polynome.

$$v^{(1)}(x) = 1$$
 $v^{(2)}(x) = x^4 + x^3 + x^2 + x$ $v^{(3)}(x) = x^5 - 2x^3 - 4x^2$

 \rightarrow a(x) faktorisiert sich in drei irreduzible Faktoren.

►
$$GGT(a(x), v^{(2)}(x)) = x + 1 \rightsquigarrow f_1(x) = x + 1$$

$$\frac{a(x)}{x+1} = x^5 - 4x^4 + 5x^3 - 4x + 1$$

►
$$GGT(a(x), v^{(2)}(x) + \frac{1}{3.4.5.6}) = 1$$
 $GGT(a(x), v^{(2)}(x) + 2) = 1$

► GGT(
$$a(x)$$
, $v^{(2)}(x) + 7$) = $x^3 + 2x^2 + 3x + 4 = f_2(x)$
 $a(x)/g_1(x) = x^2 + 5x + 3 = f_3(x)$

ロ ト 4 同 ト 4 三 ト 4 三 ト 9 0 0

< ロ ト → 付 ト → 三 ト → 三 → り Q (~)

Prof. Dr. K. Madlener: Computeralgebra

393

Anwendungen modularer und p-adischer Methoder

Beispiel (Forts.)

Zufalls-Element aus B

$$v(x) = 3v^{(1)}(x) - 2v^{(2)}(x) + 5v^{(3)}(x) = 5x^2 - 2x^4 - x^3 - 2x + 3$$

$$GGT(a(x), v(x)^5 - 1) = x^5 - 4x^4 + 5x^3 + 3x^2 - 4x + 1$$

►
$$a(x) = (x+1)\underbrace{(x^5 - 4x^4 + 5x^3 + 3x^2 - 4x + 1)}_{2 \text{ irr. Faktoren}} \rightsquigarrow f_1(x) = x+1$$

$$v(x) = 2v^{(1)}(x) + 3v^{(2)}(x) + 4v^{(3)}(x) = 4x^5 + 3x^4 - 5x^3 - 2x^2 + 3x + 2$$

$$\mathsf{GGT}(x^5 - 4x^4 + 5x^3 - 2x^2 - 4x + 1, v(x)^5 - 1) = 1 \; \mathsf{Pech!}$$

▶
$$v(x) = v^{(1)}(x) + 3v^{(2)}(x) - 4v^{(3)}(x) = -4x^5 + 3x^4 - 3x^2 + 3x + 1$$

 $GGT(x^5 - 4x^4 \cdots, v(x)^5 - 1) = x^2 + 5x + 3$
 $(x^2 + 5x + 3)(x^3 + 2x^2 + 3x + 4)$ restlichen Faktoren.

Anwendungen modularer und p-adischer Methoden

Anwendung: Irreduzible Polynome: Test und Konstruktion

Anwendung:

Irreduzible Polynome:: Test und Konstruktion

Faktorisierungsalgorithmen können für Irreduzibilitätstests verwendet werden:

z.B. Die Getrennte-Grad Faktorisierung kann angehalten werden, falls ein echter Faktor gefunden wurde oder bis zum Grad > n/2 kein Faktor gefunden wurde.

Alternativen

6.27 Lemma Ein Polynom $a \in \mathbb{F}_q[x], n = \operatorname{grad}(a) \geq 1$ ist genau dann irreduzibel, wenn

- i) $a \mid x^{q^n} x$ und
- ii) $GGT(x^{q^{n/t}} x, a) = 1$ für alle Primteiler t von n.

←□ → ←□ → ←□ → □ → ○

Prof. Dr. K. Madlener: Computeralgebra

Anwendungen modularer und p-adischer Methoden

Anwendung: Irreduzible Polynome: Test und Konstruktion

Irreduzibilitätstest: Beweis

- ▶ Wegen Lemma 6.12 $(x^{q^d} x)$ ist Produkt aller monischen irreduziblen Polynome in $\mathbb{F}_q[x]$ deren Grad d teilt) folgen i) und ii), falls a irreduzibel ist.
- ▶ Umgekehrt falls i) gilt, so teilt der Grad eines irreduziblen Faktors von a die Zahl n. Sei g ein solcher irreduzibler Faktor mit $d = \operatorname{grad}(g) < n$, d. h. $d \mid n/t$ für einen Primfaktor t von n, d. h. $g \mid x^{q^{n/t}} x \nleq zu$ ii), also d = n und a ist irreduzibel
- Hieraus lässt sich leicht ein Irreduzibilitätstest-Algorithmus herleiten: Rabin 1980.
- ▶ Berechne $f = x^{q^n} \mod a$ $f \neq x$, so ist a reduzibel.
- ▶ Für alle Primteiler t von n. Berechne $g = x^{q^{n/t}} \mod a$. if $\mathsf{GGT}(g x, a) \neq 1 \rightsquigarrow \mathsf{reduzibel}$.
- return irreduzible.

Prof. Dr. K. Madlener: Computeralgebra



Anwendung: irreduzible Polynome (Forts.)

6.28 Satz Der Algorithmus ist korrekt, d. h. er ist Entscheidungsalgorithmus für Irreduzibilität.

Koster

 $0(M(n)\log q + (n^{(w+1)/2} + n^{1/2}M(n))\delta(n)\log n)$ Operationen in \mathbb{F}_q .

Berechne zunächst $x^q \mod a \leadsto 0(M(n)\log q)$ $s_m := x^{q^m} \mod a$ für $m \in \mathbb{N}$. In $0((n^{(w+1)/2} + n^{1/2}M(n)), \ w \approx 2.376$ (Matrizenmultipl.)

Anzahl der m ber. $s_m: 1 + \delta(n)$ (Anzahl der Primteiler von n).

Als Irreduzibilitätstest eignet sich auch die Bestimmung vom Rang von Q-I (siehe Berlekamp).

 $0(n^w + M(n) \log q)$ Körperoperationen.

Für w = 3 ist der hier vorgestellte Algorithmus schneller.



Prof. Dr. K. Madlener: Computeralgebra

39

wendungen modularer und p-adischer Methoden

Anwendung: Irreduzible Polynome: Test und Konstruktion

Arithmetik in \mathbb{F}_{p^n}

Wie findet man irreduzible Polynome vom Grad n in $\mathbb{F}_p[x]$?

Wozu: Arithmetik in $GF(p^n)$, Konstruktion von Körpererweiterungen.

Probabilistische Verfahren

6.29 Lemma Sei q Primzahlpotenz $n \ge 1$. Die Anzahl l(n,q) der monisch irreduziblen Polynome vom Grad n in $\mathbb{F}_q[x]$ erfüllt:

$$\frac{q^n-2q^{n/2}}{n}\leq I(n,q)\leq \frac{q^n}{n}$$

Insbesondere, falls $q^n \geq 16$ so erfüllt p_n - die Wahrscheinlichkeit eines zufällig gewählten monischen Polynoms vom Grad n irreduzibel zu seindie Ungleichung

$$\frac{1}{2n} \le \frac{1}{n} \left(1 - \frac{2}{q^{n/2}} \right) \le p_n \le \frac{1}{n}$$

Anwendung: Irreduzible Polynome: Test und Konstrukti

Probabilistische Verfahren: Beweis vom Lemma

Beweis: Sei f_n Produkt aller monisch irreduziblen Polynome von grad n in $\mathbb{F}_q[x]$, d. h. grad $(f_n) = nI(n,q)$. Satz 6.12 kann umgeformt werden in

$$x^{q^n} - x = \prod_{d|n} f_d = f_n \prod_{d|n,d < n} f_d$$
, d.h. $q^n = \operatorname{grad}(f_n) + \sum_{d|n,d < n} \operatorname{grad}(f_d)$,

also

$$q^n \ge \operatorname{grad}(f_n) = n \cdot I(n, q) \rightsquigarrow \text{ obere Schranke.}$$

Es gilt, da q > 2

$$\sum_{d\mid n,d < n} \operatorname{grad}(f_d) \leq \sum_{1 \leq d \leq n/2} \operatorname{grad} f_d \leq \sum_{1 \leq d \leq n/2} q^d \leq \frac{q^{n/2+1}-1}{q-1} \leq 2q^{n/2}$$

somit

$$n \cdot I(n,q) = \operatorname{grad}(f_n) = q^n - \sum_{d \mid n,d \le n} \operatorname{grad}(f_d) \ge q^n - 2q^{n/2}$$

→ untere Schranke.

<ロ > ← □ > ← □ > ← 亘 > ← 亘 → の ⊆ → の ⊆

Prof. Dr. K. Madlener: Computeralgebra

Anwendungen modularer und p-adischer Methoden

Anwendung: Irreduzible Polynome: Test und Konstruktion

Probabilistische Verfahren (Forts.)

E Es gibt insgesamt q^n monische Polynome vom Grad n in $\mathbb{F}_q[x]$, d. h.

$$\frac{1}{n} \ge \frac{I(n,q)}{q^n} \ge \frac{1}{n}(1 - 2q^{-n/2}) \ge \frac{1}{2n}$$

für $q^n \ge 16$.

▶ Genaue Formel ist $n \cdot I(n,q) = \sum_{d|n} \mu\left(\frac{n}{d}\right) q^d$ (mithilfe der Möbius

Inversion). Hierbei ist μ die Möbius Funktion, d. h.

$$\mu(\textit{n}) = \begin{cases} 1 & \text{falls } \textit{n} = 1 \\ (-1)^k & \text{falls } \textit{n} \text{ Produkt von } \textit{k} \text{ verschiedenen} \\ & \text{Primzahlen ist} \\ 0 & \text{falls } \textit{n} \text{ nicht quadratfrei} \end{cases}$$

▶ Werte I(5, q) für q = 2, 3, 4, 5, 7, 8, 9

6 48 204 624 3360 6552 11808

Anwendungen modularer und p-adischer Methoden

Anwendung: Irreduzible Polynome: Test und Konstruktion

Anwendung: Irreduzible Polynome: Test und Konstruktion

Probabilitische Algorithmen zur Erzeugung irreduzibler Polynome mit grad n

```
procedure Ben Or's_Irreduzible_Polynome \{ \text{Eingabe: Primzahlpotenz } q, n \geq 1 \} \{ \text{Ausgabe: zufälliges monisches irred. Polynom mit Grad } n \text{ aus } \mathbb{F}_q[x] \} begin 1 \text{ wähle zufällig monisches Polynom } f \in \mathbb{F}_q[x] \text{ von Grad } n 2 \text{ for } i = 1, \ldots, \lceil n/2 \rceil \text{ do} g_i := \mathsf{GGT}(x^{q^i} - x, f); if g_i \neq 1 then goto 1 3 \text{ return } f end.
```

◆□▶ ◆□▶ ◆■▶ ◆■▶ ■ り900

Prof. Dr. K. Madlener: Computeralgebra

401

Anwendungen modularer und p-adischer Methoden

Anwendung: Irreduzible Polynome: Test und Konstruktion

Probabilitische Algorithmen (Forts.)

▶ Wie beim DD-Fakt Algorithmus zeigt man, dass die Anzahl der Operationen für Schritt 2 etwa $0(sM(n)\log(nq))$, $0^{\sim}(n^2\log q)$

(Soft 0 $f \in 0^{\sim}(g)$, falls es Konstanten $N, c \in \mathbb{N}$ gibt mit $f(n) \leq g(n)(\log_2(3+g(n)))^c$ für $n \geq N$).

► Mit Lemma 6.29 → erwartete Operationenzahl

$$0^{\sim}(n^3 \log q) \quad (0(nsM(n) \log(nq)))$$

Probabilitische Algorithmen (Forts.)

Es gilt jedoch

6.30 Lemma (ohne Beweis) Der Erwartungswert für den Grad des kleinsten irreduziblen Faktors eines zufällig gewählten Polynoms von Grad n aus $\mathbb{F}_q[n]$ ist $0(\log n)$.

6.31 Satz Ben Or's Algorithmus ist korrekt und die erwartete Anzahl von Operationen in \mathbb{F}_q ist

 $0(nM(n)\log n\log(nq))$ oder $0^{\sim}(n^2\log q)$.

Beweis

Faktorisierung in $R[x_1, \ldots, x_n]$, R ZPE Ring

Für ein $i: O(M(n)\log(nq))$, für 2: $O(M(n)\log n\log n(q))$. Anzahl der versuchten f:: O(n) wegen Lemma 6.29.

Prof. Dr. K. Madlener: Computeralgebra 403

Anwendungen modularer und p-adischer Methoden

Faktorisierung in $R[x_1, \ldots, x_n]$ R ZPE Ring

Zusammenfassung: Faktorisierung in $\mathbb{F}_q[x]$. ok.

- ▶ Faktorisierung in $\mathbb{Z}[x] \cong$ Faktorisierung in $\mathbb{Q}[x]$ + Faktorisierung in \mathbb{Z} . d.h. Faktorisierung primitiver Polynome aus $\mathbb{Z}[x]$ $\rightarrow a \in \mathbb{Z}[x]$ primitiv, o.b.d.A. a quadratfrei.
- ▶ Wähle "Big Prime" $p \in \mathbb{Z}$, $p \dagger \mathsf{LKoeff}(a)$, $a \bmod p \in \mathbb{F}_p[x]$ quadratfrei. Faktorisiere $a \bmod p$ in $\mathbb{F}_p[x]$. Wenn $a \bmod p$ irreduzibel in $\mathbb{F}_p[x]$, so auch a irreduzibel in $\mathbb{Z}[x]$.
- ► Wahl der Primzahl mithilfe Mignotte's Schranke (erlaubt die Rekonstruktion der Faktoren).
- ▶ Wie findet man die modularen Faktoren von $a \mod p$, die einen echten Faktor von a in $\mathbb{Z}[x]$ entsprechen? Vorgestellte Methode: Versuche alle möglichen Faktoren-Kombinationen.

Problem: Exponentielle Laufzeit möglich.



Faktorisierung in $R[x_1, \ldots, x_n]$, R ZPE Ring

Beispiel: Swinnerton-Dyer Polynome

$$a_i(x) = \prod (x \pm \sqrt{2} \pm \sqrt{3} \pm \sqrt{5} \pm \cdots \pm \sqrt{p_i}) \in \mathbb{Z}[x]$$

 p_i *i*-te Primzahl, Produkt läuft über alle 2^i möglichen Kombinationen der Vorzeichen \pm . *a* ist irreduzibel mit Grad 2^i .

Beachte: \mathbb{F}_{p^2} enthält alle Quadratwurzeln $\sqrt{2} \mod p \cdots \sqrt{p_i} \mod p$.

- ▶ Big-Prime Version
 - $B := (n+1)^{1/2} 2^n Ab$
 - b = LC(a)
 - ▶ n = grad(a)
 - $ightharpoonup A = \max-\operatorname{norm}(a)$

p ungerade Primzahl $2B : <math>a \mod p$ QF. $a \equiv bg_1 \cdots g_r \mod p$, d. h. in $\mathbb{Z}_p[x]$.



Prof. Dr. K. Madlener: Computeralgebra

405

Zusammenfassung: Faktorisierung

- ▶ Mit Hensel Lifting: Zassenhaus Algorithmus
 - $B := (n+1)^{1/2} 2^n Ab$ wie oben.
 - $C := (n+1)^{2n} A^{2n-1}$

d. h. $\gamma \sim 2\log_2 C \sim 2 \cdot [2n\log(n+1) + (2n-1)\log A] + 4n[\log(n+1) + \log[\max-norm(a)]]$

Wähle $p \le 2\gamma \ln \gamma$: $a \mod p$ QF

 $I = \lceil \log_p(2B+1) \rceil$ Hensel-Lifting nach $\mathbb{Z}_{p'}[x]$.

Problem: exp. Kombination der möglichen Faktoren. Kann durch andere Techniken in Polynom-Zeit (# Operationen) realisiert werden (siehe Kap 16 vz. Gathen, Gerhard, Short Vectors in Lattices, Lenstra Lovasz 16.5).

Heuristiken: Wähle mehrere Primzahlen p

6.32 Beispiel $a(x) = x^{16} + 11x^4 + 121$

▶ Faktorisierung in $\mathbb{Z}_{13}[x]$ liefert

a(x) = u(x)v(x) 41 Faktorpaarungen Grade $D_{13} = \{2, 3, 4, 5, 6, 7, 8\}$ (bis $\lfloor n/2 \rfloor$)

▶ Faktorisierung von a(x) in $\mathbb{Z}_{23}[x]$ liefert 8 irreduzible Faktoren, alle mit Grad 2 \rightsquigarrow 162 Paarungen.

Grade $D_{23} = \{2, 4, 6, 8\}$

- \rightarrow $D_{13,23} = \{2,4,6,8\}$ sind die möglichen Grade mindestens eines Faktors u(x)v(x) = a(x) in $\mathbb{Z}[x]$.
- → Anzahl der Kombinationen mod13 reduziert sich auf 25.

(D) (B) (B) (B) (B) (9)

Prof. Dr. K. Madlener: Computeralgebra

407

Anwendungen modularer und p-adischer Methoden

Faktorisierung in $R[x_1, \ldots, x_n]$, R ZPE Ring

Beispiel (Forts.)

▶ Faktorisierung von a(x) in $\mathbb{Z}_5[x]$ liefert 2 irreduzible Faktoren vom Grad 4 und 12, d. h.

 $D_5 = \{4\}$, nur ein Paar muss geliftet werden.

▶ Faktorisierung von a(x) in $\mathbb{Z}_{31}[x]$ liefert 2 irreduzible Faktoren vom Grad 8, d. h.

 $D_{31} = \{8\}$

 $\rightsquigarrow D_{5.31} = \{ \}, d. h. a ist irreduzibel.$

► Eisenstein Irreduzibilitäts-Test:

 $a \in \mathbb{Z}[x]$, p Primzahl mit $p \dagger LC(a)$ $p \mid$ alle anderen Koeffizienten von a, $p^2 \dagger a(0) \rightsquigarrow a$ ist irreduzibel in $\mathbb{Q}[x]$.

z.B. $x^n - p$ ist irreduzibel in $\mathbb{Q}[x]$ für alle n, p. Übung.

- ► Für die Swinnerton-Dyer Polynome gilt: sie lassen sich für jedes *p* in linearen und quadratischen Faktoren faktorisieren.
 - → obige Heuristik bringt dafür nichts!

Anwendungen modularer und p-adischer Methoden

Faktorisierung in K[x] für K algebraischer Zahlenkörper

Multivariate Polynomfaktorisierung in $\mathbb{Q}[x_1,\ldots,x_{\nu}]$ bzw. $\mathbb{Z}[x_1,\ldots,x_{\nu}]$

▶ Siehe Homomorphismus-Diagramm Fol. 303

- ▶ Problem Liste der korrekten multivariaten Leit-Koeffizienten. $a(x_1,\ldots,x_{\nu})\in\mathbb{Z}[x_1,\ldots,x_{\nu}]$ x_1 als Hauptvariable. $a(x_1,\ldots,x_{\nu})\equiv u_1(x_1)\cdots u_n(x_1)\in\mathbb{Z}[x_1]\mod\Phi_I$
- ▶ Leitkoeffizienten von $a(x_1,...,x_{\nu})$ (als Polynome in x_1) ist multivariates Polynom in Variablen x_2, \ldots, x_{ν} .



◆ロト 4個ト 4 恵 ト 4 恵 ト . 車 . 夕 Q (*)

Prof. Dr. K. Madlener: Computeralgebra

Anwendungen modularer und p-adischer Methodei

Faktorisierung in $R[x_1, \ldots, x_n]$, R ZPE Ring

Multivariate Polynomfaktorisierung... (Forts.)

Leitkoeffizienten Problem tritt auch hier auf, die Leitkoeffizienten der Faktoren müssen korrekt gewählt werden.

(Normierungstrick: korrekte Koeffizienten auf alle Faktoren verteilen). Wang's Lösung

$$a(x_1,...,x_{\nu}) = a_d(x_2,...,x_{\nu})x_1^d + \cdots$$

- Faktorisiere $a_d(x_2, \ldots, x_{\nu})$ (rekursiver Aufruf) Verteile die Faktoren von $a_d(x_2, \dots, x_{\nu})$ auf die $u_1(x_1), \dots, u_n(x_1)$ Geeignete Wahl von Φ_I . Auswertungspunkte: $\alpha_2, \ldots, \alpha_{\nu} \in \mathbb{Z}$ mit
 - 1. $a_d(\alpha_2,\ldots,\alpha_{\nu})\neq 0$.

Prof. Dr. K. Madlener: Computeralgebra

- 2. $a(x_1, \alpha_2, \dots, \alpha_{\nu})$ quadratfrei.
- 3. Jeder Faktor von $a_d(x_2, \ldots, x_{\nu})$ wenn ausgewertet in $\alpha_2, \ldots, \alpha_{\nu}$ hat Primzahlfaktor, der nicht in den anderen Auswertungen der restlichen Faktoren vorkommt.

Faktorisierung in K[x] für K algebraischer Zahlenkörper

Anwendung: Symbolische Integration

Trager (Kronecker).

Algebraische Zahlenkörpern, algebraische Erweiterungen von F, d. h. $F(\alpha) = F[x]/\langle m(x) \rangle$, m irreduzibles Polynom in F[x]. α ist "Wurzel" von m(x) mit Grad n (z. B. $x^2 + 1$ in $\mathbb{Q}[x]$ oder $\mathbb{R}[x]$)

$$F(\alpha) = \{ [f_0 + f_1 \alpha + \dots + f_{n-1} \alpha^{n-1}] : f_i \in F \}$$

6.33 Beispiel
$$F = \mathbb{Q}$$
, $\alpha = \sqrt{2}$, $m(x) = x^2 - 2$, dann $\mathbb{Q}[x]/\langle x^2 - 2 \rangle \simeq \mathbb{Q}(\sqrt{2}) = \{a + b\sqrt{2} : a, b \in \mathbb{Q}\}$ mit $(a + b\sqrt{2}) + (a' + b'\sqrt{2}) = (a + a') + (b + b')\sqrt{2}$ $(a + b\sqrt{2}) \cdot (a' + b'\sqrt{2}) = (aa' + 2bb') + (ab' + ba')\sqrt{2}$

Prof. Dr. K. Madlener: Computeralgebra

Anwendungen modularer und p-adischer Methoder

Faktorisierung in K[x] für K algebraischer Zahlenkörper

Grundlagen: Konjugation

- ▶ Sei m(x) eindeutiges monisches Minimalpolynom von α über F. Die Konjugierten von α über F sind die restlichen verschiedenen Nullstellen von m(x). Seien diese $\alpha_2, \ldots, \alpha_n$ z.B. $-\sqrt{2}$ ist konjugiert zu $\sqrt{2}$ über \mathbb{O}
- ▶ Sei $\beta \in F(\alpha)$ mit $\beta = f_0 + f_1\alpha + \cdots + f_{n-1}\alpha^{n-1}$. Die Konjugierten von β sind β_2, \ldots, β_n , wobei

$$\beta_i = f_0 + f_1 \alpha_i + \dots + f_{n-1} \alpha_i^{n-1}$$

► Konjugation induziert Isomorphismen:

$$\sigma_i : F(\alpha) \to F(\alpha_i) \text{ mit } \sigma_i(\beta) = \beta_i$$

6.34 Satz Sei $\beta \in F(\alpha_1, \ldots, \alpha_n)$.

 $\beta \in F$ gdw β invariant unter allen Permutationen der α_i .

Beweis: Fundamentalsatz für symmetrische Funktionen liefert: β kann eindeutig durch die elementarsymmetrischen Funktionen von $\alpha_1, \ldots, \alpha_n$ dargestellt werden (z. B. $\alpha_1 + \cdots + \alpha_n, \alpha_1 \cdot \alpha_2 \cdots \alpha_n, \cdots$). Diese können als Terme der Koeffizienten des Minimalpolynoms von α dargestellt werden, die dann in F liegen.

6.35 Definition Sei Norm: $F(\alpha) \to F$ definiert durch

- Norm $(\beta) = \beta \cdot \beta_2 \cdots \beta_n$ (d. h. Produkt aller Konjugierten). Da invariant unter Konjugation, folgt $Norm(\beta) \in F$.
- Norm kann auch mithilfe der Resultante beschrieben werden: $a[x], b[x] \in R[x]$ nicht null.

$$a(x) = \sum_{i=0}^{m} a_i x^i \quad b(x) = \sum_{i=0}^{n} b_i x^i$$

Prof. Dr. K. Madlener: Computeralgebra

Prof. Dr. K. Madlener: Computeralgebra

Anwendungen modularer und p-adischer Methoder

Faktorisierung in K[x] für K algebraischer Zahlenkörpe

Sylvester Matrix: Eigenschaften

Sylvester Matrix von a und b ist

$$M_{n+m,n+m} = \begin{cases} & \begin{cases} a_m & a_{m-1} & \cdots & a_1 & a_0 \\ & a_m & \cdots & \cdots & a_1 & a_0 \end{cases} \\ & & \vdots & & & & \\ & & a_m & \cdots & \cdots & a_0 \end{cases} \\ b_n & b_{n-1} & \cdots & b_1 & b_0 & & & \\ & & b_n & \cdots & \cdots & b_0 & & \\ & & & \vdots & & & \\ & & & b_n & \cdots & \cdots & b_0 \end{cases}$$

 $\operatorname{Res}_{\kappa}(a,b)$ ist die Determinante von M.

Wobei Res(0, b) = 0 für $b \neq 0$ und Res(a, b) = 1 für $a, b \in R^*$ definiert wird.

Sylvester Matrix: Beispiel

6.36 Beispiel Seien $a = 3vx^2 - v^3 - 4$, $b = x^2 + v^3x - 9 \in \mathbb{Z}[v][x]$ dann

$$\operatorname{Res}_{x}(a,b) = \det \begin{bmatrix} 3y & 0 & -y^{3} - 4 & 0 \\ 0 & 3y & 0 & -y^{3} - 4 \\ 1 & y^{3} & -9 & 0 \\ 0 & 1 & y^{3} & -9 \end{bmatrix}$$

$$= -3y^{10} - 12y^7 + y^6 - 54y^4 + 8y^3 + 729y^2 - 216y + 16$$

Man beachte, dass $\operatorname{Res}_{x}(a,b) \in \mathbb{Z}[y] = R$. Es gilt für $a(x) = a_{m} \prod_{i=1}^{m} (x - \alpha_{i})$ und $b(x) = b_{n} \prod_{i=1}^{n} (x - \beta_{i})$

$$\operatorname{Res}_{\mathbf{x}}(a,b) = a_m^n b_n^m \prod_{i=1}^m \prod_{j=1}^n (\alpha_i - \beta_j) = (-1)^{mn} b_n^m \prod_{j=1}^n a(\beta_i)$$

Prof. Dr. K. Madlener: Computeralgebra

Anwendungen modularer und p-adischer Methoden

Faktorisierung in K[x] für K algebraischer Zahlenkörpe

Grundlagen (Forts.)

- ▶ Eigenschaft: Zwei Polynome haben einen nichttrivialen gemeinsamen Faktor, falls Res(a, b) = 0.
- ightharpoonup Eigenschaft: Ist q(x) monisch, so gilt

$$\operatorname{Res}_{x}(p,q) = \prod_{x: q(x)=0} p(x), \text{ d. h.}$$

 $Norm(\beta) = Res_x(b(x), m(x)) = \beta \cdot \beta_2 \cdots \beta_n$ wobei β durch "Polynom" $b(\alpha)$ dargestellt wird.

▶ Fortsetzung der Norm auf $F(\alpha)[z]$. Sei $p \in F(\alpha)[z]$, d. h. p kann als bivariates Polynom in Variablen α und z betrachtet werden. Setze

 $Norm(p) = Res_x(p(x, z), m(x))$ liefert als Ergebnis ein Polynom in F[z].

Faktorisierung in K[x] für K algebraischer Zahlenkörpe

Grundlagen (Forts.)

- ▶ Beachte: $p(\alpha, z) \mid \text{Norm}(p) \text{ in } F(\alpha)[z].$
- ▶ Die Norm-Funktion ist multiplikativ, d. h. Norm(pq) = Norm(p)Norm(q) d.h. Jedes Polynom p welches in $F(\alpha)[z]$ in Faktoren zerfällt, liefert eine Faktorisierung von Norm(p) in F[z]
- ► Trager's Algorithmus basiert auf einer Umkehrung dieser Eigenschaft. Prozedur: Faktorisiere Norm(p) in F[z] und lifte diese Faktoren von F[z] zu Faktoren von p(z) in $F(\alpha)[z]$.
- ▶ Benötigt wir noch Irreduzibilitätstest in $F(\alpha)[z]$.

4日 → 4周 → 4 目 → 4 目 → 9 Q (*)

Prof. Dr. K. Madlener: Computeralgebra

Prof. Dr. K. Madlener: Computeralgebra

417

Anwendungen modularer und p-adischer Methoden

Faktorisierung in K[x] für K algebraischer Zahlenkörpe

Irreduzibilitätstest für $a(z) \in F(\alpha)[z]$

6.37 Satz Sei $a(z) \in F(\alpha)[z]$ irreduzibel über $F(\alpha)$. Dann ist Norm(a) Potenz eines irreduziblen Polynoms über F.

Beweis: Angenommen Norm(a) = b(z)c(z) mit teilerfremde Polynome $b, c \in F[z]$. Da $a(z) \mid Norm(a)$ in $F(\alpha)[z]$ und a(z) irreduzibel ist, gilt

- ▶ $a(z) \mid b(z)$ oder $a(z) \mid c(z)$ in $F(\alpha)[z]$.
- ▶ o.b.d.A. $a(z) \mid b(z)$: d.h. b(z) = a(z)d(z) mit $d(z) \in F(\alpha)[z]$ und teilerfremd zu a(z) da irreduzibel.
- ► Konjugation liefert $b(z) = \sigma_i(a(z))\sigma_i(d(z))$, d. h. $\sigma_i(a(z))$ ist Faktor von b(z) für alle i. Dann aber
- Norm(a) = $\prod_{i} \sigma_i(a) \mid b(z)$ also c(z) = 1. d.h. Norm(a) = b(z) und b(z) ist entweder irreduzibel oder Potenz eines irreduziblen Elements.

Faktorisierung in K[x] für K algebraischer Zahlenkörper

Folgerung

Satz 6.37 hat als Folgerung:

Angenommen $a(z) \in F(\alpha)[z]$ habe die Eigenschaft, dass

- ▶ Norm(a) quadratfrei in F[z]. Dann
 - ► a(z) irreduzibel gdw Norm(a) irreduzibel.
 - Falls a(z) in $F(\alpha)[z]$ sich faktorisieren lässt als $a(z) = a_1(z) \cdots a_k(z)$ mit $a_i(z)$ irreduzibel, so ist $\operatorname{Norm}(a) = \operatorname{Norm}(a_1)\operatorname{Norm}(a_2) \cdots \operatorname{Norm}(a_k)$, wobei jedes $\operatorname{Norm}(a_i)$ irreduzibel ist.
- ▶ Wenn Norm(a) quadratfrei ist, muss Norm(a_i) \neq Norm(a_j) für $i \neq j$ gelten. d.h.:
- Es gibt eine eindeutige Korrespondenz zwischen den Faktoren von a(z) über $F(\alpha)$ und den Faktoren von Norm(a) über F.

∢ロト→御ト→恵ト→恵・・恵・の

Prof. Dr. K. Madlener: Computeralgebra

Anwendungen modularer und p-adischer Methoden

Faktorisierung in K[x] für K algebraischer Zahlenkörper

Umkehrung

6.38 Satz Sei $a(z) \in F(\alpha)[z]$, Norm(a) quadratfrei. Ist $p_1(z), \ldots, p_k(z)$ eine vollständige Faktorisierung von Norm(a) über F[z], so ist

$$a(z) = \prod_{i=1}^{k} \mathsf{GGT}(a(z), p_i(z))$$

eine vollständige Faktorisierung von a(z) über $F(\alpha)[z]$.

Beweis:

Angenommen $a(z) = a_1(z) \cdots a_k(z)$ vollständige Faktorisierung von a(z) in $F(\alpha)[z]$, dann ist $\operatorname{Norm}(a) = \operatorname{Norm}(a_1) \cdots \operatorname{Norm}(a_k)$ eine Faktorisierung von $\operatorname{Norm}(a)$ in F[x]. d.h. Für jedes i gilt $p_i(z) = \operatorname{Norm}(a_j)$ für ein geeignetes j. Da $\operatorname{Norm}(a)$ quadratfrei, folgt $\operatorname{Norm}(a_i) \neq \operatorname{Norm}(a_h)$ für $h \neq j$.

◆□ > ◆@ > ◆差 > ◆差 > 差 め Q @

Prof. Dr. K. Madlener: Computeralgebra 4

Anwendungen modularer und p-adischer Methoden

Faktorisierung in K[x] für K algebraischer Zahlenkörper

Faktorisierung in K[x] für K algebraischer Zahlenkörpe

Umkehrung (Forts.)

▶ Behauptung: Gilt $p_i(z) = \text{Norm}(a_i)$, so

(*)
$$a_i(z) = \mathsf{GGT}(a(z), p_i(z))$$

wobei der GGT in $F(\alpha)[z]$ genommen wird.

- ▶ Aus der Definition der Norm und den Eigenschaften folgt $a_i(z) \mid a(z)$ und $a_i(z) \mid p_i(z)$ in $F(\alpha)[z]$. Ein größerer Teiler wurde bedeuten: es gibt $a_h(z) \mid a(z)$ und $p_i(z)$ in $F(\alpha)[z]$ für ein $h, h \neq j$.
- ▶ Da $a_h(z) \mid p_i(z)$, gilt Norm $(a_h) \mid \text{Norm}(p_i)$. Aber $p_i(z) \in F[z]$ so folgt $Norm(p_i) = p_i(z)^n$
- Aber Norm (a_h) ist irreduzibel, d. h. Norm $(a_h) = p_i(z)$ falls $h \neq i$, d. h. (*) gilt.

4□ ト 4団 ト 4 豆 ト 4 豆 ト 9 Q @

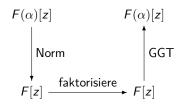
Prof. Dr. K. Madlener: Computeralgebra

Anwendungen modularer und p-adischer Methoder

Faktorisierung in K[x] für K algebraischer Zahlenkörpe

Faktorisierung in $F(\alpha)[z]$

 \blacktriangleright Wenn Norm(a) quadratfrei, so kann a(z) wie folgt faktorisiert werden:



- ▶ Ist $a(z) \in F(\alpha)[z]$ nicht quadratfrei, so kann man wie gehabt o.b.d.A. auf quadratfreien Fall reduzieren. Benötigt wird aber Norm(a) quadratfrei.
- ▶ Finde $s \in F$, so dass $b(z) = a(z + s\alpha)$ quadratfrei. Dann faktorisiere $b(z) = b_1(z) \cdots b_k(z)$.
- ▶ Die Faktorisierung für $a(z) = a_1(z) \cdots a_k(z)$ mit $a_i(z) = b_i(z s\alpha)$.

Faktorisierung in $F(\alpha)[z]$ (Forts.)

6.39 Satz Sei a(z) quadratfreies Polynom in $F(\alpha)[z]$. Dann ist $Norm(a(z - s\alpha))$ quadratfrei bis auf endlich vielen $s \in F$.

Beweis:

- ightharpoonup Sei $\operatorname{Norm}(a) = \prod p_i(z)^i$ quadratfreie Faktorisierung von $\operatorname{Norm}(a)$ in F[z].
- ▶ Da a(z) quadratfrei ist und a(z) | Norm $(a) \rightsquigarrow$ $a(z) \mid p(z) = p_1(z) \cdots p_r(z)$. Beachte hierbei $p(z) \in F[z]$.
- ▶ Seien β_1, \ldots, β_k die Wurzeln von p(z), d.h. $p(z) = \prod (z \beta_i)$.
- ▶ Da p(z) quadratfrei ist, so sind alle β_i verschieden.

Anwendungen modularer und p-adischer Methoden

Prof. Dr. K. Madlener: Computeralgebra

Faktorisierung in K[x] für K algebraischer Zahlenkörpe

Beweis (Forts.)

- $c_s(z) := \operatorname{Norm}(p(z s\alpha)) = \prod_i \prod_i (z (s\alpha_i + \beta_i)) \in F[z].$
- ▶ Dieses Polynom kann mehrfach Wurzeln haben gdw $s \cdot \alpha_j + \beta_i = s \cdot \alpha_u + \beta_v$ für j, i, u, v geeignet gdw $s = \frac{\beta_v - \beta_i}{\alpha_v - \alpha_v}$
- ▶ d. h. für fast alle $s \in F$ gilt Norm $(a(z s\alpha)) \mid p(z s\alpha) \mid c_s(z)$ mit $c_s(z)$ Polynom ohne Mehrfachwurzeln.
- ▶ In diesen Fällen ist $a_s(z) = a(z s\alpha)$ Polynom in $F(\alpha)[z]$ mit quadratfreier Norm.



Faktorisierung über alg. Zahlenkörpern

procedure Alg_Factorization($a(z), m(x), \alpha$) {Eingabe: quadratfreis $a(z) \in F(\alpha)[z]$ } $\{\alpha \text{ Alg-Zahl mit minimal Polynom } m(x), \text{ grad } n, \text{ a als Polynom in } \alpha, z\}$ {Ausgabe: Faktorisierung von a} 1 //Finde s : Norm($a_s(z)$) quadratfrei // s := 0; $a_s(\alpha, z) := a(\alpha, z)$; $Norm(a_s) := Res_x(m(x), a_s(x, z))$ while Grad (GGT (Norm (a_s) , Norm $(a_s)'$)) $\neq 0$ do s := s + 1; $a_s(\alpha, z) := a_s(\alpha, z - \alpha)$; $Norm(a_s) := Res_x(m(x), a_s(x, z))$ 2 //Faktorisiere Norm(a_s) in F[z] und lifte Ergebnis // $b := factors(Norm(a_s));$ if sizeof(b) = 1 then return a(z)else for each $a_i(z) \in b$ do $a_i(\alpha, z) := \mathsf{GGT}(a_i(z), a_s(\alpha, z))$ $a_i(\alpha, z) := a_i(\alpha, z + s\alpha)$ $substitute(a_i(z) \leftarrow a_i(\alpha, z), b)$ return (b)

Prof. Dr. K. Madlener: Computeralgebra

425

Anwendungen modularer und p-adischer Methoden

Faktorisierung in K[x] für K algebraischer Zahlenkörper

Beispiel

6.40 Beispiel Sei

$$a_{\alpha}(z) = z^4 + z^3 + (2 + \alpha - \alpha^2)z^2 + (1 + \alpha^2 - 2\alpha^3)z - 2 \in Q(\alpha)[z]$$

Mit $\alpha = 3^{1/4}$ minimal Polynom für α , $m(x) = x^4 - 3$.

► Norm(
$$a_{\alpha}(z)$$
) = Res_x($a_{\alpha}(z)$, $m(x)$) =
= $z^{16} + 4z^{15} + 14z^{14} + 32z^{13} + 47z^{12} + 92z^{11} + 66z^{10} + 120z^{9}$
-50 $z^{8} - 24z^{7} - 132z^{6} - 40z^{5} - 52z^{4} - 64z^{3} - 64z^{2} - 32z + 16$
Ist QF in $\mathbb{Q}[z]$

▶ Faktorisiert man in $\mathbb{Q}[z]$ so $Norm(a_{\alpha}(z)) = g(z) \cdot h(z)$ mit

$$g(z) = z^8 + 4z^7 + 10z^6 + 16z^5 - 2z^4 - 8z^3 - 20z^2 - 8z + 4$$

 $h(z) = z^8 + 4z^6 + 9z^4 + 4z^2 + 4$

Beispiel (Forts.)

▶ Berechne GGT in $\mathbb{Q}(\alpha)[z]$

$$\mathsf{GGT}(a_{\alpha}(z), g(z)) = z^2 + (1 - \alpha)z + (1 - \alpha^2)$$
 und

$$\mathsf{GGT}(a_{\alpha}(z),h(z)) = z^2 + \alpha z + (1 + \alpha^2)$$

Faktorisierung von $a_{\alpha}(z)$ in $\mathbb{Q}(\alpha)[z]$ ist

$$a_{\alpha}(z) = (z^2 + (1 - \alpha)z + (1 - \alpha^2))(z^2 + \alpha z + (1 + \alpha^2))$$

Problem:

- Gradwachstum von Norm(a).
- Koeffizientenwachstum

Forschungsgegenstand

- Vermeidung von Wachstum
- Berechnungen mit Körpererweiterungen
- Galoiskorrespondenz

◆ロト ◆部ト ◆恵ト ◆恵ト ・恵 ・ 夕 ♀

Prof. Dr. K. Madlener: Computeralgebra

427

Primzahltesten und Faktorisierung in Z

Inhalt Kapitel 7

Primzahltesten und Faktorisierung in $\ensuremath{\mathbb{Z}}$

Primzahltesten

Primzahlen mit vorgebenen Eigenschaften

Faktorisierung in $\ensuremath{\mathbb{Z}}$

Anwendung: Cryptosysteme

Prof. Dr. K. Madlener: Computeralgebra

Primzahltesten und Faktorisierung in $\ensuremath{\mathbb{Z}}$

Primzahltester

Primzahltesten und Faktorisierung in $\mathbb Z$ Eine Übersicht

- ▶ Gelöst: Komplexität vom Primzahltest:: In $P(O(n^6))$
- ▶ Offenes Problem: Komplexität der Faktorisierung.
- ▶ Vorhanden: pol. schnelle probabilitische Primzahltests.
- Spezielle Zahlen: Mersenne Zahlen: $M_n = 2^n 1$. Größte bekannte Mersenne Primzahl
 - ▶ M₃₀₂₁₃₇₇ Zahl mit 909526 Dezimalstellen.
 - ► Gimps: Great Internet Mersenne Prime Search.
 - ► Woltman/Kurowski → Internet Computing Paradigma
- ▶ $N \in \mathbb{Z}^+$ ist N Primzahl? (Trivialtest bis \sqrt{N} auf Teilbarkeit prüfen, exponentiell in $\log N$).

↓□▶ ◆□▶ ◆□▶ ◆□▶ □ ♥९९

Primzahltesten und Faktorisierung in Z

Primzahltesten

Primzahltesten: Grundlagen

$$\begin{split} & \mathbb{Z}_N^{\mathsf{x}} = \{ a \bmod N \in \mathbb{Z}_N : \mathsf{GGT}(a,N) = 1 \} \ \mathsf{mult}. \ \mathsf{Einheiten} \ \mathsf{Gruppe} \ \mathsf{in} \ \mathbb{Z}_N, \\ & \mathsf{Ordnung} \ \varphi(N) = |\mathbb{Z}_N^{\mathsf{x}}| \ \mathsf{Euler} \ \mathsf{Funktion}. \\ & N = p_1^{e_1} \cdots p_r^{e_r} \ \mathsf{paarweise} \ \mathsf{verschieden} \ p_i : \mathbb{Z}_N^{\mathsf{x}} \cong \mathbb{Z}_{p_1^{e_1}}^{\mathsf{x}} \times \cdots \times \mathbb{Z}_{p_r^{e_r}}^{\mathsf{x}} \\ & \begin{cases} N \ \mathsf{Primzahl}, \ \mathsf{so} \ \varphi(N) = N - 1 \\ N = p^I, \ \mathsf{so} \ \varphi(N) = p^{I-1}(p-1) \end{cases} \\ & \overset{}{\leftarrow} \varphi(N) = \prod_{i=1}^r p_i^{e_i-1}(p_i-1) \\ \mathsf{Fermat:} \ a^{N-1} \equiv 1 \ \mathsf{mod} \ N, \quad N \ \mathsf{Primzahl}, \quad \mathsf{GGT}(a,N) = 1 \end{split}$$

- Für $a, N \in \mathbb{Z}^+$ mit $\mathsf{GGT}(a, N) = 1$. Sei die Ordnung von $a \bmod N$ $\mathsf{ord}_N(a) = \min_{k > 1} \ a^k \equiv 1 \bmod N$.
- ▶ Euler: $a^{\varphi(N)} \equiv 1 \mod N$, d. h. ord_N(a) | $\varphi(N)$
- ▶ Falls p Primzahl, $l \ge 2$, $N = p^l$, $a = 1 + p^{l-1}$. Dann $\operatorname{ord}_N(a) = p$, da $a^p \equiv \sum_{0 \le i \le p} \binom{p}{i} p^{(l-1)} \equiv 1 \mod p^l$

- (ロ) (部) (注) (注) (注) (9)

Prof. Dr. K. Madlener: Computeralgebra

Primzahltesten und Faktorisierung in $\ensuremath{\mathbb{Z}}$

Primzahltester

Algorithmus Fermat's Test

```
\label{eq:procedure}  \mbox{ Fermat\_Test $N$ }  \mbox{ Eingabe: ungerade Zahl $N \geq 3$ }  \mbox{ Ausgabe: "Nicht Primzahl" oder "mögliche Primzahl"}
```

begin

- 1 Wähle $a \in \{2, \dots, N-2\}$ zufällig (Gleichverteilung)
- 2 Berechne $b := a^{N-1} \mod N$ durch wiederholtes Quadrieren
- 3 if $b \neq 1$ then return "nicht Primzahl" else return "mögliche Primzahl"

end

- ▶ Ist $GGT(a, N) \neq 1$ so $GGT(b, N) \neq 1$ d. h. $b \neq 1 \rightsquigarrow$ nicht Primzahl ist korrekt.
- ▶ Angenommen GGT(a, N) = 1, $b \neq 1$ Fermat \rightsquigarrow Antwort korrekt
- ▶ b = 1 Antwort "mögliche Primzahl".
 Wann ist für diese Antwort N keine Primzahl?

 ▶ Falls a im Test aus $\mathbb{Z}_N^{\times} \backslash L_N$ gewählt wurde, so liefert der Test "nicht Primzahl" \leadsto Fermat Zeuge für die nicht Primalität von N

▶ Falls $a \in L_N$, so a Fermatlügner für N.

Fermatsche Zahlen: $F_n = 2^{2^n} + 1$ sind Primzahlen für n = 0, 1, 2, 3, 4

► Fermat Vermutete: alle diese Zahlen sind Primzahlen (August 1640) kleiner Fermat Oktober 1640

4□ > 4圖 > 4 = > 4 = > = 900

Prof. Dr. K. Madlener: Computeralgebra

433

Primzahltesten und Faktorisierung in $\ensuremath{\mathbb{Z}}$

Primzahltesten

Fermat's Test: Ergebnisse

▶ $3^{2^{32}} \equiv 1461798105 \not\equiv 1 \mod 2^{32} + 1$, d.h. sein Satz wiederlegt seine Vermutung.

Problemzahlen: N nicht Primzahl, aber $L_N = \mathbb{Z}_N^{\times}$. Carmichael Zahlen (Zahlen ohne fermatsche Zeugen).

7.1 Satz

- ▶ Ist *N* Primzahl oder eine Carmichael Zahl, so liefert Fermats Test "mögliche Primzahl".
- ▶ Ist *N* nicht Primzahl und nicht Carmichael Zahl, so ist die Antwort "nicht Primzahl" mit Wahrscheinlichkeit mindestens 1/2.

Der Test benötigt $O(\log N \cdot M(\log N))$ Wortoperationen. (Z. B. $M(n) \in O(n \log n \log \log n)$ oder $\in O(n^2)$)

Fermat's Test: Verfeinerung

▶ Beachte: Carmichael Zahlen sind quadratfrei und $p-1 \mid N-1$ für jeden Primfaktor p von N, sie sind ungerade und haben mindestens 3 Primfaktoren:

 $561 = 3 \cdot 11 \cdot 17$, $1105 = 5 \cdot 13 \cdot 17$, $1729 = 7 \cdot 13 \cdot 19$.

- ► Fermat's Test liefert keine Teiler von *N*, falls die Antwort "nicht Primzahl" ist.
- ▶ Verfeinerung: Teiler für Carmichael Zahlen finden.
- ▶ Starke Fermat Zeugen und Starke Fermat Lügner

Starker Pseudoprimalitäts-Test

```
{Eingabe: ungerade Zahl N \geq 3} {Ausgabe: "Nicht Primzahl", "mögliche Primzahl", "echter Teiler von N"} begin 1 Wähle a \in \{2, \ldots, N-2\} zufällig 2 g := \mathsf{GGT}(a, N) if g > 1 then return g 3 Sei N-1=2^k m mit k, m \in N, k \geq 1, m ungerade. Berechne b_0 := a^m \mod N //durch wiederholtes quadrieren if b_0 = 1 then return "mögliche Primzahl" 4 for 1 \leq i \leq k do b_i := b_{i-1}^2 \mod N 5 if b_k = 1 then j := \min\{0 \leq i < k : b_{i+1} = 1\} else return "nicht Primzahl" 6 g := GGT(b_j + 1, N); if g = 1 or g = N then return "mögliche Primzahl" else return g end
```

< ロ ト < 個 ト < 重 ト く 重 ト 、 重 ・ 夕 Q (や)

Prof. Dr. K. Madlener: Computeralgebra 43

4 D F 4 B F 4 B F 9 9 9 9

Starker Pseudoprimalitäts-Test (Forts.)

7.2 Satz

Ist N Primzahl, so liefert der Algorithmus "mögliche Primzahl".

Ist N nicht Primzahl und nicht Carmichael Zahl, so liefert der

Algorithmus "nicht Primzahl" mit Wahrscheinlichkeit mindestens 1/2.

Ist N eine Carmichael Zahl, so liefert der Algorithmus einen echten Teiler von N mit Wahrscheinlichkeit mindestens 1/2.

Die Anzahl der Wortoperationen ist $O(\log N \cdot M(\log N))$.

Beweis: Zeitschranke folgt aus der Vorüberlegung: 2 und 6 kosten O(M(logN)loglogN) Wortoperationen, 3 und 4 O(logN) Multiplikationen mod N d.h. $O(logN \cdot M(logN))$

Beachte: $b_i \equiv a^{2^i m} \mod N$ für $0 \le i \le k$.

Insbesondere $b_k \equiv a^{N-1} \mod N$.

Ist $b_{i-1} = 1$, so auch $b_i = 1$ für alle i.

Ist N nicht Primzahl und nicht Carmichael, so a Fermat Zeuge mit Wahrscheinlichkeit $\geq 1/2$, $b_k \neq 1$.



Prof. Dr. K. Madlener: Computeralgebra

437

Primzahltesten und Faktorisierung in $\ensuremath{\mathbb{Z}}$

Prof. Dr. K. Madlener: Computeralgebra

Primzahltester

Starker Pseudoprimalitäts-Test (Forts.)

- ▶ Angenommen N ist Primzahl, dann $b_k = 1$, falls
 - ▶ $b_0 = 1 \rightsquigarrow$ "mögliche Primzahl" in 3.
 - ▶ Sonst $b_j \neq 1$ und $b_i^2 \equiv b_{i+1} = 1 \mod N$ in 5.
- ▶ $x^2 1 \in \mathbb{Z}[x]$ hat höchstens 2 Nullstellen, d. h. Quadratwurzeln von 1 mod N sind 1 und -1 d. h. $b_j = N 1$ und somit $g = N \rightsquigarrow$ korrekte Antwort "mögliche Primzahl" in 6.
- Sei N Carmichael Zahl, P Menge der Primteiler von N. Da N quadratfrei ist, gilt $N = \prod_{p \in P} p$. Betrachte $I = \{i : 0 \le i \le k \text{ und } \forall u \in \mathbb{Z}_N^\times \ u^{2^i m} = 1\}$ $k \in I$ nach Definition der Carmichael Zahlen, $i \in I$, so auch $i + 1 \in I$. Für i < k

Starker Pseudoprimalitäts-Test (Forts.)

- ▶ Sei $p \in P$ und $a \in \mathbb{Z}$ nicht Quadrat $\operatorname{mod} p$ mit $a^{(p-1)/2} \not\equiv 1 \operatorname{mod} p$. (existiert nach Lemma 6.18). Da $a^{p-1} \equiv 1 \operatorname{mod} p$, folgt $\operatorname{ord}_p(a) \mid p-1$, aber teilt nicht (p-1)/2, d. h. $\operatorname{ord}_p(a)$ ist gerade. Da m ungerade ist, folgt $a^m \not\equiv 1 \operatorname{mod} p$. Chinesischer Restsatz liefert ein $u \in \mathbb{Z}_N^n$ mit $u^m \not\equiv 1$, d. h. $0 \not\equiv l$. Es gibt somit l < k mit $l \not\in l$ und $l+1 \in l$.
- Sei

$$G = \{ u \in \mathbb{Z}_N^{\times} : u^{2^{\prime} m} = \pm 1 \} \le \mathbb{Z}_N^{\times}$$

Behauptung: $G \neq \mathbb{Z}_N^{\times}$. Es gibt $p \in P$ und $b \in \mathbb{Z}$ mit $\mathsf{GGT}(b,p) = 1$ mit $b^{2^l m} \not\equiv 1 \mod p$, da sonst $l \in I$. Wähle ein solches Paar.

Chinesischer RS liefert:

Es gibt $c \in \mathbb{Z}$ mit $c \equiv b \mod p$ und $c \equiv 1 \mod N/p$. Also $c \mod N \in \mathbb{Z}_N^{\times} \backslash G$, d.h. G hat höchstens $|\mathbb{Z}_N^{\times}|/2 = \varphi(N)/2$ Elemente wählt man a mit $a \in \mathbb{Z}_N^{\times} \backslash G$, so liefert Algorithmus echten Teiler. (S 496)

<ロ > < 個 > < 国 > < 国 > < 国 > < 国 > の

4 D > 4 B > 4 B > 4 B > B 990

Prof. Dr. K. Madlener: Computeralgebra

Primzahltesten und Faktorisierung in Z

Primzahltesten

Andere Primzahltests

▶ Solovay-Srassen (74) basiert auf Legendre Symbol: Für $a, N \in \mathbb{Z}, N$ Primzahl sei

$$(\frac{a}{N}) = \begin{cases} 1 & \text{falls } \mathsf{GGT}(a,N) = 1 \text{ und } a \text{ ist Quadrat } modN \\ -1 & \text{falls } \mathsf{GGT}(a,N) = 1 \text{ und } a \text{ ist nicht Quadrat } modN \\ 0 & \text{falls } \mathsf{GGT}(a,N) \neq 1 \end{cases}$$

- ▶ Gauß quadratisches Reziprozitätsgesetz: $(\frac{a}{N}) = (\frac{N}{a})$ für ungerade Primzahlen, bis auf a, N beide kongruent 3 mod 4.
- ▶ Jacobi Symbol: Für $N = p_1^{e_1}...p_r^{e_r}$ definiere

$$(\frac{a}{N}) = (\frac{a}{p_1})^{e_1} ... (\frac{a}{p_r})^{e_r}$$

"leicht" zu berechnen. Analogie zu EEA (siehe Übung 18.23 vzG,G)

Prof. Dr. K. Madlener: Computeralgebra 440

Andere Primzahltests (Fort.)

▶ Ist N Primzahl, so gilt für alle $a \in \mathbb{Z}$

$$(*) \ (\frac{a}{N}) \equiv a^{(N-1)/2} \ modN$$

- ▶ Solovay-Strassen zeigen: Ist N zusammengesetzt und nicht Primzahlpotenz, so gilt (*) nicht für mindestens die Hälfte der Zahlen $a \in 1...N-1$. Ihr Algorithmus testet (*) für zufällig gewählte a's und kostet $O(logN \cdot M(logN))$ Wortoperationen pro Test.
- ► Spezielle Primzahltests:
 - ▶ Pepin (1877) für Fermat Zahlen $F_n = 2^{2^n} + 1$ F_n ist Primzahl gdw $3^{(F_n-1)/2} \equiv -1 \mod F_n$
 - Lucas-Lehmer Test (1878,1935) für Mersenne-Zahlen $M_n=2^n-1$ 2^n-1 ist Primzahl gdw $l_{n-1}\equiv 0$ $mod M_n$. Hierbei l_i rek. definiert durch $l_1=4$ und $l_i=l_{i-1}^2-2$ $(i\geq 1)$



4□ > 4問 > 4 = > 4 = > 9 Q ○

Prof. Dr. K. Madlener: Computeralgebra

441

Primzahltesten und Faktorisierung in Z

Primzahltesten

Primzahltesten ist in P

Siehe

http://www.cse.iitk.ac.in/news/primality.html

Primzahltesten und Faktorisierung in Z

Primzahlen mit vorgebenen Eigenschaften

Finden von Primzahlen mit vorgebenen Eigenschaften

- z. B. Primzahl mit *n*-Bits, Big-Prime Methoden, Hensel Lifting Methoden.
- ► Primzahlsatz:
 - ▶ Seien $\Pi(x) = |\{p \in \mathbb{N} : p < x, p \text{ Primzahl}\}|, p_n = n \text{ te Primzahl}$
 - ▶ Dann gilt:: $\Pi(x) \approx \frac{x}{\ln x}$ und $p_n \approx n \ln n$
 - d.h. Eine zufällige Zahl nahe bei x ist eine Primzahl mit Wahrscheinlichkeit $1/\ln x$.
 - d. h. wählt man zufällige n-Bit Zahl und testet sie auf Primalität, so erwartet man eine Primzahl zu finden mit etwa $n \ln 2$ Versuchen.
- ▶ Wie findet man Primzahl p mit B (<math>B vorgegeben). Wähle zufälliges Element im Intervall, teste auf Primalität k mal und gebe erste Zahl aus, die diesen Test erfüllt.
- Ist p nicht Primzahl, so ist die Wahrscheinlichkeit Antwort "mögliche Primzahl" $\leq 2^{-k}$. Dies bedeutet aber nicht, dass die Ausgabe Primzahl mit Wahrscheinlichkeit $1-2^{-k}$ ist.

Prof. Dr. K. Madlener: Computeralgebra 443

Primzahltesten und Faktorisierung in Z

Primzahlen mit vorgebenen Eigenschafter

Finden von Primzahlen mit vorgebenen Eigenschaften

- **7.3 Satz** Gegeben B, k, die Ausgabe des Verfahrens ist eine Primzahl mit Wahrscheinlichkeit mindestens $1-2^{-k+1} \ln B$ und benötigt im Mittel $O(k(\log^2 B) M(\log B))$ Wortoperationen.
- ▶ Beachte $|P| = \Pi(2B) \Pi(B) \ge B/2 \ln B$ $(B \ge 6) \leadsto \frac{|P|}{B} \ge \frac{1}{2 \ln B}$
- ▶ Erwarteten Wahlen: 2InB und jede kostet $O(klogB \cdot M(logB))$
 - **7.4 Lemma** Sei $P \subset \mathbb{N}$ Menge von Primzahlen, $a = \min P$. Sei $M \in \mathbb{Z}$ mit $0 \neq |M| \leq C$. Wählt man p zufällig aus P, so gilt

$$\mathsf{Prob}\{M \equiv 0 \bmod p : p \in P\} \le \frac{\log_a C}{|P|}$$

4 D > 4 B > 4 B > 4 B > B 990

(es gibt höchstens $\log_a |M| \le \log_a C$ Primzahlen in P, die M teilen).

Prof. Dr. K. Madlener: Computeralgebra 442 Prof. Dr. K. Madlener: Computeralgebra 444

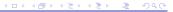
Finden von Primzahlen mit vorgebenen Eigenschaften

7.5 Satz

- ▶ Es gibt einen probabilitischen Algorithmus, der mit Wahrscheinlichkeit mindestens 3/4 eine Primzahl p zwischen B+1 und 2B berechnet für jede Zahl $B \in \mathbb{N}$ mit Wortlänge β .
- ▶ Ist $M \in \mathbb{Z}$, $M \neq 0$ mit $6 \ln |M| \leq B$, so gilt: p Primzahl und $p \nmid M$ mit Wahrscheinlichkeit mindestens 1/2.
- ▶ Der Algorithmus benötigt $O(\beta^2 M(\beta) \log(\beta))$ Wortoperationen.
 - ▶ Wähle $k = 2 + \lceil \log_2 \ln B \rceil$ (dann Satz)
 - ▶ p | M für p prim mit Wahrscheinlichkeit

$$\frac{\log_B C}{|P|} \leq \frac{\ln C \cdot 2 \ln B}{\ln B \cdot B} \leq \frac{1}{3} \quad (B \geq 6) \rightsquigarrow \frac{3}{4}(1 - \frac{1}{3}) = \frac{1}{2}$$

▶ Die ersten r Primzahlen k\u00f6nnen deterministisch in O(rlog²rloglogr) Wortoperationen berechnet werden.



Prof. Dr. K. Madlener: Computeralgebra

445

Primzahltesten und Faktorisierung in $\ensuremath{\mathbb{Z}}$

Primzahlen mit vorgebenen Eigenschaften

Prof. Dr. K. Madlener: Computeralgebra

Kosten und Voraussetzungen für Primzahlen in Algorithmen

			C	ost for
	modular	r prime	prime	algorithm
a	lgorithn	n requirements	finding	
deter-	big	$p>2n^{n/2}A^n$	$n^3 \log^3 A$	$n^4 \log A$
minant	small	$p_i < 2r \ln r$	$n\log^3 A$	$n^4 \log A$
gcd	big	$p > \sqrt{n+1} 2^{n+1} A^2$	$n^3 + \log^3 A$	$n^2 + n \log A$
	small	$p_i < 2r \ln r$	$n \log A$	$n^2 + n \log A$
EEA	small	$p_i < 2r \ln r$	$n \log A$	$n^3 \log A$
root	big	$p>2n(A^2+A)$	log ³ A n	$(2 \log A + n \log^2 A)$
finding	power	$p < 2r \ln r$	$n \log A$	$n^2 \log A$
factor-	big	$p > \sqrt{n+1} 2^{n+1} A^2$	$n^3 + \log^3 A$	$n^{10} + n^8 \log^2 A$
ization	power	$p < 2r \ln r$	$n \log A$	$n^{10} + n^8 \log^2 A$

Primzahltesten und Faktorisierung in Z

Faktorisierung in Z

Faktorisierung in \mathbb{Z} (Übersicht)

Methode	Jahr	Zeit
Trialdivision	$-,\infty$	$O^{\sim}(2^{n/2})$
Pollard's $p-1$ Methode	1974	$O^{\sim}(2^{n/4})$
Pollard's ϱ -Methode	1975	$O^{\sim}(2^{n/4})$
Pollard-Strassen	1976	$O^{\sim}(2^{n/4})$
Morrison/Brillhart	1975	$\exp(O^{\sim}(n^{1/2}))$
Dixon's Random Squares	1981	$\exp(O^{\sim}(n^{1/2}))$
Lenstra's Elliptic Curves	1987	$\exp(O^{\sim}(n^{1/2}))$
Number Field Sieve	1990	$\exp(O^{\sim}(n^{1/3}))$

Prof. Dr. K. Madlener: Computeralgebra

Primzahltesten und Faktorisierung in Z

Faktorisierung in $\ensuremath{\mathbb{Z}}$

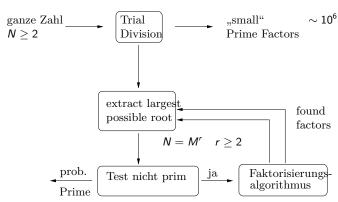
Faktorisierung in \mathbb{Z} (Übersicht) (Forts.)

- ▶ Faktorisierung von Polynome über \mathbb{F}_2 Grad 200000.
- ► Faktorisierung von Zahlen der Länge 150 noch als extrem hart. "Praktische Relevanz von *P*" → wichtig für Cryptosysteme.
- ▶ Tabellen: Cunningham Projekt 1925. Faktorisierungen von $b^n \pm 1$.
- ▶ "Most Wanted" Factorizations.
 - ▶ RSA Cryptosystem: p_1 , p_2 größe Pzahl p_1p_2 . 130-Digits RSA Zahlen geknackt $\sim 2 \times 65$ 96
 - ▶ Faktorisierung Fermat'scher Zahlen $F_y = 2^{29} + 1$ F_{10} , F_{11} ellipt. Kurven, "most wanted 98" $2^{569} - 1/(PF)$ 23 Dig. $F_{12} = 114689 \cdot 26017793 \cdot 63766529 \cdot 190274191361 \cdot 1256132134125569 \cdot C$ (C mit 1187 Digits).
 - http://www.cerias.purdue.edu/homes/ssw/cun/index.htm (siehe Wagstaff).

Prof. Dr. K. Madlener: Computeralgebra 448

Faktorisierung in $\mathbb Z$

Format der Faktorisierungsalgorithmen



siehe Kap. 19, vz. Gathen/Gerhard + Kap.20: Anwendung Public Key Cryptography.

□▶→□▶→重▶→重→□

Prof. Dr. K. Madlener: Computeralgebra

449

Primzahltesten und Faktorisierung in $\ensuremath{\mathbb{Z}}$

Prof. Dr. K. Madlener: Computeralgebra

Faktorisierung in $\ensuremath{\mathbb{Z}}$

Faktorisierungsalgorithmen in \mathbb{Z}

Annahme: N ist keine perfekte Potenz, d.h. $N \neq M^k$ für $M \in \mathbb{Z}$, $k \geq 2$

- ▶ Ganzzahlige Wurzeln berechnen: Gegeben $a, n \in \mathbb{N}$ Entscheide ob a eine n-te Potenz einer Zahl ist und berechne diese gegebenfalls.
- ▶ Gesucht Lösung von $y^n a = 0$:: Verwende hierfür Newtons Iteration (2-adisch) für a, n ungerade in $O(M(\log N))$.
- ▶ Bestimme b, d, e, r mit $N = 2^d 3^e b^r$ GGT(b, 6) = 1 r maximal in $O(\log N \cdot M(\log N))$ Wortoperationen. (Siehe Aufgaben 9.44 und 18.6 in vzG,G).

◆□▶ ◆□▶ ◆■▶ ◆■▶ ● 900

Primzahltesten und Faktorisierung in $\mathbb Z$

Faktorisierung in Z

Trial Division Faktorisierungsalgorithmus

1. Trial Division Fakt Algorithmus

{Eingabe: $N \in \mathbb{N}_{\geq 3}$, weder Prim noch perfekte Potenz, $b \in \mathbb{N}$ } {Ausgabe: kleinster Primfaktor von N falls kleiner b sonst "Failure"}

begin

- 1 **for** $p = 2, 3, \dots, b$ **do**
- 2 if $p \mid n$ then return p
- 3 return "Failure"

end

▶ Um alle p-Faktoren zu finden, dividiere durch p so oft wie möglich dann weiter. Verwende: nächster Primteiler > p. Ist $S_1(N)$ bzw. $S_2(N)$ der grösste bzw. zweitgrösste P-Faktor von N. So $S_2(N) < \sqrt{N}$, d.h $S_2(n)(\log N)^{O(1)}$ Schritte. Für zufällige Zahlen N gilt ϱ

Prob
$$(S_1(N) > N^{0.85}) \approx 0.20$$
 Prob $(S_2(N) > N^{0.30}) \approx 0.20$

 $ightharpoonup O^{\sim}(N^{0.30})$ erwartete Schrittkomplexität für 1.

◆□▶ ◆□▶ ◆□▶ ◆□▶ ◆□ ◆○

Primzahltesten und Faktorisierung in Z

Prof. Dr. K. Madlener: Computeralgebra

Faktorisierung in ${\mathbb Z}$

Pollard und Strassen Methode

Sei $a \mapsto \overline{a}$ die mod N Reduktion und $1 \le c \le \sqrt{N}$. Betrachte $F = (x+1)(x+2) \cdots (x+c) \in \mathbb{Z}[x]$ $f = \overline{F} \in \mathbb{Z}_N[x]$ Dann gilt $\overline{c^2}! = \prod_{0 \le i < c} f(\overline{ic})$. Strategie: "baby step/giant step"::

2. Pollard_ Strassen_Faktorisierung

{Eingabe: $N \in \mathbb{N}_{\geq 3}$, weder Prim noch perfekte Potenz, $b \in \mathbb{N}$ } {Ausgabe: kleinster Primfaktor von N falls < b sonst "Failure"}

begin

- 1 $c \leftarrow \lceil b^{1/2} \rceil$; Berechne Koeffizienten von $f = \prod_{1 \le j \le c} (x + \bar{j}) \in \mathbb{Z}_N[x]$;
- 2 Berechne $g_i \in \{0, \dots, N-1\}$ mit $g_i \mod N = \overline{f(ic)}$ für $0 \le i < c$;
- 3 Falls $\mathsf{GGT}(g_i, N) = 1$ für $0 \le i < c$ then return "Failure" $k \leftarrow \mathsf{Min} \{0 \le i < c : \mathsf{GGT}(g_i, N) > 1)\}$
- 4 return Min $\{kc + 1 \le d \le kc + c : d \mid N\}$ end

Prof. Dr. K. Madlener: Computeralgebra 452

Fakt. Alg. Pollard/Strassen (Forts.)

7.6 Satz

Algorithmus 2. ist korrekt und benötigt $O(M(b^{1/2})M(\log N)(\log b + \log\log N))$ Wortoperationen und Platz für $O(b^{1/2}\log N)$ Wörter. Beweis: Für 0 < i < c gilt:

- ▶ Ein Primteiler p von N teilt F(ic) und somit auch $\mathsf{GGT}(g_i,N) = \mathsf{GGT}(F(ic) \bmod N,N)$ gdw. p teilt Zahl im Intervall $\{ic+1,\cdots,ic+c\}$ \leadsto Korrektheit.
- ► Kosten für 1. und 2. $O(M(c) \log c)$ Add., Mult. in \mathbb{Z}_N Schritt 3 $O(cM(\log N) \log \log N)$ Wortoperationen Schritt 4 $O(cM(\log N))$ Wortoperationen Add., Mult. in \mathbb{Z}_N kostet $O(M(\log N))$.
- ightharpoonup Platz für $O(b^{1/2})$ Zahlen der Größe $O(\log N)$
- ▶ Schleife mit $b = 2^i$, $(i = 1, 2, ..., b > S_2(N))$ liefert vollständige Faktorisierung in $O(M(S_2(N)^{1/2})M(log\ N)logN)$.



Prof. Dr. K. Madlener: Computeralgebra

453

Primzahltesten und Faktorisierung in $\ensuremath{\mathbb{Z}}$

Prof. Dr. K. Madlener: Computeralgebra

Faktorisierung in $\ensuremath{\mathbb{Z}}$

Pollards *o*-Methode (1975)

Idee

Wähle Funktion $f: \mathbb{Z}_N \to \mathbb{Z}_N$ und Startwert $x_0 \in \mathbb{Z}_N$ Setze $x_i = f(x_{i-1})$ für i > 0. Betrachte die Folge (x_i) ::

- ▶ Hoffe nun die Folge x_0, x_1, x_2, \cdots verhält sich wie eine Folge unabhängiger Zufallselemente von \mathbb{Z}_N .
- ▶ Ist p ein unbekannter Primfaktor von N so findet eine Kollision mod p statt, falls es t, l gibt mit l > 0 und $x_t \equiv x_{t+l} \mod p$
- ▶ Ist *N* keine Primzahlpotenz und *q* ein weiterer p-Teiler von *N*, so sind, für unabhängige Reste modulo *N*, *x_i* mod *p* und *x_i* mod *q* ebenfalls unabhängige Zufallsvariablen (Chin. Restsatz).
- ▶ D.h. mit großer Wahrscheinlichkeit $x_t \not\equiv x_{t+1} \mod q$ und somit $GGT(x_{t+1} x_t, N)$ ist nicht trivialer Faktor von N.

Primzahltesten und Faktorisierung in Z

Faktorisierung in Z

Pollards ϱ -Methode (Forts.)

- ► Frage: Wie groß sin t, l?
 Offenbar $t + l \le p$ und der erwartete Wert ist $O(\sqrt{p})$ für eine Zufallsfolge $(x_i)_{i \in \mathbb{N}}$.
- ► Geburtstagproblem: Wieviel Personen benötigt man um eine Wahrscheinlichkeit (zwei personen mit gleichem Geburtstag zu haben) ≥ 1/2 zu erhalten (23 reichen 50,7%)
- Auswahl (mit Wiederholung) aus Urne mit p Marken. Die erwartete Anzahl von Wahlen bis zu einer Kollision is $O(\sqrt{p})$
- ▶ Wie bestimmt man Zykel : Floyd's Trick. Sei $x_0 \in \{0, \dots, p-1\}$ $f: \{0, \dots, p-1\} \rightarrow \{0, \dots, p-1\}$ Betrachte $(x_i)_{i \geq 0}$ mit $x_{i+1} = f(x_i)$.
- ▶ Zykel der Länge l > 0 mit $x_i = x_{i+1}$ für alle $i \ge t$ für $t \in \mathbb{N}$

4 □ > 4

Prof. Dr. K. Madlener: Computeralgebra

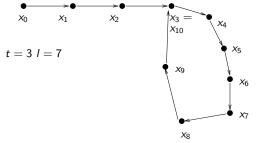
4

Primzahltesten und Faktorisierung in $\ensuremath{\mathbb{Z}}$

Faktorisierung in $\ensuremath{\mathbb{Z}}$

Pollards *Q*-Methode (Forts.)

1. t seien minimal



Speichere Folge bis $x_i = x_j$ O(t + I) zuviel Platz

4 D > 4 B > 4 B > 4 B > B 990

Floyd's 1-step/2-step cycle detection method::

Führe zweite sequenz mit $y_i = x_{2i}$ speichere nur x_i, y_i bis $x_i = y_i = x_{2i}$

 ${\sf FLOYD_Cycle_Det_ALG}$

 $y_0 \leftarrow x_0$; $i \leftarrow 0$;

repeat $i \leftarrow i + 1$; $x_i \leftarrow f(x_{i-1})$; $y_i \leftarrow f(f(y_{i-1}))$ until $x_i = y_i$; return i

Prof. Dr. K. Madlener: Computeralgebra 456

Faktorisierung in ℤ

Floyd's 1-Step/2-Step cycle Detection Method

7.7 Lemma FLOYD_Cycle_Det_ALG hält nach höchstens t + I Iterationen.

Beweis: Da $x_{2i} = y_i$ für alle i gilt:

- ▶ $x_i = y_i$ gdw. $i \ge t$ und $I \mid (2i i) = i$, und der Kleinste Index ist i = t + (-t REM I) < t + I falls t > 0 und i = I falls t = 0.
- ★ Pollard's ϱ -Methode zur Faktorisierung von N: Erzeuge Folge $x_0, x_1, \dots \in \{0, \dots N-1\}$ wie folgt: x_0 wird zufällig gewählt, $x_{i+1} = f(x_i) = x_i^2 + 1$ REM N.
- ▶ Sei p Kleinste Primzahl die N teilt $\rightsquigarrow x_{i+1} \equiv x_i^2 + 1 \mod p$ für $i \geq 0$. Kollision $\mod p$ kann nach $O(\sqrt{p})$ Schritte erwartet werden. Verwende hierfür FLOYD'S-ALG.

<ロ > → □ > → □ > → □ → ○ へ ○

Prof. Dr. K. Madlener: Computeralgebra

Primzahltesten und Faktorisierung in Z

Faktorisierung in $\ensuremath{\mathbb{Z}}$

Pollards ρ-Methode zur Faktorisierung

3. Pollard_ ϱ _Faktorisierung

{Eingabe: $N \in \mathbb{N}_{\geq 3}$, weder Prim noch perfekte Potenz} {Ausgabe: entweder echter Teiler oder "Failure"}

begin

- 1 Wähle $x_0 \in \{0, \dots, N-1\}$ zufällig; $y_0 \leftarrow x_0$; $i \leftarrow 0$;
- 2 repeat
- 3 $i \leftarrow i+1$; $x_i \leftarrow x_{i-1}^2 + 1 \mod N$; $y_i \leftarrow (y_{i-1}^2 + 1)^2 + 1 \mod N$;
- 4 $g \leftarrow \mathsf{GGT}(x_i y_i, N);$
 - if 1 < g < N then return g
- else if g = N then return "Failure"

end

7.8 Satz Ist p der kleinste P-Teiler von $N \rightsquigarrow$ erwartete Laufzeit ersten Teiler zu finden $O(\sqrt{p}M(\log N)\log\log N)$.

Vollständige Faktorisierung $S_2(N)^{1/2}0^{\sim}(\log^2 N)\approx 0 \ (N^{1/4})$

4□ > 4Ē > 4Ē > 4Ē > 4□ >

Primzahltesten und Faktorisierung in ℤ

Faktorisierung in Z

Pollards S-Methode zur Faktorisierung (Forts.)

7.9 Beispiel N = 82123 $x_0 = 631$

i	$x_i \mod N$	$x_i \mod 41$	$y_i \mod N$	$GGT\ (x_i-y_i,N)$
0	631	16	631	N
1	69670	11	28986	1
2	28986	40	13166	1
3	69907	2	40816	1
4	13166	5	20459	1
5	64027	26	6685	1
6	40816	21	75835	1
7	80802	32	17539	41
8	20459	0		
9	71874	1		
10	6685	2		

 $N = 41 \cdot 2003$ $x_{38} \equiv 4430 \equiv x_{143} \mod N$

(ㅁ) (큠) (필) (필) 및 50

Prof. Dr. K. Madlener: Computeralgebra

Faktorisierung in Z

Dixon's Random Square Faktorisierungsmethode

▶ Erstes Verfahren mit Aufwand kleiner als $\exp(\varepsilon \cdot \log N)$ für jedes $\varepsilon > 0$ ldee: Die Gleichungen

$$N = s^{2} - t^{2} = (s+t)(s-t)$$

$$N = a \cdot b = \left(\frac{a+b}{2}\right)^2 - \left(\frac{a-b}{2}\right)^2$$

Beschreiben Bijektion zwischen Faktorisierungen von *N* und Darstellungen von *N* als Differenz zweier Quadrate.

- Naiver Faktorisierungsalgorithmus: Für $t = \lceil \sqrt{N} \rceil, \lceil \sqrt{N} \rceil + 1, \cdots$ Teste ob $t^2 - N$ perfektes Quadrat ist. Findet man solch ein Quadrat so Faktorisierung erfolgreich!
- ▶ Gut Falls N = ab mit |a b| klein, da Laufzeit abhängig von |a b| ist. Fermat kannte dieses Argument: $N = 2027651281 \sqrt{N} \approx 45029$

$$N = 45041^2 - 1020^2 = 46061 \cdot 44021$$

Prof. Dr. K. Madlener: Computeralgebra 458 Prof. Dr. K. Madlener: Computeralgebra

Dixon's Random Square Methode (Forts.)

- ▶ Variante: Wähle $k \ll N$ $t = \lceil \sqrt{kN} \rceil, \lceil \sqrt{kN} \rceil + 1, \cdots$ und teste ob $t^2 kN$ perfektes Quadrat. Falls $t^2 kN = s^2$ so $\mathsf{GGT}(s+t,N)$ ist hoffentlich nichttrivialer Faktor von N, so dass $s \not\equiv \pm t \mod N$
- ▶ Das finden von Relationen der Form $s^2 \equiv t^2 \mod N$ auf dieser Weise ist für große N sehr unwahrscheinlich.

7.10 Beispiel

N=2183 Angenommen wir haben folgende Kongruenzen $453^2\equiv 7 \mod N \quad 1014^2\equiv 3 \mod N \quad 209^2\equiv 21 \mod N$ Dann $(453\cdot 1014\cdot 209)^2\equiv 21^2 \mod N$ oder $687^2\equiv 21^2 \mod N \quad \leadsto \ 37=\mathsf{GGT}(687-21,N) \quad 59=\mathsf{GGT}(687+21,N)$ Dieses ist auch die Faktorisierung von N

◆□▶◆□▶◆■▶◆■▶ ■ 99(

Prof. Dr. K. Madlener: Computeralgebra

461

Primzahltesten und Faktorisierung in $\ensuremath{\mathbb{Z}}$

Faktorisierung in $\ensuremath{\mathbb{Z}}$

Dixon's Random Square Methode (Forts.)

- ► Systematisches Vorgehen:
 - Wähle b zufällig und hoffe, dass b^2 REM N Produkt kleiner Primzahlen ist. Sind genügend solcher gefunden, so erhält man eine Kongruenz $s^2 \equiv t^2 \mod N$. Dann GGT(s-t,N) bzw. GGT(s+t,N)
- Faktorisierungsbasis Primzahlen p_1, \dots, p_h bis zu einer Schranke $B \in \mathbb{R}^+$ Eine Zahl b heißt B-Zahl falls b^2 REM N (Rest der Division von b^2 durch N) Produkt der P-Zahlen p_1, \dots, p_h ist.
- ▶ Im Beispiel sind 453, 1014, 209 B-Zahlen für jedes $B \ge 7$ und N = 2183
- Für eine *B*-Zahl *b* sei $b^2 \equiv p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_h^{\alpha_h} \mod N$ mit $\alpha_1 \cdots \alpha_h \in \mathbb{N}$. Assoziere dazu Binären Exponenten Vektor

$$\varepsilon = (\alpha_1 \bmod 2, \alpha_2 \bmod 2, \cdots, \alpha_h \bmod 2) \in \mathbb{F}_2^h$$

$$lacksquare$$
 Für $B ext{-}\mathsf{Zahl}\ b_i$, sei $b_i^2\equiv\prod_{1\leq i\leq h}p_i^{lpha_{ij}}\ \mathsf{mod}\ N$



Dixon's Random Square Methode (Forts.)

▶ Angenommen man hat b_1, \dots, b_l mit $\varepsilon_1 + \varepsilon_2 + \dots + \varepsilon_l = 0$ in \mathbb{F}_2^h dann

$$\left(\prod_{1\leq i\leq l}b_{j}\right)^{2}=\prod_{1\leq j\leq h}p_{j}^{\sum_{1\leq i\leq l}\alpha_{ij}}=\prod_{1\leq j\leq h}p_{j}^{2\gamma_{j}}=\left(\prod_{1\leq j\leq h}p_{j}^{\gamma_{j}}\right)^{2}\bmod N$$
wobei $\gamma_{j}=\frac{1}{2}\sum_{1\leq i\leq l}\alpha_{ij}$ (durch 2 teilbar nach Voraussetzung)

▶ Dann $s^2 \equiv t^2 \mod N$ mit

$$s = \prod_{1 \leq i \leq l} b_i \qquad t = \prod_{1 \leq j \leq h} p_j^{\gamma_j}$$

! Man benötigt nicht mehr als h+1 B-Zahlen, d.h. $l \le h+1$, da jede Menge von h+1 Vektoren in \mathbb{F}_2^h linear abhängig ist.

Prof. Dr. K. Madlener: Computeralgebra

463

Primzahltesten und Faktorisierung in $\ensuremath{\mathbb{Z}}$

Faktorisierung in 2

Dixon's Random Square Methode (Forts.)

- ! Die Hoffnung ist nun s,t gefunden zu haben mit $s\not\equiv \pm t \bmod N$
- ▶ Ist N keine Primzahlpotenz mit $r \ge 2$ verschiedene Primfaktoren, so folgt aus Chinesischer-Restsatz, dass jedes Quadrat in \mathbb{Z}_N^{\times} genau 2^r Quadratwurzeln in \mathbb{Z}_N hat.
- ▶ Ist somit s eine zufällige Quadratwurzel von t^2 so gilt

Prob
$${s \equiv \pm t \bmod N} = \frac{2}{2^r} \le \frac{1}{2}$$

▶ Im Beispiel mit $B = \{2, 3, 5, 7\}$ gilt $\varepsilon_1 = (0, 0, 0, 1)$ $\varepsilon_2 = (0, 1, 0, 0)$ $\varepsilon_3 = (0, 1, 0, 1)$ $\varepsilon_1 + \varepsilon_2 + \varepsilon_3 = 0$ in \mathbb{F}_2^4 und $\gamma_1 = \gamma_3 = 0$ $\gamma_2 = \gamma_4 = 1$ $s = 453 \cdot 1014 \cdot 209$ $t = 2^0 \cdot 3^1 \cdot 5^0 \cdot 7^1$



Faktorisierung in $\mathbb Z$

Dixon's Random Square Methode (Forts.)

- ▶ Analyse: Sei $L(N) = e^{\sqrt{\ln N \ln \ln N}}$ Dixon's Random Square Methode faktorisiert eine Zahl N mit erwarteten Wert von $O(L(N)^{\sqrt{9/2}})$ Wortoperationen.
- ▶ Zu den Kosten des folgenden Dixon's Random Squares Algorithmus:
 - ▶ Sei $n = \log N$. Kosten für Basis von Primzahlen $O(h \log^2 h \log \log h)$ Wortoperationen.
 - ▶ Teilbarkeitstest $O(h \cdot M(n))$.
 - ▶ Kosten für eine Iteration in Schleife 2 sind $O(M(n) \log n)$, für GGT, O(M(n)) Wortoperationen für b^2 REMN und O((h+n)M(n)) Operationen um B-Zahl Check durchzuführen.
 - ▶ Ist k die Anzahl der Iterationen von 2, so kosten O(k(h+n)M(n)) Wortoperationen.
 - ▶ Lösen des linearen Gleichungssystems: $O(h^3)$
 - \rightarrow $O(h^3 + k(h+n)M(n))$ Wortoperationen.

ロ ト ◆ 個 ト ◆ 恵 ト ◆ 恵 ・ 夕 Q C

Prof. Dr. K. Madlener: Computeralgebra

465

Primzahltesten und Faktorisierung in Z

Faktorisierung in $\ensuremath{\mathbb{Z}}$

Dixon's Random Squares Methode zur Faktorisierung

4. Dixon's_Random_Squares_Faktorisierung

{Eingabe: Ungerade Zahl $N \ge 3$, weder Prim noch perfekte Potenz $B \in \mathbb{R}^+$ } {Ausgabe: entweder echter Teiler oder "Failure"}

1 Berechne alle P-Zahlen $p_1, \dots, p_h \le B$ if $p_i \mid N$ für ein $i \in \{1, \dots, h\}$ then return p_i

2 $A \leftarrow \emptyset$ //Initialisiere Menge der B-Zahlen//

repeat

3 Wähle zufällig $b \in \{2, \dots, N-2\}$

 $g \leftarrow \mathsf{GGT}(b, N)$ if g > 1 then return g

4 $a \leftarrow b^2 \text{ REM } N$ //Faktorisiere a über $\{p_1, \dots p_h\}$ // **for** $i = 1 \dots h$ **do** //Bestimme Vielfachheit von p_i in a//

5 $\alpha_i \leftarrow 0$; while $p_i \mid a$ do $a \leftarrow \frac{a}{p_i}$, $\alpha_i \leftarrow \alpha_i + 1$

6 if a = 1 then $\alpha \leftarrow (\alpha_1, \cdots \alpha_h)$, $A \leftarrow A \cup \{(b, \alpha)\}$

7 until #A = h + 1

<ロ > < 個 > < 量 > < 達 > の < で

Primzahltesten und Faktorisierung in $\mathbb Z$

Faktorisierung in Z

Dixon's Random Squares Methode zur Faktorisierung

- 8 Finde verschiedene Paare $(b_1,\alpha^{(1)}),\cdots,(b_l,\alpha^{(l)})\in A$ mit $\alpha^{(1)}+\cdots+\alpha^{(l)}\equiv 0$ mod 2 in \mathbb{F}_2^h für ein $l\geq 1$ durch lösen eines $(h+1)\times h$ System von linearen Gleichungen in \mathbb{F}_2
- 9 $(\gamma_1, \dots, \gamma_h) \leftarrow \frac{1}{2}(\alpha^{(1)} + \dots + \alpha^{(I)});$ $s \leftarrow \prod_{1 \leq i \leq I} b_i; \ t \leftarrow \prod_{1 \leq i \leq I} p_j^{\gamma_i}; \ g \leftarrow \mathsf{GGT}(s+t, N);$

 $1 \le i \le l$ $1 \le j \le h$ if g < N then return g else return "Failure"

- ! Abschätzung für *k* und geeignete Wahl von *B* siehe vz. Gathen/Gerhard 19.5 (S.527 530)
- ► Fixiere $r \in \mathbb{N}$, $n = \ln N$ $B = N^{1/2r}$, d.h. $\ln B = N/2r$, $h = \Pi(B) > B/\ln (B)$ für $B \ge 59$ nach PZ-Satz
- ▶ Die Erwartete Anzahl k erfüllt dann $\frac{N}{h^{2r}}(2r)! < n^{2r} \rightsquigarrow Schranke$.
- ► Sei $\Psi(x,y) = \{a \in \mathbb{N} : 1 \le a \le x, \forall p \text{ prim } p \mid a \Rightarrow p \le y\}$ die Menge der y glatten Zahlen. b ist B-Zahl gdw. b^2REM $N \in \Psi(N,B)$.

Prof. Dr. K. Madlener: Computeralgebra

...

Primzahltesten und Faktorisierung in $\ensuremath{\mathbb{Z}}$

Faktorisierung in Z

Pollard's p-1 Methode

- Als Einführung in der Methode der elliptischen Kurven. Annahme N hat Primfaktor p, mit p-1 B-Zahl, d.h. Primpotenzen $I^e \mid p-1$ erfüllen $I^e < B$, (sie sind also B zahm für geeignet gewähltes B).
 - 5. Pollard's p 1 Faktorisierung

{Eingabe: $N \ge 3$, B > 0}

{Ausgabe: entweder echter Teiler von *N* oder "Failure"}

- $1 k \leftarrow \mathsf{KGV}\{i : 2 < i < B\}$
- 2 Wähle $a \in \{2, \dots N-2\}$ zufällig
- $3 b \leftarrow a^k \mod N; d \leftarrow \mathsf{GGT}(b-1, N);$
- 4 if 1 < d < N then return d else return "Failure"
- ▶ Hoffnung d ist nichttrivialer Teiler von N. Die Annahme garantiert $a^k \equiv 1 \mod p$, da $p-1 \mid k$, d.h. d>1. Um d < N zu garantieren, genügt es, wenn N einen weiteren Primfaktor q hat, mit $a^k \not\equiv 1 \mod q$
- ▶ Grundlage ist die Gruppe $G = \mathbb{Z}_N^{\times}$. Hoffnung ist $|G \mod p|$ ist B-Zahl.

Prof. Dr. K. Madlener: Computeralgebra 466 Prof. Dr. K. Madlener: Computeralgebra

Lentra's Elliptische-Kurven Methode (1987)

- ▶ Statt $G = \mathbb{Z}_N^{\times}$, werden Gruppen von Elliptischen-Kurven und deren Ordnung als Testkandidat B-Zahl zu sein, verwendet. Algebraische Geometrie
- ▶ Sei F Körper mit char $\neq 2,3$ und $x^3 + ax + b \in F[x]$ quadratfrei. Dann ist

$$E = \{(u, v) \in F^2 : v^2 = u^3 + au + b\} \cup \{O\} \subseteq F^2 \cup \{O\}$$

eine elliptische Kurve über F. O Punkt im Unendlichen von E.

- $> x^3 x = x(x-1)(x+1), x^3 x + b$ definieren elliptische Kurven.
- ► Gruppenstruktur: E wird zu einer abelschen Gruppe mit + wie folgt:: Ist $P = (u, v) \in E$, so sei der Spiegelpunkt -P = (u, -v) wobei -O = O Sind $P, Q \in E$, so schneidet die Gerade durch P, Q E in Punkt S. Dann R = P + Q = -S. Spezialfälle P = Q (Tangente), P + O = -(-P) = P und P + (-P) = -O = O.



Prof. Dr. K. Madlener: Computeralgebra

469

Primzahltesten und Faktorisierung in $\ensuremath{\mathbb{Z}}$

Faktorisierung in $\ensuremath{\mathbb{Z}}$

Lentra's Elliptische-Kurven Methode (1987)

- ▶ Die Größe elliptischer Kurven über \mathbb{F}_q ::
- ▶ Sei E elliptische Kurve über \mathbb{F}_q , char > 3, dann gilt $|E| \le 2q + 1$.
- ▶ Hasse's Schranke:: $||E| (q+1)| \le 2\sqrt{q}$
- ▶ Sei $y^2 = x^3 + x$ mit q = 7. E enthält (0,0), (1,0), (4,2), (4,5), (5,1), (5,6), (6,0), O. Die Gruppe wird von (4,2) mit Ordnung 4 und (0,0) mit Ordnung 2 erzeugt, d.h. isomorph zu $\mathbb{Z}_4 \times \mathbb{Z}_2$.
- ightharpoonup | $E \mid P = O$. Die Ordnung eines Elements ist wie üblich definiert.
- ▶ Rationale Funktionen zur Berechnung der Summe $(x_1, y_1) + (x_2, y_2)$

$$x_1 \neq x_2 \rightsquigarrow x_3 = (\frac{y_2 - y_1}{x_2 - x_1})^2 - x_1 - x_2, y_3 = -y_1 + \frac{y_2 - y_1}{x_2 - x_1} \cdot (x_1 - x_3)$$

$$(x_1, y_1) = (x_2, y_2) \rightsquigarrow x_3 = (\frac{3x_1^2 + a}{2y_1})^2 - 2x_1, y_3 = -y_1 + \frac{3x_1^2 + a}{2y_1} \cdot (x_1 - x_3)$$

◆ロト ◆団ト ◆量ト ◆量ト = りなで

Primzahltesten und Faktorisierung in Z

Faktorisierung in Z

Lenstra's Elliptische-Kurven Faktorisierungsalgorithmus

```
5. Lenstra's_Elliptic_Curve_Faktorisierung
   {Eingabe: Ungerade Zahl N, 3 \nmid N, keine perfekte Potenz, B \in \mathbb{R}^+, min_{p|N} < C}
                               {Ausgabe: entweder echter Teiler oder "Failure"}
  1 Wähle zufällig (a, u, v) \in \{0, ..., N-1\}^3;
     b \leftarrow v^2 - u^3 - au; g \leftarrow GGT(4a^3 + 27b^2, N);
     if 1 < g < N then return g else if g = N return "Failure";
  2 //Sei E die "elliptische Kurve" über \mathbb{Z}_N mit Koeffizienten a, b//
      Berechne die Primzahlen p_1 = 2 < ... < p_h < B:
     P \leftarrow (u, v); Q \leftarrow P; t \leftarrow 1;
  3 for 1 < j < h do
          e_i \leftarrow \lfloor \log_{p_i}(C * 2\sqrt{C} + 1) \mid;
          for 0 \le j < e_i do //Invarianten: t = p_i^j \prod_{1 \le r \le i} p_r^{e_r} und Q = tP//
  4 Versuche p_iQ in E über \mathbb{Z}_N zu berechnen;
          if ein Zähler w Nullteiler in \mathbb{Z}_N then return GGT(w, N)
                                                  else Q \leftarrow p_i Q: t \leftarrow p_i t:
  5 return "Failure"
```

Prof. Dr. K. Madlener: Computeralgebra

I > 《집 > 《호 > 《호 > · 호 · 》

Primzahltesten und Faktorisierung in Z

Faktorisierung in ${\mathbb Z}$

Analyse von Lenstra's Faktorisierungsalgorithmus

- ▶ Benötigt wird: *E* mod *p* ist elliptische Kurve für jeden Primteiler *p* | *N*. *E* ist i.A. keine Gruppe mit + definiert durch die Gleichungen, d.h die rationalen Ausdrücke müssen nicht mod *N* wohldefiniert sein.
- ▶ Sei $p \mid N$ prim, dann $p \nmid 4a^3 + 27b^2$, da sonst Ausgang in 1. Sei E_p die Reduktion von E mod p, d.h. die elliptische Kurve über \mathbb{Z}_p mit Koeffizienten a, b mod p. Zu $P \in E$ sei P_p der korrespondierende Punkt P mod p. O_p korrespondiert zu O und für alle $P \in E \setminus \{O\}$ gilt $P_p \neq O_p$.
- ▶ Bis der Teiler p in Schritt 4 gefunden wird $(p \mid GGT(w, N))$, implementieren die Berechnungen die Arithmetik von E_p in folgenden Sinn:: Ein partielles Ergebnis Q = tP in E liefert (mod p) das partielle Ergebnis $Q_p = tP_p$ in E_p , d.h. $tP_p = (tP)_p$.
- ▶ Eine Faktorisierung wird gefunden wenn für zwei Primteiler p, q von N ein Vielfaches der Ordnung von P_p in E_p erreicht wird, der nicht Vielfaches der Ordnung von P_a in E_a ist.



Analyse von Lenstra's Faktorisierungsalgorithmus

7.11 Lemma Angenommen (E,P) ist gewählt, $p,q \mid N$ verschieden, N sei der größte Primfaktor der Ordnung von P_p in E_p , $p \leq C$, $|E_p|$ sei B-glatt und $I \nmid |E_q|$. Dann wird N vom Algorithmus faktorisiert. **Beweis**: Sei $k = \prod_{1 \leq r \leq p} p_r^{e_r}$, e_r wie in 3.

- ▶ Da | E_p | B-glatt ist und $p \le C$, folgt aus der Hasse Schranke:: | E_p || k.
- ▶ Sei d die Ordnung von P_p in E_p . Dann $d \mid\mid E_p \mid$ und somit $l \leq B$ und $d \mid k$.
- Sei $p_i = l$ und e der Exponent von l in d, d.h. $1 \le e \le e_i$. Ist j = e 1 so $t = l^{e-1} \prod_{1 \le r < i} p_r^{e_r}$ und Q = tP vor Schritt 4. $t \not\equiv 0 \mod d$ und $lt \equiv 0 \mod d$

Somit $Q_p = tP_p \neq O_p$ und $IQ_p = ItP_p = O_p$. Wir zeigen, der Algorithmus kommt **nicht** bis zu dieser Stelle. Angenommen IQ = O, dann auch $IQ_q = (ItP)_q = O$. Da aber $I \nmid |E_q|$ muss bereits $Q_q = tP_q = O_q$ gelten und somit Q = O. Aber dann $Q_p = O_p \not = O$



Prof. Dr. K. Madlener: Computeralgebra

473

Primzahltesten und Faktorisierung in $\ensuremath{\mathbb{Z}}$

Prof. Dr. K. Madlener: Computeralgebra

Faktorisierung in $\ensuremath{\mathbb{Z}}$

Analyse von Lenstra's Faktorisierungsalgorithmus

7.12 Satz (Lenstra) Sei p Primzahl, $S\subseteq (p+1-\sqrt{p},p+1+\sqrt{p})\subset \mathbb{N}$ und seien $a,b\in \mathbb{F}_p$ zufällig gewählt. Sei

$$E_p = \{(u, v) : v^2 = u^3 + au + b\} \cup \{O\}$$

eine elliptische Kurve über \mathbb{F}_p . Dann gibt es eine Konstante $c \in \mathbb{R}^+$ mit

$$prob\{\mid E_p\mid\in S\}\geq \frac{c\mid S\mid}{\sqrt{p}\log p}$$

7.13 Folgerung Sei $p \leq C$ ein Primteiler von N und $\sigma = \mid \{B - \text{glatte Zahlen in } (p+1-\sqrt{p},p+1+\sqrt{p})\} \mid$. Dann erfüllt die Anzahl M der Tripel $(a,u,v) \in \{0,...,N-1\}^3$ für die der Algorithmus N faktorisiert

$$rac{ extit{M}}{ extstyle extstyle N^3} \geq rac{c_1 \sigma}{\sqrt{p} \log p} ext{ für ein } c_1 \in \mathbb{R}^+$$

4□ > 4ⓓ > 4ಠ > 4ಠ > 戛 900

Primzahltesten und Faktorisierung in Z

Faktorisierung in Z

Laufzeitanalyse von Lenstra's Faktorisierungsalgorithmus

- ▶ Die Laufzeit h\u00e4ngt wesentlich ab von der Anzahl der Auswahlen die der Algorithmus ben\u00f6tigt um mit gro\u00dfer Wahrscheinlichkeit erfolgreich zu faktorisieren (Siehe Seite 540 vzG,G).
- ▶ **Vermutung**: Für $x, u \in \mathbb{R}^+$ und $d \in \mathbb{Z}$ zufällig gewählt aus Intervall $(x \sqrt{x}, x + \sqrt{x})$ gilt

$$prob\{d \text{ ist } x^{\frac{1}{u}} \text{ glatt}\} = u^{-u(1+o(1))}$$

 Unter der Annahme der Vermutung, kann man eine erwartete Laufzeitschranke von

$$L(p)^{\sqrt{2}+o(1)}$$
 wobei $L(p)=e^{\sqrt{InpInInp}}$

zeigen. Praxis: Wähle C "klein" und bestimme $B = e^{\sqrt{(InClnlnC)/2}}$. Verdopple C falls nicht erfolgreich.

4ロト 4部ト 4 差ト 4 差ト 達 めり

Prof. Dr. K. Madlener: Computeralgebra

Primzahltesten und Faktorisierung in $\mathbb Z$

Anwendung: Cryptosystem

Moderne Anwendung: Public Key Cryptography

Cryptosysteme

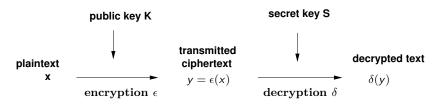
▶ Szenario:: Bob will Nachricht an Alice senden, so dass ein Lauscher (Eve) die Nachricht nicht verstehen kann. Dies wird durch eine Chiffrierung der Nachricht erreicht, so dass nur Alice, mit den richtigen Schlüssel, die Nachricht leicht entschlüsseln kann aber Eve ohne den richtigen Schlüssel die Nachricht nicht verstehen kann.



▶ Klassische Chiffrierungen: Caesar Chiffrierung: Permutationen vom Alphabet mit 26 Buchstaben oder One -Time Pad: Um eine Nachricht der Länge n zu Verschlüsseln wird ein zufälliges Wort gleicher Länge buchstabenweise mod26 aufaddiert. Symmetrisch.

Public Key Cryptosysteme: Diffie & Hellman

▶ Idee: Zwei verschiedene Schlüssel *K* für die Verschlüsselung und *S* für die Entschlüsselung, beide "einfach" aber Entschlüsselung ohne *S* "hart".



K kann öffentlich bekannt sein. Da $x=\delta(y)=\delta(\epsilon(x))$ ist δ Inverse von ϵ . Funktionen die leicht zu berechnen sind aber eine harte Inverse besitzen heißen **trapdoor Funktionen**. K ist der öffentliche Schlüssel und S der geheime Schlüssel. Hier genügen n öffentliche-geheime Schlüsselpaare um sichere Kommunikation zwischen je zwei Partnern unter n zu realisieren.

(미) (중) (본) (본) (본)

Primzahltesten und Faktorisierung in $\mathbb Z$

Anwendung: Cryptosysteme

Prof. Dr. K. Madlener: Computeralgebra

Public Key Cryptosysteme: Anforderungen

- ▶ Ein Cryptosystem gilt als geknackt, wenn es ein Boolsches Prädikat *B*(*x*) gibt z.B. die Parität von *x* falls *x* eine Zahl ist und ein pol. Zeit probabilistischer Algorithmus der *B*(*x*) mit einer etwas besseren Wahrscheinlichkeit als Zufallsraten berechnet.

 Andernfalls gilt das System als semantisch sicher.
- ▶ Was bedeutet ein Cryptosystem ist "hart"? Möglichkeiten::
 - ▶ Der Erfinder (oder niemand) kennt keinen pol. Zeit Algorithmus.
 - ► Falls das System geknackt wird, so wird möglicherweise ein bekanntes "harte" Problem gelöst werden.
 - ► Falls das System geknackt wird, so ist ein bekanntes "harte" Problem gelöst.
 - Falls das System geknackt wird, so ist eine pol. Zeit Lösung für ein NP-vollständiges Problem gefunden.
 - Es gibt nachweisbar keinen (prob.) pol. Zeit Algorithmus wie oben.

Das RSA Cryptosystem

Rivest, Shamir & Adleman (1978) basiert auf die vermeintliche "Härte" der Faktorisierung in \mathbb{Z} . Das Prinzip ist einfach: Alice wählt zufällig zwei große Primzahlen (etwa 150 stellig) $p \neq q$ und setzt N = pq. Jeder der N faktorisieren kann, kann auch das System knacken.

Nachrichten werden als Folgen von Elementen aus $\mathbb{Z}_N=\{0,1,...,N-1\}$ kodiert. Verwendet man als Alphabet $\Sigma=\{A,...,Z\}$ mit $\mid \Sigma\mid=26$, so kann man Nachrichten mit bis zu $212=\lfloor log_{26}10^{300}\rfloor$ Buchstaben mit einer Zahl darstellen, durch Verwendung der 26-adischen Darstellung. Etwa "CAESAR" wird dargestellt durch

$$2 \cdot 26^{0} + 0 \cdot 26^{1} + 4 \cdot 26^{2} + 18 \cdot 26^{3} + 0 \cdot 26^{4} + 17 \cdot 26^{5} = 200233302466 \in \mathbb{Z}_{N}$$

Will Alice Nachrichten von Bob erhalten, so wählt sie zufällig $e \in \{2,...,\varphi(N)-2\}$ mit $GGT(e,\varphi(N))=1, (\varphi(N)=(p-1)(q-1)).$ Dann berechnet sie $d \in \{2,...,\varphi(N)-2\}$ mit $de \equiv 1 \bmod \varphi(N).$ K=(N,e) ist ihr öffentlicher Schlüssel und S=(N,d) ihr geheimer Schlüssel.

Prof. Dr. K. Madlener: Computeralgebra

Primzahltesten und Faktorisierung in $\ensuremath{\mathbb{Z}}$

Anwendung: Cryptosystem

477

Das RSA Cryptosystem (Fort.)

- ▶ Die Ver- und Entschlüsselungsfunktion $\epsilon, \delta : \mathbb{Z}^{\times} \to \mathbb{Z}^{\times}$ werden durch $\epsilon(x) = x^{\epsilon}$ bzw. $\delta(y) = y^{d}$ definiert.
- ▶ Bob schickt eine Nachricht x an Alice in dem er $y = \epsilon(x)$ berechnet und versendet. Alice berechnet dann $\delta(y)$ mit ihren geheimen Schlüssel.
- ▶ Ist $u \in \mathbb{Z}$ mit $de 1 = u\varphi(N)$, so gilt

$$(\delta \circ \epsilon)(x) = x^{de} = x^{1+u\varphi(N)} = x(x^{\varphi(N)})^u \equiv x \mod N$$

(Man sollte darauf achten, dass GGT(x, N) = 1, da sonst eine Faktorisierung von N ermöglicht wird. Dies ist aber sehr unwahrscheinlich)

► RSA kann auch zur Authentifizierung verwendet werden, d.h der Sender muss nachweisen, dass es seine Nachricht ist. → Digitale Signaturen.



Prof. Dr. K. Madlener: Computeralgebra 480 Prof. Dr. K. Madlener: Computeralgebra 480

Das RSA Cryptosystem (Fort.)

▶ Bob versendet entweder $y = \delta_B(x)$ oder $y = \epsilon_A(\delta_B(x))$, da Alice ϵ_B kennt kann sie diese Nachrichten entschlüsseln.

7.14 Satz Folgende Probleme sind polynom-Zeit äquivalent:

- N zu faktorisieren
- $\triangleright \varphi(N)$ zu berechnen
- ▶ Berechnung von $d \in \mathbb{N}$ mit $de \equiv 1 \mod \varphi(N)$ aus K = (N, e)



Prof. Dr. K. Madlener: Computeralgebra

Prof. Dr. K. Madlener: Computeralgebra

Primzahltesten und Faktorisierung in Z

Anwendung: Cryptosysteme

Das Diffie-Hellman Schlüsselaustauschprotokoll (1976)

- ► Zweck: Protokoll zum Austausch von Schlüsseln zum Versenden von Nachrichten mit einem symmetrischen Cryptosystem.
- ▶ sei $q \in \mathbb{N}$ eine große Primzahlpotenz (etwa 1000 bits) und g ein Erzeuger (Generator) von \mathbb{F}_q^{\times} . Dann ist \mathbb{F}_q^{\times} isomorph zur additiven (zyklischen) Gruppe \mathbb{Z}_{q-1} via $g^i \longleftrightarrow i$.
- ► Das Protokoll arbeitet wie folgt:
 - ightharpoonup Alice und Bob einigen sich auf q und g die öffentlich sein können.
 - lacktriangle Alice wählt für sich $a\in\mathbb{Z}_{q-1}$, berechnet und sendet $u=g^a\in\mathbb{F}_q^ imes$ an Bob.
 - ▶ Bob wählt für sich $b \in \mathbb{Z}_{q-1}$, berechnet und sendet $v = g^b \in \mathbb{F}_q^{\times}$ an Alice.
 - ▶ Alice und Bob berechnen $v^a = g^{ab} = u^b$ und benützen dies als gemeinsamen Schlüssel.

Primzahltesten und Faktorisierung in Z

Anwendung: Cryptosysteme

Das Diffie-Hellman Schlüsselaustauschprotokoll: Probleme

▶ Problem 1: Diffie-Hellman Problem:: DH

Gegeben
$$g^a, g^b \in \mathbb{F}_q^{\times}$$
, berechne g^{ab} .

▶ Problem 2: Diskreter Logarithmus Problem:: DL

Gegeben
$$g^a \in \mathbb{F}_q^{\times}$$
, berechne a .

- Es wird vermutet, dass DH ein hartes Problem ist. Die bisher schnellsten Algorithmen haben Laufzeiten wie die Faktorisierung in \mathbb{Z} . Scheint nicht NP-vollständig zu sein. Ein Lauscher der q,g,u,v kennt muss DH lösen um g^{ab} (den Schlüssel) zu berechnen. Dies ist pol-reduzibel auf DL (die Umkehrung ist nicht bekannt).
- ▶ Die beste Schranke für die Berechnung von DL in \mathbb{F}_q^{\times} ist $exp(O((nlog^2n)^{1/3}))$ Wortoperationen mit $n \approx log_2q$.

(□) (部) (息) (息) (息) (2) (2) (483

Prof. Dr. K. Madlener: Computeralgebra

Primzahltesten und Faktorisierung in $\ensuremath{\mathbb{Z}}$

Anwendung: Cryptosysteme

Das ElGamal Cryptosystem

- $lackbox{ Wie gehabt } \mathbb{F}_q^{ imes} \ ext{groß und } g \ ext{Generator}.$
- ▶ Um Nachrichten von Bob zu erhalten wählt Alice zufällig $S=b\in\mathbb{Z}_{q-1}$ als ihr geheimer Schlüssel und gibt $K=(q,g,g^b)$ als ihr öffentlicher Schlüssel bekannt.
- ▶ Will Bob eine Nachricht x an Alice senden, wählt er zufällig $k \in \mathbb{Z}_{q-1}$, berechnet g^k und xg^{kb} und sendet $y = (u, v) = (g^k, xg^{kb})$ an Alice.
- ▶ Alice berechnet $x = v/u^b$
- Die Berechnung von x aus y ohne Kenntnis von S ist pol Zeit äquivalent zu DH.





Rabin's Cryptosystem

- ▶ Basiert auf die Härte der Berechnung von Quadratwurzeln modulo *N* = *pq*, hierbei sind *p*, *q* Primzahlen wie bei RSA.
- ▶ Die Faktorisierung von N kann auf die Berechnung von Quadratwurzeln wie folgt reduziert werden: Wähle $x \in \mathbb{Z}_N$ und berechne $y = \sqrt{x^2}$. Dann $x^2 \equiv y^2 \mod N$ oder $pq = N \mid (x+y)(x-y)$. Ist $x \not\equiv \pm y \mod N$, so erhält man eine Faktorisierung von N (Siehe Dixons's random square A).
- ▶ Um eine Nachricht x an Alice zu senden, verwendet Bob ihr öffentlicher Schlüssel N und sendet x² mod N. Alice berechnet die zwei Quadratwurzeln von y mod p und q mit EDF und kombiniert sie mit CRA. Es gibt Tricks um mit den 4 Möglichkeiten umzugehen.
- System ist anfällig für aktive Attacken: Falls Eve ein zufälliges x wählt und Alice dazu bringt eine Nachricht $y \equiv x^2 \mod N$ zu signieren, in dem sie die Quadratwurzel z von $y \mod N$ zurückgibt, so ist mit Wahrscheinlichkeit 1/2, GGT(x-z,N) ein echter Teiler von N.

◆ロト ◆個ト ◆重ト ◆重ト 重 めので

Prof. Dr. K. Madlener: Computeralgebra

485

Primzahltesten und Faktorisierung in \mathbb{Z}

Prof. Dr. K. Madlener: Computeralgebra

Anwendung: Cryptosysteme

Cryptosystem die auf elliptische Kurven basieren

- ▶ Die Idee ist analog zum ElGamal System, man verwendet an Stelle von \mathbb{F}_q^{\times} die additive Gruppe einer elliptischen Kurve E über \mathbb{F}_q .
- ▶ Die Kurve E und ein Punkt $P \in E$ sind öffentlich. Sei n die Ordnung von P in E.
- ▶ Alice wählt zufällig $a \in \{2, ..., n-2\}$ und gibt aP bekannt.
- ▶ Wenn Bob eine Nachricht $(m_1, m_2) \in \mathbb{F}_q^2$ an Alice senden, so wählt er ein zufälliges k, berechnet $kP = (r_1, r_2)$ und $k \cdot aP = (s_1, s_2)$ und sendet $(y_1, y_2, y_3, y_4) = (r_1, r_2, s_1m_1, s_2m_2) \in \mathbb{F}_q^4$ an Alice.
- ▶ Alice berechnet $a \cdot (r_1, r_2) = a \cdot kP = (s_1, s_2)$ und berechnet die Nachricht als $(m_1, m_2) = (s_1^{-1} y_3, s_2^{-1} y_4)$.
- lacktriangle Kommerzielle Implementierung auf Chip mit $\mathbb{F}_{2^{155}}$.

Inhalt Kapitel 8

Polynomideale und Gröbner-Basen

Motivation: Buchberger's Algorithmus

Reduktionstechniken zur Lösung des Wortproblems

Termordnungen und multivariate Division mit Rest: Reduktion in

 $K[x_1, \dots, x_n]$ Polynom-Reduktion

Längen von Reduktionsketten

Term-Ideale und Hilbert's Basissatz

Gröbnerbasen und S-Polynome

Buchberger's Algorithmus

Anwendungen von Gröbner Basen

Eliminationseigenschaften - Polynomgleichungen (Syzygien)

Zur Komplexität der Berechnung von G-Basen

4 11 1 4 4 12 1 4 12 1 1 2 1 9 0

Prof. Dr. K. Madlener: Computeralgebra

Polynomideale und Gröbner-Basen

Motivation: Buchberger's Algorithmus

Motivation: Buchberger's Algorithmus

Polynomringe in mehreren Variablen. Werkzeug zur Modellierung vieler Probleme aus der (Alg)-Geometrie, Robotik (eukl. Geometrie), Invarianten, Singularitäten.

Polynomideale: $\{p_1, \ldots, p_k\} \subseteq R[x_1, \ldots, x_n] = \mathfrak{R}, \quad \langle p_1, \ldots, p_k \rangle_{\mathfrak{R}}$ Ideal. Wortproblem: $p \in \mathfrak{R}$ gilt $p \in \langle p_1, \ldots, p_k \rangle$.

Gegebenenfalls berechne Darstellung

$$p = \sum_{i=1}^k f_i p_i \text{ mit } f_i \in \mathfrak{R}$$

Division durch $\langle p_1, \dots, p_k \rangle$ mit Rest: $p = \sum_{i=1}^k f_i p_i + \text{Rest.}$

- ▶ Polynomideale als Kern von Homomorphismen
- ► Rechnen im Quotientenring



Polynomideale und Gröbner-Basen Motivation: Buchberger's Algorithmus

Buchberger's Algorithmus

- ightharpoonup z. B. in $\mathbb{Q}[x,y,z]$
 - $p_1 = x^3 vz xz^2$
 - $p_2 = xv^2z xvz$
 - $p_3 = x^2 v^2 z^2$

Frage: Liegt $q = x^2yz - z^3 \in \langle p_1, p_2, p_3 \rangle$.

Gegebenenfalls finde $f_1, f_2, f_3 \in \mathbb{Q}[x, y, z]$ mit $q = \sum_{i=1}^{3} f_i p_i$.

► Spezialfall vom *O*-Äquivalenzproblem:

 $q \approx 0 \mod \langle p_1, p_2, p_3 \rangle$ \approx Kongruenz, die von $i = \langle p_1, p_2, p_3 \rangle$ induziert wird.

 $ightharpoonup q_1 \approx_{\mathfrak{i}} q_2 \text{ gdw } q_1 - q_2 \in \mathfrak{i}$



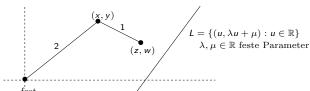
Prof. Dr. K. Madlener: Computeralgebra

Polynomideale und Gröhner-Rasen

Motivation: Buchberger's Algorithmus

Beispiele

8.1 Beispiel Einfacher Roboter



- ▶ Die möglichen Positionen $(x, y, z, w) \in \mathbb{R}^4$ des Roboters sind charakterisiert durch
 - $x^2 + v^2 = 4$

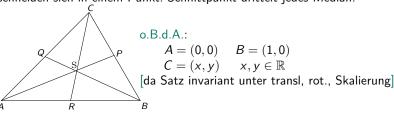
$$(z-x)^2 + (w-v)^2 = 1$$

4□ > 4個 > 4 = > 4 = > = 90

- ► Frage: Kann der Roboter die Linie erreichen?
 - (*) muss erfüllt sein und $w = \lambda z + \mu$.

Beispiele

8.2 Beispiel Euklidische Geometrie:: Die drei Medianen eines Δ schneiden sich in einem Punkt. Schnittpunkt drittelt jedes Median



- ▶ d.h. P = ((x+1)/2, y/2), Q = (x/2, y/2), R = (1/2, 0).Sei S = (u, v) Schnittpunkt \overline{AP} und \overline{BQ} .
- ▶ S liegt auf AP ist äquivalent zur Aussage \overline{AS} und \overline{AP} haben gleiche Steigung, d.h.

$$\frac{u}{v} = \frac{x+1}{v}$$

$$f_1 := uy - v(x+1) = 0$$

ightharpoonup S liegt auf \overline{BQ} analog

$$f_2 := (u-1)y - v(x-2) = 0$$

Prof. Dr. K. Madlener: Computeralgebra

Polynomideale und Gröhner-Rasen

Motivation: Buchberger's Algorithmus

Beispiele (Forts.)

 \triangleright S liegt auch auf \overline{CR} :

$$g_1 := -2(u-x)y - (v-y)(1-2x)$$

= $-2uy - (v-y) + 2vx = 0$

► S drittelt die drei Medianen: d. h.

$$(u, v) = \overline{AS} = 2\overline{SP} = (x + 1 - 2u, y - 2v)$$

$$(u-1,v) = \overline{BS} = 2\overline{SQ} = (x-2u, y-2v)$$

•
$$(u-x, v-y) = \overline{CS} = 2\overline{SR} = (2u-1, 2v)$$

oder äquivalent dazu:

- $partial g_2 := 3u x 1 = 0$
- $parbox{0.5}{\ }g_3 := 3v v = 0$
- ▶ Offenbar gilt $g_1 = -f_1 f_2$, d. h. $g_1 = 0$ folgt aus $f_1 = f_2 = 0$, d. h. die drei Medianen schneiden sich in S.

Erinnerung: Grundlagen

▶ Sei F Körper $R = F[x_1, ..., x_n]$ Polynomring in n-Variablen. $f_1, ..., f_s \in R$. Die Polynome $f_1, ..., f_s$ erzeugen ideal I.

$$I = \langle f_1, \dots, f_s \rangle = \left\{ \sum_{1 \le i \le s} q_i f_i : q_i \in R \right\}$$

 $V(I) := \{u \in F^n : f(u) = 0 \text{ für alle } f \in I\}$ Die Varietät $= \{u \in F^n : f_1(u) = \dots = f_s(u) = 0\}$ von I.

- ▶ Schreibe $V(f_1, ..., f_s)$ statt $V(\langle f_1, ..., f_s \rangle)$.
- ► Fragen über Varietäten bzw. *I*:
 - ▶ Ist $V(I) \neq \emptyset$?
 - ▶ Ist *V*(*I*) endlich?
 - ▶ Wortproblem: $f \in R$ gilt $f \in I$.
 - ▶ Trivialitätsproblem: Gilt I = R?



Prof. Dr. K. Madlener: Computeralgebra

49

Polynomideale und Gröbner-Basen

Motivation: Buchberger's Algorithmus

Beispiele (Forts.)

Fortsetzung Beispiel

$$g_1 = -f_1 - f_2 \in \langle f_1, f_2 \rangle \subseteq \mathbb{R}[u, v, x, y]$$

d.h. $V(f_1, f_2) \subseteq V(g_1)$

▶ S drittelt Medianen: d.h. AS = 2SP BS = 2SQ CS = 2SR oder

$$\begin{pmatrix} u-(x+1-2u)\\v-(y-2v)\end{pmatrix}=\begin{pmatrix} u-1-(x-2u)\\v-(y-2v)\end{pmatrix}=$$

$$\begin{pmatrix} u - x - (1 - 2u) \\ v - y + 2v \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

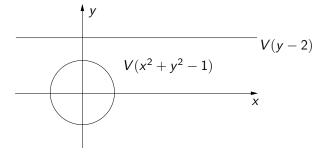
d.h.
$$g_2 = 3u - x - 1 = 0$$
 und $g_3 = 3v - y = 0$

▶ Zeige also $g_2, g_3 \in \langle f_1, f_2 \rangle$ oder $V(f_1, f_2) \subseteq V(g_2)$ und $V(g_3)$.

Beispiele (Forts.)

8.3 Beispiel

i) Sei
$$f_1 = x^2 + y^2 - 1$$
 $f_2 = y - 2 \in \mathbb{R}[x, y]$ $I = \langle f_1, f_2 \rangle$. Dann $V(I) = \{(u, v) \in \mathbb{R}^2 : u^2 + v^2 - 1 = v - 2 = 0\}$ $= \{(u, 2) \in \mathbb{R}^2 : u^2 = -3\} = \emptyset$



Prof. Dr. K. Madlener: Computeralgebra

.

Polynomideale und Gröbner-Basen

Motivation: Buchberger's Algorithmus

Beispiele (Forts.)

▶ Betrachtet man f_1 , f_2 als Polynome in $\mathbb{C}[x,y]$ und V(I) als Varietät über \mathbb{C}^2 , so

$$V(I) = \{(u,2) \in \mathbb{C}^2 : u^2 = -3\} = \{(\sqrt{3}i, 2), (-\sqrt{3}i, 2)\}$$

▶ d.h. 2 Punkte (mit $i = \sqrt{-1} \in \mathbb{C}$).

▶ ii) Sei
$$f = (y^2 + 6)(x - 1) - y(x^2 + 1)$$

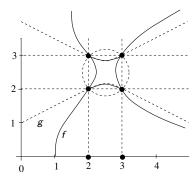
 $g = (x^2 + 6)(y - 1) - x(y^2 + 1)$ in $\mathbb{C}[x, y]$
 $h = (x - 5/2)^2 + (y - 5/2)^2 - 1/2$

$$I = \langle f, g \rangle$$

Motivation: Buchberger's Algorithmus

Beispiele (Forts.)

Maple Implicitplot



 $h = -f - g \in I$, d. h. h ist 0 auf V(I).

$$V(I) = \{(2,2), (2,3), (3,2), (3,3), \left(\frac{1\pm\sqrt{15}i}{2}, 1\mp\sqrt{15}i\right)\} \subseteq \mathbb{C}^2$$

◆ロト 4個ト 4 重ト 4 重 ト 9 Q ®

Prof. Dr. K. Madlener: Computeralgebra

497

Polynomideale und Gröbner-Basen

Motivation: Buchberger's Algorithmus

Beispiel (Forts.)

• Sei $h^* = x^2 + y^2 - 5x - 5y + 11 \in \mathbb{C}[x, y]$.

 $V(h^*) \cap \mathbb{R}^2$ ist Kreis mit Mittelpunkt (5/2, 5/2) und größerem Radius als der von $V(h) \cap \mathbb{R}^2$, enthält somit keine Punkte aus V(I). D. h. $h^* \notin I$. Es gilt $-f - g - h^* = 1$ in $\mathbb{C}[x, y]$.

Also $\langle f, g, h^* \rangle = \mathbb{C}[x, y]$ und somit $V(f, g, h^*) = \emptyset$.

Da jede gemeinsame Wurzel von f, g, h auch Nullstelle von 1 wäre.

Hilberts Nullstellensatz besagt genau dies:

 $V(J)=\varnothing \leadsto 1\in J$ (dies gilt über alg. abg. Körper) aber nicht über $\mathbb R$ wie Beispiel i) zeigt.

► Studium der Varietäten + Eigenschaften: alg. Geometrie. Gröbner Basen (oder Standard Basen Hironaka) sind spezielle Idealbasen für die viele der obigen Fragen leicht zu lösen sind. Polynomideale und Gröbner-Basen

Motivation: Buchberger's Algorithmus

Erinnerung: Grundlagen

- ▶ Der Fall n=1, d. h. F[x] ist leicht da F[x] euklidischer Bereich ist (und somit PID Hauptidealring), d. h. $\langle f_1, \ldots, f_s \rangle = \langle GGT(f_1, \ldots, f_s) \rangle$, d. h. o.B.d.A. s=1.
 - ▶ $f, g \in F[x], f = qg + r$ mit grad r < grad g. Dann
 - $f \in \langle g \rangle$ gdw r = 0 und
 - $V(g) = \{u_1, \dots, u_s\}$, falls $x u_1, \dots, x u_d$ die verschiedenen linearen Faktoren von g in F[x] sind.

Beachte:

```
\langle f_1,\ldots,f_s \rangle = \langle GGT(f_1,\ldots,f_s) \rangle ist nicht mehr gültig für F[x,y]. z. B. GGT(x,y)=1 in F[x,y] aber \langle x,y \rangle \neq \langle 1 \rangle = F[x,y] (F[x,y] ist nicht euklidisch \nexists p,q \quad p \cdot x + q \cdot y = 1).
```

- ▶ Gröbner Basen erlauben es einige der Eigenschaften zu erhalten (Division mit Rest).
- ▶ Erinnerung Reduktionstechniken.

←□ → ←□ → ← □ → ← □ → ○ ○

Prof. Dr. K. Madlener: Computeralgebra

Polynomideale und Gröbner-Basen

Reduktionstechniken zur Lösung des Wortproblems

Reduktionstechniken zur Lösung des Wortproblems

 U, \approx Äquivalenzrelation auf U (Kongruenz) U/\approx Struktur effektiv?

 $WP(U, \approx) : u, v \qquad u \stackrel{?}{\approx} v \text{ Wortproblem}.$

- ► Simplifikation, kanonische Funktionen.
 - $u \mapsto \operatorname{rep}(u)$ effektiv mit rep : $U \to U$ $u \approx v$ $\operatorname{gdw} \operatorname{rep}(u) = \operatorname{rep}(v)$
- ▶ Reduktionsmethode:: Schrittweise: $u \rightarrow u'$ mit u' "einfacher" als u. $u \rightarrow u_1 \rightarrow u_2 \rightarrow \cdots \rightarrow \text{rep}(u)$
- ► Termersetzungssysteme, Wortersetzungssysteme, Polynomersetzungssysteme



Reduktionstechniken zur Lösung des Wortproblems

Reduktionstechniken zur Lösung des Wortproblems (Forts.)

▶ Reduktions relation $\rightarrow \subseteq U \times U : a \rightarrow b$

Komposition $\rightarrow \circ \rightarrow'$: $a \rightarrow \rightarrow' b$: $a \rightarrow c \rightarrow' b$

Inverse Relation $\leftarrow: a \leftarrow b \text{ gdw } b \rightarrow a$

Symmetrischer Abschluß \leftrightarrow : $\rightarrow \cup \leftarrow$

Potenz $\rightarrow^i: \rightarrow^0 = id \rightarrow^{i+1} = \rightarrow^i \circ \rightarrow$

Transitive Hülle $\stackrel{+}{\rightarrow} = \bigcup_{i=1}^{\infty} \rightarrow^{i}$ Reflex. trans. Hülle $\stackrel{*}{\rightarrow} = \bigcup_{i=0}^{\infty} \rightarrow^{i}$

Reflex. trans. symm. Hülle $\stackrel{*}{\longleftrightarrow}$ Äquivalenzrelation

i. Allg. → rekursiv

► Spezialfälle: WP Monoide, WP Gruppen

$$B=\langle a,b;ab=1
angle$$
 monoid (byzykl. Monoid)

$$G = \langle \{a, b\} \mid \{a^2, b^2, aba^{-1}b^{-1}\} \rangle$$



Prof. Dr. K. Madlener: Computeralgebra

Polynomideale und Gröbner-Basen

Reduktionstechniken zur Lösung des Wortproblems (Forts.)

- ▶ R kommutativer Ring $I \subseteq R$ ideal gdw $p, q \in I \leadsto p q \in I$ $p \in I, r \in R \leadsto rp \in I$.
- ▶ $R[x_1, ..., x_n]$, R Ring (i.A. kommutativ), $I = \langle p_1, ..., p_m \rangle$ von $p_1, ..., p_m$ erzeugt Ideal in $R[x_1, ..., x_n]$.

$$f,g\in R[x_1,\ldots,x_n],\,f\equiv g\pmod l$$
 oder $f-g\in I$, d. h. f,g stellen gleiche Elemente im Quotientenring $R[x_1,\ldots,x_n]/I$ dar.

▶ Idee: Finde Reduktionsrelation \rightarrow_I mit

$$\stackrel{*}{\longleftrightarrow}_I \equiv \equiv_I$$

Reduktionstechniken zur Lösung des Wortproblems (Forts.)

- ► Standardbegriffe für → Reduktionsrelation:
 - ▶ $x \rightarrow x$ ist reduzibel d. h. $\exists y : x \rightarrow y$
 - x x ist irreduzibel oder in Normalform
 - ► $x \downarrow y$ ($x \uparrow y$) x, y haben gemeinsamen direkten Nachfolger / Vorgänger, d.h.

$$\exists z: x \to z \leftarrow y \ / \ x \leftarrow z \to y$$

- ▶ x ist eine \rightarrow -Normalform von y gdw y \rightarrow * \underline{x}
 - Wichtige Eigenschaften von \rightarrow :
- Noethersch:

Jede Reduktionsfolge terminiert

d. h. es gibt keine ∞ Folge $x_1 \rightarrow x_2 \rightarrow \cdots$

- ► Church Rosser $a \stackrel{*}{\longleftrightarrow} b \rightsquigarrow a \mid_{*} b$
- ▶ konfluent $a \uparrow^* b \rightsquigarrow a \downarrow^* b$
- ▶ lokal-konfluent $a \uparrow b \rightsquigarrow a \downarrow^* b$

<ロ > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > の の

Prof. Dr. K. Madlener: Computeralgebra

Polynomideale und Gröbner-Basen

Reduktionstechniken zur Lösung des Wortproblems

Reduktionstechniken zur Lösung des Wortproblems (Forts.)

8.4 Satz \rightarrow noethersch und Church Rosser, so WP für $\stackrel{*}{\longleftrightarrow}$ entscheidbar. d.h. Kanonische Simplifikationsfunktion

$$x \longmapsto y \text{ mit } x \stackrel{*}{\to} y \text{ NF für } x \text{ (Beachte} \to \text{effektiv)}.$$

- 8.5 Beispiel
- 1) Kommutative Halbgruppe mit Erzeugenden a, b, c, f, s. Relationen

$$as = c^2s$$
 $bs = cs$ $s = f$:: E

Reduktions relation auf der freien kommutativen Halbgruppe in a, b, c, f, s

$$R: s \to f \quad cf \to bf \quad b^2f \to af$$

auf Wörter in $a, b, c, f, s: u \to v$, so $ut \to vt$.

 \rightarrow_R ist noethersch, Church-Rosser und äquivalent zu E

$$a^{3}bcf^{3} =_{F} a^{2}b^{4}fs^{2}$$

Reduktionstechniken zur Lösung des Wortproblems

Reduktionstechniken: Beispiel (Forts.)

2) I ideal in $\mathbb{Q}[x, y]$ erzeugt von $x^3 - x^2$ $x^2y - x^2$ \Rightarrow Def auf $\mathbb{Q}[x, y]$: ledes "Vorkommen" von x^3 oder x^2

 \to Def. auf $\mathbb{Q}[x,y]$: Jedes "Vorkommen" von x^3 oder x^2y kann durch x^2 ersetzt werden.

Behauptung: \rightarrow ist noethersch + Church Rosser.

8.6 **Satz**

- a) \rightarrow Church Rosser $gdw \rightarrow$ konfluent.
- b) Newman Lemma. Sei \rightarrow noethersch: \rightarrow konfluent $gdw \rightarrow$ lokal konfluent.
- Verbunden unterhalb.

ロ > 4周 > 4 m > 4

Prof. Dr. K. Madlener: Computeralgebra

50

Polynomideale und Gröbner-Basen

Termordnungen und multivariate Division mit Rest: Reduktion in $K[x_1,\ldots,x_n]$

Termordnungen

▶ Partialordnung < auf S ist eine irreflexive transitive Relation < $\subseteq S \times S$. d.h. $\neg(\alpha < \alpha) \land \alpha < \beta < \gamma \Rightarrow \alpha < \gamma$ für alle $\alpha, \beta, \gamma \in S$ d. h. < ist asymmetrisch.

▶ Partialordnung ist total, falls $\alpha = \beta \lor \alpha < \beta \lor \beta < \alpha$ $(\alpha, \beta \in S)$

 Ordnung ist eine Wohlordnung, falls jede nicht leere Menge ein kleinstes Element besitzt.

▶ Schreibe $\alpha \leq \beta$, falls $\alpha = \beta$ oder $\alpha < \beta$.

 $lackbox{} < \mathsf{auf} \ \mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R} \ \mathsf{sind} \ \mathsf{total}, \ \mathsf{nur} \ \mathsf{auf} \ \mathbb{N} \ \mathsf{Wohlordnung}.$

▶ $X = \{x_1, ..., x_n\}$: freie kommutative Halbgruppe (Monoid) über X ist die Menge der Terme über X.

Darstellung der Terme:: Sei $\alpha=(\alpha_1,\ldots,\alpha_n)\in\mathbb{N}^n$. Identifikation $\alpha\to x^\alpha=x_1^{\alpha_1}\cdots x_n^{\alpha_n}\in F[x_1,\ldots,x_n]$, d. h. $\mathbb{N}^n\simeq$ Menge der Terme über X:=T(X). Operationen: \cdot,\cdot , KGV,\ldots

4□ > 4₫ > 4 ≧ > 4 ≧ > ½ 9 Q @

Termordnungen

8.7 Definition Eine Termordnung auf T(X) ist eine Relation \prec auf \mathbb{N}^n mit

- i) \prec ist totale Ordnung.
- ii) $\alpha \prec \beta \Rightarrow \alpha + \gamma \prec \beta + \gamma$ für alle $\alpha, \beta, \gamma \in \mathbb{N}^n$ ($s \prec t \Rightarrow su < tu$ für alle $s, t, u \in T(X)$ kompatibel mit Multiplikation)
- iii) ≺ ist Wohlordnung

!(insbesondere
$$1 = x_1^0 \cdots x_n^0 \prec t$$
 für alle $t \in T(X) \setminus 1$).!

- ► Falls i) gilt so ist iii) äquivalent zu, es gibt keine ∞-fallende Ketten.
- ▶ n = 1 Standard Ordnung auf \mathbb{N} ist die übliche Grad Ordnung auf T(X).

◆ロト ◆部ト ◆注ト ◆注 → りゅ(

Prof. Dr. K. Madlener: Computeralgebra

Polynomideale und Gröbner-Basen

Termordnungen und multivariate Division mit Rest: Reduktion in $\mathit{K}[\mathit{x}_1,\ldots,\mathit{x}_n]$

Beispiel: Termordnungen

- 8.8 Beispiel 3 Standard Termordnungen
- i) Lexikographische Ordnung
 - ▶ $\alpha \prec_{lex} \beta$ gdw erste nicht Null-Eintrag in $\alpha \beta$ ist negativ (von links). (Entspricht Präzedenz $x_1 \succ x_2 \succ \cdots \succ x_n$).
 - $\begin{array}{ll} {\color{red} \blacktriangleright} & n=3 & \alpha_1=(0,4,0) \ \alpha_2=(1,1,2) \ \alpha_3=(1,2,1) \ \alpha_4=(3,0,0) \\ {\color{red} {\rm Dann}} \ \alpha_1 \prec_{\mathit{lex}} \ \alpha_2 \prec \alpha_3 \prec \alpha_4. \end{array}$
- ii) Graduierte lexikographische Ordnung: $\alpha = (\alpha_1, \dots, \alpha_n)$ $\beta = (\beta_1, \dots, \beta_n) \in \mathbb{N}^n$
 - $\qquad \qquad \alpha \prec_{\mathit{grlex}} \beta \ \mathrm{gdw} \ \sum_{1 \leq i \leq n} \alpha_i < \sum_{1 \leq i \leq n} \beta_i \ \mathrm{oder} \ \Big(\sum \alpha_i = \sum \beta_i \wedge \alpha <_{\mathit{lex}} \beta \Big)$ (oft noch mit Gewichtsfunktion $W : \{1, \ldots, n\} \to \mathbb{R}^+$).
 - ▶ Es gilt $\alpha_4 \prec_{grlex} \alpha_1 \prec_{grlex} \alpha_2 \prec_{grlex} \alpha_3$.

Beispiel:Termordnungen (Fort.)

iii) Graduierte inverse lexikographische Ordnung:

- $\begin{array}{c} \quad \alpha \prec_{\mathit{grevlex}} \beta \ \mathrm{gdw} \ \sum_{1 \leq i \leq n} \alpha_i < \sum_{1 \leq i \leq n} \beta_i \ \mathrm{oder} \\ \\ (\sum \alpha_i = \sum \beta_i \quad \wedge \quad \text{am weitesten rechts stehende} \\ \quad \mathrm{nicht \ Null-Eintrag \ in} \ \alpha \beta \\ \quad \mathrm{ist \ positiv}). \end{array}$
- ▶ Es gilt $\alpha_4 \prec_{grevlex} \alpha_2 \prec_{grevlex} \alpha_3 \prec_{grevlex} \alpha_1$.
- \hookrightarrow n = 1 $\prec_{lex} = \prec_{grlex} = \prec_{grevlex}$



Prof. Dr. K. Madlener: Computeralgebra

509

Polynomideale und Gröbner-Basen

Termordnungen: Zentrales Lemma

8.9 Lemma

- a) $\prec_{lex}, \prec_{grlex}, \prec_{grevlex}$ sind Termordnungen.
- b) $s, t \in T[X]$, $s \mid t$ dann ist $s \leq t$ für jede Termordnung \prec .
- c) Die antilexikographische Ordnung \prec_{alex} auf \mathbb{N}^2 mit $\alpha \prec_{\mathit{alex}} \beta \ \mathrm{gdw} \ \beta \prec_{\mathit{lex}} \alpha$ ist Ordnung für die Bedingung iii) nicht gilt.

z. B.
$$S = \mathbb{N} \times \{0\}$$
 hat kein kleinstes Element, da $(0,0) \succ_{\mathit{alex}} (1,0) \succ_{\mathit{alex}} (2,0) \succ \cdots$

Wichtige Begriffe für Polynome und deren Reduktion

8.10 Definition Sei
$$f = \sum_{\alpha \in \mathbb{N}^n} c_{\mathsf{x}} x^{\alpha} \in R = K[x_1, \dots, x_n], f \neq 0.$$
 $c_{\alpha} \in F$ (nur endlich viele $\neq 0$), \prec Termordnung.

- i) $c_{\alpha}x^{\alpha}$ ist Monom in f für $c_{\alpha} \neq 0$ Coeff $(f, \alpha) = c_{\alpha}$.
- ii) Der Multigrad von f ist $mdeg(f) = max_{\prec} \{ \alpha \in \mathbb{N}^n : c_{\alpha} \neq 0 \}$
- iii) $LT(f) := x^{\mathsf{mdeg}(f)} = \mathsf{max}_{\prec} \{ t \in T[X] \mid \mathsf{Coeff}(f, t) \neq 0 \}$ Leitterm (Hauptterm) von f.
- iv) $LC(f) := c_{\mathsf{mdeg}(f)} \in F \setminus \{0\}$ Leitkoeffizient.
- v) $LM(f) = LC(f) \cdot LT(f) \in R$ Leitmonom.
- vi) Red(f) = f LM(f) Redukt von f.

◆ロト ◆母 ト ◆ き ト ◆ き ・ り へ

Prof. Dr. K. Madlener: Computeralgebra

E1

Polynomideale und Gröbner-Basen

Termordnungen und multivariate Division mit Rest: Reduktion in $\mathit{K}[\mathit{x}_1,\ldots,\mathit{x}_n]$

Beispiel

- \hookrightarrow \prec induziert noethersche Partialordnung << auf R: (Beweis!)
 - $f \ll g \text{ gdw } f = 0 \text{ und } g \neq 0 \text{ oder}$
 - $f \neq 0, g \neq 0 \land LT(f) \prec LT(g)$ oder
 - $\qquad \qquad f \neq 0, g \neq 0 \land LT(f) = LT(g) \land Red(f) << Red(g)$
 - **8.11 Beispiel** Sei $f = 4xyz^2 + 4x^3 5y^4 + 7xy^2z \in \mathbb{Q}[x, y, z]$

Termordnungen und multivariate Division mit Rest: Reduktion in $K[x_1,\ldots,x_n]$

Lemma

8.12 Lemma Sei \prec Termordnung auf T[X], $f, g \in R \setminus \{0\}$.

- i) mdeg(fg) = mdeg(f) + mdeg(g) $(LT(fg) = LT(f) \circ LT(g) \text{ in } T[X])$
- ii) $f + g \neq 0$ so $mdeg(f + g) \leq max\{mdeg(f), mdeg(g)\}$ d. h. $LT(f+g) \leq \max\{LT(f), LT(g)\}$ Gleichheit, falls $mdeg(f) \neq mdeg(g)$.
- ▶ Verallgemeinerung der Division mit Rest: Reduktion
- $f, f_1, \ldots, f_s \in R$ gesucht Darstellung von f mit $f = q_1 f_1 + \cdots + q_s f_s + r$ mit $q_1, \ldots, q_s, r \in R$.
- ? Wie bestimmt man q_i , welche Eigenschaften hat r?



Prof. Dr. K. Madlener: Computeralgebra

Polynomideale und Gröbner-Basen

Termordnungen und multivariate Division mit Rest: Reduktion in $K[x_1,\ldots,x_n]$

Beispiel

8.13 Beispiel

a) Sei
$$\prec = \prec_{lex}$$
 $f = xy^2 + 1$ $f_1 = xy + 1$

$$f = xy^2 + 3$$

$$f_1 = xy + 1$$

$$f_2 = y + 1$$

	xy + 1	y + 1
$xy^2 + 1$ $-(xy^2 + y)$	У	
-y + 1		-1
-(-y-1)		
2		

$$f=yf_1-1\cdot f_2+2$$

Prof. Dr. K. Madlener: Computeralgebra

$$f = 0 \cdot f_1 + (xy - x) \cdot f_2 + (x + 1)$$

▶ Kein Term in 2 bzw. -x + 1 ist durch ein $LT(f_i)$ teilbar.

Termordnungen und multivariate Division mit Rest: Reduktion in $K[x_1, \ldots, x_n]$

b) Sei
$$\prec = \prec_{lex}$$

$$f_1 = xy - 1$$

$$f_2 = v^2 - 1$$

	xy - 1	$y^{2} - 1$	Rest
$x^2y + xy^2 + y^2$ $-(x^2y - x)$	x		
$xy^2 + x + y^2$ $-(xy^2 - y)$	у		
$ \begin{array}{c} x + y^2 + y \\ -x \end{array} $			х
$y^2 + y$ $-(y^2 - 1)$		1	
y+1			

$$f = (x + y) \cdot f_1 + 1 \cdot f_2 + (x + y + 1)$$

▶ Kein Term in x + v + 1 durch ein $LT(f_i)$ teilbar.

Prof. Dr. K. Madlener: Computeralgebra

Polynomideale und Gröbner-Basen

Polynom-Reduktion

Polynom-Reduktion

procedure Algorithmus multivariate Division mit Rest {Eingabe: Polynome $f, f_1, \ldots, f_s \in R = F[x_1, \ldots, x_n], F$ Körper} {Termordnung \prec auf T[x].} {Ausgabe: $q_1, \ldots, q_s, r \in R \text{ mit } f = q_1 f_1 + \cdots + q_s f_s + r.}$ { Kein Monom in r ist durch ein $LT(f_1), \ldots, LT(f_s)$ teilbar.} begin

Prof. Dr. K. Madlener: Computeralgebra

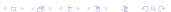
$$\begin{array}{l} 1 \; r := 0; p := f; \\ \text{for } i = 1, \cdots, s \; \text{do} \; q_i := 0 \\ 2 \; \text{while} \; p \neq 0 \; \text{do} \\ 3 \qquad \text{if } \mathsf{LM} \; (f_i) \; | \; \mathsf{LM}(p) \; \mathsf{f\"{u}r} \; \mathsf{ein} \; i \in \{1, \dots, s\} \\ \quad \quad \mathsf{then} \; \mathsf{choose} \; \mathsf{some} \; \mathsf{such} \; i : \; q_i := q_i + \frac{\mathsf{LM}(p)}{\mathsf{LM}(f_i)}; p := p - \frac{\mathsf{LM}(p)}{\mathsf{LM}(f_i)} f_i; \\ \quad \quad \mathsf{else} \; r := r + \mathsf{LM}(p); p := p - \mathsf{LM}(p) \\ 4 \; \mathsf{return} \; q_1, \dots, q_s, r \\ \mathsf{end} \end{array}$$

Polynom-Reduktion (Forts.)

8.14 Satz Bei Schritt 3 gelten folgende Invarianten

- i) $mdeg(p) \leq mdeg(f)$ $f = p + q_1f_1 + \cdots + q_sf_s + r$.
- ii) $q_i \neq 0 \Rightarrow \mathsf{mdeg}(q_i f_i) \leq \mathsf{mdeg}(f) \quad 1 \leq i \leq s$.
- iii) Kein Term in r ist teilbar durch ein $LM(f_i)$.
- ▶ Ist p_j der Wert von p in Durchgang j, so $p_{j+1} \prec p_j$.
- ► Der Algorithmus terminiert

Frage: Platz und Zeit Bedarf für den Algorithmus? Wovon hängen diese ab?.



4□ > 4問 > 4 = > 4 = > 9 Q ○

Prof. Dr. K. Madlener: Computeralgebra

517

Polynomideale und Gröbner-Basen

Polynom-Reduktion

Einschrittreduktion mit einer Menge $P = \{f_1, \dots, f_s\}$

► Einschritt Reduktionsrelation:: $f, g, h \in K[x]$ $g \xrightarrow{f} h$ g reduziert sich nach h mit f gdw. es gibt $s, t \in T[X]$ Coeff $(g, s) = c \neq 0$ s = LT(f)t (d.h. $LT(f) \mid s$) und

$$h = g - \frac{c}{LC(f)} \cdot t \cdot f$$
 Ein "Monom" in g wird ersetzt

- $ightharpoonup g \xrightarrow{P} h \text{ gdw } \exists f_i \in P \quad g \xrightarrow{f_i} h$
- $\longrightarrow \xrightarrow{*}, \xleftarrow{*}$ wie üblich.

Prof. Dr. K. Madlener: Computeralgebra

- ▶ Beachte: Multivariate Division mit Rest liefert ein r mit $r \xrightarrow{P}$ irreduzibel und $g \xrightarrow{*}_{P} r$. Strategie: Left-Most-Reduktion.
- ▶ Es gilt $\stackrel{*}{\underset{P}{\longleftrightarrow}} = \equiv_{\langle P \rangle}$ (Übung)

Beispiel

8.15 Beispiel $\prec = \prec_{lex}$, $f = x^2y + xy^2 + y^2$, $f_1 = xy - 1$, $f_2 = y^2 - 1$ f ist mit f_1 reduzibel in x^2y und xy^2 f ist mit f_2 reduzibel in xy^2 und y^2

	xy-1	$y^{2} - 1$	Rest	
$x^2y + xy^2 + y^2$	X	X		$x^2y + xy^2 + y^2$
$-(x^2y-x)$				$-(xy^2-x)$
$xy^2 + x + y^2$	у	1		$x^2y + y^2 + x$
$-(xy^2-y)$				$-(y^2-1)$
$x + y^2 + y$	X		X	$x^2y + x + 1$
-x				$-(x^2y-x)$
$y^2 + y$		1		
$-(y^2-1)$				
x+y+1				2x + 1

d.h. Rest muss nicht eindeutg sein, d.h. i. Allg. keine Konfluenz.

←□ → ←□ → ←필 → ←필 → → 필 → 今○

Prof. Dr. K. Madlener: Computeralgebra

Polynomideale und Gröbner-Basen

Polynom-Reduktio

Beispiel (Forts.)

Beachte i. Allg. Wahl von i mit $HT(f_i) \mid HT(P)$. Wähle kleinstes $i \rightsquigarrow$ die Quotienten q_1, \ldots, q_s und der Rest r sind eindeutig festgelegt, schreibe $r = f \operatorname{rem}(f_1, \ldots, f_s)$ für diese Wahl. Gewünscht wird:

- $f \in \langle f_1, \dots, f_s \rangle$ gdw $r = f \text{ rem}(f_1, \dots, f_s) = 0$
 - ▶ Dies stimmt. falls s = 1 ist.
 - ▶ Für $s \ge 2$ für Gröbner Basen! Sonst i. Allg. nicht

8.16 Beispiel
$$f = xy^2 - x$$
, $f_1 = xy + 1$, $f_2 = y^2 - 1$ $xy^2 - x \xrightarrow{f_1} -x - y = r$, d. h. $f = yf_1 + 0f_2 + (-x - y)$, aber $f = 0f_1 + xf_2 + 0$, d. h. $f \in \langle f_1, f_2 \rangle$.

518 Prof. Dr. K. Madlener: Computeralgebra 520

Längen von Reduktionsketten

Nachtrag: Längen von Reduktionsketten

ightharpoonup K = K(G, f) Max. Länge einer Reduktionsfolge

$$f = h_1 \stackrel{G}{\rightarrow} h_2 \stackrel{G}{\rightarrow} h_3 \stackrel{G}{\rightarrow} \cdots \stackrel{G}{\rightarrow} h_k$$

Siehe: Dube, Mishra, Yap Report 88 Courant Institute NY University 1986 Yap Fundamental Problems of algorithmic Algebra. Oxford.



4□ > 4個 > 4 = > 4 = > = 90

Prof. Dr. K. Madlener: Computeralgebra

Polynomideale und Gröbner-Basen

Längen von Reduktionsketten

Längen von Reduktionsketten (Forts.)

$$\mathsf{Term}(f) = (t_1, t_2, \cdots, t_l) \; \mathsf{mit} \; t_1 > t_2 > \cdots > t_l,$$

$$\emph{I}:= \mathsf{L"ange} \;\mathsf{von}\; \emph{f} \qquad \qquad \Delta:= \mathsf{Max}\{\emph{W}_i(t_1): i=1,\cdots,n\}$$

$$\mu_0 := \operatorname{\mathsf{Min}}\{\mu(g) : g \in G\} \quad \operatorname{\mathsf{M}}_0 := \operatorname{\mathsf{Max}}\{1, \operatorname{\mathsf{Max}}\{M(g) : g \in G\}\}$$

 $(W_i, \mu, M \text{ hängen von der Ordnung ab.})$

 $Robbiano's \ Charakterisierung \ von \ Termordnungen:$

$$W_k(x_1^{\alpha_1}...x_n^{\alpha_n}) = \sum_{i=1...n} w_{k,i}\alpha_i \qquad (k = 1,...,n), \text{ mit } w_{k,i} \in \mathbb{R}.$$

Z.B.max. Grad eines Terms bzw. max. Grad einer Variable in Term, z.B. für lex-Ordnung ist d max. Grad einer Variable in den Termen von f, so $1 \leq \mu(f) \leq d$ $1 \leq M(f) \leq d$

$$ightharpoonup K(G, f) \le 2^{1 + (\Delta S^n/M_0)} \quad \text{mit } S = (M_0/\mu_0) + 1 \quad \text{(Lex)}$$

Polynomideale und Gröbner-Basen

Längen von Reduktionsketten

Längen von Reduktionsketten (Forts.)

▶ Seien
$$d, \Delta, I, L$$
 mit $d \ge I - 2 > 0$ $\Delta > L$

$$f = x_1^{\Delta} x_n^L + x_1^{\Delta} x_n^{L-1} + \dots + x_1^{\Delta} x_n \quad \text{mit } G$$

$$g_1 = x_1 - (x_2^d x_3^d \cdots x_{n-1}^d)(x_n^d + x_n^{d-1} + \dots + x_n^{d-I+2})$$

$$g_2 = x_2 - (x_3^d x_4^d \cdots x_{n-1}^d)(x_n^d + x_n^{d-1} + \dots + x_n^{d-I+2})$$

$$\vdots$$

$$g_n = x_n^I - x_n^{I-1} - \dots - x_n$$

$$g_{n+1} = x_n^{I-1} - x_n^{I-2} - \dots - x_n$$

$$\vdots$$

$$g_{n+I-2} = x_n^2 - x_n$$

$$g_{n+I-1} = x_n - 1$$

Reduktionsfolge, die min</le>

Monom zum Reduzieren wählt, hat Länge der oberen Schranke

Andere Schranken (genauere) Dube, Mishra, Yap.

Prof. Dr. K. Madlener: Computeralgebra

Polynomideale und Gröhner-Rasen

Längen von Reduktionsketter

Längen von Reduktionsketten (Forts.)

Für jede Termordnung gilt:

$$\mathsf{K}(\mathsf{G},\mathsf{f}) \leq \begin{cases} \mathsf{L} & \mathsf{I} = 1\\ (1 + \mathsf{R}_{\mathsf{F}}\overline{u})\mathsf{L} & \mathsf{I} = 2\\ 2^{\mathsf{R}_{\mathsf{F}}\overline{u}}\mathsf{L} & \mathsf{I} \geq 3 \end{cases}$$

- L Anzahl der Monome in f
- ▶ / Max. Anzahl von Monomen in Polynom aus G
- $ightharpoonup R_F$ Konstante, die von > und G abhängt
- $ightharpoonup \overline{u}$ Maximum der "Gewichte" der Monomen in f $(\overline{u} = O(\operatorname{grad}(f)))$
- ► Head-Reduktion

$$lacksquare$$
 $<_{lex}$: $\le (d+1)^{\frac{n^2+n}{2}}D^n$ •d Max. Grad eines Mon. in f

• D Max. Grad eines Mon. in $g \in G$

$$<_{grlex} : \le (d+1)^n D^n$$

bzw.
$$(D+1)^n$$

Prof. Dr. K. Madlener: Computeralgebra 522 Prof. Dr. K. Madlener: Computeralgebra 524 September 524 Prof. Dr. K. Madlener: Computeralgebra 525 Prof. Dr. K. Madlener: Computeralgebra 526 Prof. Dr. K. Madlener: Computeralgebra 526 Prof. Dr. K. Madlener: Computeralgebra 527 Prof. Dr. K. Madlener: Computeralgebra 528 Prof. Dr. K. Madlener: Computeralgebra 529 Prof. Dr. K

Polynomideale und Gröbner-Basen

Term-Ideale und Hilbert's Basissatz

8.17 Definition Ein Termideal $I \subseteq R = F[x_1, \dots, x_n]$ ist ein von Terme erzeugtes Ideal in R, d. h. es gibt eine Teilmenge $A \subseteq \mathbb{N}^n$ mit

$$I = \langle x^A \rangle = \langle \{ x^\alpha : \alpha \in A \} \rangle$$

D.h. es wird von Monomen mit Koeffizienten 1 erzeugt.

8.18 Lemma Sei $I = \langle x^A \rangle \subseteq R$ Termideal, $\beta \in \mathbb{N}^n$, dann

$$x^{\beta} \in I \text{ gdw } \exists \alpha \in A : x^{\alpha} \mid x^{\beta}$$

Beweis: " \Leftarrow " klar, " \Rightarrow " sei $x^{\beta} \in I$, dann $x^{\beta} = \sum_{i \in E} q_i x^{\alpha_i}$ für eine endliche Menge E mit $q_i \in R = F[x_1, \ldots, x_n]$. Jeder Term, der in der rechten Summe vorkommt, ist teilbar durch ein $\alpha \in A$. x^{β} muss als Term in der rechten Seite vorkommen, also folgt die Behauptung.



< ロ ト → 付き → → き ト → ま → り Q (~)

Prof. Dr. K. Madlener: Computeralgebra

Polynomideale und Gröbner-Basen

Term-Ideale und Hilbert's Basissatz

Term-Ideale und Hilbert's Basissatz (Forts.)

8.19 Lemma Sei $I \subseteq R = F[x_1, \dots, x_n]$ Termideal, $f \in R$, dann sind äquivalent

- i) $f \in I$.
- ii) Jedes Monom von f liegt in I.
- iii) f ist eine F-Linearkombination von Terme in I.

Beweis:

i) \Rightarrow ii) nach Voraussetzung $f = \sum_{i \in E} q_i x^{\alpha_i}$ $\alpha_i \in A$

Jeder Term in f ist teilbar durch ein x^{γ} mit $\gamma \in A$ also ist jedes Monom von f in I.

 $ii) \Rightarrow iii) \Rightarrow i)$ klar. (gilt sogar für beliebige Ideale).

Term-Ideale und Hilbert's Basissatz (Forts.)

8.20 Beispiel $I = \langle x^3, x^2y \rangle \subseteq \mathbb{Q}[x, y] \rightsquigarrow 3x^4 + 5x^2y^3 \in I \ 2x^4y + 7x^2 \notin I.$

Die Implikation i) \Rightarrow ii) ist i. Allg. falsch. z. B. $g = x^3 - 2xy$, $h = x^2y - 2y^2 + x$, $I = \langle g, h \rangle$ $x^2 = -yg + xh$, dann $x^2 \in \langle LT(I) \rangle$, $x^2 \notin \langle LT(g), LT(h) \rangle$.

8.21 Folgerung Gleichheit von Termidealen::

Zwei Termideale sind gleich gdw sie enthalten die gleichen Terme.

8.22 Satz Dickson's Lemma

Termideale sind endlich erzeugt, d.h. für $A \subseteq \mathbb{N}^n$ gibt es eine endliche Teilmenge $B \subseteq A$ mit $\langle x^A \rangle = \langle x^B \rangle$.

Beweis: $A = \emptyset$ so klar. Sei $A \neq \emptyset$

◆ロト ◆個ト ◆見ト ◆見ト ・ 見 ・ 夕 ♀

Prof. Dr. K. Madlener: Computeralgebra

Polynomideale und Gröbner-Basen

Term-Ideale und Hilbert's Basissatz

Dickson's Lemma (Forts.)

- ▶ Betrachte ≤ auf \mathbb{N}^n mit $\alpha \le \beta$ gdw $\alpha_i \le \beta_i$, $1 \le i \le n$ (d. h. $x^\alpha \mid x^\beta$). Schreibe $\alpha < \beta$, falls $\alpha \le \beta$ und $\alpha \ne \beta$.
- ▶ < ist Partialordnung auf \mathbb{N}^n die i.Allg. nicht total ist $n \neq 1$.
- ▶ Sei $B = \{\alpha \in A : \forall \beta \in A, \beta \not< \alpha\}$ die Menge der minimalen Elemente von A bzgl. <.

Behauptung: B ist endlich, $B \subseteq A$,

(*)
$$\forall \alpha \in A \ \exists \beta \in B, \beta \leq \alpha$$

▶ Für $\alpha \in \mathbb{N}^n$ gibt es nur endlich viele $\beta \in \mathbb{N}^n$ mit $\beta \leq \alpha$. d.h. Es gibt keine ∞ fallende Kette

$$\alpha^{(1)} > \alpha^{(2)} > \alpha^{(3)} > \cdots$$
 in \mathbb{N}^n

► Insbesondere folgt (*).

Prof. Dr. K. Madlener: Computeralgebra



Polynomideale und Gröbner-Basen

Dickson's Lemma (Forts.)

- ▶ z. Z. B ist endlich. Induktion nach n.
 - ightharpoonup n = 1, dann ist < total $B = \{$ kleinstes Element von $A\}$.
 - ▶ $n \ge 2$, sei $A^* = \{(\alpha_1, \dots, \alpha_{n-1}) \in \mathbb{N}^{n-1} : \exists \alpha_n \in \mathbb{N} : (\alpha_1, \dots, \alpha_n) \in A\}$ nach Induktionvorraussetzung ist die Menge B^* der minimalen Elemente von A^* endlich.
- ► Für jedes $\beta = (\beta_1, \dots, \beta_{n-1}) \in B^*$ wähle $b_\beta \in \mathbb{N}$ mit $(\beta_1, \dots, \beta_{n-1}, b_\beta) \in A$ und sei $b = \max\{b_\beta : \beta \in B^*\}$.
- ▶ Behauptung: $(\alpha_1, ..., \alpha_n) \in B$, so $\alpha_n \leq b$.
- ▶ Sei $\alpha = (\alpha_1, \dots, \alpha_n) \in A$, dann gibt es ein minimales Element $\beta = (\beta_1, \dots, \beta_{n-1}) \in B^*$ von A^* mit $\beta \leq (\alpha_1, \dots, \alpha_{n-1})$.



◆ロト 4個ト 4 恵 ト 4 恵 ト . 車 . 夕 Q (*)

Prof. Dr. K. Madlener: Computeralgebra

529

Polynomideale und Gröbner-Basen

Term-Ideale und Hilbert's Basissatz

Dickson's Lemma (Forts.)

▶ Ist $\alpha_n > b$, so

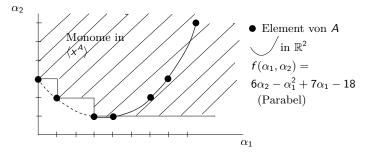
$$(\beta_1,\ldots,\beta_{n-1},b_{\beta}) \leq (\beta_1,\ldots,\beta_n,b) < \alpha$$

 α ist nicht minimal, d. h. $\alpha_n \leq b$.

- ▶ Analog zeigt man, dass alle Komponenten beschränkt sind, es gibt nur endlich viele $(\alpha_1, \dots, \alpha_n) \in B$.
- ▶ $\alpha \leq \beta \text{ gdw } x^{\alpha} \mid x^{\beta} \rightsquigarrow x^{A} \subseteq \langle x^{B} \rangle \text{ und somit } \langle x^{A} \rangle \subseteq \langle x^{B} \rangle.$ ⊃ folgt aus $B \subseteq A$.
- ► Beachte: Ideale können auch in Monoiden betrachtet werden. Ideale in e.e. kommutativen Monoiden sind endlich erzeugt (als Ideal).

Beispiel

8.23 Beispiel
$$n = 2$$
, $A = \{(\alpha_1, \alpha_2) \in \mathbb{N}^2 : 6\alpha_2 = \alpha_1^2 - 7\alpha_1 + 18\}$ Die Menge der minimalen Elemente ist $B = \{(0, 3), (1, 2), (3, 1)\}$, d. h. $\langle x^A \rangle = \langle y^3, xy^2, x^3y \rangle$



Prof. Dr. K. Madlener: Computeralgebra 53

Polynomideale und Gröbner-Basen

Term-Ideale und Hilbert's Basissatz

Folgerung

8.24 Folgerung Sei \prec eine totale Ordung auf \mathbb{N}^n mit $\forall \alpha, \beta, \gamma \in \mathbb{N}^n$, $\alpha \prec \beta \Rightarrow \alpha + \gamma \prec \beta + \gamma$.

ightharpoonup \prec ist wohlfundiert $\operatorname{gdw} \alpha \succeq 0$ für $\alpha \in \mathbb{N}^n$.

Beweis: " \Leftarrow " $A \subseteq \mathbb{N}^n$, $A \neq \emptyset$, $I = \langle x^A \rangle \subseteq R$ ist endlich erzeugt nach Dickson's Lemma, d. h. $\exists \alpha_1, \ldots, \alpha_s \in A$:

 $I = \langle x^{\alpha_1}, \dots, x^{\alpha_s} \rangle$ seien sie als $\alpha_1 \prec \alpha_2 \prec \dots \prec \alpha_s$ geordnet. Behauptung $\min_{\prec} A = \alpha_1$.

▶ Sei $\alpha \in A$ beliebig, da $x^{\alpha} \in I \exists i \leq s, \ \gamma \in \mathbb{N}^n : \alpha = \alpha_i + \gamma, \ d. \ h.$ $\alpha = \alpha_i + \gamma \succeq \alpha_1 + \gamma \succeq \alpha_1 + 0 = \alpha_1 \leadsto \alpha_1 = \min_{\prec} A.$

◆ロト 4個ト 4 恵 ト 4 恵 ト . 重 . 夕久の

▶ D.h. die Bedingung iii) der Termordnungen kann durch (iii)* $\forall \alpha \in \mathbb{N}^n$, $\alpha \succeq 0$ ersetzt werden.



Term-Ideale und Hilbert's Basissatz

Notation-Beispiel

Schreibweise: $G \subseteq R = F[x_1, ..., x_n]$ $LM(G) = \{LM(g) : g \in G\}, LT(G) = \{LT(g) : g \in G\}$

- ▶ Ist $I \subseteq R$ ideal, dann gibt es eine endliche Teilmenge $G \subseteq I$ mit $\langle LT(G) \rangle = \langle LT(I) \rangle$ nach Dickson's Lemma.
- ▶ Es kann aber endliche Mengen G die I erzeugen geben mit

$$\langle LT(G)\rangle \subsetneq \langle LT(I)\rangle$$

▶ Beispiel: $g = x^3 - 2xy$ $h = x^2y - 2y^2 + x$ $\prec = \prec_{grley}$ $G = \{g, h\}$ $I = \langle G \rangle$ $x^2 = -yg + xh$, d. h. $x^2 \in \langle LT(I) \rangle$, aber $x^2 \notin \langle LT(G) \rangle = \langle x^3, x^2y \rangle$.

◆ロト ◆個ト ◆恵ト ◆恵ト 連 めのの

4□ > 4問 > 4 = > 4 = > 9 Q ○

Prof. Dr. K. Madlener: Computeralgebra

533

Polynomideale und Gröbner-Basen

Term-Ideale und Hilbert's Basissatz

Hilbert's Basissatz

8.25 Lemma Sei *I* ideal in $R = F[x_1, ..., x_n]$.

Ist $G \subseteq I$ endlich mit $\langle LT(G) \rangle = \langle LT(I) \rangle$, so gilt $\langle G \rangle = I$.

Beweis: Sei $G = \{g_1, \dots, g_s\}$ $f \in I$ beliebig.

Division mit Rest liefert

- ▶ $f = q_1g_1 + \cdots + q_sg_s + r$ mit $q_1, \ldots, q_s, r \in R$. Wobei r = 0 oder kein Term in r ist durch $LT(f_i)$ für ein i teilbar.
- $r = f q_1g_1 \cdots q_sg_s \in I \leadsto LT(r) \in LT(I) \subseteq \langle LT(G) \rangle.$
- ▶ Wegen Lemma 8.18 folgt r = 0. Also $f \in \langle g_1, \dots, g_s \rangle = \langle G \rangle$.

8.26 Satz Hilbert's Basissatz

Jedes Ideal I in $R = F[x_1, \dots, x_n]$ ist endlich erzeugt. Genauer, es gibt endliche Teilmenge $G \subseteq I$ mit $\langle G \rangle = I$ und $\langle LT(G) \rangle = \langle LT(I) \rangle$.

▶ Dickson's Lemma angewendet auf $\langle LT(I) \rangle$.

Folgerungen

8.27 Folgerung Aufsteigende Kettenbedingung (E.Noether)

Sei $I_1 \subseteq I_2 \subseteq I_3 \subseteq \cdots$ eine aufsteigende Kette von Idealen in R. Dann gibt es ein n mit

$$I_n = I_{n+1} = I_{n+2} = \cdots$$
 für ein $n \in \mathbb{N}$.

Beweis: $I = \bigcup_{i>1} I_i$ ist ideal, endlich erzeugt d.h.

$$I = \langle g_1, \ldots, g_s \rangle$$
. Wähle $n = \min\{j \geq 1, g_1, \ldots, g_s \in I_j\}$.

- ▶ Ringe die diese Bedingung erfüllen heißen noethersch, d. h. $F[x_1, ..., x_n]$ ist noethersch.
- ightharpoonup Allgemeiner gilt: Ist R noethersch so auch R[x]

(ロ) (型) (E) (E) (E) の

Prof. Dr. K. Madlener: Computeralgebra

Polynomideale und Gröbner-Basen

Gröbnerbasen und S-Polynome

Gröbnerbasen bezüglich Termordnungen

8.28 Definition Sei \prec eine Termordnung und $I \subseteq R$ ein Ideal. Eine endliche Teilmenge $G \subseteq I$ heißt Gröbner Basis für I bzgl. \prec , falls $\langle LT(G) \rangle = \langle LT(I) \rangle$.

Beachte: Jede Gröbner Basis für I ist eine Idealbasis von I nach Lemma 8.25, es gilt

$$f \in I$$
 gdw $r = f \operatorname{rem}(G) = 0$
gdw $f \xrightarrow{*}_{G} 0$

- ▶ d. h. \xrightarrow{G} ist konfluent auf I.
 - **8.29 Folgerung** Jedes Ideal I in $R = F[x_1, \ldots, x_n]$ hat eine Gröbner Basis (Satz 8.26 Hilbert's Basissatz).
 - **8.30 Beispiel** $g = x^3 2xy$, $h = x^2y 2y^2 + x$ ist keine *G*-Basis von $\langle g, h \rangle$.

Gröbnerbasen und S-Polynome

Konfluenz von \longrightarrow für Gröbner Basen

8.31 Lemma Sei G G-Basis für $I \subseteq R$. $f \in R$. Dann gibt es ein eindeutiges Polynom $r \in R$ mit

- i) $f r \in I$.
- ii) Kein Monom in r ist teilbar durch ein Element in LT(G).

Beweis:

- Existenz folgt aus Algorithmus multivariate Division mit Rest.
- Eindeutigkeit: Angenommen $f = h_1 + r_1 = h_2 + r_2$ Kein Monom in r_1, r_2 ist teilbar durch ein Element in LT(G). $r_1 - r_2 = h_2 - h_1 \in I \leadsto LM(r_1 - r_2)$ ist teilbar durch LT(g) mit $g \in G$ nach Lemma 8.18 $\rightsquigarrow r_1 - r_2 = 0$.

Prof. Dr. K. Madlener: Computeralgebra

Polynomideale und Gröbner-Basen

Gröbnerbasen und S-Polynome

Konfluenz von \longrightarrow für Gröbner Basen (Forts.)

- ▶ Folgerung: Wir können für Gröbnerbasen *G* schreiben $f \text{ rem } G = r \in R$ r ist die einzige Normalform von f bzgl. \longrightarrow .
- $ightharpoonup \sim \longrightarrow_{G}$ ist konvergent.
 - **8.32** Satz Sei G Gröbner-Basis für $I \subseteq R$ bzgl. Termordnung \prec , $f \in R$.
- $f \in I \text{ gdw } f \text{ rem } G = 0 \text{ gdw } f \xrightarrow{*} 0.$

Umgekehrt: Falls diese Eigenschaft für G gilt, so ist G eine Gröbner Basis (Beweis!).

Beachte: Beweis von Hilbert's Basissatz nicht konstruktiv, auch Dickson's Lemma liefert uns keine Konstruktion für die Menge der minimalen Elemente für $A = \{\alpha : x^{\alpha} \in \langle LT(I) \rangle \}.$

4□ > 4個 > 4 = > 4 = > = 90

S-Polynome: Konfluenztest

8.33 Definition

Seien $g, h \in R$ nicht Null, $\alpha = (\alpha_1, \dots, \alpha_n) = \mathsf{mdeg}(g)$ $\beta = (\beta_1, \dots, \beta_n) = \mathsf{mdeg}(h) \text{ und } \gamma = (\mathsf{max}\{\alpha_1, \beta_1\}, \dots, \mathsf{max}\{\alpha_n, \beta_n\})$ das S-Polynom von g und h ist

$$S(g,h) = \frac{x^{\gamma}}{LM(g)}g - \frac{x^{\gamma}}{LM(h)}h \in R$$

- ightharpoonup Offenbar S(g,h) = -S(h,g).
- ▶ Da $\frac{x^{\gamma}}{LM(g)}$, $\frac{x^{\gamma}}{LM(h)} \in R$ gilt, folgt $S(g,h) \in \langle g,h \rangle$.
- ▶ Beachte: $LT(S(g,h)) \prec x^{\gamma}$. (Wichtig für Noethersche Induktionsbeweise nach \prec).

Prof. Dr. K. Madlener: Computeralgebra

Polynomideale und Gröhner-Baser

Gröbnerbasen und S-Polynome

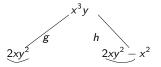
S-Polynome: Konfluenztest (Forts.)

8.34 Beispiel
$$g = x^3 - 2xy$$
 $h = x^2y - 2y^2 + x$ $\in \mathbb{Q}[x, y] \prec_{grlex}$

$$\hookrightarrow \alpha = (3,0), \beta = (2,1), \gamma = (3,1)$$

•
$$S(g,h) = \frac{x^3y}{x^3}g - \frac{x^3y}{x^2y}h = y(x^3 - 2xy) = -x^2$$

 $-x(x^2y - 2y^2 + x)$



Differenz ist $-x^2$

- ▶ Um Konfluenz auf $\langle g, h \rangle$ zu erreichen müssen S-Polynome sich nach 0 reduzieren lassen. Frage: Folgt aus $u - v \rightarrow_{(\sigma, h)}^* 0$ auch $u \downarrow_{(\sigma, h)}^* v$
- ▶ Die Leitmonome bei Linearkombinationen können sich wegheben, dieses kann durch die S-Polynome charakterisiert werden.

Polynomideale und Gröbner-Basen Gröbnerbasen und S-Polynome

S-Polynome: Hauptlemma

8.35 Lemma Sei $g_1, \ldots, g_s \in R$, $\alpha_1, \ldots, \alpha_s \in \mathbb{N}^n$, $c_1, \ldots, c_s \in F \setminus \{0\}$

- ▶ $\delta \in \mathbb{N}^n$ mit $\alpha_i + \mathsf{mdeg}(g_i) = \delta$ für 1 < i < s und $\mathsf{mdeg}(f) < \delta$ (d. h. x^{δ} ist nicht Leitterm von f).

Dann gilt $x^{\gamma_{ij}}$ teilt x^{δ} für 1 < i < j < s wobei $x^{\gamma_{ij}} = \mathsf{KGV}(LT(g_i), LT(g_i))$ und es gibt $c_{ii} \in F$ mit

$$(*) f = \sum_{1 \le i \le j \le s} c_{ij} x^{\delta - \gamma_{ij}} S(g_i, g_j)$$

und $\mathsf{mdeg}(x^{\delta - \gamma_{ij}} S(g_i, g_i)) \prec \delta$ für alle 1 < i < j < s.



4□ > 4個 > 4 = > 4 = > = 90

Prof. Dr. K. Madlener: Computeralgebra

Gröhnerhasen und S-Polynom

Prof. Dr. K. Madlener: Computeralgebra

Polynomideale und Gröbner-Basen

Gröbnerbasen und S-Polynom

S-Polynome: Hauptlemma-Beweis

Beweis: O.b.d.A. $LC(g_i) = 1$ (sonst verändere die c_i) und somit $LT(g_i) = LM(g_i) = x^{\mathsf{mdeg}(g_i)}$ für alle i.

- ▶ Sei $1 \le i < j \le s$. Der Term $x^{\delta} = x^{\alpha_i} LT(g_i) = x^{\alpha_j} LT(g_i)$ ist gemeinsamer Vielfacher von $LT(g_1)$ und $LT(g_i)$
- \hookrightarrow d. h. $x^{\gamma_{ij}} \mid x^{\delta}$ und $\alpha_i + \mathsf{mdeg}(g_i) = \alpha_i + \mathsf{mdeg}(g_i) = \delta$ Wegen

$$S(g_i, g_j) = \frac{x^{\gamma_{ij}}}{LT(g_i)}g_i - \frac{x^{\gamma_{ij}}}{LT(g_j)}g_j$$

- ▶ Also $\mathsf{mdeg}(S(g_i,g_i)) \prec \gamma_{ii}$, da die Leitterme in dieser Summe sich wegheben, es gilt somit
- ▶ (*) Wird nun durch Induktion nach s bewiesen.
- s = 1 nicht möglich, Behauptung richtig.

S-Polynome: Hauptlemma-Beweis

▶ Sei *s* > 2

$$g = f - c_{1}x^{\delta - \gamma_{12}}S(g_{1}, g_{2})$$

$$= c_{1}x^{\alpha_{1}}g_{1} + c_{2}x^{\alpha_{2}}g_{2} + \sum_{3 \leq i \leq s} c_{i}x^{\alpha_{i}}g_{i} - c_{1}x^{\delta - \gamma_{12}} \left(\frac{x^{\gamma_{12}}}{LT(g_{1})}g_{1} - \frac{x^{\gamma_{12}}}{LT(g_{2})}g_{2}\right)$$

$$= c_{1}\underbrace{(x^{\alpha_{1}} - x^{\delta - \mathsf{mdeg}(g_{1})})}_{= 0}g_{1} + (c_{2}x^{\alpha_{2}} + c_{1}x^{\delta - \mathsf{mdeg}(g_{2})})g_{2} + \sum_{3 \leq i \leq s} c_{i}x^{\alpha_{i}}g_{i}$$

$$= (c_{1} + c_{2})x^{\alpha_{2}}g_{2} + \sum_{3 \leq i \leq s} c_{i}x^{\alpha_{i}}g_{i}$$

▶ Hierbei verwende $\alpha_1 + \mathsf{mdeg}(g_1) = \delta = \alpha_2 + \mathsf{mdeg}(g_2)$

Polynomideale und Gröhner-Baser

S-Polynome: Hauptlemma-Beweis

- ▶ $\mathsf{mdeg}(g) \leq \mathsf{max}_{\prec} \{ \mathsf{mdeg}(f), \mathsf{mdeg}(x^{\delta \gamma_{12}} S(g_1, g_2)) \} \prec \delta,$ g hat die ursprüngliche Gestalt mit s-1 Summanden (falls $c_1+c_2\neq 0$) sonst s-2 Summanden.
- \hookrightarrow Nach Induktionsvor. $g = \sum_{2 \le i \le i \le s} c_{ij} x^{\delta \gamma_{ij}} S(g_i, g_i)$ Für $c_{ii} \in F$ $(2 \le i < j \le s)$. g = 0, falls s = 2. Setzt man $c_{12} = c_1$ und $c_{1i} = 0$ für $3 \le i \le s$, so

$$f=g+c_1x^{\delta-\gamma_{ij}}S(g_1,g_2)=\sum_{1\leq i< j\leq s}c_{ij}x^{\delta-\gamma_{ij}}S(g_i,g_s)$$



Charakterisierungssatz mit S-Polynome

8.36 Satz Eine endliche Menge $G=\{g_1,\ldots,g_s\}\subseteq R$ ist eine Gröbner Basis für $\langle G \rangle \mod$

$$S(g_i, g_j)$$
 REM $(g_1, \dots, g_s) = 0$ für $1 \le i < j \le s$

$$\mathrm{gdw} \qquad S(g_i, g_j) \xrightarrow{\ ^* \ } 0 \text{ für } 1 \leq i < j \leq S.$$

Beweis: " \Rightarrow " klar, " \Leftarrow " sei $f \in I \setminus \{0\}$ zeige $LT(F) \in \langle LT(G) \rangle$ $f = \sum_{1 \le i \le s} q_i g_i$ $\delta = \max_{1 \le i \le s} \{ \operatorname{mdeg}(q_i g_i), 1 \le i \le s \}$

Angenommen $mdeg(f) \prec \delta$, d. h. δ Monome heben sich weg.

- ▶ $f^* = \sum_{1 \leq i \leq s, mdeg(q_ig_i) = \delta} LM(q_i)g_i$ hat die Gestalt, wie sie in Lemma 8.35 vorausgesetzt wird.
- ▶ f^* lässt sich als Linearkombination von Polynomen der Form $x^{\alpha_{ij}}S(g_i,g_j)$ mit $\alpha_{ij} \in \mathbb{N}^n$ darstellen, wobei $\alpha_{ij} + \mathsf{mdeg}(S(g_i,g_j)) \prec \delta$ nach Lemma 8.35.

9P = (=) (=)

Prof. Dr. K. Madlener: Computeralgebra

545

Polynomideale und Gröbner-Baser

Gröbnerbasen und S-Polynome

Charakterisierungssatz mit S-Polynome

Nach Voraussetzung gilt $S(g_i, g_i)$ rem $(g_1, \ldots, g_s) = 0$, d. h.

$$f^* = \sum_{1 \leq i \leq s} q_i^* g_i \text{ mit } \max_{\prec} \{ \mathsf{mdeg}(q_i^* g_i) : 1 \leq i \leq s \} \prec \delta$$

- ▶ $f f^*$ und f^* haben Darstellungen der Form $\sum q_i, g_i \text{ mit max}_{\prec} \{ \mathsf{mdeg}(q_i g_i) : 1 \leq i \leq s \} \prec \delta \text{ also auch } f.$
- ▶ Wiederholte Anwendung liefert Darstellung von f mit $f = \sum_{1 \leq i \leq s} q_i g_i$ und mdeg $f = \delta = \max\{\text{mdeg } q_i g_i\}$ d. h. $\text{mdeg}(f) = \text{mdeg}(q_i g_i)$ für mindestens ein $f = \delta$ und somit $f = \delta$ un

Polynomideale und Gröbner-Basen

Gröbnerbasen und S-Polynome

Beispiel

8.37 Beispiel Twisted Cubic

$$C = V(G)$$
 mit $G = \{y - x^2, z - x^3\}$, d. h. $C = \{(a, a^2, a^3) : a \in F\}$. In \mathbb{R}^3 Schnitt von $V(y - x^2)$ und $V(z - x^3)$.

G ist Gröbner Basis für
$$\langle G \rangle$$
 bzgl. lex. Ordnung $y \succ z \succ x$.
• $S(y-x^2,z-x^3) = z(y-x^2) - y(z-x^3) = yx^3 - zx^2 \xrightarrow{*}_G 0$
 $= x^3(y-x^2) + (-x^2)(z-x^3) + 0$

(ロ) (部) (目) (目) (目) (9)

Prof. Dr. K. Madlener: Computeralgebra

Polynomideale und Gröbner-Basen

Buchberger's Algorithmus

Buchberger's Algorithmus

```
\{ \text{Ausgabe: Gr\"{o}bner Basis } G \subseteq R \text{ f\"{u}r } I = \langle f_1, \dots, f_s \rangle \text{ bzgl. } \prec \}  \{ \text{ mit } \{ f_1, \dots, f_s \} \subseteq G. \}  begin 1 \ G := \{ f_1, \dots, f_s \}  2 \ \text{repeat}  S := \varnothing  Ordne die Elemente von G als g_1, \dots, g_t for i \leq j \leq t do  r := S(g_i, g_j) \text{ rem } (g_1, \dots, g_t)  if r \neq 0 then S := S \cup \{ r \}  if S = \varnothing then return G else G := G \cup S end
```

{Eingabe: $f_1, \ldots, f_s \in R = F[x_1, \ldots, x_n], \prec \text{Termordnung.}}$

Polynomideale und Gröbner-Basen

Buchberger's Algorithmus: Korrektheit

8.38 Satz Algorithmus ist korrekt und terminierend.

Beweis: Es gilt stets $\langle G \rangle = I$ (nur Elemente aus I hinzu), falls Terminierung, so korrekt.

- $G_i \subseteq G_{i+1}$, d. h. $\langle LT(G_i) \rangle \subseteq \langle LT(G_{i+1}) \rangle$ aufsteigende Kette, die stabil werden muss. D.h., wenn $G_i = G_{i+1}$, so erfüllen alle S Polynome von G_i : $S(\cdot, \cdot)$ rem $(G_i) = 0$.
- ▶ Frage: Platz und Zeitbedarf, Implementierungen.
 - **8.39 Folgerung** Folgende Probleme sind mit *G*-Basen entscheidbar
- i) Wortproblem $(f \in \langle G \rangle)$
- ii) $\langle G \rangle \subseteq \langle H \rangle$
- iii) $\langle G \rangle = \langle H \rangle$

Prof. Dr. K. Madlener: Computeralgebra

Polynomideale und Gröbner-Baser

Buchberger's Algorithmus

Prof. Dr. K. Madlener: Computeralgebra

Beispiel

$$f_1 = x^3 - 2xy$$
 $f_2 = x^2y - 2y^2 + x$ $\in \mathbb{Q}[x, y], y < x \prec = \prec_{grlex}$

- ▶ $S(f_1, f_2) = -x^2$ $LT(S(f_1, f_2)) = -x^2 \notin \langle x^3, x^2y \rangle$ • $f_3 := S(f_1, f_2)$ rem $(f_1, f_2) = -x^2$. Dann $S(f_1, f_2)$ rem $(f_1, f_2, f_3) = 0$
- ► $S(f_1, f_3) = 1f_1 (-x)f_3 = -2xy$ $S(f_1, f_3) \text{ rem } (f_1, f_2, f_3) = -2xy =: f_4 \bullet$ $S(f_1, f_3) \xrightarrow[f_1, f_2, f_3]{*} 0$
- ▶ $S(f_2, f_3) = 1f_2 (-y)f_3 = -2y^2 + x$ irred. • $f_5 = S(f_2, f_3)$ rem $(f_1, \dots, f_4) = -2y^2 + x$ $\rightarrow S(f_i, f_j)$ rem $(f_1, \dots, f_5) = 0$ für $1 \le i < j \le 5$, d. h. $\{f_1, \dots, f_5\}$ ist Gröbner Basis.

Buchberger's Algorithmus (Forts.)

- ▶ Varianten des Buchberger Algorithmus um:
 - 1. Gewisse *S*-Polynome nicht zu reduzieren.
 - 2. Basis so klein wie möglich zu halten.
 - 3. Wiederholungen zu vermeiden.
- ► Ziel:: Implementierung zu optimieren

Beachte:

Ist G Gröbner Basis und $f \in \langle G \rangle$, so ist $G \cup \{f\}$ auch Gröbner Basis.

8.40 Lemma Ist *G* Gröbner Basis von $I \subset R$, $g \in G$.

 $LT(g) \in \langle LT(G \setminus \{g\}) \rangle$, dann ist $G \setminus \{g\}$ Gröbner Basis von I.

Beweis: z.z. $\langle LT(G) \rangle = \langle LT(G \setminus \{g\}) \rangle = \langle LT(I) \rangle$ wegen $LT(g) \in \langle LT(G \setminus \{g\}) \rangle$ folgt die Behauptung.

Prof. Dr. K. Madlener: Computeralgebra

Polynomideale und Gröbner-Basen

Buchberger's Algorithmus

Minimale- und reduzierte- Gröbner Basen

8.41 Definition Eine Menge $G \subseteq R$ heißt minimale (bzw. reduzierte) Gröbner Basis für $I = \langle G \rangle$, falls G eine G-Basis ist und für alle $g \in G$

- i) LC(g) = 1
- ii) $LT(g) \notin \langle LT(G \setminus \{g\}) \rangle$ (minimal)
- iii) g ist irreduzibel bzgl. $G \setminus \{g\}$. (reduzierte)

8.42 Satz Eindeutigkeitssatz

Jedes Ideal hat eine eindeutige reduzierte Gröbner Basis bzgl. \prec .

Beweis: Existenz: Anwendung von Lemma 8.40 o.B.d.A.

 $G = \{g_1, \ldots, g_s\}$ minimal.

Sei
$$h_i = g_i \text{ rem } \{h_1, \dots, h_{i-1}, g_{i+1}, \dots, g_s\}$$
 $i = 1, \dots, s$
= $\mathsf{NF}(g_i, G \setminus \{g_i\})$

Buchberger's Algorithmus

Eindeutigkeit reduzierter Gröbner Basen

- ▶ Seien G, G^* reduzierte Gröbner Basen für I. Dann $LT(G) = LT(G^*)$: für $t \in LT(G) \subseteq \langle LT(G) \rangle = \langle LT(I) \rangle = \langle LT(G^*) \rangle$ gibt es $g^* \in G^*$ mit $LT(g^*) \mid t$ und es gibt $g \in G$ mit $LT(g) \mid LT(g^*) \mid t$. G ist reduziert $\leadsto LT(g) = LT(g^*) = t \leadsto LT(G) \subseteq LT(G^*)$. = aus Symmetrie
- ▶ Sei $g \in G$ und $g^* \in G^*$ mit $LT(g) = LT(g^*)$. Da beide reduziert sind, ist kein Monom in $g g^* \in I$ reduzibel also $g = g^*$, d. h. $G \subseteq G^*$ und umgekehrt.

8.43 Lemma Translationslemma:

$$p - q \xrightarrow{F} h \rightsquigarrow \exists p', q' : p \xrightarrow{*}_{F} p', q \xrightarrow{*}_{F} q', h = p' - q'$$

$$p - q \xrightarrow{*}_{} 0 \rightsquigarrow \exists g : p \xrightarrow{*}_{} g, q \xrightarrow{*}_{} g$$

$$p \rightarrow q \rightsquigarrow tp \rightarrow tq \land p + h \downarrow^{*}_{} q + h$$

Frage: Wieviele Polynome müssen hinzugenommen werden, um eine Gröbner Basis zu bekommen?



Prof. Dr. K. Madlener: Computeralgebra

553

Polynomideale und Gröbner-Basen

Prof. Dr. K. Madlener: Computeralgebra

Buchberger's Algorithmus

Äquivalente Charakterisierungen für GB

8.44 Satz Chakterisierungssatz für Gröbner Basen bzgl. \prec Sei $G \subseteq R$ endlich $I = \langle G \rangle$. Dann sind äquivalent

- 1. (Normalform) jedes $f \in R$ hat eindeutige $\underset{G}{\longrightarrow}$ Normalform
- 2. (Standard Basen) $\langle LT(G) \rangle = \langle LT(I) \rangle$
- 3. (Buchberger Krit.) für alle $f,g\in G$ gilt $S(f,g)\stackrel{*}{\longrightarrow} 0$
- 4. (Church Rosser) \xrightarrow{G} ist Church-Rosser, d. h. $f \xleftarrow{*}_{G} g$, so $f \downarrow_{G}^{*} g$
- 5. (Extended Standard) Jedes $f \in I$ hat eine Darstellung

$$f = \sum_{1 \leq i \leq s} q_i g_i$$
 mit $\mathsf{mdeg}(f) \succeq \mathsf{mdeg}(q_i g_i)$ $i = 1, \dots, s$

6. (Ideal Membership) für alle $f \in I$ gilt $f \xrightarrow{*}_{G} 0$ oder $f \operatorname{rem}(G) = 0$.

Buchberger's Algorithmus

Beispiele von GB

8.45 Beispiel
$$\{f_1, ..., f_5\}$$
, Beispiel ist Gröbner Basis $x^3 - 2xy$, $x^2y - 2y^2 + x$, $-x^2$, $-2xy$, $-2y^2 + x$

 $\hookrightarrow \{x^2, xy, y^2 - \frac{1}{2}x\}$ ist minimale reduzierte Gröbner Basis, sie ist eindeutig bzgl. \prec_{grlex} .

Beachte: Bzgl. einer anderen Ordnung kann die *G*-Basis mehr oder weniger Polynome enthalten.

z.B.
$$x \succ y : \{x - 2y^2, y^3\}$$

 $y \succ x : \{y^2 - \frac{1}{2}x, yx, x^2\}$

Beachte: Termideale, Monomideale und homogene Ideale haben reduzierte Gröbner Basen, die unabhängig von der Ordnung sind.

Prof. Dr. K. Madlener: Computeralgebra

Polynomideale und Gröbner-Basen

Buchberger's Alzorithmus

Buchberger's Algorithmi

Homogene Ideale

Prof. Dr. K. Madlener: Computeralgebra

Ein Polynom $f = \sum_{i=1}^{n} \alpha_e x^e$ ist homogen, falls alle Terme x^e gleichen Grad haben, d. h. $\sum_{i=1}^{n} e_i = k :: f$ ist dann homogen vom Grad k.

- ▶ Jedes Polynom lässt sich eindeutig als Summe homogener Polynome (die homogenen Komponenten) darstellen.
- ▶ *I* ist homogen, falls $I = \langle G \rangle$, *G* enthält homogene Polynome $gdw \ \forall f \in I$ jede homogene Komponente von *f* liegt in *I*.

Anwendungen von Gröbner Basen

1. Berechnungen in Quotientenringen: $K[x_1, \ldots, x_n]/I$

$$a \in R \mapsto \bar{a} = a + I \in S = R/I$$

- ā Restklasse von a.
- ▶ Ist G eine reduzierte G-Basis von I, so kann man als Repräsentant von a die Normalform von a bzgl. \xrightarrow{G} wählen, d. h. $rep_G(a) = a$ rem (G).

Offenbar

$$\bar{a} = \bar{b} \text{ gdw } a - b \in I \text{ gdw } rep_G(a - b) = 0 \text{ gdw } rep_G(a) = rep_G(b)$$

Arithmetik in s:

$$\bar{a} \pm \bar{b} \leftrightarrow rep_G(a) \pm rep_g(b)$$

 $\bar{a} \cdot \bar{b} \leftrightarrow rep_G(rep_G(a) \cdot rep_G(b))$

◆ロト ◆御 ト ◆ 恵 ト ◆ 恵 ・ 夕 ♀ (

Prof. Dr. K. Madlener: Computeralgebra
Polynomideale und Gröbner-Basen

Anwendungen von Gröbner Basei

Anwendungen von Gröbner Basen (Forts.)

8.46 Lemma Sei $I \subseteq R = K[x_1, \dots, x_n]$, G Gröbnerbasis von I.

- i) V = R/I ist K-Vektorraum.
- ii) $B = \{t \in T[X] : t = rep_G(t)\}$. Dann ist $\bar{B} = \{\bar{t} : t \in B\}$ eine K-Vektorraumbasis von V.
- iii) V hat endliche K-Dimension $\operatorname{gdw} I$ nulldimensional.
- ▶ Eine Teilmenge $S \subseteq R$ ist nulldimensional, falls $Zero(S) \subseteq \overline{K}^n$, $(\overline{K} \text{ alg. Abschluss von } K)$ endlich ist
- ▶ hier $Var_{\bar{K}}(I)$ endlich. (S Ideal)

Anwendungen von Gröbner Basen (Forts.)

8.47 Satz $I \subseteq K[x_1, \dots, x_n]$. *I* ist nulldimensional gdw für jedes $i = 1, \dots, n$ $I \cap K[x_i] \neq \{0\}$.

Beweis:

- "\(= " \) Sei $0 \neq f_i \in I \cap K[x_i]$ i = 1, ..., n. Sei $\xi \in Var(I)$, ξ muss Nullstelle von f_i sein, d. h. ξ_i muss Nullstelle von f_i sein und davon gibt es nur endlich viele. $\rightsquigarrow Var(I)$ ist endlich.
- "⇒" angenommen $\operatorname{Var}_{\bar{K}}(I) = \{\xi^1, \dots, \xi^m\}$ mit $\xi^j = (\xi^j_1, \dots, \xi^j_n)$. Da $\xi^j_i \in \bar{K}$, gibt es Polynome $f_i \in K[x_i]$ mit $f_i(\xi^j_i) = 0$ für $j = 1, \dots, m$, d. h. f_i ist null auf $\operatorname{Var}(I)$.

 Nach Hilbert's Nullstellensatz gibt es $m_i \in \mathbb{N}^+$ mit $f_i^{m_i} \in I$, d. h.

Prof. Dr. K. Madlener: Computeralgebra

 $I \cap K[x_i] \neq \{0\}.$

Polynomideale und Gröbner-Basen

Anwendungen von Gröbner Basen

Anwendungen von Gröbner Basen (Forts.)

8.48 Folgerung Sei *G* eine Gröbner Basis für *I*.

I ist nulldimensional gdw für jedes $i=1,\ldots,n$ es ein $g_i\in G$ gibt mit $LT(g_i)\in T[x_i]$ gdw V=R/I hat endliche K-Dimension.

Beweis: $\langle LT(G) \rangle = \langle LT(I) \rangle$, da Gröbner Basis. Für $i = 1, \dots, k$: $I \cap K[x_i] \neq \{0\}$ $\text{gdw } \langle LT(I) \rangle \cap K[x_i] \neq \{0\}$ $\text{gdw } LT(G) \cap K[x_i] \neq \emptyset$ $\text{gdw } B = \{t \in T[x] : t = rep_G(t)\}$ ist endlich.

Bedingung: " \Rightarrow " klar, da x_i^k , $k > k_i$ reduzibel.

" \Leftarrow " Ist I nicht nulldimensional, so gibt es eine Variable x_i , so dass $x_i^I \in B$ für alle $I \geq 0$, d. h. B ist unendlich und V ist ∞ -dimensional als K-Vektorraum.

Beispiele

- **8.49 Beispiel** $I = \langle x^3y 2y^2 1, x^2y^2 + x + y \rangle$ in $\mathbb{Q}[x, y]$.
- $\prec = \prec_{grlex}$ mit $x \succ y$. $GB(I, \prec) = \{ \underbrace{x^4 + x^2y - 2xy + 1, \underline{x^3y} - 2y^2 - 1, \underline{x^2y^2} + x + y,}_{x^2 + xy + 2y^3 + y} \}$
- $\prec = \prec_{lex}$ mit $y \succ x$. $GB(I, \prec) = \{ 34y + 8x^8 - x^7 - 2x^6 - 4x^5 + 7x^3 - 35x^2 + 30x - 8, \}$ $x^9 + x^5 + 3x^4 - 4x^3 + 4x^2 + 2$
- ► Dimension als **O-Vektorraum**:
 - 1) $\{x^3, x^2, x, x^2y, xy, y^2, y, xy^2, 1\}$ irr. Terme 2) $\{x^8, x^7, x^6, x^5, x^4, x^3, x^2, x, 1\}$ irr. Terme.

 - d.h. $DIM_{\mathbb{Q}}(\mathbb{Q}[x,y]/I) = 9$
 - Übung: Berechne Polynome in $I \cap \mathbb{Q}[x]$ und $I \cap \mathbb{Q}[y]$.



< ロ ト → 付き → → き ト → ま → り Q (~)

Prof. Dr. K. Madlener: Computeralgebra

Polynomideale und Gröbner-Basen

Anwendungen von Gröbner Basei

Beispiele

▶ Man kann auch eine Multiplikationstafel für B berechnen: für $p, q \in B$ berechne Normalform von pq (ist dann Polynom), z. B. oben $x \cdot x^3 = -x^2 y + 2xy - 1$

Wortproblem für / bzw. \equiv_I Idealmembership. Problem für e.e. Ideale $I \subseteq R = K[x_1, \dots, x_n]$

Gegeben: $f, f_1, \ldots, f_m \in R$.

Frage: $f \in \langle f_1, \dots, f_m \rangle$ ggf. Darstellung.

Lösung: Berechne G-Basis $G = \{g_1, \dots, g_s\}$ für I, wobei die Darstellungen von $g_i = \sum_{j=1}^m q_j^i f_j$ berechnet werden. $f \in \langle f_1, \dots, f_m \rangle$ gdw f res (G) = 0 + Darstellung in $g_i \rightsquigarrow D$ arstellung in f_i berechnen.

Beispiele (Forts.)

8.50 Beispiel Folgende Polynomrelationen seien gegeben.

$$f_1 = 4z - 4xy^2 - 16x^2 - 1 = 0$$

$$I :: f_2 = 2y^2z + 4x + 1 = 0 \quad f_i \in \mathbb{R}[x, y, z]$$

$$f_3 = 2x^2z + 2y^2 + x = 0$$

Folgt hieraus

$$g(x, y) = 4xy^2 + 16x^2y^2 + y^2 + 8x + 2 = 0$$

Für (x, y) mit $\exists z : (x, y, z) \in Var(I)$?

▶ Ist $g \in I = \langle f_1, f_2, f_3 \rangle$? g ist irreduzibel bzgl. $\{f_1, f_2, f_3\}$

Polynomideale und Gröhner-Rasen

Anwendungen von Gröbner Baser

Beispiele (Forts.)

Prof. Dr. K. Madlener: Computeralgebra

▶ Berechne Gröbner Basis bzgl. lex. Ordnung x < y < z liefert:

$$\begin{array}{ll} g_1 = & 32x^7 - 216x^6 + 34x^4 - 12x^3 - x^2 + 30x + 8 \\ g_2 = & 2745y^2 - 112x^6 - 812x^5 + 10592x^4 - 61x^3 - 812x^2 + 988x + 2 \end{array}$$

- $g_3 = 4z 4xy^2 16x^2 1$
- → Var(I) ist endlich.

$$g \xrightarrow{*}_{G} 0$$
 , d.h. $g(x,y) = 0$ auf Var. *I*.

▶ Lösung von Systemen von Polynomgleichungen

$$V = \{ a \in K^n : f_1(a) = \cdots = f_m(a) = 0 \}$$

Fragen: $V \neq \emptyset$? Gegebenenfalls finde $a \in V$. $g \in K[x_1, \dots, x_n]$, ist g(a) = 0 für alle $a \in V$.

→ Hilbert's Nullstellensatz, Radikal, Eliminationsmethoden.

Eliminationseigenschaften - Polynomgleichungen (Syzygien)

Eliminationseigenschaften - Polynomgleichungen

8.51 Definition Sei $I \subseteq K[x_1, \dots, x_n] = R$, radikal(I) ist ideal in R mit:

 $f \in \operatorname{radikal}(I) \operatorname{gdw} f^n \in I \operatorname{für ein} n \in \mathbb{N}^+$

Schreibweise Lit: \sqrt{I} (Übung \sqrt{I} ist Ideal). Motivation: Sei $\mathbb{Z} \subseteq R_0 \subseteq R_1 \subseteq \mathbb{C}$ (oder alg. abg. Körper).

- ▶ $R_1^d = \mathbb{A}^d(R_1)$ d-dimensionaler affiner Raum von R_1 .
- ▶ $U \subseteq \mathbb{A}^d(R_1)$, $f \in R_0[x_1, \dots, x_d]$, f verschwindet auf U, falls f(a) = 0 für alle $a \in U$.
- Ideal(U) $\subseteq R_0[x_1, \dots, x_d]$ sei definiert durch Ideal(U) = $\{f \in R_0[x_1, \dots, x_d] \mid f \text{ verschwindet auf } U\}$ ist Ideal!.
- ► $I = \langle f_1, \dots, f_n \rangle$, $f_i \in R_0[x_1, \dots, x_d]$ $\mathsf{Zero}_{R_1}(I) = \mathsf{Var}_{R_1}(I) = \{ a \in \mathbb{A}^d(R_1) : f_i(a) = 0, i = 1, \dots, n \}$

|ロト 4回 ト 4 E ト 4 E ト 9 Q (や

Prof. Dr. K. Madlener: Computeralgebra

565

Polynomideale und Gröbner-Basen

Elizabeth strain and the Delarant interaction (Courts)

Eliminationseigenschaften - Polynomgleichungen (Syzygien)

Eliminationseigenschaften - Polynomgleichungen (Forts.)

- $U \mapsto \mathsf{Ideal}(U)$ für $U \subseteq \mathbb{A}^d(R_1)$
- $I \mapsto \mathsf{Zero}_{R_1}(I)$ für $I = \langle f_1, \dots, f_n \rangle \subseteq R_0[x_1, \dots, x_d]$

Algebraische Teilmengen von $\mathbb{A}^d(R_1)$ (Zariski Topologie)

- ► Es gilt: für $I \subseteq R_0[x_1, ..., x_d]$ und $U \subseteq \mathbb{A}^d(R_1)$.
 - $\quad \blacktriangleright \ \, \textit{I} \subseteq \mathsf{Ideal}[\mathsf{Zero}_{\textit{R}_1}(\textit{I})]$
 - ▶ $U \subseteq \mathsf{Zero}_{R_1}[\mathsf{Ideal}(U)]$
 - ▶ Zero[Ideal[Zero(I)]] = Zero(I) $I \subseteq R_0[x_1, ..., x_d]$
 - ▶ Ideal[Zero[Ideal(U)]] = Ideal(U) $U \subseteq \mathbb{A}^d(R_1)$
- Schränkt man die Abbildungen auf ideale und alg. Mengen, sind sie dann invers zueinander?

Nur für ideale, die radikal sind, d. h. $f^n \in I$, $n \ge 1 \leadsto f \in I$. Da Ideal(U) stets radikal ist.

Hilbert's Nullstellensatz

- ▶ Hilberts Nullstellensatz (schwache Form) Sei D noetherscher ZPE-Ring, \bar{D} alg. Abschluss. Ein Ideal $I \subseteq D[x_1, \ldots, x_d]$ hat keine Nullstellen in $\mathbb{A}^d(\bar{D})$ gdw I enthält nichtriviales Element von D.
- ▶ Hilberts Nullstellensatz (starke Form) Sei D wie oben. $I \subseteq D[x_1, ..., x_d]$ Ideal und $f \in D[x_1, ..., x_d]$. f verschwindet auf $Var_{\bar{D}}(I)$ gdw es gibt $m \ge 0$, $0 \ne a \in D$ mit $a \cdot f^m \in I$.
- \hookrightarrow d.h. $f \in \sqrt{I}$ (Körperfall).
- $\hookrightarrow f_1(\bar{x}) = 0 \land \dots \land f_m(\bar{x}) = 0 \Rightarrow f(\bar{x}) = 0$ $\leadsto f \in \sqrt{\langle f_1, \dots, f_m \rangle}$

(ロ) (部) (目) (目) (目) (9)

Prof. Dr. K. Madlener: Computeralgebra

507

Polynomideale und Gröbner-Basen

reliablished and Published and Constitution

Eliminationseigenschaften - Polynomgleichungen (Syzygien

Hilbert's Nullstellensatz: Motivation

- Offenbar gilt:
 - ▶ Starke Form → schwache Form.
 - Schwache Form \leadsto starke Form: Sei $f \in D[x_1, \dots, x_d]$, f verschwindet auf $\text{Var}_{\bar{D}}(I)$. Sei $I = \langle f_1, \dots, f_m \rangle$ "Rabinowitz-Trick" neue Var. z: Setze $g := 1 - z \cdot f$, dann hat das Ideal $\langle f_1, \dots, f_m, g \rangle$ keine Nullstellen, da g nicht null an den Nullstellen von $f_1, \dots, f_m \leadsto$ es gibt $0 \neq a \in D \cap \langle f_1, \dots, f_m, g \rangle$

$$a = \sum_{i=1}^{m} \alpha_i f_i + \beta (1 - zf) \qquad \alpha_i, \beta \in D[x_1, \dots, x_d, z]$$

- ▶ Setze $z = 1/f \rightsquigarrow a = \sum_{i=1}^{m} \alpha'_i f_i \text{ mit } \alpha'_i \in D(x_1, \dots, x_d).$ Rationale Funktionen mit Nenner Potenz von $f \rightsquigarrow$
- $ightharpoonup a \cdot f^n = \sum (\alpha'_i f^{m_i}) f_i \text{ mit } \alpha'_i f^{m_i} \in D[x_1, \dots, x_d].$

(□▶ ◀♬▶ ◀불▶ ◀불▶ _ 亳 _ 쒸९♂

Prof. Dr. K. Madlener: Computeralgebra 566 Prof. Dr. K. Madlener: Computeralgebra

Eliminationseigenschaften - Polynomgleichungen (Syzygien)

Anwendungen von Hilberts's Nullstellensatz

Anwendung: Radikal-Membership-Problem

Seien
$$f_1, \ldots, f_m \in K[x_1, \ldots, x_d]$$
, $f \in K[x_1, \ldots, x_d]$
Frage: Gilt $f \in \sqrt{\langle f_1, \ldots, f_m \rangle}$?
$$f \in \sqrt{\langle f_1, \ldots, f_m \rangle} \quad \text{gdw} \quad f \text{ verschwindet auf } \text{Var}_K(\langle f_1, \ldots, f_m \rangle)$$

$$\text{gdw} \quad f_1(\bar{x}) = f_1(\bar{x}) = \cdots = (1 - z \cdot f)(\bar{x}) = 0$$

$$\text{hat keine Lösungen in } \bar{K}^d$$

$$\text{gdw} \quad 1 \in \langle f_1, \ldots, f_m, 1 - zf \rangle_{K[x_1, \ldots, x_d, z]}$$

$$\text{d.h. } f \in \sqrt{\langle f_1, \ldots, f_m \rangle} \quad \text{gdw} \quad 1 \in GB(f_1, \ldots, f_m, 1 - zf)$$

► Frage: Gilt $Var_{\bar{K}}(\langle f_1, \dots, f_m \rangle) = \emptyset$ Lösung: Ja, falls $1 \in GB(f_1, \dots, f_m)$.

(ロ) (B) (토) (토) (B) (C)

Prof. Dr. K. Madlener: Computeralgebra

Polynomideale und Gröbner-Basen

Eliminationseigenschaften - Polynomgleichungen (Syzygien)

Anwendungen: Eliminationseigenschaften, Varietäten

- ▶ Frage: Gilt $\operatorname{Var}_{\bar{K}}(\langle f_1, \dots, f_m \rangle)$ ist endlich, $I = \langle f_1, \dots, f_m \rangle$ ist nulldimensional. Lösung: Berechne GB und K-Dimension von $K[x_1, \dots, x_d]/I$. Angenommen $\operatorname{Var}_{\bar{K}}(\langle f_1, \dots, f_m \rangle)$ sei endlich:
- ► Frage: Kann man die Lösungen explizit darstellen?

8.52 Satz Eliminationseigenschaft für Gröbner Basen. Sei G Gröbner Basis von $I=\langle f_1,\ldots,f_m\rangle$ bzgl. der Lex-Ordnung mit $x_1< x_2<\cdots< x_n$. Dann gilt

$$J = I \cap K[x_1, \ldots, x_i] = \langle G \cap K[x_1, \ldots, x_i] \rangle_{K[x_1, \ldots, x_i]}$$

Insbesondere ist $G \cap K[x_1, ..., x_i]$ eine G-Basis für J.

Polynomideale und Gröbner-Basen

Eliminationseigenschaften - Polynomgleichungen (Syzygien

Eliminationseigenschaften: Beweis

Beweis:

"⊃" klar.

" \subseteq " $LT(f) \in T[x_1, \ldots, x_i]$ gdw $f \in K[x_1, \ldots, x_i]$, $f \in I \cap K[x_1, \ldots, x_i] \leadsto f \stackrel{*}{\longrightarrow} 0$ und die Leitterme der Basispolynome bei der Reduktion liegen in $T[x_1, \ldots, x_i]$, d. h. diese Basispolynome liegen in $G \cap K[x_1, \ldots, x_i]$.

8.53 Folgerung Sei *G G*-Basis von *I* bzgl. der Lex Ordnung mit $x_1 < x_2 < \cdots < x_n$.

I ist nulldimensional gdw für jedes i = 1, ..., n gibt es ein $g_i \in G$ mit $HT(g_i) \in K[x_1, ..., x_i] \setminus K[x_1, ..., x_{i-1}].$

(日) (部) (目) (目) (単) の(9)

Prof. Dr. K. Madlener: Computeralgebra

Polynomideale und Gröbner-Basen

Eliminationseigenschaften - Polynomgleichungen (Syzygien

Eliminationseigenschaften: Beweis (Forts.)

Beweis:

"
$$\curvearrowright$$
" klar
" \curvearrowright " $\xi = (\xi_1, \dots, \xi_n) \in \mathsf{Var}_{\bar{K}}(I)$. So $g_i(\xi_1, \dots, \xi_i) = 0$ $i = 1, \dots, n$.

- i=1 g_1 Polynom in einer Var. \leadsto endlich viele Nullstellen, d.h. nur endlich viele Möglichkeiten für ξ_1 .
- Induktiv \leadsto $g_i(\xi_1,\ldots,\xi_{i-1},\xi_i)=0$ für ξ_1,\ldots,ξ_{i-1} nur endlich viele Möglichkeiten \leadsto für ξ_i nur endlich viele Möglichkeiten.



Eliminationseigenschaften - Polynomgleichungen (Syzygien)

Anwendung: Polynomgleichungen

8.54 Beispiel Pol. Gleichungssystem

$$f_1 :: 4xz -4xy^2 -16x^2 -1 = 0$$

 $f_2 :: 2y^2z +4x +1 = 0$ $\mathbb{Q}[x, y, z]$
 $f_3 :: 2x^2z +2y^2 +x = 0$ $x < y < z$

Gröbner Basis bzgl. lex. Ordnung:

$$g_1 = 65z +64x^4 -423x^3 +168x^2 -354x +104$$

 $g_2 = 26y^2 -16x^4 +108x^3 -16x^2 +17x$
 $g_3 = 32x^5 -216x^4 +64x^3 -42x^2 +32x +5$



4□ > 4問 > 4 = > 4 = > 9 Q ○

Prof. Dr. K. Madlener: Computeralgebra

573

Polynomideale und Gröbner-Basen

Eliminationseigenschaften - Polynomefleichungen (Syzveien)

Anwendung: Polynomgleichungen (Forts.)

- $\hookrightarrow \mathsf{Var}_{\mathbb{C}}(\langle f_1, f_2, f_3 \rangle)$ ist endlich, $\mathsf{DIM}_{K}(\mathbb{Q}[x, y, z]/I) = 10$, d. h. $|\mathsf{Zero}(f_1, f_2, f_3)| = 10$ (Nullstellen mit Vielfachheit zählen).
- ▶ Lösungen von g_3 als Parameter: α (5-Nullstellen)

$$(\alpha, \pm \frac{1}{\sqrt{26}} \sqrt{\alpha} \sqrt{16\alpha^3 - 108\alpha^2 + 16\alpha - 17}, \frac{1}{65} (64\alpha^4 - 423\alpha^3 + \cdots))$$

 \hookrightarrow Darstellung von $Var_{\mathbb{C}}(\langle f_1, f_2, f_3 \rangle)$.

Beispiel: Polynomgleichungen

8.55 Beispiel Parametrisiertes Gleichungssystem.

$$f_1 := x_4 + b - d = 0$$

$$f_2 := x_4 + x_3 + x_2 + x_1 - a - c - d = 0$$

$$f_3 := x_3x_4 + x_1x_4 + x_1x_3 - ad - ac - cd = 0$$

$$f_4 := x_1x_3x_4 - acd = 0$$

• Parameter: $a, b, c, d \in K$, $<:: x_1 < x_2 < x_3 < x_4$

G-Basis bzgl. lex Ordnung: $\mathbb{Q}(a, b, c, d)[x_1, x_2, x_3, x_4]$

$$g_1 = x_4 + b - d$$

$$g_2 = x_3 - \frac{b^2 - 2bd + d^2}{acd}x_1^2 - \frac{abc + abd - 2acd - ad^2 + bcd - cd^2}{acd}x_1 - a - c - d$$

$$g_3 = x_2 + \cdots$$
 $g_4 = x_1^3 + \cdots$

Eine Wurzel von g_4 ist $\frac{-ad}{b-d}$ (?)

$$\leadsto \left(\frac{-ad}{bd}, \frac{ab+b^2-bd}{b-d}, c, -b+d \right)$$

► Anwendungen: Schaltungsentwurf: Verstärker usw.

◆□ ▶ ◆昼 ▶ ◆ 豊 ▶ ・ 豊 ・ 夕 ♀

Prof. Dr. K. Madlener: Computeralgebra

575

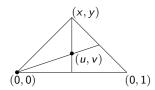
Polynomideale und Gröbner-Basen

Eliminationseigenschaften - Polynomgleichungen (Syzygien

Automatisches Beweisen in der Geometrie

8.56 Beispiel Dreieck, Schnitt der Medianen, Formulierung:

$$f_1, f_2 \in \mathbb{R}[u, v, x, y]$$



- $f_1 = 0 \land f_2 = 0 \Rightarrow g_1 = 0 \land g_2 = 0 \land g_3 = 0$,
- falls $g_1, g_2, g_3 \in \langle f_1, f_2 \rangle$, so ok. g_1 bereits gezeigt.
- ► $GB(\langle f_1, f_2 \rangle)$ $u \succ v \succ x \succ y$ lex Ordnung. $f_1 = uy - vx - v$ $f_2 = uy - vx + 2v - v$
- $\hookrightarrow S(f_1, f_2) = f_1 f_2 = -3v + y = -g_3$ $\{f_1, f_2, g_3\}$ ist Gröbner Basis.

Automatisches Beweisen in der Geometrie (Forts.)

▶ Die eindeutige reduzierte *G*-Basis ist

$$G = \{uy - \frac{1}{3}xy - \frac{1}{3}y, v - \frac{1}{3}y\}$$

$$\begin{array}{lll} g_1 = & -2uy - (v - y) + 2vx & \xrightarrow{*} & 0 \\ g_2 = & 3u - x - 1 & \text{irreduzibel d. h. } g_2 \not\in \langle G \rangle \\ g_3 = & 3v - y & \xrightarrow{*} & 0 \end{array}$$

- Beachte aber $yg_2 = 3uy xy y \longrightarrow 0$, d. h. $yg_2 \in I$
- \hookrightarrow d.h. $g_2(x,y)=0$, falls $(x,y)\in V(I)$ und $y\neq 0$ nicht Degeneriertheitsbedingung $y \neq 0$.
- ightharpoonup Nimmt man 1 yz zu G hinzu, z neue Variablen garantiert. $y \neq 0$: $g_2 = g_2 \cdot (1 - yz) + zyg_2 \in \langle f_1, f_2, 1 - yz \rangle$

Prof. Dr. K. Madlener: Computeralgebra

Polynomideale und Gröbner-Basen

Eliminationseigenschaften - Polynomgleichungen (Syzygien

Implizitierung (Implicitation)

▶ Seien $f_1, \ldots, f_n \in K[t_1, \ldots, t_m]$ und eine affine Alg. Varietät $V \subseteq K^n$ sei in parametrisierte Form gegeben, d.h.

$$x_1 = f_1(t_1, \dots, t_m)$$

$$\vdots$$

$$x_n = f_n(t_1, \dots, t_m)$$

- $\hookrightarrow V = \{a \in K^n : \exists b \in K^m \mid a = (f_1(b), \dots, f_n(b))\}$
- ▶ Finde Polynome $g_1, \ldots, g_s \in K[x_1, \ldots, x_n]$, so dass V = Var(I) mit $I = \langle g_1, \dots, g_s \rangle$ "implizite Darstellung".

8.57 Beispiel

- Twisted Cubic: x = t $v = t^2$ $z = t^3$ Implizitierung: $g_1 = y - x^2$ $g_2 = z - x^3$
- $x = t^2$, $y = t^3$, $z = t^4$ Implizitierung: $g_1 = z - x^2$ $g_2 = v^2 - x^3$

< ロ ト → 付き → → き ト → ま → り Q (~)

Implizitierung: Lösungsansatz

Lösung mit *G*-Basen:

Eliminationseigenschaften - Polynomgleichungen (Syzygien)

- ▶ Betrachte $J = \langle x_1 f_1, \dots, x_n f_n \rangle \subseteq K[t_1, \dots, t_m, x_1, \dots, x_n]$. Wähle Ordnung $t_1 \succ \cdots \succ t_m \succ x_1 \succ \cdots \succ x_n \quad \prec_{lex}$
- \hookrightarrow Einige der g in GB(J) hängen nur von x_1, \ldots, x_n ab, dies sind Kandidaten für die Implizitierung.

8.58 Beispiel

- $t \succ z \succ v \succ x$
- ► $GB\{x-t, y-t^2, z-t_3\}$ ist $\{t-x, z-x^3, y-x^2\}$
- ► $GB(x-t^2, v-t^3, z-t^4)$ ist $\{t^2-x, ty-x^2, tx-y, z-x^2, y^2-x^3\}$
- ▶ Die Varietät, die von $G \cap K[x_1, ..., x_n]$ definiert wird, ist die kleinste Varietät (Alg-Menge), die das Bild der Parametrisierung enthält.

Prof. Dr. K. Madlener: Computeralgebra

Polynomideale und Gröhner-Basen

Eliminationseigenschaften - Polynomgleichungen (Syzygien

Lösung linearer Gleichungen in $K[\bar{x}]$:: Syzygien

Lösung linearer Gleichungen in $K[\bar{x}]$

Gegeben:
$$f_1, \ldots, f_s, f \in K[\bar{x}] = R$$
.
Gesucht: Lösungen von $f_1z_1 + \cdots + f_sz_s = f$
bzw. $f_1z_1 + \cdots + f_sz_s = 0$ (*)
mit $z_i \in K[\bar{x}]$.

- ▶ Jede Lösung von (*) heißt eine Syzygie von f_1, \ldots, f_s .
- ▶ Beachte die Menge der Lösungen von (*) ist ein R-Modul, hat eine endliche Basis.
- ▶ Gesucht wird eine Modul-Basis für $syz(\{f_1, ..., f_s\})$.



Basis für Syszygienmodul für Gröbner-Basen

8.59 Satz Basis für Syszygienmodul für Gröbner-Basen. Sei $G = \{f_1, \dots, f_s\}$ Gröbner-Basis. Eine Basis S für syz(G) erhält man wie folgt:

- ▶ Für $1 \le i \le s$ sei $e_i = (0, ..., 0, 1, 0, ..., 0) \in \mathbb{R}^s$ *i*-te Einheitsvektor.
- Für $1 \le i < j \le s$ $t_{ij} = KGV(LT(f_i), LT(f_j))$ $p_{ij} = \frac{t_{ij}}{LM(f_i)}$ $q_{ij} = \frac{t_{ij}}{LM(f_j)}$
 - $S(f_i, f_j) = p_{ij}f_i q_{ij}f_j \xrightarrow{*}_{G} 0$ $= \sum_{l=1}^{s} k_{ij}^{l}f_l \text{ mit } k_{ij}^{l} \in R$
 - $S = \{ p_{ij}e_i q_{ij}e_j (k_{ii}^1, \dots, k_{ii}^s) \mid 1 \le i < j \le s \}$

Prof. Dr. K. Madlener: Computeralgebra

58

Polynomideale und Gröbner-Basen

Eliminationseigenschaften - Polynomgleichungen (Syzygien)

Basis für Syszygienmodul für Gröbner-Basen (Forts.)

Beweis:

Prof. Dr. K. Madlener: Computeralgebra

• Jedes Element in S ist eine Syzygie von G. Sei $s_{ij} = p_{ij}e_i - q_{ij}e_j - (k_{ii}^1 \cdots k_{ii}^s)$ als Zeilenvektor.

$$\hookrightarrow s_{ij} \cdot \begin{pmatrix} f_1 \\ \vdots \\ f_s \end{pmatrix} = p_{ij}f_i - q_{ij}f_j - \sum_{l=1}^s k_{ij}^l f_l = 0$$

▶ Sei $z = (z_1, \ldots, z_s) \in R^s$ nichttriviale Syzygie von G, d. h. $\sum z_i f_i = 0$, und p maximaler Term in dieser Summe. Lemma 8.35 liefert das Ergebnis: Durch Abziehen geeigneter Vielfacher von $s_{i,j}$ von z lässt sich Summe mit kleinerem p erreichen, d. h. z ist Linearkombination der s_{ij} und somit bildet S eine Basis für syz(G).

Basis für Syszygienmodul für Gröbner-Basen (Forts.)

8.60 Satz Sei $F = (f_1, \ldots, f_s)^T$ mit $f_i \in K[\bar{x}]$ und $G = (g_1, \ldots, g_m)^T$ eine Gröbner-Basis für $\langle F \rangle$. (Betrachte F, G als Spaltenvektoren aus R^s bzw. R^m).

▶ Die Matrix $R_{r \times m}$ bestehe aus r Zeilen, die eine Basis für syz(G) bilden (S von 8.59). Weiterhin seien die Matrizen A, B definiert durch $G = A_{m \times s} F$ bzw. $F = B_{s \times m} G$ (Darstellungen der g_i in den f_i und umgekehrt.)

Sei
$$Q:=\left(egin{array}{c} I_s-B\cdot A \ R\cdot A \end{array}
ight)_{s+r,s}$$

Dann bilden die Zeilen von Q eine Basis für syz(F).

∢ロト∢御ト∢恵ト∢恵ト 連 めの

Prof. Dr. K. Madlener: Computeralgebra

E0

Polynomideale und Gröbner-Basen

Eliminationseigenschaften - Polynomgleichungen (Syzygien

Beweis

Beweis Seien
$$b_1, \ldots, b_{s+r}$$
 Polynome $b = (b_1 \ldots b_{s+r})$.

$$(b \cdot Q)F = ((b_1, \ldots, b_s)(I_s - BA) + (b_{s+1} \cdots b_{s+r})RA)F$$

$$= (b_1 \cdots b_s)(F - \underbrace{BAF}_{=F}) + (b_{s+1} \cdots b_{s+r})R\underbrace{AF}_{=G}$$

$$= 0$$

- \hookrightarrow d.h. Jede Linearkombination der Zeilen von Q ist eine Syzygie von F.
- Sei $H=(h_1\dots h_s)$ eine Syzygie von F. Dann ist $H\cdot B$ eine Syzygie von G. Für ein H' gilt dann $H\cdot B=H'\cdot R$ und somit $H\cdot B\cdot A=H'\cdot R\cdot A$, d. h. $H=H\cdot (I_S-BA)+H'\cdot R\cdot A=(H,H')Q$, also ist H Linearkombination der Zeilen von Q.

Lösung inhomogener Gleichungen

Prof. Dr. K. Madlener: Computeralgebra

▶ $f_1z_1 + \cdots + f_sz_s = f$ Existenz gdw $f \in \langle F \rangle$ Gröbner Basis für F, $G = A \cdot F$, $f \xrightarrow{*}_{G} f' \neq 0$ nicht lösbar, sonst $f \xrightarrow{*}_{G} 0$, $g_1h'_1 + \cdots + g_mh'_m = f \hookrightarrow H = (h'_1 \cdots h'_m)A$ ist partikuläre Lösung.

Polynomideale und Gröbner-Basen

Zur Komplexität der Berechnung von G-Baser

Effektive Operationen mit Idealen

Seien $I = \langle f_1, ..., f_r \rangle$ und $J = \langle g_1, ..., g_s \rangle$ Ideale in K[X]

- ▶ $I + J := \{f + g : f \in I, g \in J\} = \langle f_1, ..., f_r, g_1, ..., g_s \rangle$
- $I \cdot J := \{ fg : f \in I, g \in J \} = \langle f_i g_i : 1 \le i \le r, 1 \le j \le s \rangle$
- $I \cap J := \{ f : f \in I \text{ und } f \in J \} = (\langle t \rangle I + \langle 1 t \rangle J) \cap K[X]$
- ▶ $I: J := \{f : fg \in I, \forall g \in J\} = \cap_{j=i,...,s} (I: \langle g_j \rangle)$ wobei $I: \langle g \rangle = \langle h_1/g, ..., h_m/g \rangle$ mit $I \cap \langle g \rangle = \langle h_1, ..., h_m \rangle$
- ► Transformation von G-Basen bzgl. verschiedener Termordnungen (Lazard)
- ► GGT-Berechnung mit G-Basen (Gianni, Trager)



4日 > 4間 > 4目 > 4目 > 1目

Prof. Dr. K. Madlener: Computeralgebra

58

Polynomideale und Gröbner-Basen

Zur Komplexität der Berechnung von G-Basen

Zur Komplexität der Berechnung von G-Basen

Probleme:

- ▶ Ordnungen, Längen von Ketten bei Reduktion.
- ▶ Wachstum der Größen bei der Berechnung: Eingabe weniger Polynome, kleine Grade, kleine Koeffizienten: Ausgabe Polynome mit großen Graden, große Koeffizienten.
 - D. h. Ergebnisse können groß werden.
 - ► Klassen *P*, *BPP*, *NP*, EXPSPACE
- EXPSPACE-vollständige Probleme benötigen $2^{2^{O(n)}}$ Zeit. IM (Wortproblem für Ideale über $\mathbb{Q}[x_1,\ldots,x_s]$.)
- Mayr & Mayer 82: IM ist EXPSPACE-hart für allg. Ideale.
- Mayr 89,92: IM ist in EXPSPACE, dh. IM ist vollständig.

Zur Komplexität der Berechnung von G-Basen (Forts.)

- $f \xrightarrow{*}_{GB} 0$ kann in EXPSPACE berechnet werden (für *G*-Basen).
- Entscheidungsproblem: Ist $\{f_1, \dots, f_m\}$ Gröbner Basis ist EXPSPACE-hart.

8.61 Satz Kühnle, Mayr 96 Die Berechnung einer reduzierten *G*-Basis kann in EXPSPACE erfolgen. (Beachte EXPSPACE = DSPACE(2^{lin}) wird nur Platz auf Arbeitsband

▶ Gleiche Ergebnisse gelten für binomial-ideale, d. h. Ideale werden durch Binome $x^{\alpha} - x^{\beta}$ erzeugt.

∢ロト∢御ト∢恵ト∢恵ト 連 めの

Prof. Dr. K. Madlener: Computeralgebra

gemessen).

58

Polynomideale und Gröbner-Basen

Zur Komplexität der Berechnung von G-Baser

Zur Komplexität der Berechnung von G-Basen (Forts.)

- ▶ Bürgisser (98) K ∞-Körper. IM benötigt exponentielle parallele Zeit.
- ► Für homogene Ideale: Mayr 95: IM ist PSPACE-vollständig. Berechnung der *G*-Basis bleibt EXPSPACE-hart.

Gradschranken

- ▶ Hermann 1926: $f \in \langle f_1, \dots, f_s \rangle$, $f = \sum_{1 \leq i \leq s} q_i f_i$ Grade der q_i doppeltexponentiell. Siehe auch Mayr & Mayer 82.
- ▶ Die Grade der Polynome in einer reduzierten Gröbner Basis für $\langle f_1, \ldots, f_s \rangle \subseteq F[x_1, \ldots, x_n]$ sind höchstens

$$2\left(\frac{d^2}{2}+d\right)^{2^{n-1}}$$

wobei $deg(f_i) \leq d$ für alle i.

Zur Komplexität der Berechnung von G-Basen (Forts.)

Beachte: die Schranke hängt nicht von der Anzahl der Polynome oder von den Koeffizienten ab. Sie ist polynomial im Grad und doppelt exponentiell in der Anzahl der Variablen.

- ▶ Es gibt Ideale, für die jede *G*-Basis mindestens $2^{2^{c^n}}$ Elemente enthält mit Elementen vom Grad mindestens $2^{2^{c^{*n}}}$ $(c, c^* \in \mathbb{R}^+)$.
- ▶ Varietäten: $V(f_1, ..., f_s)$ ist leer $gdw 1 \in \langle f_1, ..., f_s \rangle$ in PSPACE.
- ► Für nulldimensionale Ideale einfache Exp-Schranken. Lakshman (90), Mayr (97) Übersicht.
- ▶ Systeme: Macaulay, Singular. Projekte: Posso Frisco.

