

# Computeralgebra

Prof. Dr. K. Madlener

9. Juli 2007

## 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

# Moderne CA-Systeme

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

## Ziele:

- ▶ Breite Funktionalität
- ▶ Einfache Bedienung
- ▶ Effizienz
- ▶ Erweiterbarkeit

## Probleme:

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



# Probleme bei der Implementierung von CA-Systemen

- ▶ 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

# Symbolische      Numerische Berechnungen

1.1 **Beispiel** Chebyshev-Polynome. Rekursive Definition.

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

**Liste der Polynome:**  $1, x, 2x^2 - 1, 4x^3 - 3x, 8x^4 - 8x^2 + 1, \dots$

Werte die Polynome an bestimmten Stellen aus.

Etwas für  $x = 0.3$ :  $1, 0.3, -0.82, -0.792, 0.3448, \dots$

**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.

# 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 · x · T[n - 1] - T[n - 2];  
for n := 0 to 4 do  
    writeln('T'n[x] = T[n])  
end.
```

# Maple Version für Chebyshev Polynome

$$T[0] := 1;$$

$$T[1] := x;$$

**for**  $n = 2$  **to**  $4$  **do**

$$T[n] := \text{expand}(2 \cdot x \cdot T[n-1] - T[n-2]);$$

Interne Darstellung

$$T[0] := 1$$

$$T[1] := x$$

$$T[2] := 2x^2 - 1$$

$$T[3] := 4x^3 - 3x$$

$$T[4] := 8x^4 - 8x^2 + 1$$

Externe Darstellung.

# 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).

## 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**, **SYMBOLICS MACSYMA**, **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.







# Grundlegende Konzepte der Algebra

## 2.2 Beispiel Einheiten, assoziierte Elemente:

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

Assoziiert sein ist Äquivalenzrelation, Klassen assoziierter Elemente.

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

Auswahl von Repräsentanten aus assoziierten Klassen:  $\rightsquigarrow$

**Einheits-normale Elemente**::  $n : D \rightarrow D$       $n(a)$  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**:  $0 \neq a \in D$  ist Nullteiler, falls  $ab = 0$  für ein  $0 \neq b \in D$ .



# Polynomringe

Polynomring über  $R$ :  $R[x]$

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

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

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

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

Monisch, falls  $a_n = 1$

Addition, Multiplikation definiert wie üblich.

## Eigenschaften von $R[x]$

1.  $R$  ist kommutativ, so auch  $R[x]$ , 0 add. Einheit, 1 mult. Einheit.
2.  $D$  Integerbereich, so auch  $D[x]$ .  
Einheiten sind die konstanten Polynome  $a_0$  mit  $a_0$  ist Einheit in  $D$ .
3. Ist  $D$  ein ZPE-Ring, so auch  $D[x]$ .

Polynome mit einheitsnormalen Leitkoeffizienten heißen einheitsnormal.



# Polynomringe: Beispiele

## 2.4 Beispiel

$\mathbb{Z}[x]$ : 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

$$\text{grad}(a(x) + b(x)) \leq \max(\text{grad}(a(x), \text{grad}(b(x)))$$

(= falls  $\text{grad}(a(x)) \neq \text{grad}(b(x))$ )

$$\text{grad}(a(x) \cdot b(x)) = \text{grad}(a(x)) + \text{grad}(b(x))$$

# Euklidische Bereiche

## 2.5 Definition

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

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

## 2.6 Beispiel

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

Beachte:

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

# 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  $\mathbb{Z}$ 
  - a) Wähle  $q, r$  mit  $r = 0$  oder  $\text{sign}(r) = \text{sign}(a)$
  - b) Wähle  $q, r$  mit  $r = 0$  oder  $\text{sign}(r) = \text{sign}(b)$
- ▶ In  $F[x]$  sind  $q, r$  eindeutig. (warum?)

Euklidische Ringe sind ZPE-Ringe.

$g = \text{GGT}(a, b)$ , 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.

# Euklidischer Algorithmus

## 2.8 Beispiel

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

$$\begin{aligned} 126 &= 3 \cdot 35 + 21 \\ 35 &= 1 \cdot 21 + 14 \\ 21 &= 1 \cdot 14 + 7 \\ 14 &= 2 \cdot 7 + 0 \end{aligned} \quad 7 \text{ ist GGT}(126, 35)$$

**Anwendung:** Simplifikation rationaler Ausdrücke:  $35/126 \rightsquigarrow 5/18$

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

$$\begin{aligned} \text{quo}(a, b) &= q \quad (\text{auch } a \text{ quo } b) \text{ und} \\ \text{rem}(a, b) &= r \quad (\text{auch } a \text{ rem } b \text{ oder } a \text{ mod } b) \end{aligned}$$

Es gilt dann  $\text{GGT}(a, b) = \text{GGT}(b, r)$

# Grundlage für Euklidischen Algorithmus

## 2.9 Lemma $GGT(a, b) = GGT(b, r)$

**Beweis:** Sei  $a = bq + r$ , dann gilt  
 $GGT(b, r) \mid a$  und  $b \rightsquigarrow GGT(b, r) \mid GGT(a, b)$ , wegen  $r = a - bq$  folgt  
 $GGT(a, b) \mid r$  und  $b \rightsquigarrow GGT(a, b) \mid GGT(b, r)$ , d. h.  
 $GGT(a, b)$  und  $GGT(b, r)$  sind assoziiert, da EN sind sie gleich.

Seien  $a, b \in D, b \neq 0, v(a) \geq v(b)$ .  
Eine **Restfolge** für  $a, b$  ist definiert durch die Folge  $\{r_i\}$  mit  
 $r_0 := a, r_1 := b$  und  $r_i = \text{rem}(r_{i-2}, r_{i-1}), i = 2, 3, 4, \dots$

Es gilt  $v(r_0) \geq v(r_1) > v(r_2) > v(r_3) \dots$   
Es gibt ein  $k$  mit  $r_{k+1} = 0$  ( $k \leq v(b)$ ) und  $GGT(a, b) = n(r_k)$ .

# Procedure Euclid

```
procedure Euclid (a,b)
  {Berechne  $g = \text{GGT}(a, b)$   $a, b \in D$  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 Restfolgeneigenschaften. Komplexitätsanalyse folgt.

## Erweiterter euklidischer Algorithmus (EEA)

```

procedure EEA( $a, b; s, t$ )
  {Berechne  $g = \text{GGT}(a, b)$  und  $s, t \in D$  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
 $q := \text{quo}(c, d); r := c - q \cdot d;$ 
 $r_1 := c_1 - q \cdot d_1; r_2 := c_2 - q \cdot d_2;$ 
 $c := d; c_1 := d_1; c_2 := d_2;$ 
 $d := r; d_1 := r_1; d_2 := r_2;$ 
  end
 $g := n(c);$ 
 $s := c_1 / (u(a) \cdot u(c)); t := c_2 / (u(b) \cdot u(c));$  return ( $g$ );
end.

```

{*Invariante::*}

{ $c = c_1 n(a) + c_2 n(b) \wedge$

{ $d = d_1 n(a) + d_2 n(b)$ }

**Beachte:**

$$n(c) = c_1 \cdot \frac{n(a)}{u(c)} + c_2 \cdot \frac{n(b)}{u(c)}: \quad \text{d.h. } s, t \text{ sind die Bezout-Koeffizienten.}$$

# Erweiterter euklidischer Algorithmus: Beispiel

**2.10 Beispiel** In  $\mathbb{Z}$ ::  $a = 18$   $b = 30$

Wertefolge::

Iteration	$q$	$c$	$c_1$	$c_2$	$d$	$d_1$	$d_2$
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	-5	3

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

## Erweiterter euklidischer Algorithmus: Beispiel

$$\text{In } \mathbb{Q}[x]:: \quad a = 12x^3 - 28x^2 + 20x - 4, \quad b = -12x^2 + 10x - 2$$

$$u(a) = 12 \qquad u(b) = -12$$

Iter.	$q$	$c$	$c_1$	$c_2$	$d$
–	–	$x^3 - \frac{7}{3}x^2 + \frac{5}{3}x - \frac{1}{3}$	1	0	$x^2 - \frac{5}{6}x + \frac{1}{6}$
1	$x - \frac{3}{2}$	$x^2 - \frac{5}{6}x + \frac{1}{6}$	0	1	$\frac{1}{4}x - \frac{1}{12}$
2	$4x - 2$	$\frac{1}{4}x - \frac{1}{12}$	1	$-x + \frac{3}{2}$	0

$$g = n(c) = x - \frac{1}{3}, \quad s = \frac{c_1}{u(a)u(c)} = \frac{1}{12 \cdot \frac{1}{4}} = \frac{1}{3}$$

$$t = \frac{-x + \frac{3}{2}}{(-12) \cdot \frac{1}{4}} = \frac{x - \frac{3}{2}}{3} = \frac{x}{3} - \frac{1}{2}$$

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

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

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

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

Diese ist  $l$ -mal durchzuführen:  $l \leq v(b) + 1 = m + 1$ .

Sei  $R = F[x]$ ,  $F$  Körper, dann  $v(a) = \text{grad}(a)$ .

Zähle **Grundoperationen** (go) in  $F$ :

Kosten der Division mit Rest: Seien  $\text{grad}(a) = n$ ,  $\text{grad}(b) = m$ .

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

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

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

Sei  $n_i = \text{grad}(c)$  in Durchlauf  $i$  ( $0 \leq i \leq l + 1$ ), wobei  $d$  in Durchlauf  $l$  Null wird. Dann gilt  $n_0 = n \geq n_1 = m > n_2 > \dots > n_l$  und  $\text{grad}(q_i) = n_{i-1} - n_i$  für  $1 \leq i \leq l$  ( $q_i$  Wert von  $q$  in Durchlauf  $i$ ). Kosten der Division mit Rest:  $(2n_i + 1)(n_{i-1} - n_i + 1)$  arithm. Operationen in  $F$ .

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

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

**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

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

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

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

$i = 2$ ::  $s_2 = (s_0 - q_1 s_1) = 1 - q_1 \cdot 0$ ,  $\text{grad } s_1 = -\infty < 0 = \text{grad } s_2$ .

Sei  $i \geq 2$  Behauptung richtig für  $2 \leq j \leq i$ , dann

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

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

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

und

$$\text{grad } s_{i+1} = \text{grad } q_i + \text{grad } s_i = \sum_{2 \leq j < i} \text{grad } q_j + \text{grad } q_i = \sum_{2 \leq j < i+1} \text{grad } q_j$$

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  $\text{grad } n, m : \leq 2(n+1)(m+1)$  Operationen.

$2(\text{grad } t_i + 1)(\text{grad } q_i + 1) + (\text{grad } t_{i+1} + 1)$ , d. h.

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

Normalfall

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

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

## Kostenanalyse für $\mathbb{Z}$ : Langzahlarithmetik

**Darstellung von Zahlen:** Wort 64 Bits.  **$2^{64}$ -Standard Darstellung:** Zahl als Feld von Wörtern. Erstes Wort für Vorzeichen und Länge des Feldes, d. h.  $a \in \mathbb{Z}$

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

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

**Als Feld:**  $s 2^{63} + n + 1, a_0, \dots, a_n$  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  $2 \leq b < \frac{|w|}{2}$ , wobei  $|w|$  Wortlänge ist (Multiplikation der Koeffizienten in Wort).

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

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

# Langzahlarithmetik: Klassische Algorithmen

**Klassische Algorithmen für:**  $+$ ,  $-$ ,  $\cdot$ , quo, Exponentiation

**Maß** in **Grundoperationen** (go):

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

# Algorithmen: Addition

**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**

$j := n; k := 0$

$\{k = \text{Übertrag}\}$

**while**  $j > 0$  **do**

**begin**

$w_j := (u_j + v_j + k) \bmod b;$

$\{k \in \{0, 1\}\}$

$k := \lfloor (u_j + v_j + k) / b \rfloor;$

$j := j - 1;$

**end**

$w_0 := k;$

**end.**

Korrektheit! Aufwand  $\approx 2n$  go.

# Algorithmen: Subtraktion

**S:** Subtraktion nicht negativer ganzer Zahlen.

**Eingabe:**  $(u_1 \cdots u_n)_b \geq (v_1 \cdots v_n)_b$

**Ausgabe:** Nichtnegative Differenz:  $u - v = (w_1 \cdots w_n)_b$

**begin**

$j := n; k := 0$

**while**  $j > 0$  **do**

**begin**

$w_j := (u_j - v_j + k) \bmod b;$

$k := \lfloor (u_j - v_j + k) / b \rfloor$

$\{k \in \{0, -1\}\}$

$j := j - 1;$

**end**

**end.**

Korrektheit! Aufwand  $\approx 2n$  go.

# 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  $v_j = 0$  then
         $w_j := 0$ 
    else
        begin
         $i := n; k := 0;$ 
        while  $i > 0$  do
             $t := u_i v_j + w_{i+j} + k; w_{i+j} := t \bmod b; k := \lfloor t/b \rfloor; i := i - 1;$ 
             $w_j := k;$ 
        end
         $j := j - 1;$ 
    end
end
```

{Korrektheit! Aufwand  $\approx 3nm$  go}

# Algorithmen: Motivation für Multiplikationsalg.

$$(u_1 \cdots u_n)(v_1 \cdots v_m)$$


---

$$\left. \begin{array}{l} (u_1 v_m) \cdots (u_{n-1} v_m)(u_n v_m) \\ (u_1 v_{m-1}) \cdots (u_n v_{m-1}) \end{array} \right\} m$$

$$(u_1 v_1) \cdots (u_n v_1)$$


---

$$w_1 \cdots w_m w_{m+1} \quad \cdots w_{n+m}$$

# Algorithmen: Division

**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  $0 \leq \lfloor \frac{u}{v} \rfloor < b$ .

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

z. B.

$$\underline{3142} : \underline{47} = \underline{66} \text{ Rest } 40$$

$$\begin{array}{r} \underline{282} \\ 322 \\ \underline{282} \\ 40 \end{array}$$

# Algorithmen: Division

## Problem

**Eingabe:**  $u = (u_0 u_1 \cdots u_n)_b$   $v = (v_1 \cdots v_n)_b$  mit  $\lfloor \frac{u}{v} \rfloor < b$  (einstellig).

**Bestimme:**  $q = \lfloor \frac{u}{v} \rfloor$  mit  $u = qv + r$ , wobei  $0 \leq r < v$ .

**Schätzung für  $q$ :**  $\hat{q} = \min \left( \left\lfloor \frac{u_0 b + u_1}{v_1} \right\rfloor, b - 1 \right)$  erste Stelle für  $q$ .

**2.12 Lemma** (Übung): Es gilt

1)  $\hat{q} \geq q$

2) Für  $v_1 \geq \lfloor \frac{b}{2} \rfloor$  gilt  $\hat{q} - 2 \leq q \leq \hat{q}$

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

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

**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** $d := \left\lfloor \frac{b}{(v_1+1)} \right\rfloor;$   $\{d \in \{\lfloor b/2 \rfloor, \dots, 1\}\}$  $(u_0 \cdots u_{m+n})_b := (u_1 \cdots u_{m+n}) \cdot d; (v_1 \cdots v_n)_b := (v_1 \cdots v_n) \cdot d; \{ \textit{Normierung} \}$ **for**  $j$  **from** 0 **to**  $m$  **do****begin****if**  $u_j = v_1$  **then** $\hat{q} := b - 1$ **else** $\hat{q} := \left\lfloor \frac{u_j b + u_{j+1}}{v_1} \right\rfloor$ **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** $\hat{q} := \hat{q} - 1;$  $(u_j \cdots u_{j+n})_b := (u_j \cdots u_{j+m})_b - \hat{q} \cdot (v_1 \cdots v_n)_b; q_j := \hat{q};$ **end** $(r_1 \cdots r_n)_b := (u_{m+1} \cdots u_{m+m})_b / d;$ **end.**Korrektheit! Aufwand  $O(m \cdot n)$  go.

# Algorithmen: Exponentiation

**E:** Exponentiation:: **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**

$y := x; z := 1;$

*{Ergebnis in  $z, y \rightsquigarrow x, x^2, x^4, \dots$ }*

**while**  $n > 1$  **do**

**begin**

$m := \lfloor \frac{n}{2} \rfloor;$

**if**  $n > 2m$  **then**

$z := zy;$

$y := yy; n := m;$

**end**

$z := zy;$

**end.**

## Algorithmen: Exponentiation Beispiel

$$\begin{array}{rcccccc}
 & n & 13 & 13 & 6 & 3 \\
 x^{13} & m & & 6 & 3 & 1 \\
 & y & x & x^2 & x^4 & x^8 \\
 & z & 1 & x & x & x^5 & x^{13}
 \end{array}$$

**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:**

$$N = k + e_0 + e_1 + \dots + e_k - 1 \leq 2k = 2 \log n$$

**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!

## Algorithmen: Exponentiation Analyse

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

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

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  $n = 2^k + 2^{k-1}$

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

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

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:  $y = x^n \bmod a$ ,  $n > 10^{50}$ ,

Rekurrenzgleichungen, Potenzreihenentwicklungen.

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

$$a = r_0 \geq b = r_1 > r_2 > \cdots > r_l \geq 0 \quad q_i \geq 0 \text{ alle } i$$

Darstellung der Zahlen z. B.  $2^{64}$ -Standard Darstellung

$$\text{Länge } \lambda(a) = \left\lfloor \frac{\log_2 |a|}{64} \right\rfloor + 1$$

Verwendet man  $l \leq v(b) + 1 = b + 1 \leq 2^{64\lambda(b)} \rightsquigarrow \exp \text{ in } \lambda(b)$ .

Polynomiale Schranke für  $l : 1 \leq i \leq l$

$$r_{i-1} = q_i r_i + r_{i+1} \geq r_i + r_{i+1} > 2r_{i+1}, \text{ d. h.}$$

$$\prod_{2 \leq i < l} r_{i-1} > 2^{l-2} \prod_{2 \leq i < l} r_{i+1} \text{ f\"ur } l \geq 2 \quad r_{l-1} \geq 2 \text{ folgt}$$

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

**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.

GGT Kosten für  $\mathbb{Z}$ :  $v(a) = |a|$ Alle Quotienten gleich 1, z. B.  $(a, b) = (13, 8)$  EA

$$13 = 1 \cdot 8 + 5$$

$$8 = 1 \cdot 5 + 3$$

$$5 = 1 \cdot 3 + 2$$

$$3 = 1 \cdot 2 + 1$$

$$2 = 2 \cdot 1$$

$$l \text{ für } (a, b) = (F_{n+1}, F_n)$$

$$\rightsquigarrow l = n - 1 \approx 1.44 \log F_n + O(1)$$

für  $b$  fest und  $a$  Var giltim Mittel  $l \approx 0.584 \log b$ **Beachte:** Dirichlet / Lejeune 1849    Cesaro 1881Für zufällig gewählte Zahlen  $a, b$  gilt

$$PR(\text{GGT}(a, b) = 1) = \frac{6}{\pi^2} \approx 0.6079$$

$$\text{Verwende: } PR(p \nmid n \wedge p \nmid m) = 1 - \frac{1}{p^2}$$

$$\prod_p \left(1 - \frac{1}{p^2}\right) \approx \frac{6}{\pi^2}$$

## Aufwand für EEA über $\mathbb{Z}$

Sei  $n = \lambda(a)$ ,  $m = \lambda(b) \rightsquigarrow O(nm)$  für EA

(Kosten der Div mit Rest  $a = qb + r \quad O((\lambda(a) - \lambda(b)) \cdot \lambda(b))$  go)

Für die Bezout Koeffizienten gilt analog

$$|s_i| \leq \frac{b}{r_{i-1}} \quad \text{und} \quad |t_i| \leq \frac{a}{r_{i-1}} \quad 1 \leq i \leq l + 1$$

**2.14 Satz** Der EEA für Zahlen  $a, b \in \mathbb{N}$   $\lambda(a) = n \geq \lambda(b) = m$ , 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)

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

Reduktion auf GGT-Berechnung.

## Ringkonstruktionen: $R[x]$ Polynomring

$R$  ZPE, so  $R[x]$  ZPE-Ring.  $R$  euklidisch  $\not\Rightarrow R[x]$  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  
 $F[x] : a(x), b(x), c(x)$  gesucht  $\sigma(x)$  und  $\tau(x)$  mit

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

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

Zerlegung rationaler Funktionen:

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

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

$\rightsquigarrow$  Pseudodivision primitiver EA.

# Quotienten-Körper von Integritätsbereichen

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

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

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

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

Addition + Multiplikation auf  $S/\sim$ :

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

$$(a/b) \cdot (c/d) = ac/bd$$

wohldefiniert auf Äquivalenzklassen.

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

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

$0/1$   $1/1$   $a/1$  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  $\text{GGT}(a, b) = 1$ ,  $b$  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:  $-1/2$ .

## 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.  $\mathbb{Z}(x)$   $\mathbb{Q}(x)$

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

Die Äquivalenzklasse 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.

# Potenzreihen - erweiterte Potenzreihen

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

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

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

0 alle  $a_k = 0$ ,  $a_k = 0$  für  $k \geq 1$  **Konstante PR.**

Addition + Multiplikation wie üblich!

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

**Eigenschaften:**

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

## Potenzreihen - Einheiten

$$a(x) = \sum a_k x^k \quad b(x) = \sum b_k x^k \quad a(x) \cdot b(x) = 1 \text{ so}$$

$$1 = a_0 b_0$$

$$0 = a_0 b_1 + a_1 b_0$$

$$\vdots$$

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

$$\rightsquigarrow a_0 \text{ ist Einheit}$$

Ist  $a_0$  Einheit in  $R$ , so wird  $b$  bestimmt durch

$$b_0 = a_0^{-1}, \quad b_1 = -a_0^{-1}(a_1 b_0), \quad \dots, \quad b_n = -a_0^{-1}(a_1 b_{n-1} + \cdots + a_n b_0)$$

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

### Beachte

$$\text{ord}(a(x) + b(x)) \geq \min\{\text{ord}(a(x)), \text{ord}(b(x))\}$$

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

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  $\text{ord}(a(x)) = l$      $\text{ord}(b(x)) = m$ , d. h.

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

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

## 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

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

$r(x) = 0$  falls  $\text{ord}(a(x)) \geq \text{ord}(b(x))$ ,  $r(x) = a(x)$  falls  $\text{ord}(a(x)) < \text{ord}(b(x))$ .

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

**Achtung:**  $D$  ZPE Ring  $\not\rightarrow D[[x]]$  ZPE Ring, d. h. Normalformen schwer, assoziierte Elemente!

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

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

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

sind assoziiert

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

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

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

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

Einheit. Die Monome  $x^l (l \geq 0)$  und 0 sind einheitsnormal.

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



# Standard Ringkonstruktionen

- ▶  $i \leq R, i$  ideal, so  $R/i$  Ring: Quotientenring  
Idealkongruenz:  $x \equiv_i y \quad (x \equiv y \pmod{i})$  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.

# Inhalt Kapitel 3

## Normalformen - Algebraische Darstellungen

- 3.1 Datenstrukturen - Algebraische Strukturen
- 3.2 Einfache Simplifikationsregeln in CA-Systemen
- 3.3 Wortproblem - Simplifikation
- 3.4 Formalisierung des Simplifikationsbegriffs
- 3.5 Abstraktionsebenen für algebraische Strukturen
- 3.6 Normalformen für Polynomringe, Quotientenkörper und Potenzreihen
- 3.7 Datenstrukturebene



## 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)

$$\frac{\partial c}{\partial x} \rightarrow 0 \quad \frac{\partial x}{\partial x} \rightarrow 1 \quad \frac{\partial(u+v)}{\partial x} \rightarrow \frac{\partial u}{\partial x} + \frac{\partial v}{\partial x}$$

$$\frac{\partial(uv)}{\partial x} \rightarrow u \frac{\partial v}{\partial x} + \frac{\partial u}{\partial x} v$$

$$\frac{\partial(u^v)}{\partial x} \rightarrow v u^{v-1} \frac{\partial u}{\partial x} + (\log_e u) u^v \frac{\partial v}{\partial x}$$

$\rightsquigarrow$  Simplifikation symbolischer Ausdrücke  $\rightsquigarrow$  Reduktionsmethoden.







# 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:  $u = v$  gdw  $uv^{-1} = 1$

Ringe:  $u = v$  gdw  $u - v = 0$

↔ Eigenschaften einer speziellen Äquivalenzklasse.



# Wortproblem - 0-Äquivalenz

## 3.4 Beispiel

$$\frac{1}{2} \sin\left(\frac{1}{2}\right) = \sin(x) \circ \frac{1}{2} \quad \sin(|x|)$$

$$|\sin(x)|, \quad \sin(\sin(\sin(x))), \quad \sin(x)^2, \quad \sin(x) + |x|$$

**Beweisidee:** Reduziere Hilbert's 10. Problem auf „ $E = 0$ “.

Matiassevic 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  $F : \mathbb{R} \rightarrow \mathbb{R}$ , so

$$f^0(x) = x \quad f^{(n+1)}(x) = f(f^n(x)) \quad n > 0$$

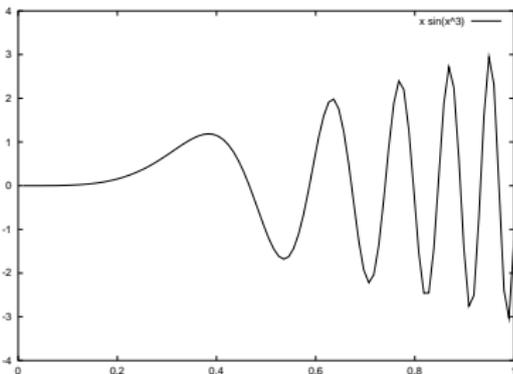
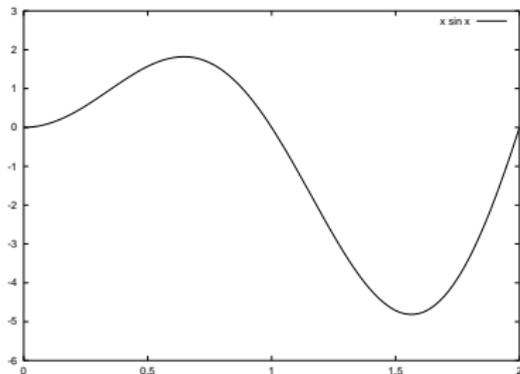
# Wortproblem - 0-Äquivalenz

**3.5 Lemma** Jedes Tupel reeller Zahlen kann durch eine reelle Zahl dargestellt werden (bis auf  $\varepsilon$ ). Seien  $h(x) = x \sin x$ ,  $g(x) = x \sin(x^3)$ .

Dann gibt es für alle  $a_1, a_2, \dots, a_n \in \mathbb{R}$ ,  $0 < \varepsilon < 1$  ein  $b \in \mathbb{R}$  mit

$$|h(g^{(k+1)}(b)) - a_k| < \varepsilon \quad 1 \leq k \leq n.$$

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



# Wortproblem - 0-Äquivalenz

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

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

**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  $(a_1, \dots, a_n) \in \mathbb{N}^n$  mit  $P(a_1, \dots, a_n) = 0$  gdw es gibt  $(b_1, \dots, b_n) \in \mathbb{R}_{\geq 0}^n$  mit  $F(b_1, \dots, b_n) < 0$ .

# Wortproblem - 0-Äquivalenz: Positive Ergebnisse

## 3.9 Satz Richardson

Betrachte Funktionenklasse, die durch Termmenge  $\Lambda$  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  $\Lambda$

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

Das Prädikat „ $F(x) \equiv 0$ “ auf  $\Lambda$  ist entscheidbar.

## Wortproblem - 0-Äquivalenz: Positive Ergebnisse

**Beweisidee:** Komplexität von Ausdrücken + Induktion. Sei etwa  $y$  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:

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

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

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

$$F_2 \equiv 0 \rightsquigarrow F_3 \equiv 0 \quad F_3 \equiv 0 \rightsquigarrow F_2 \text{ konstant.}$$

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

$$y = \log u \rightsquigarrow y' = \frac{u'}{u} \quad u', u \text{ weniger komplex.}$$

$y = e^u$  ist dies nicht der Fall, unterscheide hier

$$F \equiv a_1 y^n + \dots + a_0 \begin{cases} a_0 \equiv 0 \rightsquigarrow F_1 = a_n y^n + \dots + a_1 y = Qy \\ a_0 \neq 0 \rightsquigarrow F_2 = F/a_0 \rightsquigarrow F_2'/y \equiv 0 \rightsquigarrow F \equiv c \end{cases}$$

# Formalisierung des Simplifikationsbegriffs

## Zwei Ziele:

1. „Einfachere“ äquivalente Objekte zu definieren und sie bei Operationen zu verwenden.
2. Wenn möglich kanonische (d. h. eindeutige) Darstellung in (einigen/allen) Äquivalenzklassen 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  $\preceq$  eine Partialordnung auf  $E$ . Eine **Simplifikationsfunktion** für  $[E; \sim]$  bzgl.  $\preceq$  ist eine rekursive Funktion  $f : E \rightarrow E$  mit

$$\text{i) } f(t) \sim t \quad \text{ii) } f(t) \preceq t$$

i. Allg.  $\preceq$  wohlfundierte Partialordnung auf  $E$ , d. h. es gibt keine

$\infty$ -Ketten  $e_1 \succ e_2 \succ e_3 \succ \dots$ ,

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

## Formalisierung des Simplifikationsbegriffs

Eine **Normalisierungsfunktion** bzgl.  $\preceq$  ist eine Simplifikationsfunktion  $f$  bzgl.  $\preceq$  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.  $[0], [1] : t \sim 0$  so  $f(t) = f(0) = 0$ .

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

Eine **kanonische Funktion** ist eine Simplifikationsfunktion  $f$  mit

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

Sie berechnet eindeutige (kanonische) Repräsentanten für jede Äquivalenzklasse.

## 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  $[E; \sim]$ .

## Berechenbare Quotientenstrukturen

**3.12 Satz** Sei  $E$  entscheidbar,  $R$  berechenbare Operation auf  $E$ , d. h.  $R : E^n \rightarrow E$  und  $\sim$  eine Kongruenz bzgl.  $R$ .

Hat  $E$  eine kanonische Funktion  $f$  bzgl.  $\sim$  und ist  $\text{rep}(E) = \{t \in E : f(t) = t\}$  die Menge der kanonischen Repräsentanten, so lässt sich die Quotientenstruktur wie folgt darstellen:

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

und

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

$\text{rep}(E)$  ist entscheidbar,  $R'$  ist berechenbar.

# Beispiele: Monoide und Gruppen

## 3.13 Beispiel

- ▶  $(a, b : ba = \lambda)$  Normalformen  $a^n b^m \quad n, m \geq 0$

Wortersetzungssystem:  $ba \rightarrow \lambda$  terminierend (Längen kürz.)

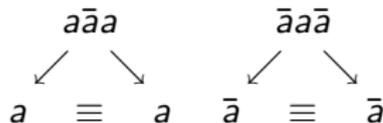
konfluent (d. h. eindeutige NF)

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

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

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

Konfluent: Kritische Paare:



## Beispiele: Monoide und Gruppen

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

WES:  $aba \rightarrow \lambda$   $bab \rightarrow \lambda$  terminierend.

Nicht konfluent  $abab$



Hinzunahme von Regeln  $b \rightarrow a \rightsquigarrow$  Knuth Bendix Vervollständigung.

Länge-Lexikographische Ordnung  $b \succ a$

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

Multiplikation:

	$\lambda$	$a$	$a^2$
$\lambda$	$\lambda$	$a$	$a^2$
$a$	$a$	$a^2$	$\lambda$
$a^2$	$a^2$	$\lambda$	$a$

# Abstraktionsebenen für algebraische Strukturen

- ▶  $\mathbb{Z}_m$       $f(n) = n \bmod m$  positive Reste  
Repr.  $0, 1, \dots, m - 1$ , Definition von  $+$ ,  $\cdot$  auf  $\mathbb{Z}_m$ .

1. **Objektebene:** Menge     Operationen = Elemente der Mengen

2. **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.

# 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

## 1. $E = \mathbb{Z}[x]$

Formebene

- ▶ Jedes Polynom  $\sum_{i=0}^n a_i x^i \in \mathcal{F}$
- ▶  $p_1, p_2 \in \mathcal{F}$ , 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:**

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

## Beispiele (2)

$f_1$  ist Normalisierungsfunktion,  $f_2$  ist kanonische Funktion.

Normalform bzgl.  $f_1$ :

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

Kanonische Form bzgl.  $f_2$ :

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

Oft gilt  $s \sim t$  gdw  $M(s, t) \sim 0$ ,  $\exists$  Normalisierungsfunktion  $\rightsquigarrow$  kanonische Funktion.

## Beispiele (3)

- b) Abelsche Halbgruppen    Varietät  
 Erzeugende                    Relationen  
 $\Sigma :: a, b, c, f, s$              $E :: as = c^2s, bs = cs, s = f$   
     + Kommutativität

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

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

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

Kongruenz, die von  $E$  erzeugt wird: **Ersetzungsregeln**

$\underline{s} \rightarrow f$     $\underline{c}f \rightarrow bf$     $\underline{b^2}f \rightarrow af$  „Modulo Kommutativität“

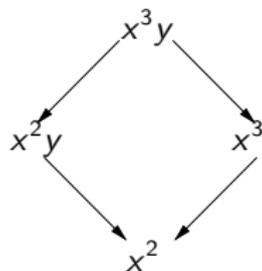
Definiere kanonische Funktion  $\xrightarrow{*}$  mit kanonischen Formen  
 $\underline{\subseteq} a^{n_1} b^{n_2} c^{n_3} f^{n_4}$

## Beispiele (4)

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

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

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



$$x^3 - x^2y \xrightarrow{*} 0$$

→ definiert Simplifikationsfunktion  $p \xrightarrow{*} NF(p)$ , sie ist kanonisch ( $\rightsquigarrow$  Gröbner Basen).

Normalformen für Polynomringe und Quotientenkörper, d. h.

Normalformen für Polynome und rationale Ausdrücke.

## Beispiele (5)

**Ringe:** Axiome kommutative Ringe mit 1.

**Signatur:**  $0, 1, -, +, *$

**Axiome:**  $+$  Komm., Ass., 0 neutr. El., Gruppe inv. -  
 $*$  Komm., Ass., Einh.  $+$  Distributivgesetz

Gleichheitsaxiome  $\rightsquigarrow$  Varietät.

**Univariate Polynome:** Formebene.

$R[x] : a_n x^n + \dots + a_1 x + a_0, n \geq 0, a_i \in R, a_n \neq 0 \cup \{0\}$

System kanonischer Formen für  $R[x]$  (**dicht**) oder **dünn** alle Koeffizienten  $\neq 0$ .

**Multivariate Polynome:** Formebene.

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

$$a(x_1, \dots, x_n) = \sum_{i=0}^{\text{grad}(a(\bar{x}))} a_i(x_1 \dots x_{n-1}) x_n^i$$

dicht/dünn

## 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**  $a(\bar{x}) \in D[x]$

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

$x^e \quad e \in \mathbb{N}^n$  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**  $x > y > z \quad x^2y^2 > x^2yz^3 > x^2z^2 > x > y^3z \dots$

oder

**Grad-Lex**  $x^2yz^2 > x^2y^2 > x^2z^2 > y^3z > z^4 > y^3 > y^2 > x$

## 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:**

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

**Kanonische distributive Darstellung:**

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

**Faktorierte Normalform:**

$$f_3(a(x, y)) =$$

$$(x^3 - x^2y + 2x^2 - xy + 4x - 3y + 3)(x^4y^2 + 2x^4y + x^4 + y^3 - 3y^2 - 9y - 5)$$

**Faktorierte kanonische Form:**

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

## Beispiele (8)

- Rekursive Darstellung
- Distributive Darstellung
- Kanonische distributive Darstellung (Ordnung auf Termen)  $f_2$

- Faktorisierte Normalform

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

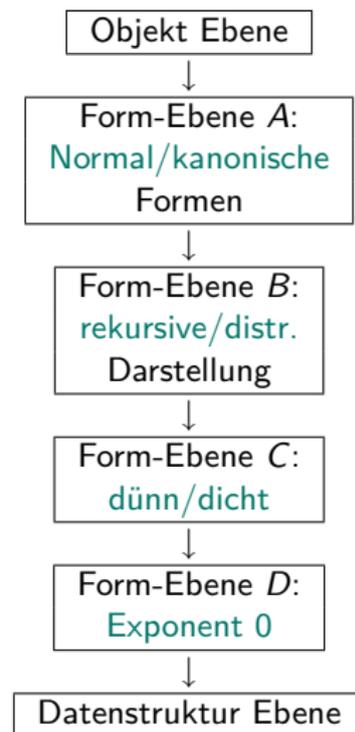
$f_3$

- Faktorisierte kanonische Form (D ZPE)

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

Faktorisiere die  $f_2(p_i)$ , fasse gleiche Faktoren zusammen.

Einheitsnormale Faktorisierung +  
Ordnung der Faktoren



## Beispiele (9)

- ▶  $f_1, f_2, f_3$  sind „einfach“ zu berechnen.
- ▶  $f_4$  kostspielig!
- ▶  $f_2, f_3$  werden am häufigsten verwendet.
- ▶  $(x + y)^{1000} - y^{1000}$  von  $f_2$  und  $f_3$  expandiert!
- ▶ Weitere Transformationen erwünscht!

# Normalformen für rationale Ausdrücke

$D$  Integritätsbereich, Quotientenkörper  $(D) \quad F_D$

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

$D[x_1, \dots, x_n] \quad D(x_1 \dots x_n)$

**Formebene:**  $\frac{a}{b} :: \text{GGT}(a, b) = 1, b \in \mathbb{N}, a, b$  kanon.

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

# 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:  $\frac{a'}{b'} \rightarrow \frac{a''}{b''}, a'' = a' \cdot (u(b'))^{-1}, b'' = b' \cdot (u(b'))^{-1}$

4.  $\frac{a''}{b''} \rightarrow \frac{f_2(a'')}{f_2(b'')}$

Andere Formen:  $a/b$

Fakt/Fakt

Fakt / erweitert

erweitert / Faktor

mit GGT( $a, b$ ) = 1,  $b$  einheitsnormal.

# 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:

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

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

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

# Datenstrukturebene

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       $\rightsquigarrow$       Wortlänge z. B. 64 Bits

Multi-Precision       $\rightsquigarrow$       Langzahlen

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

Langzahlen als Listen von SP-Zahlen.

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

Wahl von  $\beta$

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

## Datenstrukturebene (Forts.)

### Wahl von $\beta$ :

i)  $\beta - 1$  größte SPZ

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

Länge  $l$  der Liste: Dynamisch oder statisch.

**Implementierung:** Zeiger oder Felder.

Referenzierte / sequentielle Zuweisung (-Vorzeichen, -Länge)

$$d \rightarrow \boxed{d_0} \boxed{\phantom{00}} \rightarrow \boxed{d_1} \boxed{\phantom{00}} \rightarrow \cdots \rightarrow \boxed{d_{l-1}} \boxed{\phantom{00}} \rightarrow \text{nil}$$

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

$$N \rightarrow \boxed{890} \boxed{\phantom{00}} \rightarrow \boxed{567} \boxed{\phantom{00}} \rightarrow \boxed{234} \boxed{\phantom{00}} \rightarrow \boxed{1} \boxed{\phantom{00}} \rightarrow$$

Feld

890	567	234	1	0	0...
-----	-----	-----	---	---	------

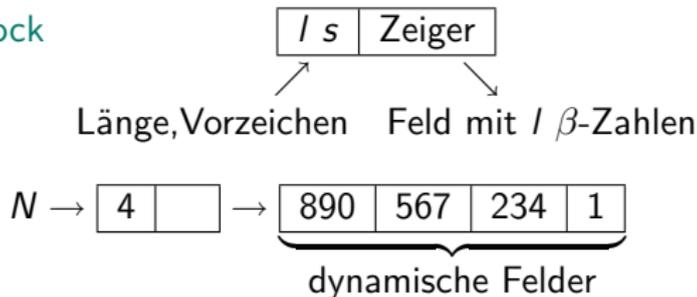
# Probleme

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

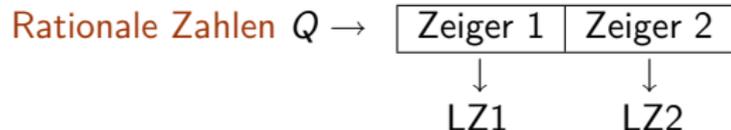
Descriptor Allocation

Mischung

Beschreibungsblock



**Problem:** Speicherverwaltung kostspielig    Garbagecollection

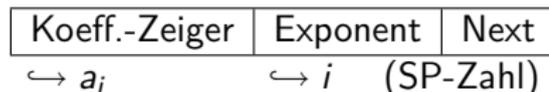


# Datenstrukturen für Polynome

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

$B$  : Rekursive, dünn  $\rightarrow$  Listen

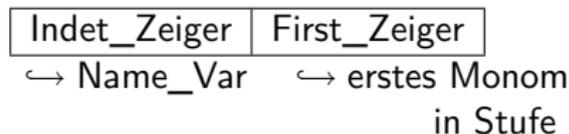
$C$  : Distributive, dünn  $\rightarrow$  Felder.



$$a_i x_1^i$$

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

Als Headerknoten









# Inhalt Kapitel 4

## Arithmetik in Polynomringen - Modulare Arithmetik

- 4.1 Arithmetik in Polynomringen
- 4.2 Pseudo-Division mit Rest-Primitiver EA
- 4.3 Modulare Arithmetik
- 4.4 Schnelle Arithmetik
- 4.5 Die schnelle Fourier Transformation (FFT) Anwendung auf Polynommultiplikation
- 4.6 Anwendung FFT auf Langzahlmultiplikation
- 4.7 Modulare Methoden
- 4.8 Chinesische Reste Algorithmen
- 4.9 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  $\mathbb{Z}[x]$

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

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

$$\rightsquigarrow \text{GGT}(a, b) = 2 \cdot (2x - 3) \\ = 4x - 6$$

## Beispiel (Forts.)

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

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

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

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

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

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

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

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

## Beispiel

### 4.2 Beispiel

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

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

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

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

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

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

Wählt man  $\bar{a}(x) = 5^2 \cdot a(x)$ , so gilt

$$5^2 \cdot (3x^3 + x^2 + x + 5) = (5x^2 - 3x + 1) \cdot (15x + 14) + (52x + 111) \text{ in } \mathbb{Z}[x], \text{ wobei } \text{grad}(52x + 111) < \text{grad}(5x^2 - 3x + 1)$$

$\rightsquigarrow$  **Pseudo-Divisions-Eigenschaft**



## Pseudo-Division mit Rest-Primitiver EA

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

- ▶  $\beta = \text{Haupt\_Koeff}(b(x))$ ,
- ▶  $l = \text{grad}(a(x)) - \text{grad}(b(x)) + 1$
- ▶  $q(x)$ :: **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.  $\text{GGT}(a_0, \dots, a_n) = 1$  für  $\text{grad}(a) = n$ ).

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

Der **Inhalt (Content)** von  $a(x)$  (Bez.  $\text{cont}(a(x))$ ) ist der eindeutige EN GGT der Koeffizienten von  $a(x)$ ,  
d. h. jedes Polynom hat eine eindeutige Darstellung

$$a(x) = u(a(x)) \cdot \text{cont}(a(x)) \cdot PP(a(x))$$

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

Für 0 setze  $\text{cont}(0) = 0$  und  $PP(0) = 0$ .







## 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	–	$8x^3 - 14x^2 + 7x - 6$	$2x^3 + 5x^2 - 22x + 15$
1	$-68x^2 + 190x = 132$	$2x^3 + 5x^2 - 22x + 15$	$34x^2 - 95x + 66$
2	$4280x - 6420$	$34x^2 - 95x + 66$	$2x - 3$
3	0	$2x - 3$	0

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

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

**Problem:** Wachstum der Koeffizienten bei PD mit Rest!



# Modulare Arithmetik

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

„Fingerprinting“ Technik:

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

$a^* = a \pmod{p}$ ,  $b^* = b \pmod{p}$ ,  $c^* = c \pmod{p}$  teste, ob  
 $a^* \cdot b^* \equiv c^* \pmod{p}$  (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:**  $\mathbb{Z} \rightarrow \mathbb{Z}_p$  oder  $\mathbb{Z}_n$ .



## Darstellungen von $\mathbb{Z}_n$

- ▶ **Positive Darstellung:** Repräsentanten  $\{0, 1, \dots, n - 1\}$
- ▶ **Symmetrische Darstellung:** Repräsentanten  $\left\{-\left\lfloor \frac{n}{2} \right\rfloor, \dots, -1, 0, 1, \dots, \left\lfloor \frac{n}{2} \right\rfloor\right\}$  bzw.  $\left\{-\frac{n}{2} + 1, \dots, -1, 0, 1, \dots, \frac{n}{2}\right\}$   
 Z.B.:  $\mathbb{Z}_7 = \{0, 1, 2, 3, 4, 5, 6\}$  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, Subtraktion  $\rightsquigarrow$  über Repräsentanten.
- ▶  $n = p$  Primzahl, so  $\mathbb{Z}_p$  Körper: d. h. **Inversen**  
 EEA,  $\text{GGT}(m, p) = 1$ , d. h.  $s \cdot m + t p = 1$ ,  $s \cdot m \equiv 1 \pmod{p}$ .
- ▶ e arith. Ausdruck    Berechnung von  $e \bmod n$ .  
 $a \equiv b \pmod{n} \rightsquigarrow a * c \equiv b * c \pmod{n}$ ,  $* \in \{+, -, \cdot\}$



## Modulare Arithmetik in $R[x]$ (Forts.)

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

Modulare Inverse kann mit EEA berechnet werden.

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

**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:**  $(\frac{1}{4}x + \frac{1}{2})(x^3 - x + 2) + (-\frac{1}{4}x^2 - \frac{1}{2}x + \frac{1}{4})x^2 = 1$

d. h.  $(-x^2 - 2x + 1)/4$  ist Inverse von  $x^2 \bmod x^3 - x + 2$ .

**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 ( $\text{grad}(f) = n$ ).



## 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  $\alpha = (x \bmod f) \in K$ .

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

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



## 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.  
 $\rightsquigarrow F_{125} = \mathbb{F}_5[x]/(f)$  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  $x^2 \bmod x^3 - x + 1$ .

Setzt man  $\alpha = x \bmod f$ , so gilt

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

- ▶ Kosten der Operationen in  $\mathbb{F}_5[x]/(f)$ : Addition, Multiplikation, Division.  
 $O(n^2)$  Operationen in  $F$ .

# Die Euler Funktion

## Erinnerung:

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

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

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

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

# 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

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

$$\boxed{\bar{u}_1 \quad \bar{u}_0}$$

$$\boxed{\bar{v}_1 \quad \bar{v}_0}$$

d. h.

$$u = \bar{u}_1 b^n + \bar{u}_0$$

$$v = \bar{v}_1 b^n + \bar{v}_0$$

mit

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

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

**Reduktion:**

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







## Modulare Darstellung großer Zahlen (Forts.)

**Zeige::** Jede Zahl  $a < m$  kann eindeutig durch Liste  $a = (a_0, \dots, a_n)$  mit  $a_j = a \bmod m_j$  dargestellt werden.

**Surjektivität:** Behauptung:

Es gibt  $l_j$  mit  $\Phi(l_j) = (0, \dots, 0, 1, 0 \dots 0)$

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

Setze  $l_0 = s \cdot m/m_0 \rightsquigarrow l_0 \equiv 0 \bmod m_j \quad 1 \leq j \leq n$

$$l_0 = s \frac{m}{m_0} \equiv s \frac{m}{m_0} + tm_0 = 1 \bmod m_0$$

d. h.  $\Phi(l_0) = (1, 0 \dots 0)$

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

# Operationen via mod. Darstellungen

$$a = (a_0, \dots, a_n)$$

$$b = (b_0, \dots, a_n)$$

$$a, b < m$$

- ▶  $a + b < m \rightsquigarrow a + b = (c_0, \dots, c_n)$  mit  $c_i = a_i + b_i \bmod m_i$
- ▶  $a \cdot b < m \rightsquigarrow a \cdot b = (c_0, \dots, c_n)$  mit  $c_i = a_i b_i \bmod m_i$ .

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

## Operationen via mod. Darstellungen (Forts.)

Für  $K[x]$   $K$  Körper geht dies genauso!

$a_0, \dots, a_n \in K$  verschiedene Elemente aus  $K$ .

$$m_i(x) = (x - a_i) \quad m(x) = m_0(x) \cdots m_n(x)$$

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

- ▶ Ist  $a(x)$  Polynom von Grad höchstens  $n$ , so gilt  
 $a(x) \bmod m(x) = a(x)$ ,  $a(x) \bmod m_i(x) = a(a_i)$
- ▶  $\text{grad}(a(x)) \leq n \quad \Phi(a(x)) = (a(a_0), \dots, a(a_n))$

## 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?



## Beispiel (Forts.)

b)  $R = F[x]$ ,  $m_i = x - a_i$ ,  $0 \leq i \leq n$ ,  $a_0, \dots, a_n \in F$  (pv).

$$f \equiv f(a_i) \pmod{x - a_i}, \quad 0 \leq i \leq n, \quad f \rightarrow (f(a_0), \dots, f(a_n))$$

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

$$l_i \equiv 1 \pmod{x - a_i} \quad l_i \equiv 0 \pmod{x - a_j}, \quad j \neq i$$

$\text{grad}(l_i) \leq n$  sind die **Lagrange Interpolanden**

$$l_i = \prod_{\substack{0 \leq j \leq n \\ j \neq i}} \frac{x - a_j}{a_i - a_j}$$

Für  $b_0, \dots, b_r \in F$ , so  $f = \sum_{0 \leq i \leq n} b_i l_i$  Lagrange Interpolationspolynom mit

$$f(a_i) = b_i \text{ für } 0 \leq i \leq n.$$

d. h. Chinesische Reste Algorithmus für  $n + 1$  Lin Polynome ist Interpolation in  $n + 1$  Werte. Polynom ist eindeutig:  $\text{Grad} \leq n$ .

# Die schnelle Fourier Transformation (FFT)

## Anwendung auf Polynommultiplikation

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

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

**Idee:** **Fourier Transformierte:** Reduktion auf einfachere Operationen

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

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

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

# Die allgemeine Fourier Transformation

Die Variablen  $t$  und  $f$  stehen für Zeit und Frequenz

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

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

# Diskrete Fourier Transformation

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

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

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

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

D.h.::  $a(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \dots + a_1x + a_0$

$A_j = a(\omega^j)$ , 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)

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

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

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

Setzt man  $a(x) = a_0 + a_1 x + \dots + a_{n-1} x^{n-1}$

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

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  $x_j$ : **prim. E.W.**

# Diskrete Fourier Transformation: Die Auswertung

Angenommen  $n$  **gerade**, dann  $a(x) = b(x^2) + x \cdot c(x^2)$ , wobei  
 $b(y) = a_0 + a_2y + \dots + a_{n-2}y^{n/2-1}$ ,  $c(y) = a_1 + a_3y + \dots + a_{n-1}y^{n/2-1}$ .  
 Hierbei haben  $b(y)$  und  $d(y)$   $\text{grad} \leq \text{grad}(a(x))/2$ .

**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

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

für geeignete Konstante  $c$ .

# Diskrete Fourier Transformation: Die Auswertung

**Beweis:** Wegen (\*) gilt

$$x_0^2 = x_{n/2}^2, \quad x_1^2 = x_{n/2+1}^2 \cdots x_{n/2-1}^2 = x_{n-1}^2,$$

d. h. es gibt nur  $n/2$  verschiedene Quadrate, d. h.

- ▶  $a$  vom Grad höchstens  $n - 1$  kann an den Stellen  $\{x_0, \dots, x_{n-1}\}$  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  $x_j^2$ ) und  $\frac{n}{2}$  Multiplikationen, Additionen und Subtraktionen, um die Werte zu kombinieren.  $\rightsquigarrow$  Behauptung.

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

# 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  $1 \leq p < n$

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

# 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  $A_{ij} = \omega^{ij}$ ,  $0 \leq i, j \leq n-1$ ,  $\mathbf{a} = [a_0, \dots, a_{n-1}]^T$ , dann

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

heißt **diskrete Fourier Transformierte von  $\mathbf{a}$**  (bzgl.  $\omega$ ).

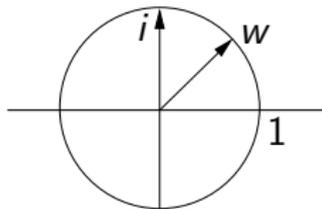
# Beispiel

## 4.14 Beispiel

1. Sei  $R = \mathbb{C}$  komplexe Zahlen,  $n = 8$

$\omega = e^{2\pi i/8} = e^{\pi i/4} = \frac{1+i}{\sqrt{2}}$  ist primitive 8-EW.

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



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

(heben sich paarweise auf!)

## Beispiel (Forts.)

b)  $R = \mathbb{Z}_{17}$ ,  $n = 4$ . 4 ist eine 4 EW, da  $4^4 = 256 \equiv 1 \pmod{17}$ .

Sie ist auch primitiv, da  $4^2 = 16$  und  $4^3 = 13$  und  $\sum_{j=0}^3 4^j = 0 \pmod{17}$ .

Fourier Punkte:  $\{1, 4, 4^2, 4^3\} = \{1, 4, 16, 13\}$

**Diskrete Fourier Transformation:**

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

# Symmetriebedingung für PEW

## 4.15 Lemma

Ist  $\omega$  eine primitive  $n$ -te EW, so erfüllen die  $n$  Fourier Punkte die Symmetriebedingung \*

### Beweis:

Da  $\omega$  primitive  $n$ -te EW ist, gilt

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

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

Ist  $\omega^{n/2+j} - \omega^j = 0$  so  $\omega^{n/2} = 1 \not\perp$ .

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  $\omega$  primitive  $n$ -te EW,  $n$  gerade. Dann

- $\omega^2$  ist primitive  $n/2$ -EW.
- 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.

## 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  $\{1, -1\}$ .

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

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

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:**  $A \leftarrow \text{FFT}(8, 14, a(x)) = (7, -1, 8, -19, 7, -7, -18, -10)$



## Schnelle Fourier Transformation (FFT): Analyse

**Aufwand:**  $O(n \log n)$  Operationen: Sei  $n = 2^m$  dann

$$T(1) = 0, \quad T(2^k) = 2T(2^{k-1}) + c2^{k-1} \quad k \geq 1$$

$$\begin{aligned} T(n) &= T(2^m) = 2T(2^{m-1}) + c2^{m-1} = 2^2 T(2^{m-2}) + c2^{m-1} \cdot 2 \\ &= 2^3 T(2^{m-3}) + c2^{m-1} \cdot 3 \cdots = 2^m T(1) + c2^{m-1} m \\ &= c2^{m-1} m = c \frac{n}{2} \log n \end{aligned}$$

**Beachte:** Eignet sich gut für Parallelisierung: Rekurrenzgleichung:

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

## (FFT):: Ergebnis

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

$$\left\{ R[x] \mid \text{grad} \leq N-1 \right\} \rightarrow R[x]/(x - \omega^0) \times \cdots \times R[x]/(x - \omega^{N-1})$$

Wie sieht es mit der **Umkehrung** aus

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

### Vandermonde Matrix



# Inverse Fourier Transformation

## 4.18 Definition

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

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

wobei

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

Hierbei ist  $\omega$  primitive  $n$ -te EW.

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

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

## Inverse Fourier Transformation

**Beweis:** Sei  $0 < p < N$ . Dann

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

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

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

Für  $p = 0$  ist der Ausdruck  $N$ .

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

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

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

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

## Inverse Fourier Transformation

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

## Beispiel

**4.20 Beispiel** In  $\mathbb{Z}_{17}$ ,  $\omega = 4$  primitive 4-te EW.

Inverse  $S_{(1,4,16,13)} : (\mathbb{Z}_{17})^4 \rightarrow (\mathbb{Z}_{17})^4$

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

Sowohl  $T$  als auch  $S$  sind Fourier Transformationen.

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

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

# Schnelle Polynommultiplikation

$$\begin{array}{l}
 a(x) \text{ grad } m \\
 b(x) \text{ grad } n
 \end{array}
 \qquad
 a(x) \cdot b(x) \qquad \text{grad } m + n$$

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

## Schnelle Fourier Polynommultiplikation

```

procedure FFT_Multiplikation( $a(x), b(x), m, n$ )
begin
    {Eingabe: Polynome  $a, b$  vom Grad  $m, n$ }
    {Berechne  $c(x) = a(x) \cdot b(x)$  mit FFT's}
     $N :=$  erste Zweierpotenz größer als  $m + n$ ;  $\omega :=$  primitive  $n$ -te EW;
     $A := FFT(N, \omega, a(x)); B := FFT(N, \omega, b(x));$ 
for  $i$  from 0 to  $N - 1$  do
    begin
         $C_i := A_i B_i;$ 
    end

     $C := N^{-1} FFT(N, \omega^{-1}, C(x)); c(x) := \sum_{i=0}^{N-1} C_i x^i;$ 

return ( $c(x)$ )
end.
  
```

Aufwand (ohne Berechnungskosten für  $\omega$ ) ( $O((m+n) \log(m+n))$ )

Grundoperationen in  $R$ .

Kommt zum Tragen erst für  $m + n \geq 600$  (Moenck)

## 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).

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

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

$$C = (5, 4, 0, -6, 20, -14, 15, -7)$$

$$= FFT(8, 14, a(x)b(x))$$

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

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

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

## 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  $\mathbb{C}$

▶  $\mathbb{Q}, \mathbb{R}, \mathbb{Z}_p$

**4.22 Satz**  $\mathbb{Z}_p$  hat primitive n-te EW 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  $\rightsquigarrow n \mid p - 1$  (Lagrange).

Die multiplikative Gruppe endlicher Körper ist zyklisch. Sei  $\alpha$  erzeugendes Element der multiplikativen Gruppe von

$\mathbb{Z}_p : \mathbb{Z}_p^\times = \{1, \alpha, \alpha^2, \dots, \alpha^{p-2}\}$  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.  $\omega$  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  $\omega$  primitive n-te EW.

## Berechnung primitiver n-ter EW

**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  $\mathbb{Z}_p^*$ .

Anwendung  $n = 2^r$  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  $\text{GGT}(a, b) = 1$ . Die Anzahl der Primzahlen  $\leq x$  der Form  $ak + b$ ,  $k = 1, 2, \dots$  ist in etwa

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

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  $\leq x$ .



## 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.

$$15: \quad 15^8 = 18 \pmod{41} \quad 15^{20} \equiv -1 \pmod{41}$$

$$\quad \neq 1 \quad \quad \quad \neq 1$$

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

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

## Beispiel (Forts.)

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

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



$$\begin{aligned}
 a \in R, N = 2^k \rightsquigarrow \sum_{i=0}^{N-1} a^i &= \prod_{i=0}^{k-1} (1 + a^{2^i}) \text{ Ind. nach } k \\
 &= (1 + a) \sum_{i=0}^{N/2-1} (a^2)^i
 \end{aligned}$$

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

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

## Beispiel (Forts.)

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

### 4.27 Satz

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

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

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

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

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



# Modulare Algorithmen: Allgemeine Methoden

## Anwendungsfälle

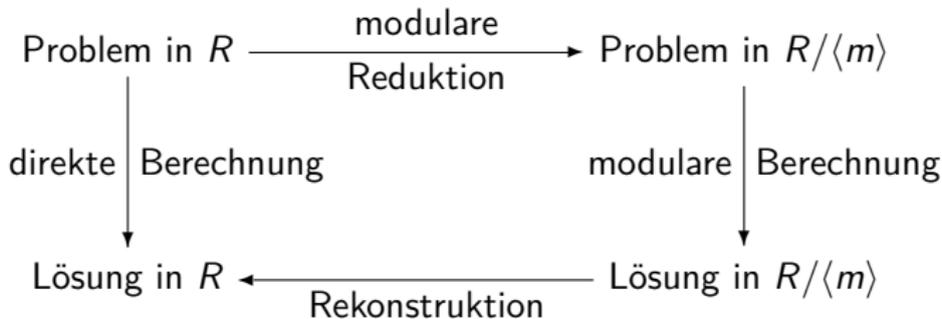
- ▶ FFT-Anwendung zur Multiplikation von Polynomen.
- ▶ GGT-Berechnung in  $\mathbb{Z}_m \cong \mathbb{Z}_{m_0} \times \cdots \times \mathbb{Z}_{m_k}$  um Koeffizientenwachstum zu vermeiden.
- ▶  $\mathbb{Z}[x] \quad \rightarrow \quad \mathbb{Z}_p[x]$   
nicht euklid            euklidisch
- ▶ Faktorisierung, Wurzelberechnung,...

## 3 Varianten:

**Big-Prime, Small-Primes, Prime-Power**

# Modulare Methoden: Big-Prime

**Big-Prime:**  $R$  euklidisch  $m = p$

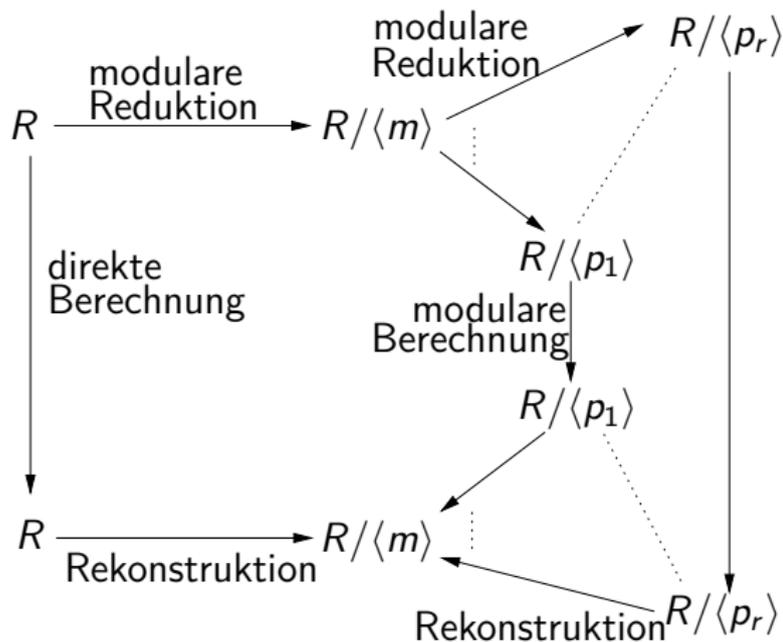


**Benötigt werden:**

- ▶ Schranke für die Lösung in  $R$ .
- ▶ Finde geeignete Moduli.

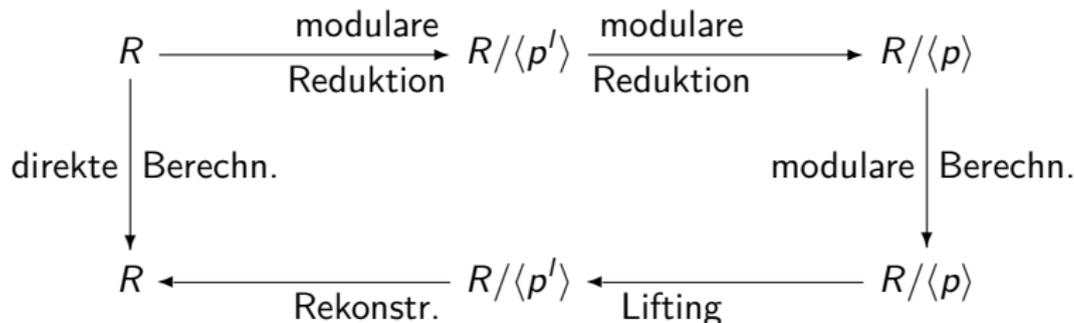
# Modulare Methoden: Small-Primes

- **Small Primes::**  $m = p_1 \dots p_r$   $p_i \neq p_j (i \neq j)$



## 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)



## Beispiel (Fort.)

Cramers Regel::

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

wobei Länge  $\text{Det}[\quad] \approx n \cdot w$ , 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

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

- $\Phi(a + b) = \Phi(a) + \Phi(b) \quad (a, b \in R)$
- $\Phi(ab) = \Phi(a)\Phi(b) \quad (a, b \in R)$
- $\Phi(1) = 1$
- $\Phi(0) = 0 \quad \Phi(-a) = -\Phi(a)$

# Ringmorphismen: Beispiele

## 4.29 Beispiel

### 1. Modulare Homomorphismen: $m \in \mathbb{Z}$

$$\Phi_m \mathbb{Z}[x_1, \dots, x_n] \rightarrow \mathbb{Z}_m[x_1, \dots, x_n]$$

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

### 2. Auswertungshomomorphismen: $\alpha \in D$

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

$$\Phi_{x_i-\alpha}(a(x_1, \dots, x_n)) = a(x_1, \dots, x_{i-1}, \alpha, x_{i+1}, \dots, x_n)$$

### 3. Komposition Homomorphismen:

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

## Ringmorphismen: Beispiel

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

$$\varphi_1 := \Phi_{x-0}\Phi_5 : \mathbb{Z}[x] \rightarrow \mathbb{Z}_5 \quad \varphi_1(a) = 0$$

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

$$\varphi_2 := \Phi_{x-1}\Phi_5 : \mathbb{Z}[x] \rightarrow \mathbb{Z}_5 \quad \varphi_2(a) = 2$$

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

$$c(0) \equiv 0 \pmod{5}, \quad c(1) \equiv 3 \pmod{5}, \quad c(2) \equiv 4 \pmod{5}$$

Analog mit  $m = 7$

$$c(0) \equiv 6 \pmod{7}, \quad c(1) \equiv 2 \pmod{7}, \quad c(2) \equiv 5 \pmod{7}.$$

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

## Ideale und Homomorphismen (Erinnerung)

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

1.  $a - b \in I$  für  $a, b \in I$ .
2.  $ar \in I$  für  $a \in I, r \in R$ .

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

►  $\Phi : R \rightarrow R'$  Homomorphismus

$\text{Kern}(\Phi) = \{r \in R : \Phi(r) = 0\}$  ist Ideal.

►  $\Phi_1 : R \rightarrow R', \Phi_2 : R \rightarrow R'', \text{Kern}(\Phi_1) = \text{Kern}(\Phi_2)$ , so ist  
 $\Phi_1(R) \cong \Phi_2(R) \cong R/\text{Kern}(\Phi_1)$

## Ideale und Homomorphismen (Erinnerung)

- ▶  $I = \langle a_1, \dots, a_n \rangle = \{a_1 r_1 + \dots + a_n r_n : r_i \in R\}$ . 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  $\rightsquigarrow$   $R$  HIR).  
 $D$  noethersch, so ist  $D[x]$  noethersch.
- ▶ Summen, Produkte, Potenzen von Idealen.

$$\begin{aligned} \langle a \rangle \cdot \langle b \rangle &= \langle ab \rangle & \langle a \rangle^i &= \langle a^i \rangle \quad i \geq 1 \\ \langle a \rangle + \langle b \rangle &\neq \langle a + b \rangle \\ &= \langle \text{GCD}(a, b) \rangle \text{ in HIR.} \end{aligned}$$

# Chinesischer Restesatz Algorithmen

$R$  euklidisch,  $m_0, \dots, m_n \in R$  paarweise teilerfremd.

$$m = m_0 \cdot \dots \cdot m_n \quad \langle m \rangle = \langle m_0 \rangle \cdot \dots \cdot \langle m_n \rangle$$

$$\Phi_i := \Phi_{m_i} : R \rightarrow R/\langle m_i \rangle :: a \rightarrow a \bmod m_i$$

$$\Phi : \Phi_0 \times \dots \times \Phi_n : R \rightarrow R/\langle m_0 \rangle \times \dots \times R/\langle m_n \rangle$$

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

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

**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:  $l_i$  mit  $\Phi_i(l_i) = 1$  sonst 0 ( $i \neq j$ ) über EEA.



# Interpolationsproblem: Anwendungsfälle

## 2 Anwendungsfälle:

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

- ▶  $\mathbb{Z}$ : Finde  $\mathbf{a} \in \mathbb{Z}$  mit  $u \leq \mathbf{a} < u + m$ ,  $\mathbf{a} \equiv \mathbf{a}_i \pmod{m_i}$  für festes  $u$  (z. B.  $u = 0$  oder  $u = -m/2$ ).

- ▶  $\mathbb{Z}[x_1, \dots, x_\nu]$ :  $\Phi_p: \mathbb{Z}[x_1, \dots, x_\nu] \rightarrow \mathbb{Z}_p[x_1, \dots, x_\nu]$   
 $\Phi_l: \mathbb{Z}_p[x_1, \dots, x_\nu] \rightarrow \mathbb{Z}_p[x_1]$  (oder  $\mathbb{Z}_p$ ).

$l$  Komposition von Auswertungshomomorphismen  $x_i \mapsto \alpha_i, \alpha_i \in \mathbb{Z}_p$ .

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

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

$$\text{grad}_x(\mathbf{a}(x)) \leq n \quad \mathbf{a}(x_j) = \mathbf{a}_i \in D \quad 0 \leq i \leq n.$$

# Garner & Newton Interpolationsalgorithmen

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

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

wobei  $\nu_k \in R/\langle m_k \rangle$   $k = 0, 1, \dots, n$ .

**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  $\langle m \rangle \subseteq \langle m_k \rangle$  für  $k = 0, 1, \dots, n$ , 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  $\beta$ :  $a = \sum_{i=0}^n \nu_i \beta^i$   $0 \leq a < \beta^{n+1}, 0 \leq \nu_i < \beta$

## 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  $\mathbb{Z}_p$ ,  $\alpha_i \in \mathbb{Z}_p$  ( $D$ ),  $\nu_k \in F_D$ ,  $0 \leq k \leq n$ .

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

Darstellung ist eindeutig unter diesen Bedingungen Existenz!

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

## Newton Koeffizienten

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

$$a = \nu_0 + \nu_1(m_0) + \nu_2(m_0 m_1) + \cdots + \nu_n \left( \prod_{i=0}^{n-1} m_i \right)$$

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

- ▶  $a \equiv \nu_i \pmod{m_0}$  (oder  $a(\alpha_0) = \nu_0 = a_0$ ), d. h.  $\nu_0 = a_0$ .
- ▶ Sind die Koeffizienten  $\nu_0, \dots, \nu_{k-1}$  bestimmt, so folgt

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

## Newton Koeffizienten (Fort.)

Wähle  $\nu_k$ , so dass

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

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

(Beachte im Polynomfall

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

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).

## Newton Koeffizienten (Forts.)

Also gilt

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

oder

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

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

# Garner's Algorithmus/Newton Interpol. Algorithmus

## Gemischte Basisdarstellung

```

procedure INTEGERCRA  $((m_0, \dots, m_n), (a_0, \dots, a_n))$ 
     $\{m_i \in \mathbb{Z}, \text{GGT}(m_i, m_j) = 1 \ (i \neq j), a_i \in \mathbb{Z}_{m_i}\}$ 
    {Ausgabe  $a \in \mathbb{Z}_m$  mit  $m = \prod m_i \quad a \equiv a_i \pmod{m_i}, i = 1, \dots, n$ }
    { Schritt 1: Berechne die benötigten Inversen }
    {Inverse( $a, q$ ) =  $a^{-1} \pmod{q}$ }
for  $k$  from 1 to  $n$  do
    begin
        product :=  $\Phi_{m_k}(m_0)$ ;
        for  $i$  from 1 to  $k - 1$  do
            product :=  $\Phi_{m_k}(\text{product} \cdot m_i)$ ;
         $\gamma_k := \text{inverse}(\text{product}, m_k)$ ;
    end

```



## Garner's Algorithmus (Forts.)

{ Schritt 2: Berechne die  $\{\nu_k\}$ }

$\nu_0 := a_0;$

**for**  $k$  from 1 to  $n$  **do**

**begin**

$\text{temp} := \nu_{k-1};$

**for**  $j$  from  $k - 2$  to 0 **do**

$\text{temp} := \Phi_{m_k}(\text{temp} \cdot m_j + \nu_j);$

$\nu_k := \Phi_{m_k}((a_k - \text{temp})\gamma_k);$

**end**

{ Schritt 3: Transformiere gemischte Radixdarstellung in Standard Darstellung }

$a := \nu_n;$

**for**  $k$  from  $n - 1$  to 0 **do**

$a := am_k + \nu_k;$

**return** ( $a$ )

## 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  $(a_0, \dots, a_n)$  kann als Langzahldarstellung interpretiert werden).

Bis auf Schritt 3 nur Operationen mit SPZ.

### Beachte Schreibweise:

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

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

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

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

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



## Bemerkungen zu Garner's Algorithmus

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

$k = 0, \dots, n$

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

Auch für  $0 \leq \nu_k \leq m_k - 1 \quad k = 0, \dots, n$

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

Berechnet wird

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

## Newton Interpolationsalgorithmus

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

## Beispiel Garner's Algorithmus

### 4.30 Beispiel

Angenommen SP-Zahlen beschränkt  $-100 < a < 100$  (2 Bit). Modullii:

$$m_0 = 99, m_1 = 97, m_2 = 95, m = m_0 m_1 m_2 = 919985.$$

Symmetrische konsistente Darstellung:  $-456142 \leq a \leq 456142$

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

#### Garner:

- Schritt 1:

$$\gamma_1 = m_0^{-1} \pmod{m_1} = 99^{-1} \pmod{97} = 2^{-1} \pmod{97} = -48$$

$$\gamma_2 = (m_0 m_1)^{-1} \pmod{m_2} = 8^{-1} \pmod{95} = 12$$

- Schritt 2: Gemischte Basiskoeffizienten für  $a$

$$\nu_0 = 49, \nu_1 = -35, \nu_2 = -28$$

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

## Beispiel (Forts.)

### 4.31 Beispiel Eingangssystem : System linearer Gleichungen.

Schwierigkeit: Muss keine Lösung in  $\mathbb{Z}$  haben!

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

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

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

- ▶ In  $\mathbb{Z}_p$  berechne  $x \bmod p, y \bmod p, z \bmod p, d \bmod p$  via Gauß  $\rightsquigarrow$  aus  $x_1 \equiv xd \bmod p, y_1 \equiv yd \bmod p, z_1 \equiv zd \bmod p$   $\rightsquigarrow$  CRA  $x_1, y_1, z_1, d$  aus  $\mathbb{Z} \rightsquigarrow \mathbb{Q}$  Lösung.

## Beispiel (Forts.)

In  $\mathbb{Z}_7$ :

$$\begin{array}{rcl}
 x + 2y - 3z = 1 & \text{Gauss} & x \equiv -1 \pmod{7} \\
 x \quad -3z = -2 & \rightsquigarrow & y \equiv -2 \pmod{7} \\
 3x \quad -z = -1 & & z \equiv -2 \pmod{7} \\
 & & d \equiv -2 \pmod{7}
 \end{array}$$

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

$$\begin{array}{rcccc}
 x_1 \equiv -5 \pmod{11} & y_1 \equiv 0 \pmod{11} & z_1 \equiv -4 \pmod{11} & d \equiv 1 \pmod{11} & \\
 -2 & 4 & 6 & 4 \pmod{13} & \\
 5 & -6 & -3 & -2 \pmod{17} & \\
 9 & 6 & 7 & -8 \pmod{19} &
 \end{array}$$

## Beispiel (Forts.)

Modulare Darstellungen für  $x_1$  und  $d$

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

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

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

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

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

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

## 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

$$a(0, 0) = -21 \quad a(0, 1) = -30$$

$$a(1, 0) = 20 \quad a(1, 1) = 17$$

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

- 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:

$$\gamma_1 = (\alpha_1 - \alpha_0)^{-1} \bmod 97 = 1^{-1} \bmod 97 = 1.$$

#### Schritt 2:

$$\text{Newton Koeff: } \nu_0 = -21, \nu_1 = -9$$





## Polynombeispiel: Berechnung von $a(x, y)$

▶ **Schritt 1:**

$$\gamma_1 = 1, \gamma_2 = [(\alpha_2 - \alpha_0)(\alpha_2 - \alpha_1)]^{-1} \bmod 97 = -48$$

▶ **Schritt 2:**

$$\nu_0 = -9y - 21, \nu_1 = 6y + 41, \nu_2 = y$$

▶ **Schritt 3:**

$$a(x, y) = (-9y - 21) + (6y + 41)(x - 0) + y(x - 0)(x - 1)$$





## 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.  $r \equiv d \pmod p$   $-\frac{p}{2} < r < \frac{p}{2}$ .

Da  $p \mid d - r$  und  $|d - r| \leq |d| + |r| < \frac{p}{2} + \frac{p}{2} = p$  folgt  $d = r$ .

Schranken für  $\det A$ : Hadamard Ungleichung

$$|\det A| \leq n^{n/2} B^n \text{ mit } B = \max_{1 \leq i, j \leq n} |a_{ij}|$$

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.
- ▶  $O(n^3 n^2 (\log n + \log B)^2)$  Wortoperationen.

## Modulare Determinantenberechnung (Forts.)

### Small Primes:

$$C = n^{n/2} B^n, r = \lceil \log_2(2C + 1) \rceil.$$

- ▶ Wähle  $r$  verschiedene Primzahlen  $m_0, \dots, m_{r-1} \in \mathbb{N}$ .
- ▶ Für  $0 \leq i < r$  berechne  $d_i \equiv \det A \pmod{m_i}$  (Gauss) in  $\mathbb{Z}_{m_i}$ .
- ▶ Chinesischer R.A  $d \equiv d_i \pmod{m_i} \quad 0 \leq i < r$ .

Dann  $\det A \equiv d \pmod{m_i}$  und somit  $\det A \equiv d \pmod{m}$  für  $m = m_0 \dots m_{r-1}$ .

Wegen  $m \geq 2^r > 2^{n/2} n B^n \geq 2|d|$  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,  $\log m_i \in O(\log r)$ .  
 $\log m = \sum_{0 \leq i < r} \log m_i \in O(r \log r)$ .
- ▶ Operationen mod  $m_i \leftrightarrow O(\log^2 m_i)$ , d. h.  $O(\log^2 r)$  Operationen.  
 $O(n^3 r \log^2 r)$  Wortoperationen,  $A \bmod m_i \rightarrow O(n^2 r \log^2 r)$ .
- ▶  $r$  Werte  $O(n^2 r^2 \log^2 r)$ .

$$O(n^4 \log^2(nB)(\log^2 n + (\log \log B)^2))$$

**Praxis:** Vorberechnung von Primzahlen mit Wortlänge.





## Newton's Iteration und Hensel's Konstruktion

- ▶ Umkehrung modularer & Auswertungs-Homomorphismen.
- ▶ Anwendung von Newton's Iterationsmethoden zur Lösung von Polynomgleichungen.

$$\mathbb{Z}[x_1, \dots, x_\nu] \rightarrow \mathbb{Z}_p[x_1, \dots, x_\nu] \rightarrow \mathbb{Z}_p[x_1]$$

- ▶ 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.













## Approximation und $p$ -adische Darstellung

$$a(x) \equiv b(x) \pmod{q(x)} \text{ gdw } a(x) - b(x) \in \langle q(x) \rangle$$

$$u(x) \equiv u_0(x) \pmod{p} \text{ und}$$

$$u(x) \equiv u_0(x) + u_1(x)p + \dots + u_{k-1}(x)p^{k-1} \pmod{p^k}$$

- $(p) > (p^2) > \dots > (p^k) > \dots$  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

$$a(x) \equiv b(x) \pmod{p^n}$$

Der **Fehler** bei der Approximation von  $a(x)$  durch  $b(x)$  ist

$$a(x) - b(x) \in \mathbb{Z}[x].$$











## 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

$$\begin{aligned} u(x_1, \dots, x_\nu) &= u(x_1, \alpha_2, \dots, \alpha_\nu) + \sum_{i=2}^{\nu} u_i(x_1)(x_i - \alpha_i) \\ &+ \sum_{i=2}^{\nu} \sum_{j=i}^{\nu} u_{ij}(x_1)(x_i - \alpha_i)(x_j - \alpha_j) \\ &+ \sum_{i=2}^{\nu} \sum_{j=i}^{\nu} \sum_{k=j}^{\nu} u_{ijk}(x_1)(x_i - \alpha_i)(x_j - \alpha_j)(x_k - \alpha_k) \\ &+ \dots \end{aligned}$$

- ▶  $d$  totaler Grad von  $u(x_1, \dots, x_\nu)$  bzgl.  $x_2, \dots, x_\nu$ , gibt Anzahl der geschachtelten Summationen.





## 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  $\mathbb{Z}[\bar{x}]/\langle p \rangle^{k+1}$ . Sie ist erlaubt durch die natürliche Einbettung des Ringes  $\mathbb{Z}[\bar{x}]/\langle p \rangle^k$  in den größeren Ring  $\mathbb{Z}[\bar{x}]/\langle p \rangle^{k+1}$

- ▶ Daher liegen die sukzessiven  $p$ -adischen Approximationen  $u^{(1)}, u^{(2)}, u^{(3)}, \dots$  von  $\tilde{u} \in \mathbb{Z}[\bar{x}]$  in einer Folge von Unterringen wachsender Größe von  $\mathbb{Z}[\bar{x}]$

$$\mathbb{Z}[\bar{x}]/\langle p \rangle \subset \mathbb{Z}[\bar{x}]/\langle p \rangle^2 \subseteq \mathbb{Z}[\bar{x}]/\langle p \rangle^3 \subseteq \dots \subseteq \mathbb{Z}[\bar{x}]$$

## 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

$$a \equiv b \pmod{I^n}$$

Der **Fehler** bei der Approximation von  $a$  durch  $b$  ist das Element  $a - b \in D$ . Mit  $\Phi_{I^n} : D \rightarrow D/I^n$  gilt  $\Phi_{I^n}(a - b) = 0$ .

## Iteration nach Newton für $F(u) = 0$

Lineare p-adische Iteration

- Inversion von  $\Phi_p : \mathbb{Z}[x] \rightarrow \mathbb{Z}_p[x]$ . 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.

$$u_0(x) + u_1(x)p + \cdots + u_{k-1}(x)p^{k-1} \in \mathbb{Z}_{p^k}[x] \quad k = 1, \dots, n+1$$

**Lifting von Bild**  $u_0(x) \in \mathbb{Z}_p[x]$ .

- 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.  $u^2 - a(x) = 0$ .



## 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  $u_i(x) \in \mathbb{Z}_p[x]$   $i = 1, 2, \dots, n$  wobei  $u^{(k)} = u_0(x) + u_1(x)p + \cdots + u_{k-1}(x)p^{k-1}$   $1 \leq k \leq n+1$  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  $u^{(k+1)} = u^{(k)} + u_k(x)p^k$   $1 \leq k \leq n$  führt.



## Newton-Verfahren:: Gleichung $F(u) = 0$ (Forts.)

- ▶ Das Polynom  $F(u) \in \mathbb{Z}[x][u]$  hat dann folgende Darstellung

$$F(u^{(k)} + u_k(x)p^k) = F(u^{(k)}) + F'(u^{(k)})u_k(x)p^k + g(u^{(k)}, u_k(x)p^k)[u_k(x)]^2 p^{2k}$$

für ein  $g(u, w) \in D[u, w]$ .

- ▶ Wegen  $u^{(k)} \equiv \tilde{u} \pmod{p^k}$  und  $F(\tilde{u}) = 0$  folgt  $F(u^{(k)}) \equiv 0 \pmod{p^k}$ .
- ▶ Analog gilt  $F(u^{(k)} + u_k(x)p^k) \equiv 0 \pmod{p^{k+1}}$  falls  $u^{(k+1)} = u^{(k)} + u_k(x)p^k$ .



## Newton-Verfahren:: Gleichung $F(u) = 0$ (Forts.)

► D. h.

$$\frac{F(u^{(k)} + u_k(x)p^k)}{p^k} = \frac{F(u^{(k)})}{p^k} + F'(u^{(k)})u_k(x) + g(u^{(k)}, u_k(x)p^k)[u_k(x)]^2 p^k$$

► Wende  $\Phi_p$  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  $u^{(k)} \equiv u^{(1)} \pmod{p}$  für  $k \geq 1$  gilt

$$F'(u^{(k)}) \equiv F'(u^{(1)}) \pmod{p}$$



## 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.  
 $u^{(1)} = u_0(x) = x^2 - 1 \in \mathbb{Z}_5[x]$ .

$$F'(u) = -2u \text{ und } \Phi_5(F'(u^{(1)})) = \Phi_5(-2u^{(1)}) = -2x^2 + 2$$



## Beispiel: Quadratwurzel

Dann

$$\begin{aligned}
 u_1(x) &= -\frac{\Phi_5\left(\frac{F(u^{(1)})}{5}\right)}{(-2x^2 + 2)} = -\frac{\Phi_5\left(\frac{35x^4 - 180x^3 + 95x^2 + 330x + 120}{5}\right)}{(-2x^2 + 2)} \\
 &= -\frac{(2x^4 - x^3 - x^2 + x - 1)}{(-2x^2 + 2)} = x^2 + 2x - 2 \in \mathbb{Z}_5[x]
 \end{aligned}$$

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.  $u^{(3)} = (x^2 - 1) + (x^2 + 2x - 2)5 + (-x)5^2 \in \mathbb{Z}_{125}[x]$

$F(u^{(3)}) = 0 \rightsquigarrow$  Terminierung, d. h.

Quadratwurzel ist  $u(x) = u^{(3)} = 6x^2 - 15x - 11 \in \mathbb{Z}[x]$ .



## Beispiel: Division mit Rest

Setze  $\text{rev}_k(a) := x^k a(1/x)$ . Für  $k = n$  erhält man

$$a = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$$

$$\text{rev}_n(a) = a_0 x^n + a_1 x^{n-1} + \cdots + a_{n-1} x + a_n$$

$$(*') \quad \text{rev}_n(a) = \text{rev}_{n-m}(q) \cdot \text{rev}_m(b) + x^{n-m+1} \cdot \text{rev}_{m-1}(r)$$

d.h.

$$\text{rev}_n(a) \equiv \text{rev}_{n-m}(q) \cdot \text{rev}_m(b) \pmod{x^{n-m+1}}$$

- Da  $\text{rev}_m(b)$  1 als konstanten Koeffizienten hat, ist es invertierbar mod  $x^{n-m+1}$  also

$$\text{rev}_{n-m}(q) \equiv \text{rev}_n(a) \text{rev}_m(b)^{-1} \pmod{x^{n-m+1}}$$

hieraus lassen sich  $q$  und  $r$  berechnen:

$$q = \text{rev}_{n-m}(\text{rev}_{n-m}(q)) \quad \text{und} \quad r = a - qb$$

## Beispiele (Forts.)

$$\text{z.B. } a = 5x^5 + 4x^4 + 3x^3 + 2x^2 + x \quad b = x^2 + 2x + 3 \quad \mathbb{F}_7[x]$$

$$\text{rev}_5(a) = x^4 + 2x^3 + 3x^2 + 4x + 5$$

$$\text{rev}_2(b) = 3x^2 + 2x + 1$$

Wie berechnet man  $\text{rev}_2(b)^{-1} \bmod x^4$  ?

$$\text{rev}_2(b)^{-1} \equiv 4x^3 + x^2 + 5x + 1 \bmod x^4 \text{ in } \mathbb{F}_7[x]$$

$$\rightsquigarrow \text{rev}_3(q) \equiv 6x^3 + x + 5 \bmod x^4$$

$$\rightsquigarrow q = 5x^3 + x^2 + 6 \text{ und } r = a - qb = 3x + 3$$

## Inversion modulo $x^l$ in $D[x]$

### Problem:

Gegeben  $f \in \mathbb{D}[x]$ ,  $l \in \mathbb{N}$  mit  $f(0) = 1$

Finde  $g \in \mathbb{D}[x]$  mit  $fg \equiv 1 \pmod{x^l}$

- ▶  $\rightsquigarrow$  Newton Iteration Lösungen von  $\Phi(g) = 0$  aus Anfangsnäherung  $g_0$ :

$$g_{i+1} = g_i - \frac{\Phi(g_i)}{\Phi'(g_i)}$$

- ▶  $\Phi(g) = \frac{1}{g} - f = 0$

$$\rightsquigarrow g_{i+1} = g_i - \frac{1/g_i - f}{-1/g_i^2} = 2g_i - fg_i^2$$



# Inversion modulo $x^l$ in $D[x]$ : Beispiel

**5.9 Beispiel**  $f = 3x^2 + 2x + 1$  in  $\mathbb{F}_7[x]$ ,  $l = 4$

Alg. berechnet mit  $g_0 = 1$   $r = 2 = \lceil \log(l) \rceil$

$$g_1 \equiv 2g_0 - fg_0^2 = 2 - (3x^2 + 2x + 1) \equiv 5x + 1 \pmod{x^2}$$

$$g = g_2 \equiv 2g_1 - fg_1^2 = 2x^4 + 4x^3 + x^2 + 5x + 1 \equiv 4x^3 + x^2 + 5x + 1 \pmod{x^4}$$

- ▶ Aufwand:  $l = 2^r$   $3M(l) + l \in O(M(l))$  Arithm. Operationen  
(siehe auch von zur Gathen/Gerhard s.246)

- ▶ Division mit Rest nach diesem Verfahren kostet

$$4M(n) + M(n) + O(n)$$

Ringoperationen

$$M(n) \in O(n \log n \log \log n)$$

# 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  $a \bmod p^l$   $l > 1$ , aus Inverse von  $a \bmod p$ .

**Gegeben:**  $b_0$  mit  $ab_0 \equiv 1 \pmod{p}$

**Gesucht:**  $b$  mit  $ab \equiv 1 \pmod{p^l}$  ::Liften von Inversen.

```

procedure InvLift ( $a, b_0, l$ )                                      $\{ab_0 \equiv 1 \pmod{p} \quad l \in \mathbb{N}\}$ 
 $r := \lceil \log l \rceil$ 
for  $i = 1$  to  $r$  do
    berechne  $b_i := (2b_{i-1} - ab_{i-1}^2) \bmod p^{2^i}$ 
return  $b_r$ 
  
```

**Behauptung:**  $ab_i \equiv 1 \pmod{p^{2^i}}$  Induktion:  $i = 0$

$$1 - ab_{i+1} \equiv 1 - a(2b_i - ab_i^2) \equiv 1 - 2ab_i + a^2b_i^2 \equiv (1 - ab_i)^2 \equiv 0 \pmod{p^{2^{i+1}}}$$

Bsp.:  $R = \mathbb{Z}, p > 1$  oder  $R = D[x]$   $p$  monisch,  $\text{grad } b < l$   $\text{grad } p$  etwa  $p = x$ .



## 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  $p^k \rightsquigarrow b \equiv a - F(a)F'(a)^{-1} \equiv a \pmod{p^k}$ .

$$\begin{aligned} F(b) &\equiv F(a) + F'(a)(b - a) + \rightsquigarrow (b - a)^2 \\ &\equiv F(a) + F'(a)(b - a) \equiv F(a) + F'(a)(-F(a)F'(a)^{-1}) \\ &\equiv 0 \pmod{p^{2k}} \end{aligned}$$

Da  $p^{2k} \mid (a - b)^2$  und  $F(a) \equiv 0 \pmod{p^k}$ .

Wegen  $a \equiv b \pmod{p^k}$  gilt  $a \equiv b \pmod{p}$ .  $F(a) \equiv F(b) \pmod{p}$  für alle  $F \in R[u]$ . Insbesondere für  $F'$ .

Für  $p$  Primelement im euklidischem Bereich ist die Bedingung  $F'(a)$  invertierbar mod  $p$  gdw  $F'(a) \not\equiv 0 \pmod{p}$ .



# Algorithmus $p$ -adische Newton Iteration

Korrektheit: Sei  $a_r \equiv a_{r-1} - F(a_{r-1})s_{r-1} \pmod{p^{2^r}}$ .

- ▶ Dann  $a \equiv a_r \pmod{p^l}$  und es genügt die Invarianten.

$$a_i \equiv a_0 \pmod{p}, \quad F(a_i) \equiv 0 \pmod{p^{2^i}}, \quad s_i \equiv F'(a_i)^{-1} \pmod{p^{2^i}}$$

Für  $0 \leq i \leq r$ . 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:

$$R = D[x], F \in R[u], p = x, l = 2^k, \text{grad}_u F = n, \\ \text{grad}_x F < l \rightsquigarrow O(nM(l)) + O(nl) \text{ Operationen in } D.$$

$$R = \mathbb{Z}, 0 \leq a_0 < p, F \text{ grad } n, \text{ mit Koeffizienten } < p^l \\ \rightsquigarrow O(nM(l \log p)) \text{ Wortoperationen.}$$



## 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  $a_0(0) = -1$ ,  $F(a_0) = -x \equiv 0 \pmod{x}$  sowie  $F'(a_0) = 2a_0 \equiv 1 \not\equiv 0 \pmod{x}$ , d.h.  $s_0 = 1$ .

$$a_1 \equiv a_0 - F(a_0)s_0 = -1 - (-x)1 = x - 1 \pmod{x^2}$$

$$s_1 \equiv 2s_0 - F'(a_1)s_0^2 = 2 \cdot 1 - 2(x-1) \cdot 1^2 \\ = x + 1 \pmod{x^2}$$

$$a \equiv a_1 - F(a_1)s_1 = x - 1 - x^2(x+1) \\ = -x^3 - x^2 + x - 1 \pmod{x^4}$$

Offenbar

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

## Ideal-adische Newton Iteration

- ▶ Inversion eines multivariaten Auswertungshomomorphismus

$$\Phi_I : \mathbb{Z}_p[x_1, \dots, x_\nu] \rightarrow \mathbb{Z}_p[x_1]$$

mit Kern  $I = \langle x_2 - \alpha_2, \dots, x_\nu - \alpha_\nu \rangle \quad \alpha_i \in \mathbb{Z}_p \quad 2 \leq i \leq \nu.$

- ▶ 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  $\Delta u^{(k)} \in I^k$ .



## 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

$$(*) \quad 0 = \Phi_{I^{k+1}}(F(u^{(k)})) + \Phi_{I^{k+1}}(F'(u^{(k)}))\Delta u^{(k)} \in \mathbb{Z}_p[\vec{x}]/I^{k+1}$$

und  $\Delta u^{(k)}$  muss diese Gleichung erfüllen für  $k = 1$  so  $\Delta u^{(1)} \in I$  und

$$\Delta u^{(1)} = \sum_{i=2}^v u_i(x_1)(x_i - \alpha_i) \text{ mit } u_i(x_1) \in \mathbb{Z}_p[x_1]$$





$$0 = \sum_{i=2}^{\nu} c_i(x_1)(x_i - \alpha_i) + \Phi_{I^2}(F'(u^{(1)})) \left[ \sum_{i=2}^{\nu} u_i(x_1)(x_i - \alpha_i) \right] \in \mathbb{Z}_p[\vec{x}]/I^2$$

gilt nach Gleichung (\*). Die ideal-adische Darstellung von  $\Phi_{I^2}(F'(u^{(1)}))$  hat für geeignete Koeffizienten  $d_i(x_1) \in \mathbb{Z}_p[x_1]$   $2 \leq i \leq \nu$  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

$$0 = \sum_{i=2}^{\nu} c_i(x_1)(x_i - \alpha_i) + \Phi_I(F'(u^{(1)})) \left[ \sum_{i=2}^{\nu} u_i(x_1)(x_i - \alpha_i) \right] \in \mathbb{Z}_p[\vec{x}]/I^2$$

$$u_i(x_1) = -\frac{c_i(x_1)}{\Phi_I(F'(u^{(1)}))} \in \mathbb{Z}_p[x_1] \quad 2 \leq i \leq \nu$$

Aktualisierungsformel, die den Korrekturterm  $\Delta u^{(1)}$  festlegt.



## Ideal-adische Newton Iteration–Folgeschritte (Forts.)

$$(\#) \quad u_i(x_1) = -\frac{c_i(x_1)}{\Phi_I(F'(u^{(1)}))} \in \mathbb{Z}_p[x_1]$$

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  $I$ ).

Es werden in Iterationsschritten  $k$  alle ideal-adischen Terme der Lösung  $\tilde{u}$  mit Totalgrad  $k$  (bezüglich  $I$ ) bestimmt.

## 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

$$\Phi_I(F'(u^{(1)})) = \Phi_I(-2u^{(1)}) = -2x^2 + 2$$

## Beispiel (Forts.)

Es ist nützlich  $a(x, y, z)$  in seiner ideal-adischen Darstellung bezüglich  $I$  auszudrücken: d. h.

$$\begin{aligned} a(x, y, z) = & [(x^4 - 2x^2 + 1)] + [(2x^2 - 2)z] + [(x^3 - x)y^2 \\ & + (x^2 - 1)yz + z^2] + [(x)y^2z + yz^2] \\ & + [(-x^2)y^4 + (-2x)y^3z - y^2z^2] \end{aligned}$$

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

$$\begin{aligned} \Phi_{f_3}(F(u^{(2)})) &= \Phi_{f_3}(a(x, y, z) - [(x^2 - 1) + z]^2) = \\ &= (x^3 - x)y^2 + (x^2 - 1)yz \in \mathbb{Z}_5[x, y, z]/I^3 \end{aligned}$$

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  $\Phi_{f_3}(F(u^{(2)}))$  auch null und

$$\begin{aligned} u_{22}(x) &= -\frac{c_{22}(x)}{(-2x^2 + 2)} = -\frac{(x^3 - x)}{(-2x^2 + 2)} = -2x \in \mathbb{Z}_5[x] \\ u_{23}(x) &= -\frac{c_{23}(x)}{(-2x^2 + 2)} = -\frac{(x^2 - 1)}{(-2x^2 + 2)} = -2 \in \mathbb{Z}_5[x] \end{aligned}$$





## 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  $a(x_1, \dots, x_\nu)$ :  $F(u, w) = a(x_1, \dots, x_\nu) - uw = 0$   
 $\Phi_{l,p} : \mathbb{Z}[x_1, \dots, x_\nu] \rightarrow \mathbb{Z}_p[x_1]$ , d. h. Problem der Faktorisierung in  $\mathbb{Z}_p[x_1]$   
 (relativ einfach).
- ▶ Sei etwa  $u_0(x_1)$  Faktor von  $\Phi_{l,p}(a) = a_0(x_1)$ , 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  $w_0$   $m$  pol.  $u, w \in \mathbb{Z}[x_1, \dots, x_\nu]$  mit  
 $F(u, w) = a(x_1, \dots, x_\nu) - uw = 0$  und

$$u(x_1, \dots, x_\nu) \equiv u_0(x_1) \pmod{\langle l, p \rangle}$$

$$w(x_1, \dots, x_\nu) \equiv w_0(x_1) \pmod{\langle l, p \rangle}$$

# Einführung zu Hensel's Lemma (Forts.)

Newton Iteration für nichtlineare Gleichung in 2 Variablen

- ▶ Berechnung von GCD für  $a(x_1, \dots, x_\nu), b(x_1, \dots, x_\nu) \in \mathbb{Z}[x_1, \dots, x_\nu]$   
 $\rightsquigarrow$  berechne GCD  $(a_0(x_1), b_0(x_1))$  in  $\mathbb{Z}_p[x_1]$  (Euklidisch).

- ▶ Ist  $u_0(x_1) = \text{GGT}(a_0(x_1), b_0(x_1))$ , so sei Cofaktor  $w_0(x_1) = \frac{a_0(x_1)}{u_0(x_1)}$ .  
 Lift  $u_0(x_1)$  und  $w_0(x_1)$  zu  $u(x_1, \dots, x_\nu)$  und  $w(x_1, \dots, x_\nu)$  mit  $F(u, w) = 0$ .  
 (Auch  $b$  kann verwendet werden).

$\rightsquigarrow$   $u(x_1, \dots, x_\nu)$  ist Teiler von  $a(x_1, \dots, x_\nu)$  auch von  $b(x_1, \dots, x_\nu)$ ?  
GGT?

# Allgemeine Form der Newton Iteration für $F(u, w) = 0$

- ▶ 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  $\rightsquigarrow$

$$(*) \quad F_u(u^{(k)}, w^{(k)})\Delta u^{(k)} + F_w(u^{(k)}, w^{(k)})\Delta w^{(k)} = -F(u^{(k)}, w^{(k)})$$

- ▶  $\rightsquigarrow$  Problem der Lösung Diophantischer Polynomgleichungen.



## Hensel's Lemma

Seien  $a(x) \in \mathbb{Z}[x]$  und  $u_0(x), w_0(x) \in \mathbb{Z}_p[x]$  mit

$$a(x) \equiv u_0(x)w_0(x) \pmod{p}$$

„Lifte“ nach  $\mathbb{Z}[x]$ , d. h. Inversion von  $\Phi_p : \mathbb{Z}[x] \rightarrow \mathbb{Z}_p[x]$ .  
 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

$$u(x) \equiv u_0(x) \pmod{p} \quad w(x) \equiv w_0(x) \pmod{p}$$

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)

- ▶ 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  $i = 1, 2, \dots, n$ ?
- ▶ 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

$$\Delta u^{(k)} = u_k(x)p^k \quad \Delta w^{(k)} = w_k(x)p^k$$

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.

$$-w^{(k)}\Delta u^{(k)} - u^{(k)}\Delta w^{(k)} \equiv -[a(x) - u^{(k)}w^{(k)}] \pmod{p^{k+1}}$$



# 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.

$$w^{(k)}u_k(x) + u^{(k)}w_k(x) \equiv \frac{a(x) - u^{(k)}w^{(k)}}{p^k} \pmod{p}$$

- ▶ Wendet man nun  $\Phi_p$  an, unter Verwendung von  $\Phi_p(w^{(k)}) = w_0(x)$  und  $\Phi_p(u^{(k)}) = u_0(x)$ , 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 (3)

- ▶  $p$  prim, so  $\mathbb{Z}_p[x]$  euklidisch.

d.h. Sind  $u_0(x), w_0(x)$  teilerfremd  $\rightsquigarrow$  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  $\text{grad}(\sigma(x)) < \text{grad}(w_0(x))$ .

- ▶ Man setze

$$u^{(k+1)} = u^{(k)} + \tau(x)p^k \quad w^{(k+1)} = w^{(k)} + \sigma(x)p^k$$

**Behauptung** Dies sind die richtigen  $p$ -adischen Approximationen der  
 Ordnung  $k + 1$  für  $\bar{u}$  bzw.  $\bar{w}$ .

## 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  $a(x) \equiv u^{(1)}(x)w^{(1)}(x) \pmod p$ .

Dann gibt es für  $k \geq 1$  Polynome  $u^{(k)}(x), w^{(k)}(x) \in \mathbb{Z}_{p^k}[x]$ , so dass

$$a(x) \equiv u^{(k)}(x)w^{(k)}(x) \pmod{p^k}$$

und

$$u^{(k)}(x) \equiv u^{(1)}(x) \pmod p \quad w^{(k)}(x) \equiv w^{(1)}(x) \pmod p$$

**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 (5)

Definiere

$$c^{(k)}(x) = \Phi_p \left[ \underbrace{\frac{a(x) - u^{(k)}(x)w^{(k)}(x)}{p^k}}_{\text{alle Oper. in } \mathbb{Z}_{p^{k+1}}[x]} \right] \quad (1)$$

Es gibt Polynome  $\sigma^{(k)}(x), \tau^{(k)}(x) \in \mathbb{Z}_p[x]$  mit

$$\sigma^{(k)}(x)u^{(1)}(x) + \tau^{(k)}(x)w^{(1)}(x) \equiv c^{(k)}(x) \pmod{p} \quad (2)$$

und  $\text{grad}(\sigma^{(k)}(x)) < \text{grad}(w^{(1)}(x))$ .

Setze

$$u^{(k+1)}(x) = u^{(k)}(x) + \tau^{(k)}(x)p^k \quad w^{(k+1)}(x) = w^{(k)}(x) + \sigma^{(k)}(x)p^k$$

## Hensel's Lemma (6)

Dann Multiplikation mod  $p^{k+1}$  ergibt

$$\begin{aligned}
 u^{(k+1)}(x)w^{(k+1)}(x) &\equiv u^{(k)}(x)w^{(k)}(x) + (\sigma^{(k)}(x)u^{(1)}(x) + \\
 &\quad \tau^{(k)}(x)w^{(1)}(x))p^k \pmod{p^{k+1}} \\
 &\equiv u^{(k)}(x)w^{(k)}(x) + c^{(k)}(x)p^k \pmod{p^{k+1}} \quad (2) \\
 &\equiv a(x) \pmod{p^{k+1}} \quad (1)
 \end{aligned}$$

$\rightsquigarrow$  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 (7)

**Beweis:** Induktion nach  $k$ :  $k = 1$  klar. Angenommen richtig für ein  $k \geq 1$

$u^{(k+1)}(x), w^{(k+1)}(x) \in \mathbb{Z}_{p^{k+1}}[x]$  monisch mit

$a(x) \equiv u^{(k+1)}(x)w^{(k+1)}(x) \pmod{p^{k+1}}$  und

$u^{(k+1)}(x) \equiv u^{(1)}(x) \pmod{p}, w^{(k+1)}(x) \equiv w^{(1)}(x) \pmod{p}.$

Insbesondere

$a(x) \equiv u^{(k+1)}(x)w^{(k+1)}(x) \pmod{p^k}$  nach Induktion Voraussetzung

$u^{(k+1)}(x) \equiv u^{(k)}(x) \pmod{p^k}, w^{(k+1)}(x) \equiv w^{(k)}(x) \pmod{p^k}.$

D. h.

$$u^{(k+1)}(x) = u^{(k)}(x) + \tau(x)p^k \quad w^{(k+1)}(x) = w^{(k)}(x) + \sigma(x)p^k$$

für Polynome  $\sigma(x), \tau(x) \in \mathbb{Z}_p[x]$ .

## Hensel's Lemma (7)

**Behauptung:** Diese sind eindeutig: Da  $a(x), u^{(1)}(x), w^{(1)}(x)$  monisch.  
 $\text{grad}(\sigma(x)) < \text{grad}(w^{(1)}(x)) \quad \text{grad}(\tau(x)) < \text{grad}(u^{(1)}(x))$

(Da  $u^{(k+1)}(x)$  und  $w^{(k+1)}(x)$  immer gleiche Leitertme wie  $u^{(1)}(x)$  bzw.  $w^{(1)}(x)$  haben müssen!).

$\rightsquigarrow$

$$a(x) \equiv u^{(k)}(x)w^{(k)}(x) + (\sigma(x)u^{(1)}(x) + \tau(x)w^{(1)}(x))p^k \pmod{p^{k+1}}$$

$\rightsquigarrow$

$$\sigma(x)u^{(1)}(x) + \tau(x)w^{(1)}(x) \equiv \frac{a(x) - u^{(k)}(x)w^{(k)}(x)}{p^k} \pmod{p}$$

$\rightsquigarrow$

$\sigma(x), \tau(x) \in \mathbb{Z}_p[x]$  sind eindeutig.

(Da Lösungen diophantischer Gleichungen unter diesen Voraussetzungen eindeutig sind. Siehe Geddes et al. Theorem 2.6 s.44)





## Quadratisches Hensel Lifting (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  $x^4 - 1 \equiv 0 \pmod{625}$  verwendet mit Startwert  $x = 2 \pmod{5}$ .

Dies kann als Lifting einer Faktorisierung gesehen werden: nämlich von  $x^4 - 1 \equiv (x - 2)(x^3 + 2x^2 - x - 2) \pmod{5}$  zu Faktorisierung  $\pmod{625}$ .

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  $\pmod{5}$ : EEA liefert  $s = -2, t = 2x^2 - 2x - 1$ , so dass  $su + tw \equiv 1 \pmod{5}$ .

## 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

$$a - \hat{u}\hat{w} = 25(4x^6 - 4x^5 - 8x^4 + 7x^3 + 5x^2 - 3x - 1) \equiv 0 \pmod{25}, \text{ d. h.}$$

$$a \equiv \hat{u}\hat{w} \pmod{25}.$$

**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  $\equiv 0 \pmod{m}$ .

## 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  $\text{grad } r < \text{grad } b$ .
  - ii) Sind  $a, b, q, r$  wie in i) und gilt  $a \equiv 0 \pmod{m}$  für ein  $m \in R$ , so  $q \equiv r \equiv 0 \pmod{m}$ .
- (**Beweis ii**): Sei  $\text{grad } b = n \geq 0 \rightsquigarrow$  aus  $a \equiv 0 \pmod{m}$   $b$  monisch,  $qx^n \equiv 0 \pmod{m}$ , d. h.  $q \equiv 0 \pmod{m}$  und somit auch  $r \equiv 0 \pmod{m}$ .
- ▶ Überlegung führt zu: **Algorithmus Hensel Schritt**

//Eingabe:  $m \in R$   $a, u, w, s, t \in R[x]$  mit  
 $a \equiv uw \pmod{m}$ ,  $su + tw \equiv 1 \pmod{m}$ , wobei  $w$  monisch,  
 $\text{grad } a = n = \text{grad } u + \text{grad } w$ ,  $\text{grad } s < \text{grad } w$  und  $\text{grad } t < \text{grad } u$ .

Ausgabe: Polynome  $u^*, w^*, s^*, t^* \in R[x]$  mit

$$a \equiv u^* w^* \pmod{m^2} \text{ und } s^* u^* + t^* w^* \equiv 1 \pmod{m^2}$$

wobei  $w^*$  monisch,  $u^* \equiv u \pmod{m}$ ,  $w^* \equiv w \pmod{m}$ ,  
 $s^* \equiv s \pmod{m}$ ,  $t^* \equiv t \pmod{m}$ ,  $\text{grad } u^* = \text{grad } u$ ,  
 $\text{grad } w^* = \text{grad } w$ ,  $\text{grad } s^* < \text{grad } w^*$ ,  $\text{grad } t^* < \text{grad } u^*$ .//

1. Berechne  $e, q, r, u^*, w^* \in R[x]$  mit  $\text{grad } r < \text{grad } w$  und  
 $e \equiv a - uw \pmod{m^2}$ ,  $se \equiv qw + r \pmod{m^2}$   
 $u^* \equiv u + te + qu \pmod{m^2}$ ,  $w^* \equiv w + r \pmod{m^2}$
2. Berechne  $b, c, d, s^*, t^* \in R[x]$  mit  $\text{grad } d < \text{grad } w^*$  und  
 $b \equiv su^* + tw^* - 1 \pmod{m^2}$ ,  $sb \equiv cw^* + d \pmod{m^2}$   
 $s^* \equiv s - d \pmod{m^2}$ ,  $t^* \equiv t - tb - cu^* \pmod{m^2}$
3. Return  $u^*, w^*, s^*, t^*$ .

## Algorithmus Hensel-Schritt: Korrektheit

**5.18 Satz** Der Algorithmus ist korrekt. Aufwand für:

- ▶  $R = \mathbb{Z} : 0(M(n)M(\log m))$  Wortoperationen, falls Eingaben in max-Norm  $< m^2$ .
- ▶  $R = F[y] : 0(M(n)M(\text{grad}_y m))$  Operationen in  $F$  Falls grad in  $y$  der Eingaben  $< 2\text{grad}_y m$ .

**Beweis:** Nachrechnen.  $\mathbb{Z}$ : Grad der Polynome  $\leq n$ . Koeffizienten  $\geq m^y$   
Länge  $0(\log m)$ . Division mit Rest  $0(M(n))$ .

## Beispiel

**5.19 Beispiel**  $a = x^4 - 1 = 0$  fortgesetzt.

$$u = x^3 + 2x^2 - x - 2, w = x - 2, s = -2, t = 2x^2 - 2x - 1$$

$$su + tw \equiv 1 \pmod{5}, e = a - uw = 5x^2 - 5$$

- $se = -10x^2 + 10 : x - 2 = -10x + 5$  mit Rest  $r = -5$

d. h.  $q = -10x + 5, r = -5 \pmod{25}$

$$\begin{aligned}u^* &\equiv u + te + qu \equiv x^3 + 2x^2 - x - 2 + \\ &\quad (2x^2 - 2x - 1)(5x^2 - 5) + (-10x + 5) \\ &\equiv x^3 + 7x^2 - x - 7 \pmod{25}\end{aligned}$$

$$w^* \equiv w + r \equiv x - 2 - 5 \equiv x - 7 \pmod{25}.$$

- Dann  $a \equiv u^* w^* \pmod{25}$ . Die Grade von  $u^*, w^*$  sind die von  $u$  bzw.  $w$  und die Polynome sind einfacher als zuvor.
- 7 ist somit Lösung von  $x^4 - 1 \equiv 0 \pmod{25}$  und  $7 \equiv 2 \pmod{5}$  (Startlösung).

## Beispiel (Forts.)

- ▶ Zur Berechnung von  $s^*$ ,  $t^*$  die zur nächsten Iteration benötigt werden. Berechne

$$\begin{aligned} b &\equiv su^* + tw^* - 1 \equiv -2x^3 + 11x^2 + 2x - 11 + \\ &\quad (2x^2 - 2x - 1)(x - 7) - 1 \\ &\equiv -5x^2 - 10x - 5 \pmod{25} \end{aligned}$$

$$\begin{aligned} sb = 10x^2 - 5x + 10 : x - 7 &= 10x - 10 \quad \text{Rest } -10 \pmod{25} \\ &= \quad \quad \quad = \\ &\quad \quad \quad c \quad \quad \quad d \end{aligned}$$

- ▶  $s^* \equiv s - d \equiv 8 \pmod{25}$
- ▶  $t^* \equiv t - tb - cu^* \equiv -8x^2 - 12x - 1 \pmod{25}$

Es gilt

$s^*u^* + t^*w^* \equiv 1 \pmod{25}$  und die Grade von  $s^*$ ,  $t^*$  stimmen mit denen von  $s$  bzw.  $t$  überein.

## Beispiel (Forts.)

### 5.20 Satz Hensel's Lemma Quadratisches Liften

Sei  $l \in \mathbb{N}^+$  und es gelten die Eingangsbedingungen für Hensel-Schritt Algorithmus, dann lassen sich Polynome, die die Ausgabebedingungen erfüllen mit  $m^2$  ersetzt durch  $m^l$ .

**Beweis:** Hensel-Schritt induktiv:  $m$  ersetzt durch  $m, m^2, m^4, \dots$

### 5.21 Beispiel $x^4 - 1 \equiv 0$ Fortsetzung. Sei $m = 5$

$a, u_1 = u^* \quad w_1 = w^* \quad s_1 = s^* \quad t_1 = t^*$  in  $\mathbb{Z}[x]$  aus Beispiel.

- ▶ Es gilt  $a \equiv u_1 w_1 \pmod{25}$ ,  $s_1 u_1 + t_1 w_1 \equiv 1 \pmod{25}$ .

## Beispiel (Forts.)

Anwendung von Hensel-Schritt Algorithmus liefert.

- ▶  $e_2 \equiv a - u_1 w_1 \equiv 50x^2 - 50 \pmod{625}$ .
  - ▶  $q_2 \equiv -225x + 300 \pmod{625}$  und  $r_2 \equiv -175 \pmod{625}$
  - ▶  $u_2 \equiv x^3 + 182x^2 - x - 182 \pmod{625}$  und  $w_2 \equiv x - 182 \pmod{625}$
- ▶  $b_2 \equiv s_1 u_2 + t_1 w_2 - 1 \equiv -225x^2 + 300x - 25 \pmod{625}$ 
  - ▶  $c_2 \equiv 75x - 200 \pmod{625}$  und  $d_2 \equiv 275 \pmod{625}$
- ▶  $s_2 \equiv -267 \pmod{625}$ 
  - ▶  $t_2 \equiv 267x^2 - 312x - 176 \pmod{625}$
- ▶ Dann  $s_2 u_2 + t_2 w_2 \equiv 1 \pmod{625}$ .
- ▶  $a \equiv u_2 w_2 \pmod{625}$ , 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

- ▶  $a \equiv x(x+1)(x^2+1) \pmod{3}$ .  $u_0 = x^2 + xw_0 = x^2 + 1$  teilerfremd mod 3.
- ▶  $s_0 = x + 1$        $t_0 = -x + 1$        $s_0 u_0 + t_0 w_0 \equiv 1 \pmod{3}$

Zwei Hensel Schritte liefern:

- ▶  $e_1 \equiv a - u_0 w_0 \equiv -3x^3 - 3x^2 + 3x + 3 \pmod{9}$   
 $q_1 \equiv -3x^2 + 3x + 3 \pmod{9}$      $r_1 \equiv 3x \pmod{9}$
- ▶  $u_1 \equiv x^2 + 4x + 3 \pmod{9}$
- ▶  $w_1 \equiv x^2 + 3x + 1$ ,  $b_1 \equiv 3x^2 + 3$ ,  $c_1 \equiv 3x + 3$ ,  $d_1 \equiv 0 \pmod{9}$
- ▶  $s_1 \equiv x + 1 \pmod{9}$
- ▶  $t_1 \equiv -x - 2 \pmod{9}$   $e_2 \equiv a - u_1 w_1 \equiv -9x^3 - 27x^2 - 9x \pmod{81}$   
 $q_2 \equiv -9x^2 - 9x \pmod{81}$      $r_2 \equiv 0 \pmod{81}$

## Beispiel (Forts.)

- ▶  $u_2 \equiv x^2 - 5x + 3 \pmod{81}$
- ▶  $w_2 \equiv x^2 + 3x + 1 \pmod{81}$   $b_2 \equiv -9x^2 - 9x \pmod{81}$   
 $c_2 \equiv -9x + 9 \pmod{81}$   $d_2 \equiv -27x - 9 \pmod{81}$
- ▶  $s_2 \equiv 28x + 10 \pmod{81}$
- ▶  $t_2 \equiv -28x - 29 \pmod{81}$

$$\begin{aligned}
 e_3 &= a - u_2 w_2 = x^4 - 2x^3 - 11x^2 + 4x + 3 \\
 &\quad - \underbrace{(x^2 - 5x + 3)(x^2 + 3x + 1)} \\
 &= -(x^4 + 3x^3 + x^2 - 5x^3 - 15x^2 - 5x + 3x^2 + 9x + 3) \\
 &\quad + x^4 - 2x^3 - 11x^2 + 4x + 3 \\
 &= 0
 \end{aligned}$$

d.h. Wir erhalten sogar die Faktorisierung in  $\mathbb{Z}[x]$ , da  $u_2 w_2$  irreduzibel in  $\mathbb{Z}[x]$ .

## 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  $su + tw \equiv 1 \pmod{m}$ .

Die Hauptkoeffizienten von  $u$  und  $w$  seien keine Nullteiler  $\pmod{m}$ ,  $u$  und  $u^*$  (bzw.  $w$  und  $w^*$ ) haben gleiche Hauptkoeffizienten, gleichen Grad und  $u \equiv u^* \pmod{m}$  bzw.  $w \equiv w^* \pmod{m}$ .

Gilt  $uw \equiv u^*w^* \pmod{m^l}$ , so  $u \equiv u^* \pmod{m^l}$  und  $w \equiv w^* \pmod{m^l}$ .

**Beweis:** Angenommen  $u \not\equiv u^* \pmod{m^l}$  oder  $w \not\equiv w^* \pmod{m^l}$ . Wähle  $1 \leq i < l$  maximal, so dass  $m^i \mid u^* - u$  und  $m^i \mid w^* - w$ . D. h.

$$u^* - u = gm^i, \quad w^* - w = hm^i$$

$g, h \in R[x]$  und  $m \nmid g$  oder  $m \nmid h$ . **O.b.d.A.**  $m \nmid g$

$$\begin{aligned} 0 &\equiv u^*w^* - uw = u^*(w^* - w) + w(u^* - u) \\ &= (u^*h + wg)m^i \pmod{m^l} \end{aligned}$$

## Eindeutigkeit des Hensel-Liftings (Forts.)

- ▶ Da  $m$  kein Nullteiler ist, gilt  $m \mid m^{l-i} \mid (u^*h + wg)$ .

- ▶ Bezeichne mit  $\bar{\phantom{x}}$  Reduktion mod  $m$ : Dann

$$\overline{su} + \overline{tw} = 1, \overline{u^*} = \overline{u}, \overline{u^*} \overline{h} + \overline{w} \overline{g} = 0 \text{ also}$$

$$0 = \overline{t}(\overline{u^*} \overline{h} + \overline{w} \overline{g}) = \overline{t} \overline{uh} + (1 - \overline{su})\overline{g}$$

$$= (\overline{t} \overline{h} - \overline{sg}) \overline{u} + \overline{g}, \text{ d. h. } \overline{u} \mid \overline{g}$$

- ▶ Wegen  $\text{HK}(u) = \text{HK}(u^*)$  und  $\text{grad } u = \text{grad } u^*$  gilt  $\text{grad } \overline{g} < \text{grad } \overline{u}$ .  
Da  $\text{HK}(\overline{u}) = \text{HK}(\overline{u})$  kein Nullteiler ist auch  $\overline{u}$  kein Nullteiler und  $\overline{g}$  muss 0 Polynom sein. Widerspruch zu  $m \nmid g$ .

## Folgerung

**5.24 Folgerung** Sei  $R$  euklidisch,  $p \in R$  Primelement.  $l \in \mathbb{N}^+$ ,  $f, g, u \in R[u]$  nicht Null mit  $p \nmid \text{HK}(f)$ ,  $f \bmod p$  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  $f \equiv gh \equiv uw \bmod p^l$  und  $g \equiv uv \bmod p$ . Da  $f \bmod p$  quadratfrei, ist auch  $g \bmod p$  quadratfrei und  $\text{GGT}(u \bmod p, v \bmod p) = 1$  in  $\mathbb{F}_p[x]$ .

- ▶ Hensel's Lemma liefert  $u^*, v^* \in R[x]$ , so dass  $u^* \equiv u \bmod p$ ,  $v^* \equiv v \bmod p$  und  $g \equiv u^* v^* \bmod p^l$ .
- ▶ Wegen  $uvh \equiv gh \equiv uw \bmod p$  gilt  $vh \equiv w \bmod p$ . Also  $v^* h \equiv vh \equiv w \bmod p$  und  $u^*(v^* h) \equiv gh = f \equiv uw \bmod p^l$ .
- ▶ Da  $u, v$  teilerfremd  $\bmod p$  sind, liefert die Eindeutigkeit  $u \equiv u^* \bmod p^l$  und somit  $g \equiv uv^* \bmod p^l$ , d. h.  $u \mid g \bmod p^l$ .



## Folgerung

Es gibt auch eine  $\infty$ -Version von Hensel's-Lemma.

$p$ -adische Vervollständigung von  $R$ : für  $p \in R$  prim (irreduzibel).  $R_{(p)}$

Elemente  $\sum_{i \geq 0} a_i p^i$   $0 \leq a_i < p$ , ( $R = \mathbb{Z}$ ), sonst  $F[[y]]$  falls  
 $p = y$   $R = \overline{F}[y]$ .

**5.25 Satz**  $\infty$ -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 Einzeliteration relativiert.
- ▶ Vergleich durch **Miola und Yun**: Quadratische Iteration. Teurer: Hauptsächlich wegen Berechnung von  $b, sb, s^*$  und  $t^*$ .

**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:**

## Beispiele

## 5.26 Beispiel 1

$$a(x) = x^3 + 10x^2 - 432x + 5040 \in \mathbb{Z}[x] \quad p = 5$$

$$\begin{aligned} \Phi_5(a(x)) &= x^3 - 2x \in \mathbb{Z}_5[x] \\ &= x(x^2 - 2) \end{aligned}$$

$$u_1(x) = x \quad w_1(x) = x^2 - 2 \text{ teilerfremd Hensel}$$

$$s(x) = -2x \quad t(x) = 2 \quad (-2x)x + 2(x^2 - 2) \equiv 1 \pmod{5}$$

## ► Lineare Iteration:

Iter.	$\sigma(x)$	$\tau(x)$	$u(x)$	$w(x)$	$e(x)$
0	—	—	$x$	$x^2 - 2$	$10x^2 - 430x + 5040$
1	$x - 1$	1	$x + 5$	$x^2 + 5x - 7$	$-450x + 5075$
2	$-x + 2$	1	$x + 30$	$x^2 - 20x + 43$	$125x + 3750$
3	1	0	$x + 30$	$x^2 - 20x + 168$	0

Iteration  $k$  am Ende  $5^{k+1} \mid e(x)$ .

## Beispiele (Forts.)

### 5.27 Beispiel Divergenz der Hensel-Iteration

$a(x) = x^4 + 1 \in \mathbb{Z}[x]$  ist irreduzibel über  $\mathbb{Z}[x]$ . Sei  $p = 5 \rightsquigarrow \Phi_5(a(x)) = x^4 + 1$  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 angenommen werden. Konstruktion einer Folge von Faktoren mit

$$a(x) \equiv u^{(k)}(x)w^{(k)}(x) \pmod{p^k} \text{ (bzw. } p^{2k}) \quad k = 1, 2, 3, \dots$$

- ▶  $s(x) = -1$   $t(x) = 1$   $-1(x^2 + 2) + 1(x^2 - 2) \equiv 1 \pmod{5}$   
 $a(x) - (x^2 + 2)(x^2 - 2) = x^4 + 1 - x^4 + 4 = 5 = e(x)$

## Beispiele (Forts.)

	Ende Iter. Nr.	$\sigma(x)$	$\tau(x)$	$u(x)$	$w(x)$	$e(x)$
5	0	—	—	$x^2 + 2$	$x^2 - 2$	5
$5^2$	1	-1	1	$x^2 + 7$	$x^2 - 7$	50
$5^3$	2	-2	2	$x^2 + 57$	$x^2 - 57$	3250
$5^4$	3	-1	1	$x^2 + 182$	$x^2 - 182$	33125
$5^5$	4	2	-2	$x^2 - 1068$	$x^2 + 1068$	1140625

- ▶  $\infty$ -Folge Faktoren in  $\mathbb{Z}_{5^k}[x]$  Ende von Iteration  $k$  gilt stets

$$u(x)w(x) \equiv x^4 + 1 \pmod{5^{k+1}}$$

- ▶ Dies gilt sogar für jede Primzahl  $p$ .

## Schranke für die Anzahl der Iterationen

- ▶ Apriori Schranke für die **Anzahl der Iterationen**:

$$B \geq \max\{|b| : b \text{ Koeffizienten in Polynom } a \text{ oder in jeden möglichen Faktor von } a \text{ mit Grad} \leq \max\{\text{grad}(u), \text{grad}(w)\}\}$$

$p^l > 2B$  Schranke für die Anzahl der Iterationen.

## Beispiele (Forts.)

### 5.28 Beispiel Das Leitkoeffizienten Problem

Nicht-monischer Fall: 
$$c(x) = \frac{a(x) - u(x)w(x)}{p}$$

▶  $\sigma(x)u^{(1)}(x) + \tau(x)w^{(1)} \equiv c(x) \pmod{p}$

Eindeutigkeit wird mit  $\text{grad } \sigma(x) < \text{grad } (w^{(1)}(x))$  erreicht.

▶ Updates:

$$u(x) := u(x) + \tau(x)p \quad w(x) := w(x) + \sigma(x)p$$

↪ Hauptkoeffizienten von  $w$  wird niemals verändert.

Im monischen Fall gilt auch  $\text{grad } (\tau(x)) < \text{grad } (u^{(1)}(x))$ .

↪ Hauptkoeffizienten von  $u$  wird ebenfalls niemals verändert.

i.A.  $\text{grad } (c(x)) \leq \text{grad } (a(x)) = \text{grad } (u^{(1)}(x)) + \text{grad } (w^{(1)}(x)),$

d. h.  $\text{grad } (\tau(x)) \leq \text{grad } (u^{(1)}(x)).$

## Beispiele (Forts.)

- ▶ Alle Veränderungen vom Hauptkoeffizienten müssen in  $u$  realisiert werden.

$$\begin{aligned}a(x) &= 12x^3 + 10x^2 - 36x + 35 \in \mathbb{Z}[x] \\ &= u(x)w(x) = (2x + 5)(6x^2 - 10x + 7) \in \mathbb{Z}[x]\end{aligned}$$

- ▶  $p = 5$   $\Phi_5(a(x)) = 2x^3 - x \in \mathbb{Z}_5[x] = 2(x)(x^2 + 2)$   
2 ist Einheit in  $\mathbb{Z}_5[x]$ .

$$\begin{aligned}\text{Wahl der Anfangsfaktoren: } & 2 \text{ zum Faktor } x \text{ oder} \\ & 2 \text{ zum Faktor } x^2 + 2\end{aligned}$$

$$\text{d.h. } \Phi_5(a(x)) = (2x)(x^2 + 2) = (x)(2x^2 - 1) \in \mathbb{Z}_5[x]$$

- ▶ Die richtigen Faktoren sind  
 $u^{(1)}(x) = 2x$  und  $w^{(1)}(x) = x^2 + 2$

- ▶ Hensel's Konstruktion:  $s(x) = x$   $t(x) = -2$

$$u(x) = 2x \quad w(x) = x^2 + 2 \quad e(x) = 10x^3 + 10x^2 - 40x + 35 \pmod{5}$$

## Beispiele (Forts.)

$\sigma(x)$	$\tau(x)$	$u(x)$	$w(x)$	$e(x)$
—	—	$2x$	$x^2 + 2$	$10x^3 + 10x^2 - 40x + 35$
$-2x - 1$	$2x + 1$	$12x + 5$	$x^2 - 10x - 3$	$125x^2 + 50x + 50$
$2x + 1$	$1$	$12x + 30$	$x^2 + 40x + 22$	$-500x^2 - 1500x - 625$
$-2x - 2$	$0$	$12x + 30$	$x^2 - 210x - 103$	$2500x^2 + 7500x + 312$
$2x + 1$	$0$	$12x + 30$	$x^2 + 1040x + 522$	$-12500x^2 - 37500x - 156$

- ▶ Aufspaltung von 12 in  $2 \cdot 6$ ?

d. h.

- ▶  $u^{(3)}(x) = 12x + 30$      $w^{(3)}(x) = x^2 + 40x + 22$

## Beispiele (Forts.)

- ▶ Sind bis auf Einheiten die richtigen Faktoren

$$u^{(3)}(x)w^{(3)}(x) = 12x^3 + 510x^2 + 1464x + 660 \in \mathbb{Z}[x]$$

- ▶  $6^{-1}u^{(3)}(x) \bmod \mathbb{Z}_{125}[x]$        $6w^{(3)}(x) \bmod \mathbb{Z}_{125}[x]$

$$6^{-1} = 21 \rightsquigarrow u(x) = 21u^{(3)}(x) = 2x + 5 \in \mathbb{Z}_{125}[x]$$

$$w(x) = 6w^{(3)}(x) = 6x^2 - 10x + 7 \in \mathbb{Z}_{125}[x]$$

- ▶ 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  $\mathbb{Z}$ ,  $p \nmid \text{HKoeff}(a(x))$ .

- ▶ Seien  $u^{(1)}(x), w^{(1)}(x) \in \mathbb{Z}_p[x]$  teilerfremd über  $\mathbb{Z}_p$  mit  $a(x) \equiv u^{(1)}(x)w^{(1)}(x) \pmod{p}$ .
- ▶ Seien  $u^{(k)}(x), w^{(k)}(x)$ , die von der Hensel Konstruktion bestimmten Faktoren mit  $a(x) \equiv u^{(k)}(x)w^{(k)}(x) \pmod{p^k}$ ,  
 $u^{(k)}(x) \equiv u^{(1)}(x) \pmod{p}$ ,  $w^{(k)}(x) \equiv w^{(1)}(x) \pmod{p}$ .
- ▶ Gibt es Polynome  $u(x), w(x) \in \mathbb{Z}[x]$  mit  $a(x) = u(x)w(x)$  in  $\mathbb{Z}[x]$  und  $n(u(x)) \equiv n(u^{(1)}(x)) \pmod{p}$  bzw.  $n(w(x)) \equiv n(w^{(1)}(x)) \pmod{p}$ , wobei  $n$  die Normalisierung „mache Polynom monisch als Element von  $\mathbb{Z}_p[x]$ “.
- ▶ Dann sind für alle  $k \geq 1$  die Polynome  $\Phi_{p^k}(u(x))$  und  $u^{(k)}(x)$ , sowie  $\Phi_{p^k}(w(x))$  und  $w^{(k)}(x)$  assoziiert im Ring  $\mathbb{Z}_{p^k}[x]$ .

## Liften von Faktorisierungen (Beweis)

**Beweis:** Sei  $k \geq 1$ . Nach Vor  $p \nmid \text{HKoeff}(a(x))$ , d. h.  $\text{HKoeff}(a(x))$  ist Einheit in  $\mathbb{Z}_{p^k}[x]$ .

- ▶  $\bar{a}(x) = \text{HKoeff}(a(x))^{-1}a(x) \in \mathbb{Z}_{p^k}[x]$  ist monisch. Wegen

$$\text{HKoeff}(a(x)) \equiv \text{HKoeff}(u^{(k)}(x))\text{HKoeff}(w^{(k)}(x)) \pmod{p^k}$$

ist  $p$  kein Teiler der HKoeff, d. h. **sie sind Einheiten mod  $p^k$** .

- ▶  $\bar{u}^{(k)} := \text{HKoeff}(u^{(k)}(x))^{-1}u^{(k)}(x) \in \mathbb{Z}_{p^k}[x]$  und
- ▶  $\bar{w}^{(k)} := \text{HKoeff}(w^{(k)}(x))^{-1}w^{(k)}(x) \in \mathbb{Z}_{p^k}[x]$  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.

## Liften von Faktorisierungen (Beweis)

- ▶ Angenommen es gibt  $u(x), w(x)$  mit  $a(x) = u(x)w(x)$  in  $\mathbb{Z}[x]$ .  
 $n(u(x)) \equiv n(u^{(1)}(x)) \pmod{p}$   $n(w(x)) \equiv n(w^{(1)}(x)) \pmod{p}$ .
- ▶ Betrachte  $\Phi_{p^k}(u(x))$  und  $\Phi_{p^k}(w(x)) \in \mathbb{Z}_{p^k}[x]$ . Wie eben kann man diese in  $\mathbb{Z}_{p^k}[x]$  normalisieren zu  $\bar{u}(x), \bar{w}(x)$ .
- ▶ Wegen der Eindeutigkeit gilt dann:  
 $\bar{u}^{(k)}(x) = \bar{u}(x) \quad \bar{w}^{(k)}(x) = \bar{w}(x)$  in  $\mathbb{Z}_{p^k}[x]$ ,  
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  $\Phi_{p^k}(w(x))$ .

## Beispiel (Forts.)

**5.30 Beispiel**  $a(x) = 12x^3 + 10x^2 - 36x + 35 \in \mathbb{Z}[x]$   
 $= u(x)w(x) = (2x + 5)(6x^2 - 10x + 7) \in \mathbb{Z}[x]$

- ▶  $\Phi_5(a(x)) = (2x) \cdot (x^2 + 2) = u^{(1)}(x) \cdot w^{(1)}(x)$
- ▶  $\Phi_5(u(x)) = \Phi_5(2x + 5) = 2x$
- ▶  $\Phi_5(w(x)) = \Phi_5(6x^2 - 10x + 7) = x^2 + 2$
- ▶  $u^{(2)}(x) = 12x + 5 \quad w^{(2)}(x) = x^2 - 10x - 3$
- ▶  $\Phi_{25}(u(x)) = 2x + 5 \quad \Phi_{25}(w(x)) = 6x^2 - 10x + 7$   
 Unterscheiden sich um Einheiten, z. B.  $-4(6x^2 - 10x + 7)$   
 $\equiv x^2 - 10x - 3 \quad -4(12x + 5) = 2x + 5$
- ▶  $w^{(3)}(x) = x^2 + 40x + 22 \quad u^{(3)}(x) = 12x + 30$
- ▶  $\Phi_{125}(w(x)) = 6x^2 - 10x + 7 \quad \Phi_{125}(u(x)) = 2x + 5$   
 $6^{-1}u^{(3)}(x) = 21u^{(3)}(x) = 2x + 5 \quad 6w^{(3)}(x) = 6x^2 - 10x + 7$

# Modifizierte Hensel Konstruktion: nicht monischer Fall

- ▶ Sei  $a(x) \in \mathbb{Z}[x]$  primitiv,  $a(x) \equiv u^{(1)}(x)w^{(1)}(x) \pmod p$  und es gebe  $u(x), w(x)$  mit  $a(x) = u(x)w(x) \in \mathbb{Z}[x]$ .

Es gelte  $n(u^{(1)}(x)) \equiv n(u(x)), n(w^{(1)}(x)) \equiv n(w(x)) \pmod p$ .

- ▶  $\alpha = \text{HKoeff}(a(x)), \mu = \text{HKoeff}(u(x)), \nu = \text{HKoeff}(w(x))$ ,
- ▶  $\alpha = \mu\nu$ . 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** =  $\alpha$ . Modifiziere die Hensel Konstruktion so, dass

$$\bar{u}^{(k)}(x) = \Phi_{p^k}(\bar{u}(x)) \quad \bar{w}^{(k)}(x) = \Phi_{p^k}(\bar{w}(x)) = \Phi_{p^k}(\alpha \text{HKoeff}(u^{(k)})^{-1} u^{(k)}(x))$$

$$\rightsquigarrow u(x) = pp(\bar{u}(x)) \text{ und } w(x) = pp(\bar{w}(x))$$

# Modifizierte Hensel Konstruktion: nicht monischer Fall

## 5.31 Beispiel

$$\begin{aligned} a(x) &= 12x^3 + 10x^2 - 36x + 35 \in \mathbb{Z}[x] \text{ primitiv.} \\ &\equiv u^{(1)}(x)w^{(1)}(x) \pmod{5} \end{aligned}$$

- ▶ Mit  $u^{(1)}(x) = 2x$ ,  $w^{(1)}(x) = x^2 + 2$ .
- ▶  $\bar{a}(x) = 12a(x) = 144x^3 + 120x^2 - 432x + 420$   
 $\bar{a}(x) \equiv \bar{u}^{(1)}(x)\bar{w}^{(1)}(x) \pmod{5}$  mit  
 $\bar{u}^{(1)}(x) = \Phi_5(12 \cdot 2^{-1}(2x)) = 2x$   
 $\bar{w}^{(1)}(x) = \Phi_5(12 \cdot 1^{-1}(x^2 + 2)) = 2x^2 - 1$
- ▶ Hensel Schritt auf  $\bar{a}(x)$ .  
 $u^{(2)}(x) = \bar{u}^{(1)}(x) + (-x + 1)5 = -3x + 5$   
 $w^{(2)}(x) = \bar{w}^{(1)}(x) + (x - 1)5 = 2x^2 + 5 - 6$
- ▶  $\bar{u}^{(2)}(x) = \Phi_{25}(12 \cdot (-3)^{-1}(-3x + 5)) = 12x + 5$   
 $\bar{w}^{(2)}(x) = \Phi_{25}(12 \cdot 2^{-1}(2x^2 + 5x - 6)) = 12x^2 + 5x - 11$

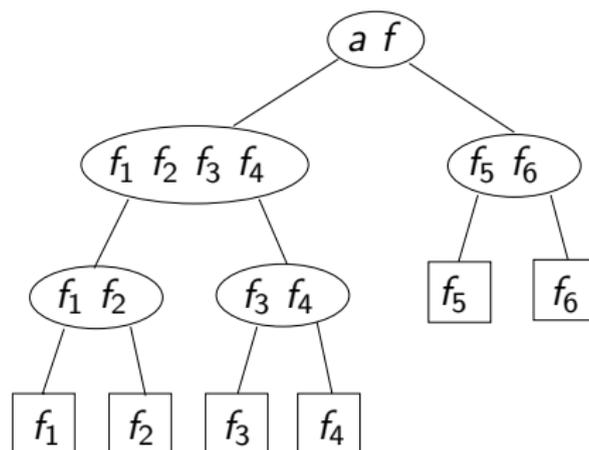


# Multifaktor Hensel Lifting

- ▶ Lifte Faktorisierung in mehr als zwei Faktoren.
- ▶ Sei  $R$  Ring,  $m \in R$ ,  $f, f_1, \dots, f_r \in R[x]$ , so dass  $\text{HKoeff}(f)$  Einheit mod  $m$  ist,  $f_1, \dots, f_r$  monisch und  $f \equiv \text{HKoeff}(f) f_1 \dots f_r \pmod{m}$ .  
Dann gibt es  $a \in R$  mit  $a \cdot \text{HKoeff}(f) \equiv 1 \pmod{m}$
- ▶ Ordne die monischen Faktoren  $\nu$  von  $f$  modulo  $m$  als Binärbaum  $\tau$  der Tiefe  $d = \lceil \log_2 r \rceil$ ,  
mit Blättern  $f_1, \dots, f_r$ , 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 (Forts.)

- Mehrere Anordnungen sind möglich. z. B.  $r = 6$ .





## Algorithmus Liften eines Faktorisierungsbaumes mod $m$

**Eingabe:**  $m \in R$ ,  $f \in R[x]$  mit Grad  $n$ ,  $a_0 \in R$  mit  $a_0 \text{HKoeff}(f) \equiv 1 \pmod{m}$ ,  $l \in \mathbb{N}$ , Faktorbaum  $\tau$  für  $f \pmod{m}$  mit Wurzel  $a_0 f$  und  $r$  Blätter

**Ausgabe:** Eine Inverse  $a^* \in R$  von  $\text{HKoeff}(f) \pmod{m^l}$  und ein Faktorbaum  $\tau^*$  von  $f$  modulo  $m^l$  mit Wurzel  $a^* f$ , so dass jeder Knoten  $\nu^* \in R[x]$  von  $\tau^*$  kongruent modulo  $m$  zum entsprechenden Knoten  $\nu \in R[x]$  von  $\tau$

Beachte: Baumstruktur bleibt unverändert.

# Algorithmus Multifaktor Hensel Lifting (MFHL)

- 1  $d := \lceil \log_2 l \rceil, \tau_0 := \tau$
- 2 **for**  $j = 1 \dots d$  **do**
- 3   {Lifte Inverse von  $HKoeff(f)$ }  
 $a_j := 2a_{j-1} - HKoeff(f)a_{j-1}^2 \bmod m^{2^j}$ ;  
 $\tau_j := \tau_{j-1}$ ;  
 Ersetze Wurzel von  $\tau_j$  durch  $a_j f$
- 4   {Lifte Baum}  
**for** jeden inneren Knoten  $\nu \in R[x]$  von  
 $\tau_j$  **von der Wurzel abwärts 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  $\tau_d$

# Algorithmus Multifaktor Hensel Lifting (MFHL)

**5.32 Satz** Der Algorithmus MFHL ist korrekt bzgl. seiner spec.

Er benötigt

$O(M(n) \log rM(l \log m))$  Wortoperationen, falls  $R = \mathbb{Z}$ ,  $m > 1$  alle Eingaben mit  $\max\_Norm < m^l$  und

$O(M(n) \log rM(l \text{ grad}_y m))$  Operationen in  $F$ , falls  $R = F[y]$ , Körper  $F$  und  $y$ -Grad aller Eingaben kleiner als  $l \text{ grad}_y m$ .

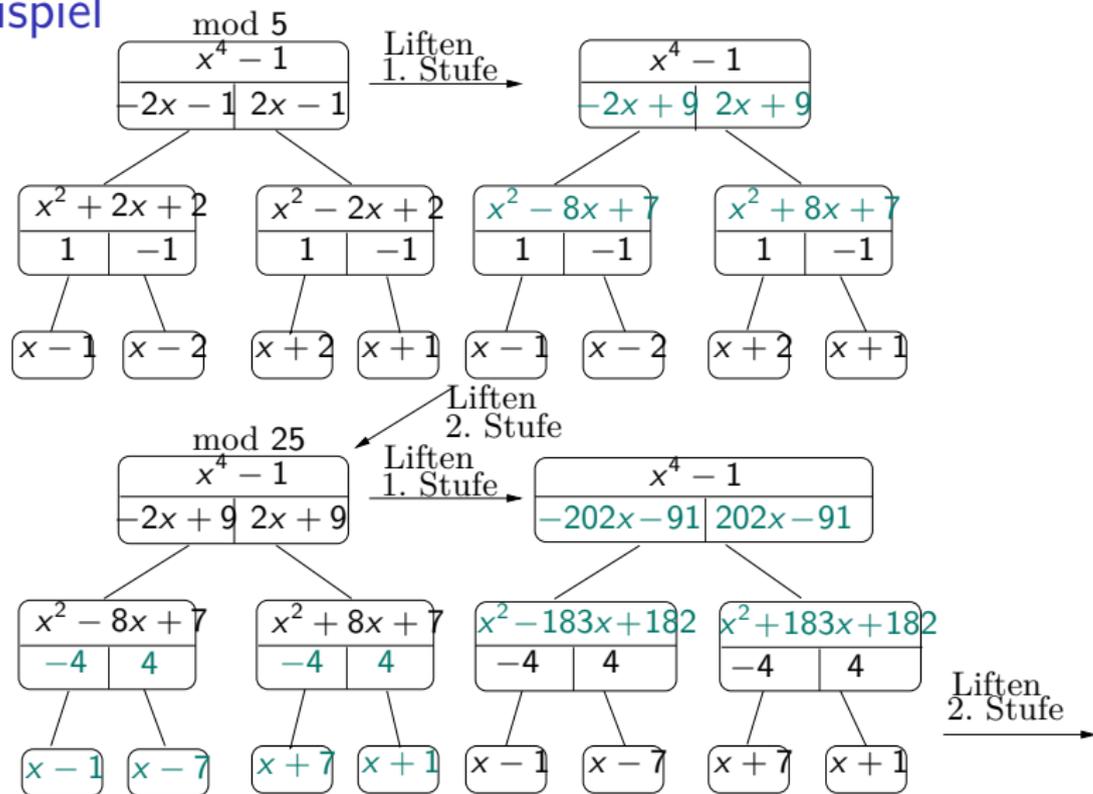
**Beweis:**

- ▶ Schritt 3 ist korrekt nach inversen Berechnung, nach Newton-Quadrat-Iteration; d. h. die Wurzel von  $\tau_j$  ist der monische Vielfache von  $f \bmod m^{2^j}$ .
- ▶ Durch Induktion nach  $j$  zeige  $\tau_j$  ist Faktorbaum von  $f \bmod m^{2^j}$  und jeder Knoten von  $\tau_j$  ist kongruent mod  $m$  zum entsprechenden Knoten von  $\tau$ .
  - ▶  $j = 0$  klar.
  - ▶  $j \geq 1$ .  $\text{Wurzel-}\tau_j \equiv \text{Wurzel-}\tau_{j-1} \pmod m$ .

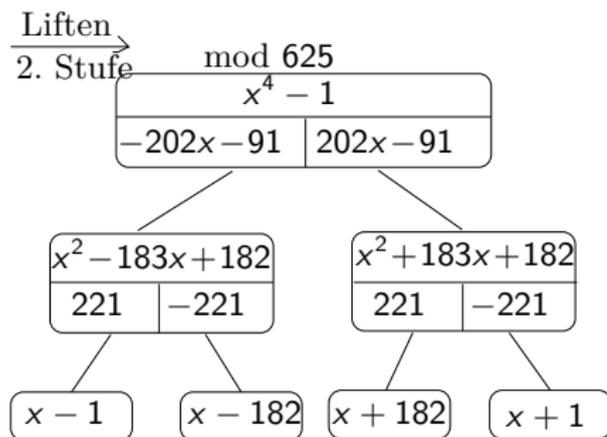
## Algorithmus MFHL Korrektheit(Forts.)

- ▶ Behauptung über  $\tau_j$  folgt nun durch Induktion über Baum und Korrektheit von Hensel Schritt. Wegen  $l \leq 2^d$  ist  $\tau_d$  auch Faktorbaum für  $f \bmod m^l$ .
- ▶  $R = \mathbb{Z}$ : Reduktion der Koeffizienten von  $f$  modulo  $m, m^2, m^4, \dots$  kann mit  $0(nM(l \log m))$  Wortoperationen durchgeführt werden.
- ▶ Schritt 5 kostet  $0(M(\text{grad } v)M(2^j \log m))$  Wortoperationen
- ▶ In einer Stufe von  $\tau_j$  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  $d \in 0(\log r)$  Stufen und die Kosten von 4 und 5 für festes  $j$  ist  $0(M(n) \log r M(2^j \log m))$  Wortoperationen. Dies dominiert Schritt 3. Behauptung folgt aus  $\sum_{1 \leq j \leq d} 2^j \leq 4l$ .
- ▶ Durch balancieren des Faktorbaums bzgl. Grad lässt sich der Faktor  $\log r$  durch die Entropie  $H(n_1/n, \dots, n_r/n)$  ( $n_i = \text{grad } f_i$ ) ersetzen.

## Beispiel

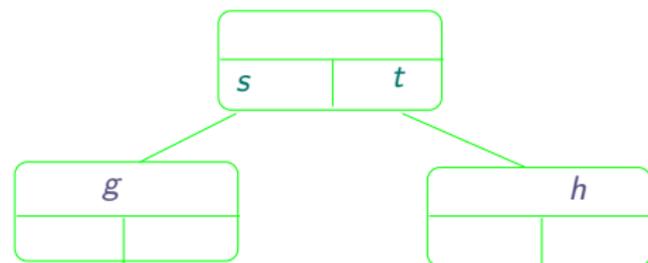


## Beispiel(Forts.)



Ausgang

$$x^4 - 1 \equiv (x - 1)(x - 2)(x + 2)(x + 1) \pmod{5}$$



$$sg + th \equiv 1 \pmod{5}$$



# Algorithmus (Forts.)

## 4 {Hensel Lifting}

$$a := b^{-1} \bmod p$$

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  $f \equiv bg_1 \cdots g_r \bmod p^l$  mit monischen Polynome  $g_1, \dots, g_r \in \mathbb{Z}[x]$  mit max\_norm höchstens  $p^l/2$  so dass  $g_i \equiv h_i \bmod p$  ( $1 \leq i \leq r$ ) 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  $f^*$ }

$$T := \{1, \dots, r\}; s := 1; G := \emptyset; f^* := f;$$

# Algorithmus (Forts.)

## 6 {Faktoren-Kombination}

- while**  $2s \leq \#T$  **do**
- 7   **for** all subsets  $S \subseteq T$  of cardinality  $\#S = s$  **do**
- 8     Compute  $g^*, h^* \in \mathbb{Z}[x]$  mit  $\max\_norm \leq p^l/2$  und  

$$g^* \equiv b \prod_{i \in S} g_i \pmod{p^l} \quad h^* \equiv b \prod_{i \in T \setminus S} g_i \pmod{p^l}$$
- 9     **if**  $\|g^*\|_1 \|h^*\|_1 \leq B$  **then**  

$$T := T \setminus S; G := G \cup \{pp(g^*)\};$$

$$f^* := pp(h^*); b := HKoeff(f^*);$$
**goto** 6;
- 10     $s := s + 1;$
- 11 **return**  $G \cup \{f^*\}$

## Algorithmus (Forts.)

- ▶ Hierbei ist

$$\|f\|_1 = \sum_{0 \leq i \leq \text{grad } f} |f_i|, \quad \|f_1\|_\infty \leq \|f\|_1 \leq (n+1)\|f\|_\infty$$

- ▶  $\|g^*\|_1 \|h^*\|_1 \leq B$  gdw.  $g^*h^* = bf^*$

„↪“ Mignotes Schranke (vzG. S. 156).

„↪“ wegen  $g^*h^* \equiv bf^* \pmod{p^l}$ .

$\|g^*h^*\|_\infty \leq \|g^*h^*\|_1 \leq \|g^*\|_1 \|h^*\|_1 \leq B < p^{l/2}$ , d. h. | alle Koeff. |  $< p^{l/2} \rightsquigarrow$  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  $f \equiv \bar{f} \pmod{5}$ .

$\bar{f}$  ist quadratfrei in  $\mathbb{Z}_5[x]$ .  $B := \sqrt{5} \cdot 2^4 \cdot 15 \cdot 6 \approx 3220$ ,

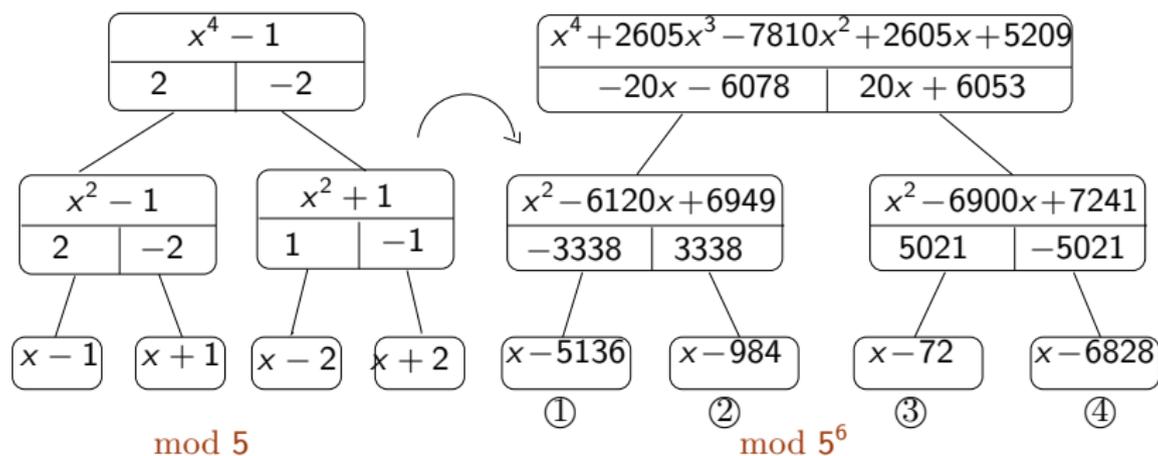
- ▶  $l = \lceil \log_5(2B + 1) \rceil = 6$ .
- ▶ Schritt 3: modulare Faktorisierung

$$f \equiv bh_1 h_2 h_3 h_4 = 1(x - 1)(x + 1)(x - 2)(x + 2) \pmod{5}$$

- ▶ Schritt 4: Liften eines Faktorbaumes für  $f \pmod{5}$  zu Faktorbaum für  $f$  modulo  $5^6$  (  $\text{mod}5^6$  aus Schranke  $l$  ).

## Algorithmus Zassenhaus: Beispiel (Forts.)

Schritt 4:



## Algorithmus Zassenhaus: Beispiel (Forts.)

- ▶ Teilmengen von  $S \subseteq \{1, 2, 3, 4\}$  mit  $S = 1$ : Keine Faktorisierung.

$$S = \{1, 3\} : g^* \equiv bg_1g_3 = 6(x - 5136)(x - 72)$$

$$\equiv 6x^2 + 2x + 2 \pmod{5^6}$$

$$h^* \equiv bg_2g_4 = 6(x - 984)(x - 6828) \equiv 6x^2 + 3x + 12 \pmod{5^6}.$$

$$\|g^*\|_1 \|h\|_1 \leq B, \text{ d. h. } g^*h^* = bf^*.$$

- ▶  $pp(g^*) = 3x^2 + x + 1$   $pp(h^*) = 2x^2 + x + 4$  sind die irreduziblen Faktoren von  $f$ .

# Multivariate Verallgemeinerung von Hensel's Lemma

Problemstellung: Finde  $u, w \in \mathbb{Z}[x_1, \dots, x_\nu]$  mit

$$F(u, w) = a(x_1, \dots, x_\nu) - uw = 0$$

und

$$\begin{aligned}u(x_1, \dots, x_\nu) &\equiv u_0(x_1) \pmod{\langle I, p \rangle} \\w(x_1, \dots, x_\nu) &\equiv w_0(x_1) \pmod{\langle I, p \rangle}\end{aligned}$$

Bei Geg.  $u_0, w_0$  mit  $a \equiv u_0 w_0 \pmod{\langle I, p \rangle}$

$I = \langle x_2 - \alpha_2, \dots, x_\nu - \alpha_\nu \rangle, p$  Primzahl.









Lösung diophantischer Polynomgleichungen in  $\mathbb{Z}_{p^l}[x_1]$ 

Lineare Iteration::

$$s^{(k+1)} = s^{(k)} + \Delta s^{(k)}, t^{(k+1)} = t^{(k)} + \Delta t^{(k)}$$

$$(\#) \quad u(x_1)s_k(x_1) + w(x_1)t_k(x_1) \equiv \frac{1 - s^{(k)}u(x_1) - t^{(k)}w(x_1)}{p^k} \pmod{p}$$

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  $k = 1, 2, \dots, l-1$  wird (#) in  $\mathbb{Z}_p[x]$  gelöst unter Verwendung von  $s^{(1)}, t^{(1)}$ .

Wir erhalten somit folgenden Satz.

Lösung diophantischer Polynomgleichungen in  $\mathbb{Z}_{p^l}[x_1]$ 

**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

- $p \nmid \text{HKoeff}(u(x_1)), p \nmid \text{HKoeff}(w(x_1))$ .
- $\Phi_p(u(x_1))$  und  $\Phi_p(w(x_1))$  teilerfremd in  $\mathbb{Z}_p[x_1]$ .

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

$$\sigma(x_1)u(x_1) + \tau(x_1)w(x_1) \equiv c(x_1) \pmod{p^l}$$

und

$$\text{grad}(\sigma(x_1)) < \text{grad}(w(x_1))$$

Ist  $\text{grad}(c(x_1)) < \text{grad}(u(x_1)) + \text{grad}(w(x_1))$  erfüllt, so gilt auch

$$\text{grad}(\tau(x_1)) < \text{grad}(u(x_1))$$

**Beweis:** Existenz klar. Eindeutigkeit: Siehe Beweis der Eindeutigkeit bei Hensel's quadratischem Lifting.





## Multivariate Taylor Darstellung

Zu lösen ist

$$(*) \quad w^{(k)} \Delta u^{(k)} + u^{(k)} \Delta w^{(k)} \equiv a(x_1 \dots x_\nu) - u^{(k)} w^{(k)} \pmod{\langle I^{k+1}, p' \rangle}$$

wobei  $u^{(k)} w^{(k)}$  die ideal-adische Approximation der Ordnung  $k$  sind, d. h.

$$a(x_1, \dots, x_\nu) - u^{(k)} w^{(k)} \in I^k$$

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

$$(**) \quad w^{(k)} u_i(x_1) + u^{(k)} w_i(x_1) \equiv c_i(x_1) \pmod{\langle I, p' \rangle}$$

Hieraus lassen sich die  $I$ -adischen Koeffizienten  $u_i(x_1), w_i(x_1) \in \mathbb{Z}_{p'}[x_1]$  bestimmen.



## Satz: Multivariate Hensel Konstruktion

- ▶ Dann gibt es für  $k \geq 1$  multivariate Polynome  $u^{(k)}, w^{(k)} \in \mathbb{Z}_{p^l}[x_1, \dots, x_\nu]/I^k$ , so dass

$$a(x_1, \dots, x_\nu) \equiv u^{(k)} w^{(k)} \pmod{\langle I^k, p^l \rangle}$$

$$\text{und } u^{(k)} \equiv u^{(1)}(x_1) \pmod{\langle I, p^l \rangle} \quad w^{(k)} \equiv w^{(1)}(x_1) \pmod{\langle I, p^l \rangle}$$

- ▶ **Eindeutigkeit:** Falls  $a(x_1, \dots, x_\nu)$  monisch bzgl.  $x_1$ , d. h. der Koeffizient in  $a(x_1, \dots, x_\nu)$  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.
- ▶ **Probleme bei der Anwendung:** Leading Coeff. Problem und Bad Zero Problem.  $\rightsquigarrow$  exponentielles Wachstum für Zwischenergebnisse.

## Beispiel

**5.37 Beispiel** Sei  $p = 5$        $I = 1$

$$a(x, y, z) = x^2 y^4 z - xy^9 z^2 + xyz^3 + 2x - y^6 z^4 - 2y^5 z$$

$I = \langle y - 1, z - 1 \rangle$  max  $x$ -Grad 2.

$$a(x, y, z) \equiv x^2 + 2x + 2 \pmod{\langle I, 5 \rangle}$$

Es gilt

$$a(x, y, z) \equiv (x - 2)(x - 1) \pmod{\langle I, 5 \rangle}.$$

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  $l$ -adische Darstellung von  $a(x, y, z)$ :

$$a(x, y, z) \equiv (x^2 + 2x + 2) - (x^2 + 1)(y - 1) + (x^2 + x - 1)(z - 1)$$

$$\begin{aligned}
 &+ (x^2 - x)(y - 1)^2 - (x^2 - 1)(y - 1)(z - 1) + (2x - 1)(z - 1)^2 \\
 &- (x^2 - x)(y - 1)^3 + (x^2 - 2x)(y - 1)^2(z - 1) - \\
 &- (x + 1)(y - 1)(z - 1)^2 + (x - 1)(z - 1)^3 + (x^2 - x)(y - 1)^4 \\
 &+ (-x^2 + 2x)(y - 1)^3(z - 1) - x(y - 1)^2(z - 1)^2 \\
 &+ (x - 1)(y - 1)(z - 1)^3 - (z - 1)^2 - (x - 2)(y - 1)^5 \\
 &+ (x^2 - 2x)(y - 1)^4(z - 1) + x(y - 1)^3(z - 1)^2 \\
 &- (y - 1)(z - 1)^4 + (x - 1)(y - 1)^6 - (2x + 1)(y - 1)^5(z - 1) \\
 &- x(y - 1)^4(z - 1)^2 - x(y - 1)^7 + (2x + 1)(y - 1)^6(z - 1) \\
 &- \dots \\
 &- \dots \\
 &- \dots \\
 &- (y - 1)^6(z - 1)^4 - x(y - 1)^9(z - 1)^2 \pmod{5}
 \end{aligned}$$

## Hensel Konstruktion für das Beispiel

$l$ -adische Darstellung enthält 38 Terme im Vergleich zu 6 Terme in der  $l$ -adischen Darstellung bzgl.  $l = \langle y, z \rangle$ .

**Problem:** Anzahl der zu lösenden polynomialen diophantischen Gleichungen ist proportional zur Anzahl der Terme in der  $l$ -adischen Darstellung von  $a(x, y, z)$ .

Die Hensel Konstruktion für dieses Beispiel liefert

$$\begin{aligned} u^{(7)} = & (x - 2) + (-x + 1)(y - 1) + (x - 2)(z - 1) + x(y - 1)^2 \\ & + (-x - 2)(y - 1)(z - 1) + (-2)(z - 1)^2 + (-x)(y - 1)^3 + \\ & + x(y - 1)^2(z - 1) + (-2)(y - 1)(z - 1)^2 + (z - 1)^3 \\ & + (x)(y - 1)^4 + (-x)(y - 1)^3(z - 1) + (1)(y - 1)(z - 1)^3 \\ & + (x)(y - 1)^4(z - 1) \\ w^{(7)} = & (x - 1) + (-1)(z - 1) + (-1)(y - 1)^5 + (-1)(y - 1)^5(x - 1) \end{aligned}$$

## Hensel Konstruktion für das Beispiel

Ausmultiplizieren mod 5 liefert

$$u^{(7)} \equiv xy^4z + yz^3 + 2 \pmod{5} \quad w^{(7)} \equiv x - y^5z \pmod{5}$$

.

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  $p \nmid \text{HKoeff}(\Phi_I(a(x_1, \dots, x_\nu)))$ .

**Möglichkeit:** Variablentransformation

$$x_j \leftarrow x_j + \alpha_j \quad 2 \leq j \leq \nu, \text{ falls } I = \langle x_2 - \alpha_2, \dots, x_\nu - \alpha_\nu \rangle$$

Problem der Zwischenergebnisse bleibt erhalten.

## Möglichkeit (Forts.)

- ▶ Möglichkeiten zur einfacheren Berechnung siehe G.C.L 262 → dünn besetzte MV Polynome.

$$c_i(x_1) = \frac{1}{n_1! \dots n_m!} \Phi_I \left( \left( \frac{\partial}{\partial x_{j-1}} \right)^{n_1} \dots \left( \frac{\partial}{\partial x_{j_m}} \right)^{n_m} e^{(k)} \right)$$

- ▶ Wang EEZ-GCD Algorithmus:  
Variablenweise

$$\mathbb{Z}_{p^l}[x_1] \rightarrow \mathbb{Z}_{p^l}[x_1, x_2] \rightarrow \mathbb{Z}_{p^l}[x_1, x_2, x_3] \dots$$



# GCD Berechnung - Faktorisierung

## GCD (GGT)-Berechnungen

- ▶ klassisch EEA (euklid. Ringe) Z.B.  $F[x]$   $O(M(n)\log n)$  Körperoperationen.
- ▶ (Pseudo-) Polynomiale Restfolgen, reduzierte PRS (primitiver EA  $\mathbb{Z}[x_1, \dots, x_\nu]$ ) (kleiner Grad  $\leq 2$ )  
Problem Koeffizientenwachstum
- ▶ Sylvester Matrix und Subresultanten
- ▶ Modularer Algorithmus (Brown) **Big-Prime, Small-Primes**
- ▶ p-adisch EZGCD (Moses u. Yun)
- ▶ EEZ-GCD (Wang).
- ▶ GCD-Heuristic

## 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  $r_5(x) = -\frac{1288744821}{543589225}$ , d.h.  $a(x), b(x)$  sind teilerfremd in  $\mathbb{Z}[x]$ .

**Problem:** Koeffizientenwachstum + Berechnung im Quotientenkörper (GGT-Berechnungen).

Modular:  $\phi_{23} : \mathbb{Z}[x] \rightarrow \mathbb{Z}_{23}[x]$ . EAA in  $\mathbb{Z}_{23}[x]$  liefert 1 als GGT.

$\phi_2 : \mathbb{Z}[x] \rightarrow \mathbb{Z}_2[x]$ . EAA in  $\mathbb{Z}_2[x]$  liefert  $x + 1$  als GGT  $\rightsquigarrow$

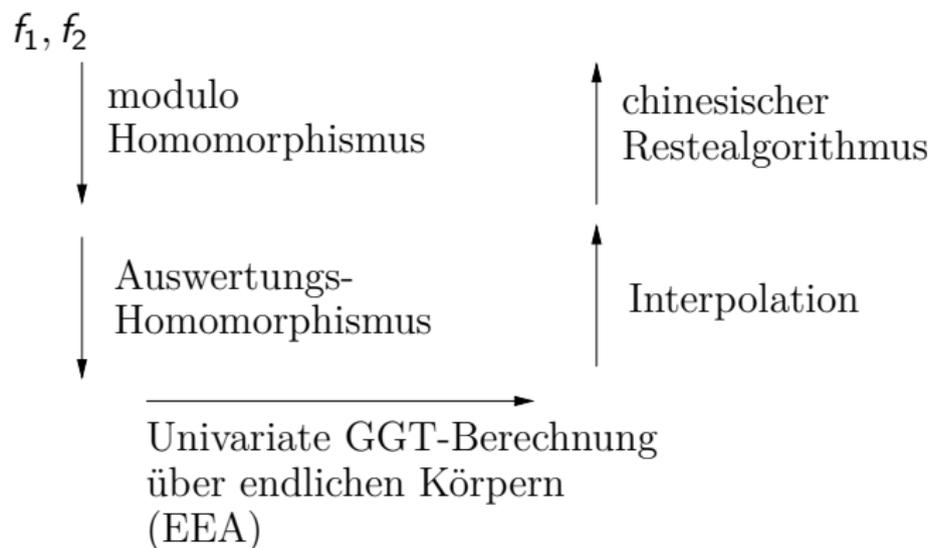
**Unglückliche Homomorphismen.**

p-adisch: EZGCD, EEZ-GCD (Wang)

**Messungen:** Siehe vz Gathen, Gerhard (S 183 Fälle  $\mathbb{Z}[x], F[x, y]$ )

# GCD Berechnungen - Schemata

**Modularer Algorithmus:**  $f_1, f_2$  Multivariate Polynome





# Faktorisierung

Welche Ringe sind interessant für die Faktorisierung

$\mathbb{Z}[x], \mathbb{Q}[x], R[x]$   $R$  ZPE Ring,  $F[x]$   $F$  endlicher Körper,  
 $\mathbb{Z}[x_1, \dots, x_\nu], \mathbb{Q}[x_1, \dots, x_\nu], \mathbb{Q}(\sqrt{2}), \mathbb{Q}(\sqrt{2}, \sqrt{3})$

$R$  ZPE-Ring,  $a \in R[x]$

$\text{cont}(a) = \text{GGT}(\text{Koeff von } a)$  (einheitsnormal)

$pp(a) = a/\text{cont}(a)$  d.h.  $a$  primitiv gdw  $\text{cont}(a) = 1$

Es gilt  $\text{cont}(ab) = \text{cont}(a)\text{cont}(b)$  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!)

## Faktorisierung (Forts.)

Insbesondere für Faktorisierung in  $\mathbb{Q}[x]$  bzw.  $\mathbb{Z}[x]$ :

Ist  $a \in \mathbb{Z}[x]$  primitiv  $\rightsquigarrow a = f_1 \dots f_k$  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  $\mathbb{Z}$ .







## Bestimmung der quadratfreien Faktorisierung

$$\text{Aus } a(x) = \prod_{i=1}^k a_i(x)^i \rightsquigarrow$$

$$a'(x) = \sum_{i=1}^k a_1(x) \cdots i a_i(x)^{i-1} a'_i(x) \cdots a_k(x)^k \text{ also}$$

$$c(x) = \text{GCD}(a(x), a'(x)) = \prod_{i=2}^k a_i(x)^{i-1}. \quad (\text{Beweis!})$$

$$\text{Setzt man } w(x) := a(x)/c(x) = a_1(x) a_2(x) \cdots a_k(x).$$

$\rightsquigarrow w(x)$  ist Produkt der quadratfreien Faktoren ohne Multiplizitäten.

Abspaltung von  $a_1(x)$ :: Sei

$$y(x) = \text{GCD}(c(x), w(x)), \text{ so gilt } a_1(x) = w(x)/y(x).$$

Dann weiter mit  $c(x)$ .  $\rightsquigarrow$  Berechnung der  $a_i(x)$ .









# Yun's quadratfreier Faktorisierungsalgorithmus

**procedure** Square\_Free\_Yun( $a(x)$ )

{Eingabe:  $a(x) \in R[x]$ , primitiv  $\text{char}(R) = 0$ ,  $R$  ZPE,  $\text{grad}(a) = n$ }

{Ausgabe: quadratfreie Faktorisierung von  $a(x)$ }

**begin**

- (1)  $i := 1$ ; **output** := 1;  
 $b(x) := a'(x)$ ;  $c(x) := \text{GGT}(a(x), b(x))$ ;
- (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**  
          $g(x) := \text{GGT}(w(x), z(x))$ ;  
          $outp := outp \sqcup g(x)$ ;  $i := i + 1$ ;  
          $w(x) := w(x)/g(x)$ ;  $y(x) := z(x)/g(x)$ ;  $z(x) := y(x) - w'(x)$   
     **end**
- (3)  $outp := outp \sqcup w(x)$ ;  
**return** ( $outp, i$ );  
**end.**



## Beispiel

### 6.6 Beispiel

Sei  $f = abc^2d^4$  für verschiedene monische irreduziblen Polynome  
 $a, b, c, d \in R[x]$ ,  $c(x) = \text{GGT}(f, f') = cd^3$ .

$$w_1 = f/c(x) = abcd, \quad y_1 = f'/c(x) = a'bcd + ab'cd + 2abc'd + 4abcd'$$

$$z_1 = y_1 - w_1' = abc'd + 3abcd'$$

$$g_1 = \text{GGT}(abcd, abc'd + 3abcd') = ab$$

$$w_2 = abcd/ab = cd, \quad y_2 = (abc'd + 3abcd')/ab = c'd + 3cd'$$

$$z_2 = 2cd'$$

$$g_2 = \text{GGT}(cd, 2cd') = c$$

$$w_3 = cd/c = d, \quad y_3 = 2cd'/c = 2d' \quad z_3 = d'$$

$$g_3 = \text{GGT}(d, d') = 1$$

$$w_4 = d/1 = d, \quad w_4' = d'/1 = d' \quad z_4 = 0$$

$$g_4 = d$$

$(ab, c, 1, d)$  Länge 4.



## 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

$$r, s \in \mathbb{F}_q$$

$$(1) r^q = r$$

$$(2) r^{1/p} = r^{q/p} = r^{p^{m-1}}$$

$$(3) (r + s)^{p^j} = r^{p^j} + s^{p^j}$$

kleiner Fermat Satz  
ist  $p$ -te Wurzel von  $r$   
 $j = 0, 1, \dots, m$

**Beweis:**  $r \in \mathbb{F}_q \rightsquigarrow \{1, r, r^2, \dots\}$  ist zyklisch und endliche Untergruppe der mult. Gruppe von  $\mathbb{F}_q$ . Diese hat die Ordnung  $q - 1 \rightsquigarrow$  Ordnung von  $r$  teilt  $q - 1$  (Lagrange),

$$r^{q-1} = 1 \rightsquigarrow (1)$$

$$(rp^{m-1})^p = r^{p^m} = r^q = r \rightsquigarrow (2)$$

$$\begin{aligned}(r + s)^{p^j} &= r^{p^j} + \binom{p^j}{1} r^{p^j-1} s + \dots + \binom{p^j}{p^j-1} r s^{p^j-1} + s^{p^j} \\ &= r^{p^j} + s^{p^j} \rightsquigarrow (3)\end{aligned}$$



## char $R \neq 0$ $R$ =endlicher Körper char $p$ (Forts.)

Sei  $a = f_1^{e_1} \cdot \dots \cdot f_r^{e_r}$  die irreduzible Faktorisierung von  $a$ . Angenommen für ein  $i$   $1 \leq i \leq r$ ,  $f_i' = 0 \rightsquigarrow f_i$  ist eine  $p$ -te Potenz, d. h.  $f_i$  ist nicht irreduzibel. (d. h. Ableitungen irreduzibler Polynome sind ungleich null), d. h.  $f_i' \neq 0$  und wegen  $\text{grad } f_i' < \text{grad } f_i$  folgt  $\text{GGT}(f_i', f_i)$  ist nicht  $f_i$  und somit 1, da  $f_i$  irreduzibel. Es kann jedoch  $e_i f_i' = 0$  gelten, wenn  $p | e_i$ .

Wegen  $a' = \sum_{1 \leq i \leq r} e_i \frac{a}{f_i} f_i'$  gilt aber  $f_i^{e_i} | a'$ .

Somit gilt

**6.9 Lemma** Sei  $F$  endlicher Körper,  $a \in F[x]$  nicht konstant.

Dann gilt

$$a \text{ ist quadratfrei gdw } \text{GGT}(a, a') = 1$$

$\rightsquigarrow$  **Algorithmus zur QFF in endlichen Körpern**







## Symbolisches Beispiel

**6.11 Beispiel** Symb-Beispiel: Seien  $A, B, C, D$  quadratfreie teilerfremde Polynome in  $\mathbb{Z}_3[x]$  und  $a = A \cdot B^3 \cdot C^5 \cdot D^9$ .

**Initialisierung:**  $b = a' = A' B^3 C^5 D^9 + 2AB^3 C^4 C' D^9$ , d. h.

GGT( $a, b$ ) =  $B^3 C^4 D^9 = c$ .

Die QF-Terme (ohne die 3 | Exponent) sind in  $w = a/c = A \cdot C$

while loop  $y = \text{GGT}(c, w) = C$   $z = w/C = A$ ,  $outp = A$

$i = 2$   $w = C$   $c = B^3 C^3 D^9$

while loop  $y = C$   $z = 1$ ,  $outp = A \sqcup 1$

$i = 3$   $w = C$   $c = B^3 C^2 D^9$

4- while loop Ausgang

$i = 5$   $w = C$   $c = B^3 D^9$   $outp = A \sqcup 1 \sqcup 1 \sqcup 1$

5-Durchgang  $y = 1$   $z = C$   $outp = A \sqcup 1 \sqcup 1 \sqcup 1 \sqcup C^5$

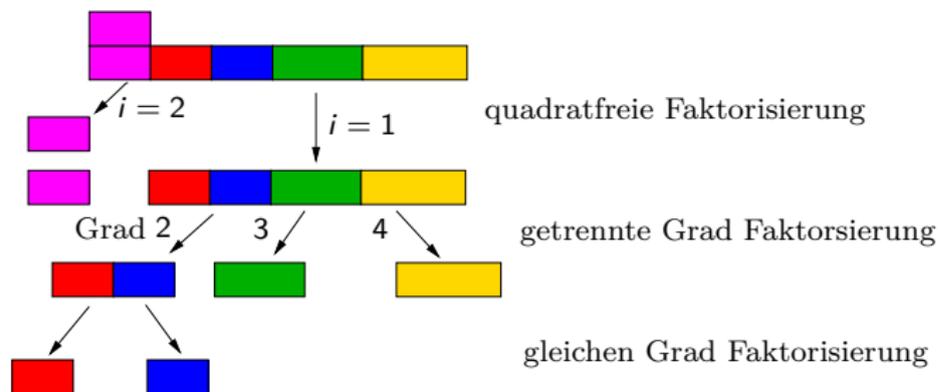
$i = 6$   $w = 1$   $c = B^3 D^9$

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



$a(x) \in \mathbb{F}_q[x]$ ,  $q = p^m$ , 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.

$\text{grad}(a_i) = k \cdot i :: k$  Faktoren mit Grad  $i$ .

## 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  $a \in \mathbb{F}_q$ , d. h.  $x^q - x = \prod_{a \in \mathbb{F}_q} (x - a)$  in  $\mathbb{F}_q[x]$ .

### Allgemeiner

**6.12 Lemma** Für  $d \geq 1$  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  $\text{Grad}(f) = n$ :

$$f \mid x^{q^d} - x \quad \text{gdw} \quad n \mid d$$

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  $a \in F$

$$a^{q^n} = a \text{ und somit } a^{q^d} = \underbrace{((a^{q^n})^{q^n} \dots)}_{s\text{-mal}})^{q^n} = a$$

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.  $h \equiv 0 \pmod{f}$ , und somit  $f \mid h$ .

## Satz von Fermat: Folgerungen

Umgekehrt sei  $f$  monisch, irreduzibel  $\text{grad}(f) = n$ ,  $f \mid x^{q^d} - x$ .

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  $\mathbb{F}_a(a)$ , d. h.  $q^d = (q^n)^s$  für ein  $s$  also  $n \mid d$ .

## 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

$$a_1(x) = \text{GGT}(a(x), x^q - x)$$

zu berechnen.

Setzt man  $a(x) = a(x)/a_1(x)$ , so hat  $a$  keine linearen irreduziblen Faktoren, d. h.

$$a_2(x) = \text{GGT}(a(x), x^{q^2} - x)$$

Usw. Hat  $a(x)$  Grad  $n$ , so muss man nur Faktoren bis zum Grad  $n/2$  bestimmen.

**6.13 Beispiel 1**  $a = x(x+1)(x^2+1)(x^2+x+2) \in \mathbb{F}_3[x]$  getrennte GF

$$(x^2+x, \quad x^4+x^3+x+2)$$

$$\uparrow$$

Grad1

$$\uparrow$$

Grad2

$$\text{GGT}(a, x^3 - x) = x^2 + x, \quad \text{GGT}(a/x^2 + x, x^9 - x) = x^4 + x^3 + x + 2$$

## Beispiel (Forts.)

2)  $a(x) = x^{63} + 1 \in \mathbb{F}_2[x]$ , dann

$$a_1(x) = \text{GGT}(a(x), x^2 - x) = x + 1 \quad \text{1-Faktor Grad 1}$$

$$a(x) = a(x)/a_1(x) = \frac{x^{63}+1}{x+1} = x^{62} + x^{61} + \dots + x^2 + x + 1$$

$$a_2(x) = \text{GGT}(a(x), x^4 - x) = x^2 + x + 1 \quad \text{1-Faktor Grad 2}$$

$$a(x) = a(x)/a_2(x) = x^{60} + x^{57} + x^{54} + \dots + x^6 + x^3 + 1$$

$$a_3(x) = \text{GGT}(a(x), x^8 - x) = x^6 + x^5 + x^4 + x^3 + x^2 + x + 1 \quad \text{2-Faktoren Grad 3}$$

$$a(x) = x^{54} + x^{53} + x^{51} + x^{50} + x^{48} + x^{46} + x^{45} + x^{42} + x^{33} + x^{30} + x^{29} + x^{27} + x^{25} + x^{24} + x^{22} + x^{21} + x^{12} + x^{11} + x^9 + x^8 + x^6 + x^4 + x^3 + x + 1$$

$$\text{GGT}(a(x), x^{16} - x) = 1, \text{GGT}(a(x), x^{32} - 1) = 1,$$

$$\text{GGT}(a(x), x^{64} - x) = a(x) = a_6(x)$$

$$x^{63} + 1 = (x + 1)(x^2 + x + 1)(x^6 + x^5 + x^4 + x^3 + x^2 + x + 1) a_6(x)$$

Grad 1
2
3 – 2Fakt
6 – 9Fakt

# Algorithmus Getrennte Grad Faktorisierung

**procedure** PARTIALFACTOR-DD( $a(x)$ ,  $q$ )

{Eingabe: Quadratfreies mon. Polynom  $a(x) \in \mathbb{F}_q[x]$ ,  $n = \text{grad}(a) > 0$ }

{Ausgabe: Getrennte Grad Zerlegung  $(a_1, \dots, a_s)$ ,  $s \leq n/2$  von  $a(x)$ }

$w := x$ ;  $a_0 := 1$ ;  $i := 0$ ;

**repeat**

$i := i + 1$ ; call wied. quadrat. Algorithm. in  $R = \mathbb{F}_q(x)/\langle a(x) \rangle$

um  $w = w^q \text{ mod } a(x)$  zu berechnen

$a_i := \text{GGT}(w - x, a(x))$ ;

**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**  $(a_1, \dots, a_i)$

Die GGT-Berechnungen  $\text{GGT}(a(x), x^{q^i} - x)$  werden durch Berechnung von  $x^{q^i} - x$  modulo  $a(x)$ , d. h. Berechnung wird in  $\mathbb{F}_q[x]/(a(x))$  durchgeführt (z. B. wiederholtes Quadrieren um  $(x^{q^{i-1}})^q \text{ mod } (a(x))$  zu berechnen).

## 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:

$$w_i \equiv x^{q^i} \pmod{f_i}, \quad f_i = G_{i+1} \cdots G_t, \quad a_i = G_i \text{ für } i \geq 1,$$

wobei  $(G_1, \dots, G_t)$  die getrennte Grad-Zerlegung von  $a$  ist.

Induktion nach  $i$ :  $i = 0$  klar,  $i > 0$  wegen

$$w_i \equiv w_{i-1}^q \equiv (x^{q^{i-1}})^q = x^{q^i} \pmod{f_{i-1}} \text{ d. h. } w_i - x \equiv x^{q^i} - x \pmod{f_i} \text{ und}$$

$$a_i = \text{GGT}(w_i - x, f_{i-1}) = \text{GGT}(x^{q^i} - x, f_{i-1})$$

Also ist  $a_i$  Produkt aller monisch irreduziblen Polynome in  $\mathbb{F}_q[x]$  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.

## Algorithmus Getrennte Grad Faktorisierung (Forts.)

Kosten für die Berechnung von  $w_i$  in Schritt (2)

$0(\log q)$  Multiplikationen mod  $a$ , d. h.  $0(M(n) \log q)$  Operationen in  $\mathbb{F}_q$ .

Die Kosten in (3) und (4) sind ebenfalls  $0(M(n) \log n)$  Operationen in  $\mathbb{F}_q[x]$ .

Berechnung kann gestoppt werden sobald  $\text{grad } f_i = \text{grad } a(x) < 2(i + 1)$ , da alle irreduziblen Faktoren von  $f_i$  grad mindestens  $i + 1$  haben, d. h.  $a(x)$  ist irreduzibel. Mit dieser Überprüfung: **early abort**

Somit  $i = \max\{m_1/2, m_2\} \leq n/2$ , 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.

## Beispiel

**6.15 Beispiel** Sei  $q = 3$  Algorithmenverlauf für

$$a(x) = x^8 + x^7 - x^6 + x^5 - x^3 - x^2 - x \in \mathbb{F}_3[x]$$

$$a'(x) = -x^7 + x^6 - x^4 + x - 1 \quad \text{GGT}(a, a') = 1, \text{ d. h. QF}$$

$$w_1 = x^3 \bmod a = x^3$$

$$a_1 = \text{GGT}(x^3 - x, a) = x \neq 1$$

$$f_1 = a/a_1 = x^7 + x^6 - x^5 + x^4 - x^2 - x - 1 \text{ (neues } a)$$

$$w_1 \text{ unverändert } x^3$$

$$w_2 = w_1^3 \bmod a = x^9 \bmod a = -x^7 + x^6 + x^5 + x^4 - x$$

$$a_2 = \text{GGT}(w_2 - x, f_1) = \text{GGT}(-x^7 + x^6 + x^5 + x^4 + x, f_1)$$

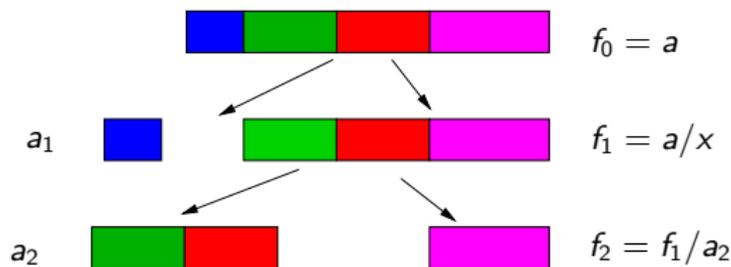
$$= x^4 + x^3 + x - 1$$

$$f_2 = f_1/a_2 = a/a_2 = \frac{x^7 + x^6 - x^5 + x^4 - x^2 - x - 1}{x^4 + x^3 + x - 1}$$

$$= x^3 - x + 1$$

## Beispiel (Forts.)

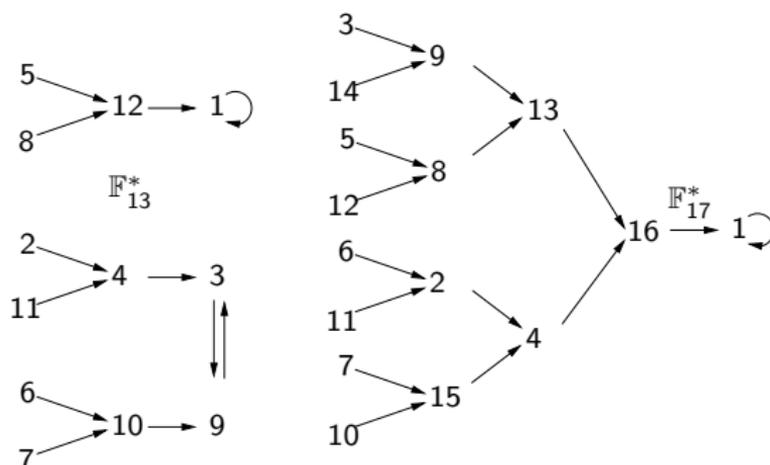
Der Algorithmus würde noch eine Iteration durchführen aber  $\text{grad}(f_2) < 2(2 + 1) = 6 \rightsquigarrow$  nicht notwendig, da  $f_2$  irreduzibel.  $a$  hat einen Lin-Faktor  $x$ , zwei verschiedene irreduziblen quadratische Faktoren, da  $\text{Grad } a_2 = 4$  und einen irreduziblen kubischen Faktor  $x^3 - x + 1$ .





## Gleiche-Grad-Faktorisierung (1)

Betrachte Quadrat-Abbildung  $\sigma : \mathbb{F}_q^* \rightarrow \mathbb{F}_q^*$  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

## 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

1.  $S$  ist eine Untergruppe der Ordnung  $(q - 1)/k$
2.  $S = \{a \in \mathbb{F}_q^* : a^{(q-1)/k} = 1\}$

**Beweis:**  $S$  als Bild eines Homomorphismus  $(\sigma_k : a \rightarrow a^k)$  ist Untergruppe von  $\mathbb{F}_q^*$ .

Der Kern von  $\sigma_k$  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

$q - 1 = |\mathbb{F}_q^*| = |\ker \sigma_k| |\text{Bild } \sigma_k| = |\ker \sigma_k| \cdot |S| \leq k(q - 1)/k = q - 1 \rightsquigarrow$

$|\ker \sigma_k| = k \quad |S| = (q - 1)/k$  und  $S = \ker \sigma_{(q-1)/k}$

## Gleiche-Grad-Faktorisierung (3)

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

1.  $S \subseteq \mathbb{F}_q^*$  ist multiplikative Ugr. der Ordnung  $(q - 1)/2$
2.  $S = \{a \in \mathbb{F}_q^* \ a^{(q-1)/2} = 1\}$
3.  $a^{(q-1)/2} \in \{1, -1\}$  für alle  $a \in \mathbb{F}_q^*$

**Faktorisierungsaufgabe:** Sei  $a \in \mathbb{F}_q[x]$ ,  $\text{grad } a = n$ , 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  $f_i$ .

## Gleiche-Grad-Faktorisierung (4)

Da  $\text{GGT}(f_i, f_j) = 1$  für  $i \neq j$ , gibt es nach chinesischem Restesatz Ring Homomorphismus

$$\chi : R = \mathbb{F}_q[x]/\langle a \rangle \rightarrow \mathbb{F}_q[x]/\langle f_1 \rangle \times \cdots \times \mathbb{F}_q[x]/\langle f_r \rangle = R_1 \times \cdots \times R_r$$

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  $f \bmod a \in R$  und

$$\chi(f \bmod a) = (f \bmod f_1, \dots, f \bmod f_r) = (\chi_1(f), \dots, \chi_r(f)), \text{ wobei } \chi_i(f) = f \bmod f_i \in R_i \text{ gilt.}$$

Es gilt für  $f \in \mathbb{F}_q[x]$ ,  $i \leq r$ ,  $f_i \mid f$  gdw  $\chi_i(f) = 0$ .

Hat man ein  $f \in \mathbb{F}_q[x]$  mit einigen  $\chi_i(f) = 0$  und anderen nicht null, so ist  $\text{GGT}(f, a)$  ein nichttrivialer Teiler von  $a$ .

↪ Probabilistisches Verfahren um Spaltungspolynom  $f$  von  $a$  zu bestimmen.





## Beispiel (Forts.)

Spaltung von  $a_2(x)$  nach Zufallsmuster  $e = (11^2 - 1)/2 = 60$

Zufallspolynom:  $x + 2$

$$\text{GGT}(a_2(x), (x+2)^{60} - 1) = x^6 + 3x^5 + 4x^4 - 2x^3 + 5x^2 + 4x - 2$$

$$a_2(x) = (x^6 + 3x^5 + 4x^4 - 2x^3 + 5x^2 + 4x - 2)(x^4 - 3x^3 + 5x^2 - x + 5)$$

Versuche mit  $x + 7$

$$\text{GGT}(x^4 - 3x^3 + 5x^2 - x + 5, (x+7)^{60} - 1) = x^2 + 3x - 2 \text{ und}$$

$$\text{GGT}(x^6 + 3x^5 + 4x^4 - 2x^3 + 5x^2 + 4x - 2, (x+7)^{60} - 1) = x^4 + 2x^3 + x^2 - 5x - 2$$

3-Faktoren Grad 2, verwende  $x^4 + 2x^3 + x^2 - 5x - 2 \rightsquigarrow$

$$(x^2 + 3x - 2)(x^2 + 5x + 3)(x^2 + 4x + 5)(x^2 - 2x + 4)(x^2 + x + 1) = a_2(x).$$



## Algorithmus (Forts.)

**6.20 Satz** Der Algorithmus ist korrekt bzgl. seiner Spezifikation.

„Failure“ wird mit der Wahrscheinlichkeit  $< 2^{1-r} \leq 1/2$  mit  $r = n/d \geq 2$  ausgegeben.

Die Anzahl der erwarteten Operationen in  $\mathbb{F}_q$  ist  $O((d \log q + \log n)M(n))$ .

**Beweis:**

- ▶ Für  $\text{GGT}(a, f) = 1$ , so  $2^{-r+1}$  als Fehlerwahrscheinlichkeit, wegen Schritt (2):  $< 2^{-r+1}$ .
- ▶ Kosten für die Schritte 2) und 4)  $O(M(n) \log n)$ .
- ▶ Schritt 3:  $2 \log_2(q^d) \in O(d \log q)$  Multiplikationen mod  $a$ , d. h.  $O(M(n)d \log q)$  Operationen in  $\mathbb{F}_q$ .
- ▶ Ruft man den Algorithmus  $k$  mal auf, so gilt  
**Failure Wahrscheinlichkeit**  $< 2^{(1-r)k} \leq 2^{-k}$ .

## 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::  $x^3 - x + 1$
- ▶  $a(x) = x^4 + x^3 + x - 1$  faktorisiert sich in  $r = 2$  irreduziblen Polynome mit Grad  $d = 4/r = 2$ .
- ▶ Angenommen  $f = x + 1$  erste Wahl. Dann ist
 
$$g_1 = \text{GGT}(f, a) = \text{GGT}(x + 1, x^4 + x^3 + x - 1) = 1$$

$$b = (x + 1)^4 \bmod a = (x + 1)^4 \bmod x^4 + x^3 + x - 1 = -1$$

$$g_2 = \text{GGT}(b - 1, a) = \text{GGT}(1, a) = 1$$
Pech gehabt!
- ▶ Zweite Wahl:  $f = x$ . Dann
 
$$g_1 = \text{GGT}(f, a) = \text{GGT}(x, x^4 + x^3 + x - 1) = 1$$

$$b = x^4 \bmod a = -x^3 - x + 1$$

$$g_2 = \text{GGT}(b - 1, a) = \text{GGT}(-x^3 - x, x^4 + x^3 + x^2 - 1) = x^2 + 1 \rightsquigarrow$$
- ▶  $x^2 + 1$  ist einer der irr. Faktoren und  $a/g_2 = x^2 + x + 1$  der andere.













## 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,  $\text{grad } f = n$ , mit  $r \geq 2$  irreduziblen Faktoren  $f_1, \dots, f_r \in \mathbb{F}_q[x]$

$R = \mathbb{F}_q[x]/\langle f \rangle$        $R_i = \mathbb{F}_q[x]/\langle f_i \rangle$        $\chi_i : R \rightarrow R_i$  wie gehabt.

- $x^{2^m} + x = T_m(T_m + 1) \rightsquigarrow T_m(\alpha) \in \mathbb{F}_2$  für  $\alpha \in \mathbb{F}_{2^m}$  ( $T_m$  ist  $\mathbb{F}_2$  linear)  
 $T_m(\alpha) = 0$  und  $T_m(\alpha) = 1$  gleichwahrscheinlich  $1/2$
- 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  $2^{1-r} \leq 1/2$ .
- Berechne  $b = T_{kd}(f) \text{ mod } a$  im Algorithmus Equal\_Degree\_Splitting.

## Eigenschaften

Die wesentlichen Eigenschaften, die verwendet wurden, sind folgende Faktorisierungen:

- Für  $q$  ungerade:

$$* \quad x^q - x = x(x^{(q-1)/2} - 1)(x^{(q-1)/2} + 1)$$

d. h. für  $W = \{v(x) \in \mathbb{F}_q[x] : v(x)^q = v(x) \pmod{a(x)}\}$  und  $v(x) \in W$  ist  $v(x)(v(x)^{(q-1)/2} - 1)(v(x)^{(q-1)/2} + 1) = v(x)^q - v(x) \equiv 0 \pmod{a(x)}$  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

$$** \quad x^{2^k} + x = T_k(x)(T_k(x) + 1)$$

↪ Wahrscheinlichkeit  $\text{GGT}\left(\overset{T_{kd}(f)}{\rightsquigarrow}, a\right)$  nicht trivial  $\geq 1/2$   
 $(f^{(q-1)/2} - 1, a)$

## Vollständiger Faktorisierungsalgorithmus für endliche Körper

**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  $\rightsquigarrow$  QFF-Faktorisierung  $\rightsquigarrow$  DD-Faktorisierung  $\rightsquigarrow$  ED-Faktorisierung.

Aufwand für  $\text{grad } a = n$ : Erwartete Anzahl von OP in  $\mathbb{F}_q$   
 $O(nM(n) \log(qn))$ , 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:  $\sigma : \mathbb{F}_{q^n} \rightarrow \mathbb{F}_{q^n}$ ,  $a \mapsto a^q$ , es gilt  $\sigma^n = id$  und kann als Automorphismus von  $R = \mathbb{F}_q[x]/\langle f \rangle$  für  $f$  quadratfrei betrachtet werden  $\rightsquigarrow$  Iterated Frobenius (Siehe S.374).

## Anwendung: Nullstellen-Bestimmung

Problem: Bestimme Nullstellen von  $a(x) \in \mathbb{F}_q[x]$ .

Es genügt die **linearen irreduziblen Faktoren** von  $a(x)$  zu berechnen, d. h.  $\text{GGT}(x^q - x, a(x)) = g$  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  $\mathbb{F}_q$ .}

- 1 call Repeated Squaring Algorithmus in  $R = \mathbb{F}_q[x]/(a(x))$  zur Berechnung von  $x^q \bmod a(x) =: h$
- 2  $g := \text{GGT}(h - x, a)$   
**if**  $g = 1$  **then** return  $\emptyset$   
**else**
- 3 call Equal\_Degree Fact( $g, 1, q$ )  
 // es werden die irreduziblen linearen Faktoren  $x - u_1, \dots, x - u_r$  mit  $r = \text{grad } g$  berechnet//
- 4 **return**  $u_1, \dots, u_r$

## Anwendung: Nullstellen-Bestimmung (Forts.)

**6.23 Folgerung** Für  $a(x) \in \mathbb{F}_q[x]$  mit  $\text{grad } a = n$  berechnet Root\_Finding alle Nullstellen von  $a(x)$  in  $\mathbb{F}_q$  die erwartete Anzahl von Operationen in  $\mathbb{F}_q$  ist:

$$O(M(n) \log n \log(nq))$$

### 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.

Anwendung: Nullstellen-Bestimmung in  $\mathbb{Z}$ 

**procedure** Root\_Finding\_ $\mathbb{Z}$  ( $a(x), p$ )

{Eingabe: nicht-konstantes Polynom  $a(x) \in \mathbb{Z}[x]$ ,  $\text{grad } a = n$ ,  
 { Max\_Norm ( $a$ ) =  $\| a \|_{\infty} = A$   
 {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  $\{u_1 \bmod p, \dots, u_r \bmod p\}$  Nullstellenmenge in  $\mathbb{F}_p$  von  
 $a(x) \bmod p$ ,  $u_i \in \mathbb{Z}$ ,  $|u_i| < p/2$  alle  $i$  //

3 **for**  $1 \leq i \leq r$

berechne  $v_i \in \mathbb{Z}[x]$  mit  $\text{grad}(v_i) = n - 1$ ,  $\| v_i \|_{\infty} < p/2$  und

$a(x) \equiv (x - u_i)v_i \bmod p$

4 **return**  $\{u_i : 1 \leq i \leq r, |u_i| \leq A \wedge \| v_i \|_{\infty} \leq nA\}$

**end**

## Anwendung: Nullstellen-Bestimmung in $\mathbb{Z}$ (Forts.)

### 6.24 Satz Korrektheit und Komplexität:

Der Algorithmus berechnet alle ganzzahligen Nullstellen von  $a(x)$ .

Die Kosten für **Schritt 2** sind

$$O(M(n) \log n \log(nA) M(\log(nA)) \log \log(nA))$$

und von **Schritt 3 pro  $u_i$**  sind  $O(nM(\log(nA)))$  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.

## 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.  $|u| \leq A < p/2$ .  
d.h. Alle verschiedenen Nullstellen von  $a(x)$  in  $\mathbb{Z}$  können eindeutig aus ihren mod  $p$  Bilder wiedergewonnen werden.
- ▶ **Zeige:**  $a(u_i) = 0$  gdw  $|u_i| \leq A$  und  $\|v_i\|_\infty \leq nA$ .
- ▶ Angenommen  $a(u_i) = 0$  dann  $|u_i| \leq A$  (siehe oben) und  $\|a/(x - u_i)\|_\infty \leq nA < p/2$ . Da  $a/(x - u_i) \equiv v_i \pmod{p}$  und beide Seiten mit  $|\text{Koeff.}| < p/2 \rightsquigarrow$  gleich  $\rightsquigarrow \|v_i\|_\infty \leq nA$ .
- ▶ Umgekehrt: Sei  $|u_i| \leq A$  und  $\|v_i\|_\infty \leq nA$ , dann  $\|(x - u_i)v_i\|_\infty \leq (1 + A)nA < p/2$ , d. h. die Kongruenz  $a(x) \equiv (x - u_i)v_i \pmod{p}$  ist eine Identität.

Beachte die Wahl von  $p \geq B + 1 > 2n(A^2 + A)$  garantiert  $p/2 > A, nA, (1 + A)nA$ .



## 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  $q^{\text{grad } f_i}$  Elementen und enthalten  $\mathbb{F}_q$  (Konstanten mod  $f_i$ ).

- ▶ Für  $f \in \mathbb{F}_q[x]$  gilt

$$f \bmod a \in \ker \beta \iff f^q \equiv f \bmod a$$

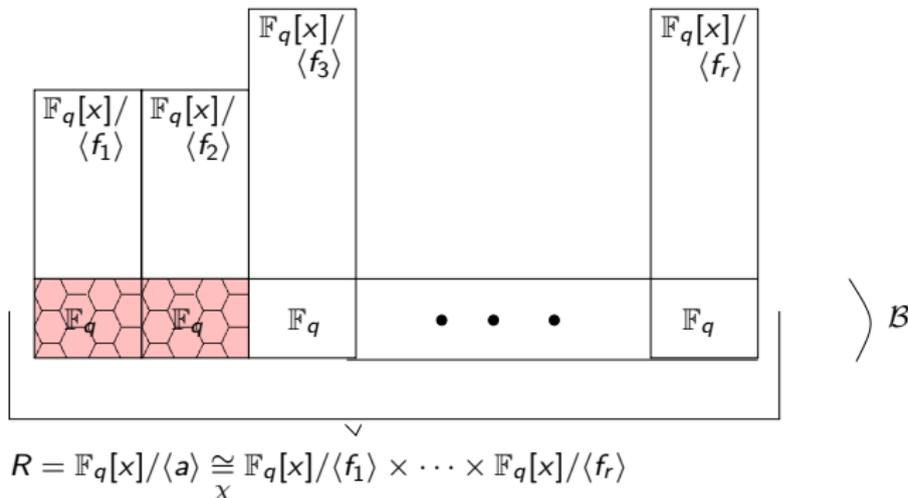
$$\iff f^q \equiv f \bmod f_i \quad \text{für } 1 \leq i \leq r$$

$$\iff f \bmod f_i \in \mathbb{F}_q \quad \text{für } 1 \leq i \leq r$$

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  $\mathbb{F}_q[x]$ ).

# 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.



## Grundlagen für Berlekamps Algorithmen

- ▶  $\mathcal{B}$  ist sogar eine  $\mathbb{F}_q$ -Unteralgebra von  $R$ : Die **Berlekamp-Unteralgebra**.
- ▶ d.h.  $f \bmod a \in \mathcal{B} \iff \chi(f \bmod a) = (a_1 \bmod f_1, \dots, a_r \bmod f_r)$  für Konstanten  $a_1, \dots, a_r \in \mathbb{F}_q$ .
- ▶ Die Matrix  $Q \in \mathbb{F}_q^{n \times n}$ , die den **Frobenius-Hom.**  $\sigma : f \rightarrow f^q$  bezüglich der **Basis**  $x^{n-1} \bmod a, \dots, x \bmod a, 1 \bmod a$  von  $R$  darstellt, heißt **Petr-Berlekamp-Matrix von  $a$** .  $x^{qj} \equiv q_{j,0} + q_{j,1}x + \dots + q_{j,n-1}x^{n-1}$

Der **Berlekamp Faktorisierungsalgorithmus** basiert nun auf folgenden Berechnungen:

- ▶ Bestimme zunächst eine Basis  $b_1 \bmod a, \dots, b_r \bmod a$  von  $\mathcal{B}$  durch **Gauss-Elimination** angewendet auf  $Q - I$ .
- ▶ Beachte:  $a$  ist irreduzibel  $\iff r = 1 \iff \text{Rang}(Q - I) = n - 1$ .  
 $r$  gibt somit die **Anzahl der irreduziblen Faktoren** von  $a(x)$  an.

## Berlekamp Faktorisierungsalgorithmus

- ▶ Angenommen  $q$  sei ungerade und sei  $b = c_1 b_1 + \dots + c_r b_r$  eine zufällige Linearkombination der Basiselemente mit  $c_1, \dots, c_r \in \mathbb{F}_q$  unabhängig gewählt, d. h.  $b \bmod a$  ist ein zufälliges Element aus  $\mathcal{B}$ .
- ▶ Wende  $(q-1)/2$  Trick, wie bei ED-Faktorisierung, an: Die  $b \bmod f_i$  sind gleichmäßig zufällig verteilte Elemente von  $\mathbb{F}_q$  für  $1 \leq i \leq r$ . Falls keines der  $f_i$   $b$  teilt, so ist  $b^{(q-1)/2} \equiv \pm 1 \pmod{f_i}$  und beide Möglichkeiten treten mit Wahrscheinlichkeit  $1/2$  auf, unabhängig für alle  $i$  (Lemma 6.18).
- ▶ **Beachte:** Falls  $b$  nicht konstant ist, so gilt  $a(x) = \prod_{s \in \mathbb{F}_q} \text{GGT}(b-s, a(x))$

$$\text{Da } x^q - x = \prod_{s \in \mathbb{F}_q} (x - s), \text{ d.h. } b^q - b = \prod_{s \in \mathbb{F}_q} (b - s)$$

$$\text{und } f_i \mid a \text{ für alle } i \rightsquigarrow f_i \mid b^q - b = \prod_{s \in \mathbb{F}_q} (b - s)$$

## Berlekamp Faktorisierungsalgorithmus (Forts.)

- Für  $s \neq t$  gilt aber  $\text{GGT}(b - s, b - t) = 1$ , d. h. für gegebenes  $i$  gilt  $f_i \mid b - s_k$  für genau ein  $s_k$ , d. h.

$$a = \text{GGT}(b^q - b, a) = \text{GGT}\left(\prod_{s \in \mathbb{F}_q} (b - s), a\right) = \prod_{s \in \mathbb{F}_q} \text{GGT}(b - s, a)$$

- Berechnung von  $Q$ :

$$\begin{aligned} x^q \bmod a(x), \quad x^{2q} \bmod a(x), \dots, x^{(n-1)q} \bmod a(x) \\ a(x) &= a_0 + a_1x + \dots + a_{n-1}x^{n-1} + x^n \\ x^m &\equiv r_{m,0} + r_{m,1}x + \dots + r_{m,n-1}x^{n-1} \bmod a(x) \\ x^{m+1} &\equiv -r_{m,n-1}a_0 + (r_{m,0} - r_{m,n-1}a_1)x + \dots \\ &\quad + (r_{m,n-2} - r_{m,n-1}a_{n-1})x^{n-1} \\ &\equiv r_{m+1,0} + r_{m+1,1}x + \dots + r_{m+1,n-1}x^{n-1} \end{aligned}$$

- $r_{m+1,0} = -r_{m,n-1}a_0$   
 $r_{m+1,i} = r_{m,i-1} - r_{m,n-1}a_i \quad i = 1, \dots, n-1 \rightsquigarrow O(qn^2)$  Körperoperationen.

## Berlekamp-Algorithmus: Ein echter Faktor

- {Eingabe: Monisches QF-Polynom  $a \in \mathbb{F}_q[x]$ ,  $\text{grad}(a) = n$ ,  $q$  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  $x^q \bmod a$ ;
  - 2 **for**  $0 \leq i < n$  berechne  $x^{qi} \bmod a = \sum_{0 \leq j < n} q_{ij} x^j$   $Q := (q_{ij})_{0 \leq i, j < n}$ ;  
//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  $b_1 \bmod a, \dots, b_r \bmod a$  der Berlekamp Algebra  $\mathcal{B}$  zu bestimmen, hierbei sind  $b_1, \dots, b_r \in \mathbb{F}_q[x]$  mit  $\text{Grad} < n$   
**if**  $r = 1$  **then return**  $a$ ;
  - 4 Wähle unabhängige zufällige  $c_1, \dots, c_r \in \mathbb{F}_q$   $b := c_1 b_1 + \dots + c_r b_r$ ;
  - 5  $g_1 := \text{GGT}(b, a)$ ; **if**  $g_1 \neq 1$  **then return**  $g_1$ ;
  - 6 Call repeated squaring algorithm in  $R = \mathbb{F}_q[x]/a$  zur Berechnung von  $f := b^{(q-1)/2} \bmod a$ ;
  - 7  $g_2 := \text{GGT}(f - 1, a)$ ; **if**  $g_2 \neq 1$  and  $g_2 \neq a$  **then return**  $g_2$   
**else return** „Failure“

## Berlekamp-Algorithmus: Ein echter Faktor

**6.25 Satz** Der Algorithmus ist korrekt und Failure kommt mit Wahrscheinlichkeit  $\leq 1/2$  vor. Er benötigt  $O(n^3 + M(n) \log q)$  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  $\text{gdw } f^{(q-1)/2} \equiv 1 \pmod{f_i}$  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  $O(M(n) \log q)$  Körper Operationen. Schritt 2 verwendet  $n - 2$  Multiplikationen  $\text{mod } a$ , d. h.  $O(nM(n))$  Operationen in  $\mathbb{F}_q$ .

Kosten für Schritt 3  $O(n^3)$  dominiert die Kosten von 2, die  $O(nr)$  Körper für Schritt 4 und die  $O(M(n) \log n)$  Operationen der GGT Berechnungen in den Schritten 5 und 7.

6 benötigt  $O(M(n) \log q)$  Körperoperationen.

# 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  $O(n^3 + M(n) \log r \log q)$ .

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  $\text{grad } n = 100$  über  $\mathbb{F}_{3^{14}}$  benötigt 191 Milliarden Körperoperationen. Generiere Matrix durch binäres Potenzieren. Variante von Zassenhaus (siehe Geddes et Al S 360).

Kaltofen und Lobo: Minimalpolynom-Berechnung.

$O(M(n^2) \log n + M(n) \log q)$ .

Dies ist wichtig, falls  $\log q$  klein im Vergleich zu  $n$  ist.

Variante, Big Prime Berlekamp Algorithm siehe Geddes et.al.

## Beispiel

### 6.26 Beispiel

$$a(x) = x^6 - 3x^5 + x^4 - 3x^3 - x^2 - 3x + 1 \in \mathbb{Z}_{11}[x] = \mathbb{F}_{11}[x]$$

$$Q \text{ Matrix } 6 \times 6, x^{q \cdot j} \equiv q_{j,0} + q_{j,1}x + \cdots + q_{j,5}x^5 \pmod{a}.$$

$$\text{Zeile 0 von } Q \ (1, 0, 0, 0, 0, 0), 1 \equiv 1 \pmod{a(x)}$$

$$x \equiv x \pmod{a(x)}$$

$$x^2 \equiv x^2 \pmod{a(x)}$$

$$x^3 \equiv x^3 \pmod{a(x)}$$

$$x^4 \equiv x^4 \pmod{a(x)}$$

$$x^5 \equiv x^5 \pmod{a(x)}$$

$$x^6 \equiv 3x^5 - x^4 + 3x^3 + x^2 + 3x - 1 \pmod{a(x)}$$

$$x^7 \equiv 3x^6 - x^5 + 3x^4 + x^3 + 3x^2 - x$$

$$\equiv -3x^5 - x^3 - 5x^2 - 3x - 3 \pmod{a(x)}$$

$$\vdots$$

$$x^{11} \equiv 5x^5 - 5x^4 - 3x^3 - 3x^2 + 5x + 3 \pmod{a(x)}$$

## 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  $\Delta$ -Form  
0,1 in diagonalen,  
falls 1 einzige 1  
in Zeile dreiecks-  
idempotenter Form.

$$Q - I = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 4 & -3 & -3 & -5 & 5 \\ 3 & -5 & 5 & 1 & -1 & 0 \\ -2 & 4 & -1 & 2 & -4 & -2 \\ -4 & -3 & -1 & 0 & -1 & -3 \\ -3 & -1 & -4 & -3 & -1 & -4 \end{bmatrix}$$

## Beispiel (Forts.)

$$\rightsquigarrow L = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & -1 & -1 & 0 & 0 \\ 0 & 0 & 4 & 2 & 0 & 0 \end{bmatrix}$$

$\Delta$  idemp-Form,  
Rang 3

$$I - L = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & -4 & -2 & 0 & 1 \end{bmatrix}$$

Basis für Nullraum ablesen,  
da  $(I - L)L = 0$

## 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.

$$v^{(1)}(x) = 1 \quad v^{(2)}(x) = x^4 + x^3 + x^2 + x \quad v^{(3)}(x) = x^5 - 2x^3 - 4x^2$$

$\rightsquigarrow a(x)$  faktorisiert sich in **drei irreduzible Faktoren**.

$$\blacktriangleright \text{GGT}(a(x), v^{(2)}(x)) = x + 1 \rightsquigarrow f_1(x) = x + 1$$

$$\frac{a(x)}{x+1} = x^5 - 4x^4 + 5x^3 - 4x + 1$$

$$\blacktriangleright \text{GGT}(a(x), v^{(2)}(x) + \underset{3,4,5,6}{1}) = 1 \quad \text{GGT}(a(x), v^{(2)}(x) + 2) = 1$$

$$\blacktriangleright \text{GGT}(a(x), v^{(2)}(x) + 7) = x^3 + 2x^2 + 3x + 4 = f_2(x)$$

$$a(x)/g_1(x) = x^2 + 5x + 3 = f_3(x)$$

## Beispiel (Forts.)

### Zufalls-Element aus $\mathcal{B}$

- ▶  $v(x) = 3v^{(1)}(x) - 2v^{(2)}(x) + 5v^{(3)}(x) = 5x^2 - 2x^4 - x^3 - 2x + 3$   
 $\text{GGT}(a(x), v(x)^5 - 1) = x^5 - 4x^4 + 5x^3 + 3x^2 - 4x + 1$
- ▶  $a(x) = (x + 1) \underbrace{(x^5 - 4x^4 + 5x^3 + 3x^2 - 4x + 1)}_{2 \text{ irr. Faktoren}} \rightsquigarrow f_1(x) = x + 1$
- ▶  $v(x) = 2v^{(1)}(x) + 3v^{(2)}(x) + 4v^{(3)}(x) = 4x^5 + 3x^4 - 5x^3 - 2x^2 + 3x + 2$   
 $\text{GGT}(x^5 - 4x^4 + 5x^3 - 2x^2 - 4x + 1, v(x)^5 - 1) = 1$  **Pech!**
- ▶  $v(x) = v^{(1)}(x) + 3v^{(2)}(x) - 4v^{(3)}(x) = -4x^5 + 3x^4 - 3x^2 + 3x + 1$   
 $\text{GGT}(x^5 - 4x^4 \dots, v(x)^5 - 1) = x^2 + 5x + 3$   
 $\rightsquigarrow (x^2 + 5x + 3)(x^3 + 2x^2 + 3x + 4)$  restlichen Faktoren.

## 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  $a \in \mathbb{F}_q[x]$ ,  $n = \text{grad}(a) \geq 1$  ist genau dann irreduzibel, wenn

1.  $a \mid x^{q^n} - x$  und
2.  $\text{GGT}(x^{q^{n/t}} - x, a) = 1$  für alle Primteiler  $t$  von  $n$ .



## Anwendung: irreduzible Polynome (Forts.)

**6.28 Satz** Der Algorithmus ist **korrekt**, d. h. er ist Entscheidungsalgorithmus für Irreduzibilität.

Kosten

$O(M(n) \log q + (n^{(w+1)/2} + n^{1/2} M(n)) \delta(n) \log n)$  Operationen in  $\mathbb{F}_q$ .

Berechne zunächst  $x^q \bmod a \rightsquigarrow O(M(n) \log q)$

$s_m := x^{q^m} \bmod a$  für  $m \in \mathbb{N}$ . In  $O((n^{(w+1)/2} + n^{1/2} M(n)))$ ,  $w \approx 2.376$   
(**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).

$O(n^w + M(n) \log q)$  Körperoperationen.

Für  $w = 3$  ist der hier vorgestellte Algorithmus schneller.

## Arithmetik in $\mathbb{F}_{p^n}$

Wie findet man irreduzible Polynome vom Grad  $n$  in  $\mathbb{F}_p[x]$ ?

Wozu: Arithmetik in  $\text{GF}(p^n)$ , Konstruktion von Körpererweiterungen.

### Probabilistische Verfahren

**6.29 Lemma** Sei  $q$  Primzahlpotenz  $n \geq 1$ . Die Anzahl  $I(n, q)$  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 sein - die Ungleichung

$$\frac{1}{2n} \leq \frac{1}{n} \left( 1 - \frac{2}{q^{n/2}} \right) \leq p_n \leq \frac{1}{n}$$



## Probabilistische Verfahren (Forts.)

- ▶ Es gibt insgesamt  $q^n$  **monische Polynome** vom Grad  $n$  in  $\mathbb{F}_q[x]$ , d. h.

$$\frac{1}{n} \geq \frac{I(n, q)}{q^n} \geq \frac{1}{n}(1 - 2q^{-n/2}) \geq \frac{1}{2n}$$

für  $q^n \geq 16$ .

- ▶ 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  $\mu$  **die Möbius Funktion**, d. h.

$$\mu(n) = \begin{cases} 1 & \text{falls } n = 1 \\ (-1)^k & \text{falls } n \text{ Produkt von } k \text{ verschiedenen} \\ & \text{Primzahlen ist} \\ 0 & \text{falls } n \text{ nicht quadratfrei} \end{cases}$$

- ▶ Werte  $I(5, q)$  für  $q = 2, 3, 4, 5, 7, 8, 9$

6   48   204   624   3360   6552   11808

## Probabilistische Algorithmen zur Erzeugung irreduzibler Polynome mit Grad $n$

**procedure** Ben Or's\_Irreduzible\_Polynome

{**Eingabe:** Primzahlpotenz  $q$ ,  $n \geq 1$ }

{**Ausgabe:** zufälliges monisches irred. Polynom mit Grad  $n$  aus  $\mathbb{F}_q[x]$ }

**begin**

1 wähle zufällig monisches Polynom  $f \in \mathbb{F}_q[x]$  von Grad  $n$

2 **for**  $i = 1, \dots, \lceil n/2 \rceil$  **do**

$g_i := \text{GGT}(x^{q^i} - x, f)$ ;

**if**  $g_i \neq 1$  **then goto** 1

3 **return**  $f$

**end.**

## Probabilistische Algorithmen (Forts.)

- ▶ Wie beim **DD-Fakt Algorithmus** zeigt man, dass die Anzahl der Operationen für Schritt 2 etwa  $O(sM(n) \log(nq))$ ,  $O \sim (n^2 \log q)$

(**Soft 0**  $f \in O \sim (g)$ , 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  $\rightsquigarrow$  erwartete Operationenzahl

$$O \sim (n^3 \log q) \quad (O(nsM(n) \log(nq)))$$

## Probabilistische 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  $O(\log n)$ .

**6.31 Satz** Ben Or's Algorithmus ist korrekt und die erwartete Anzahl von Operationen in  $\mathbb{F}_q$  ist

$$O(nM(n) \log n \log(nq)) \text{ oder } O(n^2 \log q).$$

**Beweis**

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.

## Faktorisation in $R[x_1, \dots, x_n]$ $R$ ZPE Ring

Zusammenfassung: Faktorisation in  $\mathbb{F}_q[x]$ . ok.

- ▶ Faktorisation in  $\mathbb{Z}[x] \cong$  Faktorisation in  $\mathbb{Q}[x]$  + Faktorisation in  $\mathbb{Z}$ .  
d.h. Faktorisation **primitiver Polynome** aus  $\mathbb{Z}[x]$   
→  $a \in \mathbb{Z}[x]$  **primitiv**, o.b.d.A.  $a$  **quadratfrei**.
- ▶ Wähle „Big Prime“  $p \in \mathbb{Z}$ ,  $p \nmid \text{LKoeff}(a)$ ,  $a \bmod p \in \mathbb{F}_p[x]$  **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  $a \bmod p$ , die einen echten Faktor von  $a$  in  $\mathbb{Z}[x]$  entsprechen? Vorgestellte Methode:  
**Versuche alle möglichen Faktoren-Kombinationen.**

**Problem:** Exponentielle Laufzeit möglich.

## Beispiel: Swinnerton-Dyer Polynome

$$a_i(x) = \prod (x \pm \sqrt{2} \pm \sqrt{3} \pm \sqrt{5} \pm \dots \pm \sqrt{p_i}) \in \mathbb{Z}[x]$$

$p_i$   $i$ -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  $\sqrt{2} \bmod p \cdots \sqrt{p_i} \bmod p$ .

### ► Big-Prime Version

- $B := (n+1)^{1/2} 2^n A b$
- $b = LC(a)$
- $n = \text{grad}(a)$
- $A = \text{max-norm}(a)$

$p$  ungerade Primzahl  $2B < p < 4B : a \bmod p$  QF.

$a \equiv b g_1 \cdots g_r \bmod p$ , d. h. in  $\mathbb{Z}_p[x]$ .

## Zusammenfassung: Faktorisierung

► Mit Hensel Lifting: **Zassenhaus Algorithmus**

►  $B := (n + 1)^{1/2} 2^n A b$  wie oben.

►  $C := (n + 1)^{2n} A^{2n-1}$

►  $\gamma = \lceil 2 \log_2 C \rceil$

d. h.  $\gamma \sim 2 \log_2 C \sim 2 \cdot [2n \log(n + 1) + (2n - 1) \log A]$   
 $4n[\log(n + 1) + \log[\max\text{-norm}(a)]]$

Wähle  $p \leq 2\gamma \ln \gamma$ :  $a \bmod p$  QF

$$l = \lceil \log_p(2B + 1) \rceil \text{ Hensel-Lifting nach } \mathbb{Z}_{p^l}[x].$$

**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: Beispiel

Heuristiken: Wähle **mehrere Primzahlen  $p$**

**6.32 Beispiel**  $a(x) = x^{16} + 11x^4 + 121$

- Faktorisierung in  $\mathbb{Z}_{13}[x]$  liefert

$$\begin{array}{ccccccccc}
 a(x) = & u_1(x) & u_2(x) & u_3(x) & \cdots & u_6(x) & \text{mod } 13 \\
 & \downarrow & \downarrow & \downarrow & & \downarrow & \\
 & \text{grad } 2 & 2 & 3 & \cdots & 3 & 
 \end{array}$$

$a(x) = u(x)v(x)$  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  $D_{23} = \{2, 4, 6, 8\}$

$\rightsquigarrow D_{13,23} = \{2, 4, 6, 8\}$  sind die möglichen Grade mindestens eines Faktors  $u(x)v(x) = a(x)$  in  $\mathbb{Z}[x]$ .

$\rightsquigarrow$  Anzahl der **Kombinationen mod 13** reduziert sich auf 25.

## 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  $p \nmid LC(a)$   $p \mid$  alle anderen Koeffizienten von  $a$ ,  $p^2 \nmid a(0) \rightsquigarrow a$  ist irreduzibel in  $\mathbb{Q}[x]$ .

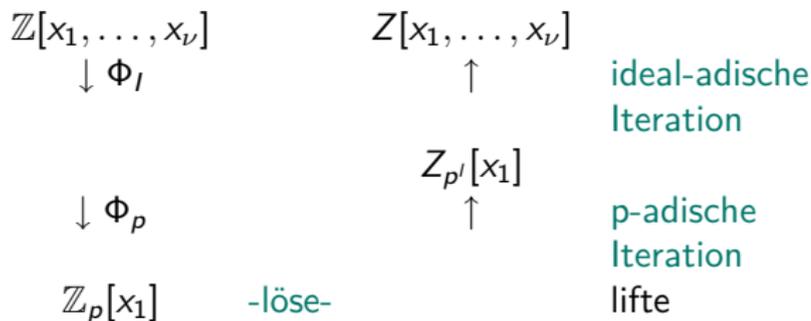
z.B.  $x^n - p$  ist irreduzibel in  $\mathbb{Q}[x]$  für alle  $n, p$ . Übung.

- ▶ Für die Swinnerton-Dyer Polynome gilt: sie lassen sich für jedes  $p$  in linearen und quadratischen Faktoren faktorisieren.

$\rightsquigarrow$  obige Heuristik bringt dafür nichts!

# Multivariate Polynomfaktorisierung in $\mathbb{Q}[x_1, \dots, x_\nu]$ bzw. $\mathbb{Z}[x_1, \dots, x_\nu]$

- ▶ Siehe Homomorphismus-Diagramm Fol. 303



- ▶ Problem Liste der korrekten multivariaten Leit-Koeffizienten.

$a(x_1, \dots, x_\nu) \in \mathbb{Z}[x_1, \dots, x_\nu]$   $x_1$  als Hauptvariable.

$a(x_1, \dots, x_\nu) \equiv u_1(x_1) \cdots u_n(x_1) \in \mathbb{Z}[x_1] \pmod{\Phi_I}$

- ▶ Leitkoeffizienten von  $a(x_1, \dots, x_\nu)$  (als Polynome in  $x_1$ ) ist multivariates Polynom in Variablen  $x_2, \dots, x_\nu$ .

## Multivariate Polynomfaktorisation... (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

$$a(x_1, \dots, x_\nu) = a_d(x_2, \dots, x_\nu)x_1^d + \dots$$

- ▶ Faktorisiere  $a_d(x_2, \dots, x_\nu)$  (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  $\Phi_j$ .** Auswertungspunkte:  $\alpha_2, \dots, \alpha_\nu \in \mathbb{Z}$  mit

1.  $a_d(\alpha_2, \dots, \alpha_\nu) \neq 0$ .
2.  $a(x_1, \alpha_2, \dots, \alpha_\nu)$  quadratfrei.
3. Jeder Faktor von  $a_d(x_2, \dots, x_\nu)$  wenn ausgewertet in  $\alpha_2, \dots, \alpha_\nu$  hat Primzahlfaktor, der nicht in den anderen Auswertungen der restlichen Faktoren vorkommt.

# Faktorisierung in $K[x]$ für $K$ algebraischer Zahlkörper

Anwendung: Symbolische Integration

Träger (Kronecker).

**Algebraische Zahlkörpern**, algebraische Erweiterungen von  $F$ , d. h.

$F(\alpha) = F[x]/\langle m(x) \rangle$ ,  $m$  irreduzibles Polynom in  $F[x]$ .

$\alpha$  ist „Wurzel“ von  $m(x)$  mit Grad  $n$  (z. B.  $x^2 + 1$  in  $\mathbb{Q}[x]$  oder  $\mathbb{R}[x]$ )

$$F(\alpha) = \{[f_0 + f_1\alpha + \cdots + f_{n-1}\alpha^{n-1}] : f_i \in F\}$$

**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

$$\begin{aligned} (a + b\sqrt{2}) + (a' + b'\sqrt{2}) &= (a + a') + (b + b')\sqrt{2} \\ (a + b\sqrt{2})(a' + b'\sqrt{2}) &= (aa' + 2bb') + (ab' + ba')\sqrt{2} \end{aligned}$$

# Grundlagen: Konjugation

- ▶ Sei  $m(x)$  eindeutiges monisches Minimalpolynom von  $\alpha$  über  $F$ . Die **Konjugierten von  $\alpha$  über  $F$**  sind die restlichen verschiedenen Nullstellen von  $m(x)$ . Seien diese  $\alpha_2, \dots, \alpha_n$   
z.B.  $-\sqrt{2}$  ist konjugiert zu  $\sqrt{2}$  über  $\mathbb{Q}$ .
- ▶ Sei  $\beta \in F(\alpha)$  mit  $\beta = f_0 + f_1\alpha + \dots + f_{n-1}\alpha^{n-1}$ . Die **Konjugierten von  $\beta$**  sind  $\beta_2, \dots, \beta_n$ , wobei

$$\beta_i = f_0 + f_1\alpha_i + \dots + f_{n-1}\alpha_i^{n-1}$$

- ▶ Konjugation induziert Isomorphismen:

$$\sigma_i : F(\alpha) \rightarrow F(\alpha_i) \text{ mit } \sigma_i(\beta) = \beta_i$$

## Charakterisierungssatz für $F$

**6.34 Satz** Sei  $\beta \in F(\alpha_1, \dots, \alpha_n)$ .

$\beta \in F$  gdw  $\beta$  invariant unter allen Permutationen der  $\alpha_j$ .

**Beweis:** Fundamentalsatz für symmetrische Funktionen liefert:

$\beta$  kann eindeutig durch die **elementarsymmetrischen Funktionen** von  $\alpha_1, \dots, \alpha_n$  dargestellt werden (z. B.  $\alpha_1 + \dots + \alpha_n, \alpha_1 \cdot \alpha_2 \cdot \dots \cdot \alpha_n, \dots$ ).  
Diese können als **Terme der Koeffizienten des Minimalpolynoms** von  $\alpha$  dargestellt werden, die dann in  $F$  liegen.

**6.35 Definition** Sei **Norm**:  $F(\alpha) \rightarrow F$  definiert durch

- ▶  $\text{Norm}(\beta) = \beta \cdot \beta_2 \cdot \dots \cdot \beta_n$  (d. h. Produkt aller Konjugierten).

Da invariant unter Konjugation, folgt  $\text{Norm}(\beta) \in F$ .

- ▶ Norm kann auch mithilfe der **Resultante** beschrieben werden:

$a[x], b[x] \in R[x]$  nicht null.

$$a(x) = \sum_{i=0}^m a_i x^i \quad b(x) = \sum_{i=0}^n b_i x^i$$

# Sylvester Matrix: Eigenschaften

Sylvester Matrix von  $a$  und  $b$  ist

$$M_{n+m, n+m} = \begin{matrix} n \\ \left\{ \begin{array}{cccccc} a_m & a_{m-1} & \cdots & a_1 & a_0 & \\ & a_m & \cdots & \cdots & a_1 & a_0 \\ & & \vdots & & & \\ & & & a_m & \cdots & \cdots & a_0 \\ m \\ \left\{ \begin{array}{cccccc} b_n & b_{n-1} & \cdots & b_1 & b_0 & \\ & b_n & \cdots & \cdots & \cdots & b_0 \\ & & & \vdots & & \\ & & & b_n & \cdots & \cdots & b_0 \end{array} \right. \end{array} \right. \end{matrix}$$

$\text{Res}_x(a, b)$  ist die Determinante von  $M$ .

Wobei  $\text{Res}(0, b) = 0$  für  $b \neq 0$  und  $\text{Res}(a, b) = 1$  für  $a, b \in R^*$  definiert wird.

## Sylvester Matrix: Beispiel

**6.36 Beispiel** Seien  $a = 3yx^2 - y^3 - 4$ ,  $b = x^2 + y^3x - 9 \in \mathbb{Z}[y][x]$   
dann gilt:

$$\operatorname{Res}_x(a, b) = \det \begin{bmatrix} 3y & 0 & -y^3 - 4 & 0 \\ 0 & 3y & 0 & -y^3 - 4 \\ 1 & y^3 & -9 & 0 \\ 0 & 1 & y^3 & -9 \end{bmatrix}$$

$$= -3y^{10} - 12y^7 + y^6 - 54y^4 + 8y^3 + 729y^2 - 216y + 16$$

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  $b(x) = b_n \prod_{i=1}^n (x - \beta_i)$

$$\operatorname{Res}_x(a, b) = a_m^n b_n^m \prod_{i=1}^m \prod_{j=1}^n (\alpha_i - \beta_j) = (-1)^{mn} b_n^m \prod_{i=1}^n a(\beta_i)$$

## Grundlagen (Forts.)

- ▶ **Eigenschaft:** Zwei Polynome haben einen nichttrivialen gemeinsamen Faktor, falls  $\text{Res}(a, b) = 0$ .

- ▶ **Eigenschaft:** Ist  $q(x)$  monisch, so gilt

$$\text{Res}_x(p, q) = \prod_{x:q(x)=0} p(x), \text{ d. h.}$$

$\text{Norm}(\beta) = \text{Res}_x(b(x), m(x)) = \beta \cdot \beta_2 \cdots \beta_n$ ,  
wobei  $\beta$  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  $\alpha$  und  $z$  betrachtet werden. Setze  
 $\text{Norm}(p) = \text{Res}_x(p(x, z), m(x))$  liefert als Ergebnis ein Polynom in  $F[z]$ .

## Grundlagen (Forts.)

- ▶ Beachte:  $p(\alpha, z) \mid \text{Norm}(p)$  in  $F(\alpha)[z]$ .
- ▶ Die Norm-Funktion ist **multiplikativ**, d. h.  $\text{Norm}(pq) = \text{Norm}(p)\text{Norm}(q)$   
d. h. Jedes Polynom  $p$  welches in  $F(\alpha)[z]$  in Faktoren zerfällt, liefert eine Faktorisierung von  $\text{Norm}(p)$  in  $F[z]$
- ▶ Trager's Algorithmus basiert auf einer Umkehrung dieser Eigenschaft.  
Prozedur:  
**Faktorisiere**  $\text{Norm}(p)$  in  $F[z]$  und lichte diese Faktoren von  $F[z]$  zu Faktoren von  $p(z)$  in  $F(\alpha)[z]$ .
- ▶ Benötigt wir noch **Irreduzibilitätstest** in  $F(\alpha)[z]$ .

## Irreduzibilitätstest für $a(z) \in F(\alpha)[z]$

**6.37 Satz** Sei  $a(z) \in F(\alpha)[z]$  irreduzibel über  $F(\alpha)$ .

Dann ist  $\text{Norm}(a)$  Potenz eines irreduziblen Polynoms über  $F$ .

**Beweis:** Angenommen  $\text{Norm}(a) = b(z)c(z)$  mit teilerfremde Polynome  $b, c \in F[z]$ . Da  $a(z) \mid \text{Norm}(a)$  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
- ▶  $\text{Norm}(a) = \prod_i \sigma_i(a) \mid b(z)$  also  $c(z) = 1$ . d.h.  $\text{Norm}(a) = b(z)$  und  $b(z)$  ist entweder irreduzibel oder Potenz eines irreduziblen Elements.

## 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  
Norm( $a$ ) = Norm( $a_1$ )Norm( $a_2$ )  $\cdots$  Norm( $a_k$ ), wobei jedes 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  $F$ .

## Umkehrung

**6.38 Satz** Sei  $a(z) \in F(\alpha)[z]$ ,  $\text{Norm}(a)$  quadratfrei. Ist  $p_1(z), \dots, p_k(z)$  eine vollständige Faktorisierung von  $\text{Norm}(a)$  über  $F[z]$ , so ist

$$a(z) = \prod_{i=1}^k \text{GGT}(a(z), p_i(z))$$

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  $\text{Norm}(a) = \text{Norm}(a_1) \cdots \text{Norm}(a_k)$  eine Faktorisierung von  $\text{Norm}(a)$  in  $F[x]$ . d.h.  
Für jedes  $i$  gilt  $p_i(z) = \text{Norm}(a_j)$  für ein geeignetes  $j$ .  
Da  $\text{Norm}(a)$  quadratfrei, folgt  $\text{Norm}(a_j) \neq \text{Norm}(a_h)$  für  $h \neq j$ .

## Umkehrung (Forts.)

- ▶ Behauptung: Gilt  $p_i(z) = \text{Norm}(a_j)$ , so

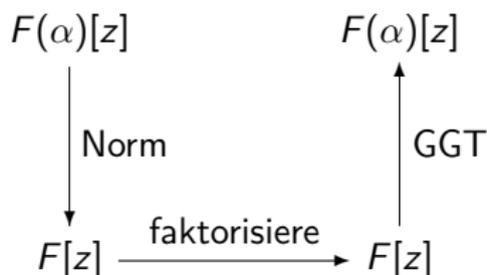
$$(*) \quad a_j(z) = \text{GGT}(a(z), p_i(z))$$

wobei der GGT in  $F(\alpha)[z]$  genommen wird.

- ▶ Aus der Definition der Norm und den Eigenschaften folgt  $a_j(z) \mid a(z)$  und  $a_j(z) \mid p_i(z)$  in  $F(\alpha)[z]$ . Ein größerer Teiler würde 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  $\text{Norm}(a_h) \mid \text{Norm}(p_i)$ . Aber  $p_i(z) \in F[z]$  so folgt  $\text{Norm}(p_i) = p_i(z)^n$
- ▶ Aber  $\text{Norm}(a_h)$  ist irreduzibel, d. h.  $\text{Norm}(a_h) = p_i(z) \nmid$  falls  $h \neq j$ , d. h.  $(*)$  gilt.

Faktorisierung in  $F(\alpha)[z]$ 

- ▶ 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  $\text{Norm}(a(z - s\alpha))$  quadratfrei bis auf endlich vielen  $s \in F$ .

**Beweis:**

- ▶ Sei  $\text{Norm}(a) = \prod_{i=1}^r p_i(z)^i$  quadratfreie Faktorisierung von  $\text{Norm}(a)$  in  $F[z]$ .
- ▶ Da  $a(z)$  quadratfrei ist und  $a(z) \mid \text{Norm}(a) \rightsquigarrow a(z) \mid p(z) = p_1(z) \cdots p_r(z)$ . Beachte hierbei  $p(z) \in F[z]$ .
- ▶ Seien  $\beta_1, \dots, \beta_k$  die Wurzeln von  $p(z)$ , d.h.  $p(z) = \prod (z - \beta_i)$ .
- ▶ Da  $p(z)$  quadratfrei ist, so sind alle  $\beta_i$  verschieden.

## Beweis (Forts.)

- ▶ Sei  $s \in F$ , dann

$$c_s(z) := \text{Norm}(p(z - s\alpha)) = \prod_j \prod_i (z - (s\alpha_j + \beta_i)) \in F[z].$$

- ▶ Dieses Polynom kann mehrfach Wurzeln haben gdw

$$s \cdot \alpha_j + \beta_i = s \cdot \alpha_u + \beta_v \text{ für } j, i, u, v \text{ geeignet gdw } s = \frac{\beta_v - \beta_i}{\alpha_j - \alpha_u}$$

- ▶ d. h. für fast alle  $s \in F$  gilt  $\text{Norm}(a(z - s\alpha)) \mid p(z - s\alpha) \mid c_s(z)$  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. Zahlkörpern

**procedure** Alg\_Factorization( $a(z), m(x), \alpha$ )

{**Eingabe:** quadratfrei  $a(z) \in F(\alpha)[z]$ }

{ $\alpha$  Alg-Zahl mit minimal Polynom  $m(x)$ , grad  $n$ ,  $a$  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 <sub>$x$</sub> ( $m(x), a_s(x, z)$ )

**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 <sub>$x$</sub> ( $m(x), a_s(x, z)$ )

2 //Faktorisiere Norm( $a_s$ ) in  $F[z]$  und lifte Ergebnis //

$b := \text{factors}(\text{Norm}(a_s))$ ;

**if** sizeof( $b$ ) = 1 **then** return  $a(z)$

**else**

**for each**  $a_i(z) \in b$  **do**

$a_i(\alpha, z) := \text{GGT}(a_i(z), a_s(\alpha, z))$

$a_i(\alpha, z) := a_i(\alpha, z + s\alpha)$

substitute( $a_i(z) \leftarrow a_i(\alpha, z), b$ )

**return** ( $b$ )

# Beispiel

## 6.40 Beispiel Sei

$$a_\alpha(z) = z^4 + z^3 + (2 + \alpha - \alpha^2)z^2 + (1 + \alpha^2 - 2\alpha^3)z - 2 \in \mathbb{Q}(\alpha)[z]$$

Mit  $\alpha = 3^{1/4}$  minimal Polynom für  $\alpha$ ,  $m(x) = x^4 - 3$ .

$$\begin{aligned} \blacktriangleright \text{Norm}(a_\alpha(z)) &= \text{Res}_x(a_\alpha(z), m(x)) = \\ &= z^{16} + 4z^{15} + 14z^{14} + 32z^{13} + 47z^{12} + 92z^{11} + 66z^{10} + 120z^9 \\ &\quad - 50z^8 - 24z^7 - 132z^6 - 40z^5 - 52z^4 - 64z^3 - 64z^2 - 32z + 16 \end{aligned}$$

Ist QF in  $\mathbb{Q}[z]$

$\blacktriangleright$  Faktorisiert man in  $\mathbb{Q}[z]$  so  $\text{Norm}(a_\alpha(z)) = g(z) \cdot h(z)$  mit

$$\begin{aligned} g(z) &= z^8 + 4z^7 + 10z^6 + 16z^5 - 2z^4 - 8z^3 - 20z^2 - 8z + 4 \\ h(z) &= z^8 + 4z^6 + 9z^4 + 4z^2 + 4 \end{aligned}$$

## Beispiel (Forts.)

- ▶ Berechne GGT in  $\mathbb{Q}(\alpha)[z]$

$$\text{GGT}(a_\alpha(z), g(z)) = z^2 + (1 - \alpha)z + (1 - \alpha^2) \text{ und}$$

$$\text{GGT}(a_\alpha(z), h(z)) = z^2 + \alpha z + (1 + \alpha^2)$$

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  $\text{Norm}(a)$ .
- **Koeffizientenwachstum**

**Forschungsgegenstand**

- Vermeidung von Wachstum
- Berechnungen mit Körpererweiterungen
- Galois Korrespondenz

# Inhalt Kapitel 7

## Primzahltesten und Faktorisierung in $\mathbb{Z}$

- 7.1 Primzahltesten
- 7.2 Primzahlen mit vorgebenen Eigenschaften
- 7.3 Faktorisierung in  $\mathbb{Z}$
- 7.4 Anwendung: Cryptosysteme





## Primzahltesten: Grundlagen

- ▶  $\mathbb{Z}_N^{\times} = \{a \bmod N \in \mathbb{Z}_N : \text{GGT}(a, N) = 1\}$  mult. Einheiten Gruppe in  $\mathbb{Z}_N$ ,  
**Ordnung**  $\varphi(N) = |\mathbb{Z}_N^{\times}|$  **Euler Funktion**.

$$N = p_1^{e_1} \cdots p_r^{e_r} \text{ paarweise verschieden } p_i : \mathbb{Z}_N^{\times} \cong \mathbb{Z}_{p_1^{e_1}}^{\times} \times \cdots \times \mathbb{Z}_{p_r^{e_r}}^{\times}$$

$$\begin{cases} N \text{ Primzahl, so } \varphi(N) = N - 1 \\ N = p^l, \text{ so } \varphi(N) = p^{l-1}(p - 1) \end{cases} \rightsquigarrow \varphi(N) = \prod_{i=1}^r p_i^{e_i-1}(p_i - 1)$$

**Fermat:**  $a^{N-1} \equiv 1 \pmod{N}$ ,  $N$  Primzahl,  $\text{GGT}(a, N) = 1$

- ▶ Für  $a, N \in \mathbb{Z}^+$  mit  $\text{GGT}(a, N) = 1$ . Sei die **Ordnung** von  $a \bmod N$   
 $\text{ord}_N(a) = \min_{k \geq 1} a^k \equiv 1 \pmod{N}$ .

- ▶ **Euler:**  $a^{\varphi(N)} \equiv 1 \pmod{N}$ , d. h.  $\text{ord}_N(a) \mid \varphi(N)$

- ▶ Falls  $p$  Primzahl,  $l \geq 2$ ,  $N = p^l$ ,  $a = 1 + p^{l-1}$ . Dann  $\text{ord}_N(a) = p$ , da  
 $a^p \equiv \sum_{0 \leq i \leq p} \binom{p}{i} p^{i(l-1)} \equiv 1 \pmod{p^l}$









# Starker Pseudoprimalitäts-Test

{Eingabe: ungerade Zahl  $N \geq 3$ }

{Ausgabe: „Nicht Primzahl“, „mögliche Primzahl“, „echter Teiler von  $N$ “}

**begin**

1 Wähle  $a \in \{2, \dots, N - 2\}$  zufällig

2  $g := \text{GGT}(a, N)$     **if**  $g > 1$  **then** return  $g$

3 Sei  $N - 1 = 2^k m$  mit  $k, m \in \mathbb{N}$ ,  $k \geq 1$ ,  $m$  ungerade.

    Berechne  $b_0 := a^m \bmod N$  //durch wiederholtes quadrieren

**if**  $b_0 = 1$  **then** return „mögliche Primzahl“

4 **for**  $1 \leq i \leq k$  **do**  $b_i := b_{i-1}^2 \bmod N$

5 **if**  $b_k = 1$  **then**  $j := \min\{0 \leq i < k : b_{i+1} = 1\}$

**else** return „nicht Primzahl“

6  $g := \text{GGT}(b_j + 1, N)$ ;

**if**  $g = 1$  or  $g = N$  **then** return „mögliche Primzahl“

**else** return  $g$

**end**



## 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  $b_j^2 \equiv b_{j+1} = 1 \pmod{N}$  in 5.
- ▶  $x^2 - 1 \in \mathbb{Z}[x]$  hat höchstens 2 Nullstellen, d. h. Quadratwurzeln von  $1 \pmod{N}$  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  $I = \{i : 0 \leq i \leq k \text{ und } \forall u \in \mathbb{Z}_N^x \ u^{2^i m} = 1\}$   
 $k \in I$  nach Definition der Carmichael Zahlen,  $i \in I$ , so auch  $i + 1 \in I$ . Für  $i < k$

## Starker Pseudoprimality-Test (Forts.)

- Sei  $p \in P$  und  $a \in \mathbb{Z}$  nicht Quadrat mod  $p$  mit  $a^{(p-1)/2} \not\equiv 1 \pmod p$ . (existiert nach Lemma 6.18). Da  $a^{p-1} \equiv 1 \pmod p$ , folgt  $\text{ord}_p(a) \mid p-1$ , aber teilt nicht  $(p-1)/2$ , d. h.  $\text{ord}_p(a)$  ist gerade.

Da  $m$  ungerade ist, folgt  $a^m \not\equiv 1 \pmod p$ .

Chinesischer Restsatz liefert ein  $u \in \mathbb{Z}_N^x$  mit  $u^m \not\equiv 1$ , d. h.  $0 \notin I$ .

Es gibt somit  $l < k$  mit  $l \notin I$  und  $l+1 \in I$ .

- Sei

$$G = \{u \in \mathbb{Z}_N^x : u^{2^l m} = \pm 1\} \subseteq \mathbb{Z}_N^x$$

**Behauptung:**  $G \neq \mathbb{Z}_N^x$ . Es gibt  $p \in P$  und  $b \in \mathbb{Z}$  mit  $\text{GGT}(b, p) = 1$  mit  $b^{2^l m} \not\equiv 1 \pmod p$ , da sonst  $l \in I$ . Wähle ein solches Paar.

Chinesischer RS liefert:

Es gibt  $c \in \mathbb{Z}$  mit  $c \equiv b \pmod p$  und  $c \equiv 1 \pmod N/p$ .

Also  $c \pmod N \in \mathbb{Z}_N^x \setminus G$ , d. h.  $G$  hat höchstens  $|\mathbb{Z}_N^x|/2 = \varphi(N)/2$  Elemente wählt man  $a$  mit  $a \in \mathbb{Z}_N^x \setminus G$ , so liefert Algorithmus echten Teiler. (S 496)

# Andere Primzahltests

- ▶ Solovay-Srassen (74) basiert auf **Legendre Symbol**: Für  $a, N \in \mathbb{Z}$ ,  $N$  Primzahl sei

$$\left(\frac{a}{N}\right) = \begin{cases} 1 & \text{falls } \text{GGT}(a, N) = 1 \text{ und } a \text{ ist Quadrat } \text{mod} N \\ -1 & \text{falls } \text{GGT}(a, N) = 1 \text{ und } a \text{ ist nicht Quadrat } \text{mod} N \\ 0 & \text{falls } \text{GGT}(a, N) \neq 1 \end{cases}$$

- ▶ Gauß quadratisches Reziprozitätsgesetz:  $\left(\frac{a}{N}\right) = \left(\frac{N}{a}\right)$  für ungerade Primzahlen, bis auf  $a, N$  beide kongruent  $3 \pmod{4}$ .
- ▶ **Jacobi Symbol**: Für  $N = p_1^{e_1} \dots p_r^{e_r}$  definiere

$$\left(\frac{a}{N}\right) = \left(\frac{a}{p_1}\right)^{e_1} \dots \left(\frac{a}{p_r}\right)^{e_r}$$

“leicht” zu berechnen. Analogie zu EEA (siehe Übung 18.23 vzG,G)



# Primzahltesten ist in **P**

Siehe

<http://www.cse.iitk.ac.in/news/primality.html>

## Finden von Primzahlen mit vorgebenen Eigenschaften

z. B. Primzahl mit  $n$ -Bits, Big-Prime Methoden, Hensel Lifting Methoden.

### ► Primzahlsatz:

- Seien  $\Pi(x) = |\{p \in \mathbb{N} : p \leq x, p \text{ Primzahl}\}|$ ,  $p_n = n \text{ te Primzahl}$
- 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  $B < p \leq 2B$  ( $B$  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.

## 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  $|P| = \Pi(2B) - \Pi(B) \geq B/2 \ln B \quad (B \geq 6) \rightsquigarrow \frac{|P|}{B} \geq \frac{1}{2 \ln B}$
- ▶ Erwarteten Wahlen:  $2 \ln B$  und jede kostet  $O(k \log B \cdot M(\log B))$

**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

$$\text{Prob}\{M \equiv 0 \pmod p : p \in P\} \leq \frac{\log_a C}{|P|}$$

(es gibt höchstens  $\log_a |M| \leq \log_a C$  Primzahlen in  $P$ , die  $M$  teilen).



# Kosten und Voraussetzungen für Primzahlen in Algorithmen

		cost for		
	modular algorithm	prime requirements	prime finding	algorithm
deter- minant	big	$p > 2n^{n/2} A^n$	$n^3 \log^3 A$	$n^4 \log A$
	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 finding	big	$p > 2n(A^2 + A)$	$\log^3 A$	$n^2 \log A + n \log^2 A$
	power	$p < 2r \ln r$	$n \log A$	$n^2 \log A$
factor- ization	big	$p > \sqrt{n+1} 2^{n+1} A^2$	$n^3 + \log^3 A$	$n^{10} + n^8 \log^2 A$
	power	$p < 2r \ln r$	$n \log A$	$n^{10} + n^8 \log^2 A$

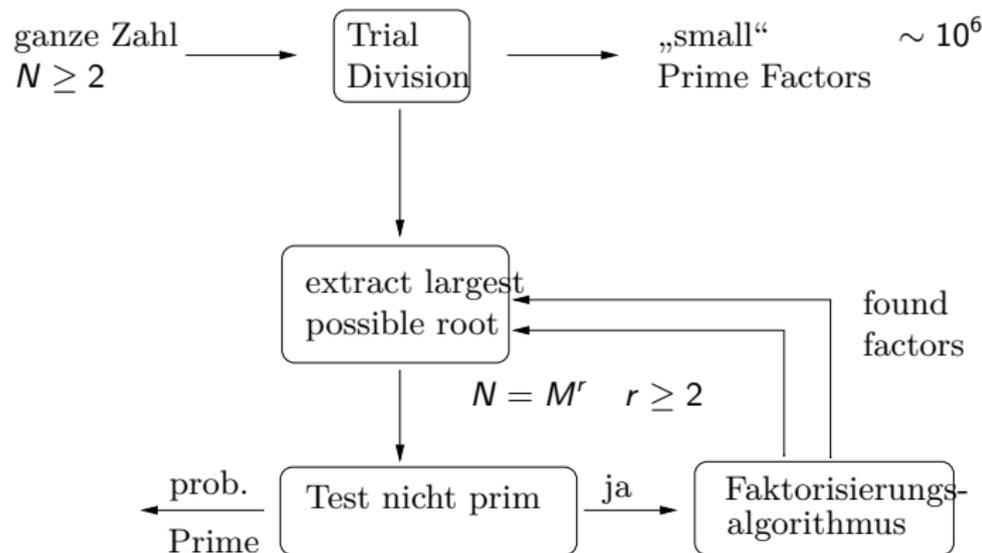
# Faktorisierung in $\mathbb{Z}$ (Übersicht)

Methode	Jahr	Zeit
Trialdivision	$-\infty, \infty$	$O \sim (2^{n/2})$
Pollard's $p - 1$ Methode	1974	$O \sim (2^{n/4})$
Pollard's $\rho$ -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	$\exp(O \sim (n^{1/3}))$

# 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  $P$ “  $\rightsquigarrow$  wichtig für Cryptosysteme.
- ▶ **Tabellen**: Cunningham Projekt 1925. Faktorisierungen von  $b^n \pm 1$ .
- ▶ **“Most Wanted“ Factorizations**.
  - ▶ RSA Cryptosystem:  $p_1, p_2$  große Pzahl  $p_1 p_2$ .  
130-Digits RSA Zahlen geknackt  $\sim 2 \times 65^96$
  - ▶ Faktorisierung Fermat'scher Zahlen  $F_y = 2^{2^y} + 1$   
 $F_{10}, F_{11}$  ellipt. Kurven, „most wanted 98“  $2^{569} - 1 / (\text{PF})$  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).

# Format der Faktorisierungsalgorithmen



siehe Kap. 19, vz. Gathen/Gerhard  
 + Kap.20: Anwendung Public Key Cryptography.

## 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 gegebenenfalls.
- ▶ 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  $N = 2^d 3^e b^r$   $\text{GGT}(b, 6) = 1$   $r$  maximal in  $O(\log N \cdot M(\log N))$  Wortoperationen.  
 (Siehe Aufgaben 9.44 und 18.6 in vzG,G).

# 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  $\rho$

$$\text{Prob}(S_1(N) > N^{0.85}) \approx 0.20 \quad \text{Prob}(S_2(N) > N^{0.30}) \approx 0.20$$

- $O\sim(N^{0.30})$  erwartete Schrittcomplexität für 1.

## Pollard und Strassen Methode

Sei  $a \mapsto \bar{a}$  die mod  $N$  Reduktion und  $1 \leq c \leq \sqrt{N}$ . Betrachte

$$F = (x+1)(x+2)\cdots(x+c) \in \mathbb{Z}[x] \quad f = \bar{F} \in \mathbb{Z}_N[x]$$

Dann gilt  $c^2! = \prod_{0 \leq i < c} f(i\bar{c})$ . 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  $f = \prod_{1 \leq j \leq c} (x + \bar{j}) \in \mathbb{Z}_N[x]$ ;

2 Berechne  $g_i \in \{0, \dots, N-1\}$  mit  $g_i \bmod N = f(i\bar{c})$  für  $0 \leq i < c$ ;

3 Falls  $\text{GGT}(g_i, N) = 1$  für  $0 \leq i < c$  **then return** „Failure“

$k \leftarrow \text{Min} \{0 \leq i < c : \text{GGT}(g_i, N) > 1\}$

4 **return**  $\text{Min} \{kc + 1 \leq d \leq kc + c : d \mid N\}$

**end**

# 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  $0 \leq i < c$  gilt:

- ▶ Ein Primteiler  $p$  von  $N$  teilt  $F(ic)$  und somit auch  $\text{GGT}(g_i, N) = \text{GGT}(F(ic) \bmod N, N)$  gdw.  $p$  teilt Zahl im Intervall  $\{ic + 1, \dots, ic + c\} \rightsquigarrow$  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  $O(M(\log N))$ .
- ▶ Platz für  $O(b^{1/2})$  Zahlen der Größe  $O(\log N)$
- ▶ Schleife mit  $b = 2^i, (i = 1, 2, \dots, b > S_2(N))$  liefert vollständige Faktorisierung in  $O(M(S_2(N)^{1/2})M(\log N) \log N)$ .

## Pollards $\rho$ -Methode (1975)

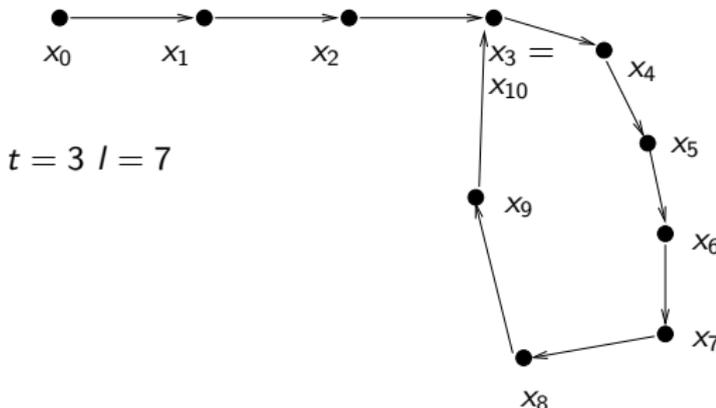
### Idee

Wähle Funktion  $f : \mathbb{Z}_N \rightarrow \mathbb{Z}_N$  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  $x_0, x_1, x_2, \dots$  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  $x_t \equiv x_{t+l} \pmod p$
- ▶ Ist  $N$  keine Primzahlpotenz und  $q$  ein weiterer  $p$ -Teiler von  $N$ , so sind, für unabhängige Reste modulo  $N$ ,  $x_i \pmod p$  und  $x_i \pmod q$  ebenfalls unabhängige Zufallsvariablen (Chin. Restsatz).
- ▶ D.h. mit großer Wahrscheinlichkeit  $x_t \not\equiv x_{t+l} \pmod q$  und somit  $\text{GGT}(x_{t+l} - x_t, N)$  ist nicht trivialer Faktor von  $N$ .

## Pollards $\varrho$ -Methode (Forts.)

- ▶ **Frage:** Wie groß sind  $t, l$ ?  
Offenbar  $t + l \leq p$  und der erwartete Wert ist  $O(\sqrt{p})$  für eine Zufallsfolge  $(x_i)_{i \in \mathbb{N}}$ .
- ▶ **Geburtstagsproblem:** Wieviel Personen benötigt man um eine Wahrscheinlichkeit (zwei Personen mit gleichem Geburtstag zu haben)  $\geq 1/2$  zu erhalten (23 reichen 50,7%)
- ▶ Auswahl (mit Wiederholung) aus Urne mit  $p$  Marken. Die erwartete Anzahl von Wahlen bis zu einer Kollision ist  $O(\sqrt{p})$
- ▶ **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  $x_i = x_{i+l}$  für alle  $i \geq t$  für  $t \in \mathbb{N}$

Pollards  $\rho$ -Methode (Forts.) $l, t$  seien minimal

Speichere Folge  
bis  $x_i = x_j$   
 $O(t + l)$   
zuviel Platz

Floyd's 1-step/2-step cycle detection method::

Führe zweite sequenz mit  $y_i = x_{2i}$  speichere nur  $x_i, y_i$  bis  $x_i = y_i = x_{2i}$ 

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**  $i$

## Floyd's 1-Step/2-Step cycle Detection Method

**7.7 Lemma** FLOYD\_Cycle\_Det\_ALG hält nach höchstens  $t + l$  Iterationen.

**Beweis:** Da  $x_{2i} = y_i$  für alle  $i$  gilt:

- ▶  $x_i = y_i$  gdw.  $i \geq t$  und  $l \mid (2i - i) = i$ , und der kleinste Index ist  $i = t + (-t \text{ REM } l) < t + l$  falls  $t > 0$  und  $i = l$  falls  $t = 0$ .

★ **Pollard's  $\rho$ -Methode zur Faktorisierung von  $N$ :**

Erzeuge Folge  $x_0, x_1, \dots \in \{0, \dots, N - 1\}$  wie folgt:

$x_0$  wird zufällig gewählt,  $x_{i+1} = f(x_i) = x_i^2 + 1 \text{ REM } N$ .

- ▶ Sei  $p$  kleinste Primzahl die  $N$  teilt  $\rightsquigarrow x_{i+1} \equiv x_i^2 + 1 \pmod{p}$  für  $i \geq 0$ . Kollision mod  $p$  kann nach  $O(\sqrt{p})$  Schritte erwartet werden. Verwende hierfür FLOYD'S-ALG.

# Pollards $\rho$ -Methode zur Faktorisierung

## 3. Pollard\_ $\rho$ \_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  $i \leftarrow i + 1$ ;  $x_i \leftarrow x_{i-1}^2 + 1 \pmod N$ ;  $y_i \leftarrow (y_{i-1}^2 + 1)^2 + 1 \pmod N$ ;

4  $g \leftarrow \text{GGT}(x_i - y_i, N)$ ;

**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  $S_2(N)^{1/2} \sim (\log^2 N) \approx 0 (N^{1/4})$

Pollards  $S$ -Methode zur Faktorisierung (Forts.)7.9 Beispiel  $N = 82123$   $x_0 = 631$ 

$i$	$x_i \bmod N$	$x_i \bmod 41$	$y_i \bmod N$	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		

$$N = 41 \cdot 2003$$

$$x_{38} \equiv 4430 \equiv x_{143} \pmod{N}$$

# Dixon's Random Square Faktorisierungsmethode

- ▶ Erstes Verfahren mit Aufwand kleiner als  $\exp(\varepsilon \cdot \log N)$  für jedes  $\varepsilon > 0$

Idee: 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  $t = \lceil \sqrt{N} \rceil, \lceil \sqrt{N} \rceil + 1, \dots$   
 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:  $N = 2027651281 \quad \sqrt{N} \approx 45029$

$$N = 45041^2 - 1020^2 = 46061 \cdot 44021$$

## Dixon's Random Square Methode (Forts.)

- ▶ Variante: Wähle  $k \ll N$   $t = \lceil \sqrt{kN} \rceil, \lceil \sqrt{kN} \rceil + 1, \dots$  und teste ob  $t^2 - kN$  perfektes Quadrat.  
Falls  $t^2 - kN = s^2$  so  $\text{GGT}(s + t, N)$  ist hoffentlich nichttrivialer Faktor von  $N$ , so dass  $s \not\equiv \pm t \pmod N$
- ▶ Das finden von Relationen der Form  $s^2 \equiv t^2 \pmod N$  auf dieser Weise ist für große  $N$  sehr unwahrscheinlich.

### 7.10 Beispiel

$N = 2183$  Angenommen wir haben folgende Kongruenzen

$$453^2 \equiv 7 \pmod N \quad 1014^2 \equiv 3 \pmod N \quad 209^2 \equiv 21 \pmod N$$

Dann  $(453 \cdot 1014 \cdot 209)^2 \equiv 21^2 \pmod N$  oder

$$687^2 \equiv 21^2 \pmod N \quad \rightsquigarrow$$

$$37 = \text{GGT}(687 - 21, N) \quad 59 = \text{GGT}(687 + 21, N)$$

Dieses ist auch die Faktorisierung von  $N$

## Dixon's Random Square Methode (Forts.)

▶ Systematisches Vorgehen:

Wähle  $b$  zufällig und hoffe, dass  $b^2 \text{ REM } N$  Produkt kleiner Primzahlen ist. Sind genügend solcher gefunden, so erhält man eine Kongruenz  $s^2 \equiv t^2 \pmod{N}$ . Dann  $\text{GGT}(s - t, N)$  bzw.  $\text{GGT}(s + t, N)$

▶ Faktorisierungsbasis Primzahlen  $p_1, \dots, p_h$  bis zu einer Schranke  $B \in \mathbb{R}^+$

Eine Zahl  $b$  heißt  **$B$ -Zahl** falls  $b^2 \text{ REM } N$  (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  $B \geq 7$  und  $N = 2183$

■ Für eine  $B$ -Zahl  $b$  sei  $b^2 \equiv p_1^{\alpha_1} p_2^{\alpha_2} \dots p_h^{\alpha_h} \pmod{N}$  mit  $\alpha_1 \dots \alpha_h \in \mathbb{N}$ .  
Assoziiere dazu **Binären Exponenten Vektor**

$$\varepsilon = (\alpha_1 \bmod 2, \alpha_2 \bmod 2, \dots, \alpha_h \bmod 2) \in \mathbb{F}_2^h$$

▶ Für  $B$ -Zahl  $b_i$ , sei  $b_i^2 \equiv \prod_{1 \leq i \leq h} p_i^{\alpha_{ij}} \pmod{N}$

## 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

$$\left( \prod_{1 \leq i \leq l} b_i \right)^2 = \prod_{1 \leq j \leq h} p_j^{\sum_{1 \leq i \leq l} \alpha_{ij}} = \prod_{1 \leq j \leq h} p_j^{2\gamma_j} = \left( \prod_{1 \leq j \leq h} p_j^{\gamma_j} \right)^2 \pmod N$$

wobei  $\gamma_j = \frac{1}{2} \sum_{1 \leq i \leq l} \alpha_{ij}$  (durch 2 teilbar nach Voraussetzung)

- ▶ Dann  $s^2 \equiv t^2 \pmod N$  mit

$$s = \prod_{1 \leq i \leq l} b_i \quad t = \prod_{1 \leq j \leq h} p_j^{\gamma_j}$$

- ! Man benötigt nicht mehr als  $h + 1$  B-Zahlen, d.h.  $l \leq h + 1$ , da jede Menge von  $h + 1$  Vektoren in  $\mathbb{F}_2^h$  linear abhängig ist.

## Dixon's Random Square Methode (Forts.)

- ! Die Hoffnung ist nun  $s, t$  gefunden zu haben mit  $s \not\equiv \pm t \pmod{N}$
- ▶ Ist  $N$  keine Primzahlpotenz mit  $r \geq 2$  verschiedene Primfaktoren, so folgt aus Chinesischer-Restsatz, dass jedes Quadrat in  $\mathbb{Z}_N^*$  genau  $2^r$  Quadratwurzeln in  $\mathbb{Z}_N$  hat.
- ▶ Ist somit  $s$  eine zufällige Quadratwurzel von  $t^2$  so gilt

$$\text{Prob } \{s \equiv \pm t \pmod{N}\} = \frac{2}{2^r} \leq \frac{1}{2}$$

- ▶ 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 $\gamma_1 = \gamma_3 = 0$ $\gamma_2 = \gamma_4 = 1$		
$s = 453 \cdot 1014 \cdot 209$	$t = 2^0 \cdot 3^1 \cdot 5^0 \cdot 7^1$	

## 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  $b^2 \text{REM} N$  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:  $O(h^3)$
  - ▶  $\rightsquigarrow O(h^3 + k(h+n)M(n))$  Wortoperationen.

# Dixon's Random Squares Methode zur Faktorisierung

## 4. Dixon's\_Random\_Squares\_Faktorisierung

{Eingabe: Ungerade Zahl  $N \geq 3$ , weder Prim noch perfekte Potenz  $B \in \mathbb{R}^+$ }  
 {Ausgabe: entweder echter Teiler oder „Failure“}

1 Berechne alle P-Zahlen  $p_1, \dots, p_h \leq B$

**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  $b \in \{2, \dots, N - 2\}$

$g \leftarrow \text{GGT}(b, N)$  **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**  $\alpha \leftarrow (\alpha_1, \dots, \alpha_h)$ ,  $A \leftarrow A \cup \{(b, \alpha)\}$

7 **until**  $\#A = h + 1$

## Dixon's Random Squares Methode zur Faktorisierung

- 8 Finde verschiedene Paare  $(b_1, \alpha^{(1)}), \dots, (b_l, \alpha^{(l)}) \in A$  mit  $\alpha^{(1)} + \dots + \alpha^{(l)} \equiv 0 \pmod{2}$  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  $(\gamma_1, \dots, \gamma_h) \leftarrow \frac{1}{2}(\alpha^{(1)} + \dots + \alpha^{(l)});$   
 $s \leftarrow \prod_{1 \leq i \leq l} b_i; t \leftarrow \prod_{1 \leq j \leq h} p_j^{\gamma_j}; g \leftarrow \text{GGT}(s+t, N);$   
**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  $B \geq 59$  nach PZ-Satz
- ▶ Die Erwartete Anzahl  $k$  erfüllt dann  $\frac{N}{h^{2r}}(2r)! < n^{2r} \rightsquigarrow$  Schranke.
- ▶ Sei  $\Psi(x, y) = \{a \in \mathbb{N} : 1 \leq a \leq x, \forall p \text{ prim } p \mid a \Rightarrow p \leq y\}$  die Menge der  **$y$  glatten Zahlen**.  $b$  ist B-Zahl gdw.  $b^2 \text{REM } N \in \Psi(N, B)$ .

## 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  $l^e \mid p - 1$  erfüllen  $l^e \leq B$ , (sie sind also  $B$  zahn für geeignet gewähltes  $B$ ).

### 5. Pollard's $p - 1$ Faktorisierung

{Eingabe:  $N \geq 3, B > 0$ }

{Ausgabe: entweder echter Teiler von  $N$  oder „Failure“}

- 1  $k \leftarrow \text{KGV}\{i : 2 \leq i \leq B\}$
  - 2 Wähle  $a \in \{2, \dots, N - 2\}$  zufällig
  - 3  $b \leftarrow a^k \bmod N$ ;  $d \leftarrow \text{GGT}(b - 1, N)$ ;
  - 4 **if**  $1 < d < N$  **then return**  $d$  **else return** „Failure“
- ▶ Hoffnung  $d$  ist nichttrivialer Teiler von  $N$ . Die Annahme garantiert  $a^k \equiv 1 \pmod{p}$ , da  $p - 1 \mid k$ , d.h.  $d > 1$ . Um  $d < N$  zu garantieren, genügt es, wenn  $N$  einen weiteren Primfaktor  $q$  hat, mit  $a^k \not\equiv 1 \pmod{q}$
  - ▶ Grundlage ist die Gruppe  $G = \mathbb{Z}_N^\times$ . Hoffnung ist  $|G \bmod p|$  ist  $B$ -Zahl.

## Lentra's Elliptische-Kurven Methode (1987)

- ▶ Statt  $G = \mathbb{Z}_N^x$ , werden Gruppen von Elliptischen-Kurven und deren Ordnung als Testkandidat  $B$ -Zahl zu sein, verwendet.

Algebraische Geometrie

- ▶ Sei  $F$  Körper mit  $\text{char} \neq 2, 3$  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  $P, Q \in E$  in Punkt  $S$ . Dann  $R = P + Q = -S$ . Spezialfälle  $P = Q$  (Tangente),  $P + O = -(-P) = P$  und  $P + (-P) = -O = O$ .

## Lentra's Elliptische-Kurven Methode (1987)

- ▶ Die Größe elliptischer Kurven über  $\mathbb{F}_q$ :
- ▶ Sei  $E$  elliptische Kurve über  $\mathbb{F}_q$ ,  $\text{char} > 3$ , dann gilt  $|E| \leq 2q + 1$ .
- ▶ Hasse's Schranke:  $||E| - (q + 1)| \leq 2\sqrt{q}$
- ▶ 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  $\mathbb{Z}_4 \times \mathbb{Z}_2$ .
- ▶  $|E| \cdot P = O$ . Die Ordnung eines Elements ist wie üblich definiert.

- ▶ Rationale Funktionen zur Berechnung der Summe  $(x_1, y_1) + (x_2, y_2)$

$$x_1 \neq x_2 \rightsquigarrow x_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_1 - x_2, y_3 = -y_1 + \frac{y_2 - y_1}{x_2 - x_1} \cdot (x_1 - x_3)$$

$$(x_1, y_1) = (x_2, y_2) \rightsquigarrow x_3 = \left(\frac{3x_1^2 + a}{2y_1}\right)^2 - 2x_1, y_3 = -y_1 + \frac{3x_1^2 + a}{2y_1} \cdot (x_1 - x_3)$$

# Lenstra's Elliptische-Kurven Faktorisierungsalgorithmus

## 5. Lenstra's\_Elliptic\_Curve\_Faktorisierung

{Eingabe: Ungerade Zahl  $N$ ,  $3 \nmid N$ , keine perfekte Potenz,  $B \in \mathbb{R}^+$ ,  $\min_{p|N} \leq C$ }  
 {Ausgabe: entweder echter Teiler oder „Failure“}

- 1 Wähle zufällig  $(a, u, v) \in \{0, \dots, N-1\}^3$ ;  
 $b \leftarrow v^2 - u^3 - au$ ;  $g \leftarrow \text{GGT}(4a^3 + 27b^2, N)$ ;  
**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  $p_1 = 2 < \dots < p_h \leq B$ ;  
 $P \leftarrow (u, v)$ ;  $Q \leftarrow P$ ;  $t \leftarrow 1$ ;
- 3 **for**  $1 \leq j \leq h$  **do**  
 $e_j \leftarrow \lfloor \log_{p_j}(C * 2\sqrt{C} + 1) \rfloor$ ;  
**for**  $0 \leq r < e_j$  **do** //Invarianten:  $t = p_i^j \prod_{1 \leq r \leq i} p_r^{e_r}$  und  $Q = tP$ //
- 4 Versuche  $p_j Q$  in  $E$  über  $\mathbb{Z}_N$  zu berechnen;  
**if** ein Zähler  $w$  Nullteiler in  $\mathbb{Z}_N$  **then return**  $\text{GGT}(w, N)$   
**else**  $Q \leftarrow p_j Q$ ;  $t \leftarrow p_j t$ ;
- 5 **return** „Failure“

## Analyse von Lenstra's Faktorisierungsalgorithmus

- ▶ Benötigt wird:  $E \bmod p$  ist elliptische Kurve für jeden Primteiler  $p \mid N$ .  $E$  ist i.A. keine Gruppe mit  $+$  definiert durch die Gleichungen, d.h die rationalen Ausdrücke müssen nicht  $\bmod N$  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  $E \bmod p$ , d.h. die elliptische Kurve über  $\mathbb{Z}_p$  mit Koeffizienten  $a, b \bmod p$ . Zu  $P \in E$  sei  $P_p$  der korrespondierende Punkt  $P \bmod p$ .  $O_p$  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 ( $p \mid \text{GGT}(w, N)$ ), implementieren die Berechnungen die Arithmetik von  $E_p$  in folgenden Sinn:: Ein partielles Ergebnis  $Q = tP$  in  $E$  liefert  $(\bmod p)$  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  $P_q$  in  $E_q$  ist.

## Analyse von Lenstra's Faktorisierungsalgorithmus

**7.11 Lemma** Angenommen  $(E, P)$  ist gewählt,  $p, q \mid N$  verschieden,  $l$  sei der größte Primfaktor der Ordnung von  $P_p$  in  $E_p$ ,  $p \leq C$ ,  $|E_p|$  sei  $B$ -glatt und  $l \nmid |E_q|$ . Dann wird  $N$  vom Algorithmus faktorisiert.

**Beweis:** Sei  $k = \prod_{1 \leq r \leq h} p_r^{e_r}$ ,  $e_r$  wie in 3.

- ▶ Da  $|E_p|$   $B$ -glatt ist und  $p \leq C$ , folgt aus der Hasse Schranke:  $|E_p| \parallel k$ .
- ▶ Sei  $d$  die Ordnung von  $P_p$  in  $E_p$ . Dann  $d \parallel |E_p|$  und somit  $l \leq B$  und  $d \mid k$ .
- ▶ Sei  $p_i = l$  und  $e$  der Exponent von  $l$  in  $d$ , d.h.  $1 \leq e \leq e_i$ . Ist  $j = e - 1$  so

$$t = l^{e-1} \prod_{1 \leq r < i} p_r^{e_r} \text{ und } Q = tP \text{ vor Schritt 4. } t \not\equiv 0 \pmod{d} \text{ und } lt \equiv 0 \pmod{d}$$

Somit  $Q_p = tP_p \neq O_p$  und  $lQ_p = ltP_p = O_p$ . Wir zeigen, der Algorithmus kommt **nicht** bis zu dieser Stelle. Angenommen  $lQ = O$ , dann auch  $lQ_q = (ltP)_q = O$ . Da aber  $l \nmid |E_q|$  muss bereits  $Q_q = tP_q = O_q$  gelten und somit  $Q = O$ . Aber dann  $Q_p = O_p \neq$

## 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

$$\text{prob}\{|E_p| \in S\} \geq \frac{c |S|}{\sqrt{p} \log p}$$

**7.13 Folgerung** Sei  $p \leq C$  ein Primteiler von  $N$  und  $\sigma = |\{B\text{-glatte Zahlen in } (p + 1 - \sqrt{p}, p + 1 + \sqrt{p})\}|$ . Dann erfüllt die Anzahl  $M$  der Tripel  $(a, u, v) \in \{0, \dots, N - 1\}^3$  für die der Algorithmus  $N$  faktorisiert

$$\frac{M}{N^3} \geq \frac{c_1 \sigma}{\sqrt{p} \log p} \text{ für ein } c_1 \in \mathbb{R}^+$$

## Laufzeitanalyse von Lenstra's Faktorisierungsalgorithmus

- ▶ Die Laufzeit hängt wesentlich ab von der Anzahl der Auswahlen die der Algorithmus benötigt um mit großer 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

$$\text{prob}\{d \text{ ist } x^{\frac{1}{u}} \text{ glatt}\} = u^{-u(1+o(1))}$$

- ▶ Unter der Annahme der Vermutung, kann man eine erwartete Laufzeitschranke von

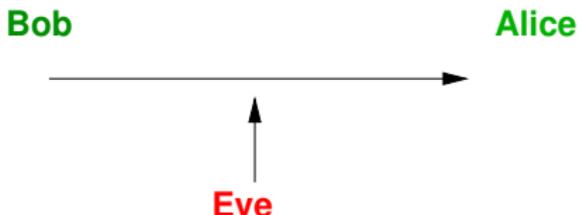
$$L(p)^{\sqrt{2}+o(1)} \text{ wobei } L(p) = e^{\sqrt{\ln p \ln \ln p}}$$

zeigen. **Praxis:** Wähle  $C$  "klein" und bestimme  $B = e^{\sqrt{(\ln C \ln \ln C)/2}}$ . Verdopple  $C$  falls nicht erfolgreich.

# Moderne Anwendung: Public Key Cryptography

## Cryptosysteme

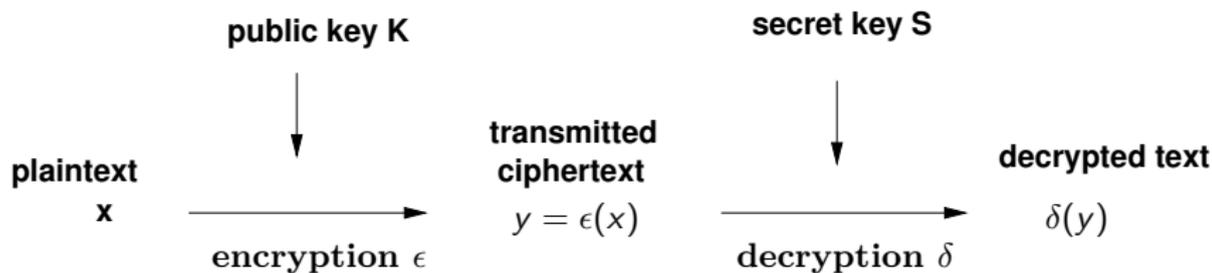
- **Szenario:** Bob will Nachricht an Alice senden, so dass ein Lauscher (Eve) die Nachricht nicht verstehen kann. Dies wird durch eine Chiffrierung der Nachricht erreicht, so dass nur Alice, mit den richtigen Schlüssel, die Nachricht leicht entschlüsseln kann aber Eve ohne den richtigen Schlüssel die 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  $\delta$  Inverse von  $\epsilon$ . 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.

## Public Key Cryptosysteme: Anforderungen

- ▶ Ein Cryptosystem gilt als **geknackt**, wenn es ein Boolesches 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  $\mathbb{Z}_N = \{0, 1, \dots, N - 1\}$  kodiert. Verwendet man als Alphabet  $\Sigma = \{A, \dots, Z\}$  mit  $|\Sigma| = 26$ , 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  $e \in \{2, \dots, \varphi(N) - 2\}$  mit  $GGT(e, \varphi(N)) = 1$ , ( $\varphi(N) = (p - 1)(q - 1)$ ). Dann berechnet sie  $d \in \{2, \dots, \varphi(N) - 2\}$  mit  $de \equiv 1 \pmod{\varphi(N)}$ .  $K = (N, e)$  ist ihr öffentlicher Schlüssel und  $S = (N, d)$  ihr geheimer Schlüssel.





# 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:
  - ▶ Alice und Bob einigen sich auf  $q$  und  $g$  die öffentlich sein können.
  - ▶ Alice wählt für sich  $a \in \mathbb{Z}_{q-1}$ , berechnet und sendet  $u = g^a \in \mathbb{F}_q^\times$  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.

# 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  $\exp(O((n \log^2 n)^{1/3}))$  Wortoperationen mit  $n \approx \log_2 q$ .

## Das ElGamal Cryptosystem

- ▶ Wie groß  $\mathbb{F}_q^\times$  groß und  $g$  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  $x^2 \equiv y^2 \pmod N$  oder  $pq = N \mid (x + y)(x - y)$ . Ist  $x \not\equiv \pm y \pmod N$ , 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  $x^2 \pmod N$ . Alice berechnet die zwei Quadratwurzeln von  $y \pmod p$  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  $y \equiv x^2 \pmod N$  zu signieren, in dem sie die Quadratwurzel  $z$  von  $y \pmod N$  zurückgibt, so ist mit Wahrscheinlichkeit  $1/2$ ,  $GGT(x - z, N)$  ein echter Teiler von  $N$ .



# Inhalt Kapitel 8

## Polynomideale und Gröbner-Basen

- 8.1 Motivation: Buchberger's Algorithmus
- 8.2 Reduktionstechniken zur Lösung des Wortproblems
- 8.3 Termordnungen und multivariate Division mit Rest: Reduktion in  $K[x_1, \dots, x_n]$
- 8.4 Polynom-Reduktion
- 8.5 Längen von Reduktionsketten
- 8.6 Term-Ideale und Hilbert's Basissatz
- 8.7 Gröbnerbasen und  $S$ -Polynome
- 8.8 Buchberger's Algorithmus
- 8.9 Anwendungen von Gröbner Basen
- 8.10 Eliminationseigenschaften - Polynomgleichungen (Syzygien)
- 8.11 Zur Komplexität der Berechnung von  $G$ -Basen

## Motivation: Buchberger's Algorithmus

Polynomringe in mehreren Variablen. Werkzeug zur Modellierung vieler Probleme aus der (Alg)-Geometrie, Robotik (eukl. Geometrie), Invarianten, Singularitäten.

**Polynomideale:**  $\{p_1, \dots, p_k\} \subseteq R[x_1, \dots, x_n] = \mathfrak{R}$ ,  $\langle p_1, \dots, p_k \rangle_{\mathfrak{R}}$  Ideal.

**Wortproblem:**  $p \in \mathfrak{R}$  gilt  $p \in \langle p_1, \dots, p_k \rangle$ .

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

## Buchberger's Algorithmus

- ▶ z. B. in  $\mathbb{Q}[x, y, z]$ 
  - ▶  $p_1 = x^3yz - xz^2$
  - ▶  $p_2 = xy^2z - xyz$
  - ▶  $p_3 = x^2y^2 - z^2$

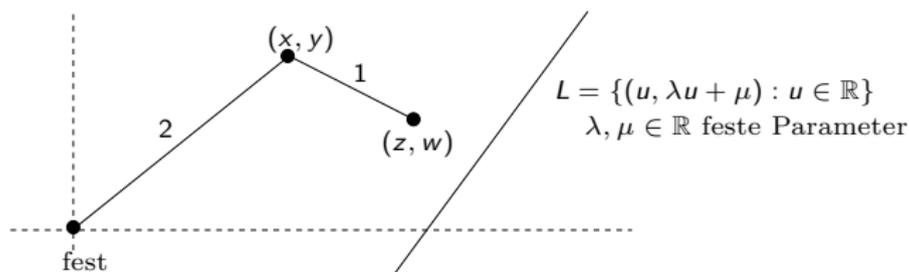
**Frage:** Liegt  $q = x^2yz - z^3 \in \langle p_1, p_2, p_3 \rangle$ .

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:
  - $q \approx 0 \text{ mod } \langle p_1, p_2, p_3 \rangle$
  - $\approx$  Kongruenz, die von  $\mathfrak{i} = \langle p_1, p_2, p_3 \rangle$  induziert wird.
- ▶  $q_1 \approx_{\mathfrak{i}} q_2$  gdw  $q_1 - q_2 \in \mathfrak{i}$

## Beispiele

### 8.1 Beispiel Einfacher Roboter



- ▶ Die möglichen Positionen  $(x, y, z, w) \in \mathbb{R}^4$  des Roboters sind charakterisiert durch

$$(*) \quad x^2 + y^2 = 4$$

$$(z - x)^2 + (w - y)^2 = 1$$

- ▶ **Frage:** Kann der Roboter die Linie erreichen?

(\*) muss erfüllt sein und  $w = \lambda z + \mu$ .







## 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.  $V(f_1, f_2) \subseteq V(g_1)$

- $S$  dreht 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.)

- ▶ 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

$$\begin{aligned}
 f &= (y^2 + 6)(x - 1) - y(x^2 + 1) \\
 g &= (x^2 + 6)(y - 1) - x(y^2 + 1) \\
 h &= (x - 5/2)^2 + (y - 5/2)^2 - 1/2
 \end{aligned}
 \quad \text{in } \mathbb{C}[x, y]$$

$$I = \langle f, g \rangle$$





## 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:

$V(J) = \emptyset \rightsquigarrow 1 \in J$  (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.











# Reduktionstechniken zur Lösung des Wortproblems (Forts.)

**8.4 Satz**  $\rightarrow$  noethersch und Church Rosser, so WP für  $\xrightarrow{*}$  entscheidbar.

d.h. Kanonische Simplifikationsfunktion

$x \mapsto y$  mit  $x \xrightarrow{*} \underline{y}$  NF für  $x$  (**Beachte  $\rightarrow$  effektiv**).

## 8.5 Beispiel

1) Kommutative Halbgruppe mit Erzeugenden  $a, b, c, f, s$ . Relationen

$$as = c^2s \quad bs = cs \quad s = f \quad :: E$$

**Reduktionsrelation** auf der freien kommutativen Halbgruppe in  $a, b, c, f, s$

$$R: \quad s \rightarrow f \quad cf \rightarrow bf \quad b^2f \rightarrow af$$

auf Wörter in  $a, b, c, f, s : u \rightarrow v$ , so  $ut \rightarrow vt$ .

$\rightarrow_R$  ist noethersch, Church-Rosser und äquivalent zu  $E$ .

$$a^3bcf^3 =_E a^2b^4fs^2$$



## Termordnungen

- ▶ **Partialordnung**  $<$  auf  $S$  ist eine irreflexive transitive Relation  $< \subseteq S \times S$ .  
d.h.  $\neg(\alpha < \alpha) \wedge \alpha < \beta < \gamma \Rightarrow \alpha < \gamma$  für alle  $\alpha, \beta, \gamma \in S$   
d. h.  $<$  ist asymmetrisch.
- ▶ Partialordnung ist **total**, falls  $\alpha = \beta \vee \alpha < \beta \vee \beta < \alpha$  ( $\alpha, \beta \in S$ ).
- ▶ Ordnung ist eine **Wohlordnung**, falls jede nicht leere Menge ein kleinstes Element besitzt.
- ▶ Schreibe  $\alpha \leq \beta$ , falls  $\alpha = \beta$  oder  $\alpha < \beta$ .
- ▶  $<$  auf  $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}$  sind total, **nur** auf  $\mathbb{N}$  Wohlordnung.
- ▶  $X = \{x_1, \dots, x_n\}$ : freie kommutative Halbgruppe (Monoid) über  $X$  ist die **Menge der Terme** über  $X$ .  
Darstellung der Terme:: Sei  $\alpha = (\alpha_1, \dots, \alpha_n) \in \mathbb{N}^n$ .  
**Identifikation**  $\alpha \rightarrow x^\alpha = x_1^{\alpha_1} \cdots x_n^{\alpha_n} \in F[x_1, \dots, x_n]$ ,  
d. h.  $\mathbb{N}^n \simeq$  Menge der Terme über  $X := T(X)$ . Operationen:  $\cdot, |, KGV, ..$

# Termordnungen

**8.7 Definition** Eine **Termordnung** auf  $T(X)$  ist eine Relation  $\prec$  auf  $\mathbb{N}^n$  mit

1.  $\prec$  ist totale Ordnung.
2.  $\alpha \prec \beta \Rightarrow \alpha + \gamma \prec \beta + \gamma$  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)
3.  $\prec$  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  $\infty$ -fallende Ketten.
- ▶  $n = 1$  Standard Ordnung auf  $\mathbb{N}$  ist die übliche Grad Ordnung auf  $T(X)$ .



## Beispiel: Termordnungen (Fort.)

### iii) Graduierte inverse lexikographische Ordnung:

- ▶  $\alpha \prec_{\text{grevlex}} \beta$  gdw  $\sum_{1 \leq i \leq n} \alpha_i < \sum_{1 \leq i \leq n} \beta_i$  oder  
 $(\sum \alpha_i = \sum \beta_i \wedge$  am weitesten rechts stehende  
 nicht Null-Eintrag in  $\alpha - \beta$   
 ist positiv).
- ▶ Es gilt  $\alpha_4 \prec_{\text{grevlex}} \alpha_2 \prec_{\text{grevlex}} \alpha_3 \prec_{\text{grevlex}} \alpha_1$ .

$$\hookrightarrow n = 1 \quad \prec_{\text{lex}} = \prec_{\text{grlex}} = \prec_{\text{grevlex}}.$$

$$\hookrightarrow f = 4xyz^2 + 4x^3 - 5y^4 + 7xy^2z \in \mathbb{Q}[x, y, z]$$

$$\text{lex} : 4x^3 + 7xy^2z + 4xyz^2 - 5y^4$$

$$\text{grlex} : 7xy^2z + 4xyz^2 - 5y^4 + 4x^3.$$



## Wichtige Begriffe für Polynome und deren Reduktion

**8.10 Definition** Sei  $f = \sum_{\alpha \in \mathbb{N}^n} c_\alpha x^\alpha \in R = K[x_1, \dots, x_n]$ ,  $f \neq 0$ .

$c_\alpha \in F$  (nur endlich viele  $\neq 0$ ),  $\prec$  Termordnung.

1.  $c_\alpha x^\alpha$  ist **Monom** in  $f$  für  $c_\alpha \neq 0$       **Coeff** $(f, \alpha) = c_\alpha$ .
2. Der **Multigrad** von  $f$  ist  $\text{mdeg}(f) = \max_{\prec} \{\alpha \in \mathbb{N}^n : c_\alpha \neq 0\}$
3.  $LT(f) := x^{\text{mdeg}(f)} = \max_{\prec} \{t \in T[X] \mid \text{Coeff}(f, t) \neq 0\}$   
**Leitterm** (Hauptterm) von  $f$ .
4.  $LC(f) := c_{\text{mdeg}(f)} \in F \setminus \{0\}$       **Leitkoeffizient**.
5.  $LM(f) = LC(f) \cdot LT(f) \in R$       **Leitmonom**.
6.  $Red(f) = f - LM(f)$       **Redukt** von  $f$ .



## Lemma

**8.12 Lemma** Sei  $\prec$  Termordnung auf  $T[X]$ ,  $f, g \in R \setminus \{0\}$ .

1.  $\text{mdeg}(fg) = \text{mdeg}(f) + \text{mdeg}(g)$   
( $LT(fg) = LT(f) \circ LT(g)$  in  $T[X]$ )
  2.  $f + g \neq 0$  so  $\text{mdeg}(f + g) \leq \max\{\text{mdeg}(f), \text{mdeg}(g)\}$   
d. h.  $LT(f + g) \leq \max\{LT(f), LT(g)\}$   
Gleichheit, falls  $\text{mdeg}(f) \neq \text{mdeg}(g)$ .
- Verallgemeinerung der Division mit Rest: **Reduktion**
- $f, f_1, \dots, f_s \in R$  gesucht Darstellung von  $f$  mit  $f = q_1 f_1 + \dots + q_s f_s + r$   
mit  $q_1, \dots, q_s, r \in R$ .
- ? Wie bestimmt man  $q_i$ , welche Eigenschaften hat  $r$ ?

## Beispiel

## 8.13 Beispiel

1. Sei  $\prec = \prec_{lex}$            $f = xy^2 + 1$            $f_1 = xy + 1$            $f_2 = y + 1$

	$xy + 1$	$y + 1$
$xy^2 + 1$	$y$	
$-(xy^2 + y)$		
$-y + 1$		$-1$
$-(-y - 1)$		
$2$		

$$f = yf_1 - 1 \cdot f_2 + 2$$

	$xy + 1$	$y + 1$
$xy^2 + 1$		$xy$
$-(xy^2 + xy)$		
$-xy + 1$		$-x$
$-(-xy - x)$		
$x + 1$		

$$f = 0 \cdot f_1 + (xy - x) \cdot f_2 + (x + 1)$$

- ▶ Kein Term in 2 bzw.  $-x + 1$  ist durch ein  $LT(f_i)$  teilbar.

b) Sei  $\prec = \prec_{lex}$ 

$$f = x^2y + xy^2 + y^2$$

$$f_1 = xy - 1$$

$$f_2 = y^2 - 1$$

	$xy - 1$	$y^2 - 1$	Rest
$x^2y + xy^2 + y^2$ $-(x^2y - x)$	$x$		
$xy^2 + x + y^2$ $-(xy^2 - y)$	$y$		
$x + y^2 + y$ $-x$			$x$
$y^2 + y$ $-(y^2 - 1)$		$1$	
$y + 1$			

$$f = (x + y) \cdot f_1 + 1 \cdot f_2 + (x + y + 1)$$

► Kein Term in  $x + y + 1$  durch ein  $LT(f_i)$  teilbar.

# Polynom-Reduktion

**procedure** Algorithmus multivariate Division mit Rest

{**Eingabe:** Polynome  $f, f_1, \dots, f_s \in R = F[x_1, \dots, x_n]$ ,  $F$  Körper}  
 {Termordnung  $\prec$  auf  $T[x]$ .}

{**Ausgabe:**  $q_1, \dots, q_s, r \in R$  mit  $f = q_1 f_1 + \dots + q_s f_s + r$ .}  
 {Kein Monom in  $r$  ist durch ein  $LT(f_1), \dots, LT(f_s)$  teilbar.}

**begin**

1  $r := 0; p := f;$

**for**  $i = 1, \dots, s$  **do**  $q_i := 0$

2 **while**  $p \neq 0$  **do**

3   **if**  $LM(f_i) \mid LM(p)$  für ein  $i \in \{1, \dots, s\}$

**then choose some such**  $i: q_i := q_i + \frac{LM(p)}{LM(f_i)}; p := p - \frac{LM(p)}{LM(f_i)} f_i;$

**else**  $r := r + LM(p); p := p - LM(p)$

4 **return**  $q_1, \dots, q_s, r$

**end**

## Polynom-Reduktion (Forts.)

**8.14 Satz** Bei Schritt 3 gelten folgende Invarianten

1.  $\text{mdeg}(p) \preceq \text{mdeg}(f) \quad f = p + q_1 f_1 + \cdots + q_s f_s + r.$
  2.  $q_i \neq 0 \Rightarrow \text{mdeg}(q_i f_i) \preceq \text{mdeg}(f) \quad 1 \leq i \leq s.$
  3. 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?.

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  $s, t \in T[X]$   
 $\text{Coeff}(g, s) = c \neq 0$   $s = \text{LT}(f)t$  (d.h.  $\text{LT}(f) \mid s$ ) und

$$h = g - \frac{c}{\text{LC}(f)} \cdot t \cdot f \quad \text{Ein „Monom“ in } g \text{ wird ersetzt}$$

- $g \xrightarrow{P} h$  gdw  $\exists f_i \in P$   $g \xrightarrow{f_i} h$
- $\xrightarrow{P}^*$ ,  $\xleftarrow{P}^*$  wie üblich.
- **Beachte**: Multivariate Division mit Rest liefert ein  $r$  mit  $r \xrightarrow{P}$  irreduzibel und  $g \xrightarrow{P}^* r$ . **Strategie: Left-Most-Reduktion.**
- Es gilt  $\xleftarrow{P}^* = \equiv_{\langle P \rangle}$  (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
$x^2y + xy^2 + y^2$	$x$	$x$	$x^2y + xy^2 + y^2$
$-(x^2y - x)$			$-(xy^2 - x)$
$xy^2 + x + y^2$	$y$	$1$	$x^2y + y^2 + x$
$-(xy^2 - y)$			$-(y^2 - 1)$
$x + y^2 + y$	$x$		$x^2y + x + 1$
$-x$			$-(x^2y - x)$
$y^2 + y$		$1$	
$-(y^2 - 1)$			
$x + y + 1$			$2x + 1$

d.h. Rest muss nicht eindeutig sein, d. h. i. Allg. keine Konfluenz.

## Beispiel (Forts.)

Beachte i. Allg. **Wahl von  $i$**  mit  $HT(f_i) \mid HT(P)$ . Wähle kleinstes  $i \rightsquigarrow$  die Quotienten  $q_1, \dots, q_s$  und der Rest  $r$  sind eindeutig festgelegt, schreibe  $r = f \operatorname{rem}(f_1, \dots, f_s)$  für diese Wahl.

Gewünscht wird:

- ▶  $f \in \langle f_1, \dots, f_s \rangle$  gdw  $r = f \operatorname{rem}(f_1, \dots, f_s) = 0$ 
  - ▶ Dies stimmt, falls  $s = 1$  ist.
  - ▶ Für  $s \geq 2$  für Gröbner Basen! Sonst i. Allg. nicht
- ▶ " $\Leftarrow$ " stimmt aber  $\Rightarrow$  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.  $f \in \langle f_1, f_2 \rangle$ .





## Längen von Reduktionsketten (Forts.)

- Seien  $d, \Delta, l, L$  mit  $d \geq l - 2 > 0$   $\Delta > L$

$$\begin{aligned}
 f &= x_1^\Delta x_n^L + x_1^\Delta x_n^{L-1} + \cdots + x_1^\Delta x_n && \text{mit } G \\
 g_1 &= x_1 - (x_2^d x_3^d \cdots x_{n-1}^d)(x_n^d + x_n^{d-1} + \cdots + x_n^{d-l+2}) \\
 g_2 &= x_2 - (x_3^d x_4^d \cdots x_{n-1}^d)(x_n^d + x_n^{d-1} + \cdots + x_n^{d-l+2}) \\
 &\vdots \\
 g_n &= x_n^l - x_n^{l-1} - \cdots - x_n \\
 g_{n+1} &= x_n^{l-1} - x_n^{l-2} - \cdots - x_n \\
 &\vdots \\
 g_{n+l-2} &= x_n^2 - x_n \\
 g_{n+l-1} &= x_n - 1
 \end{aligned}$$

- Reduktionsfolge, die  $\min_{\langle \cdot \rangle_{lex}}$  Monom zum Reduzieren wählt, hat Länge der oberen Schranke

Andere Schranken (genauere) Dube, Mishra, Yap.



## Längen von Reduktionsketten (Forts.)

Für jede Termordnung gilt:

$$\blacktriangleright K(G, f) \leq \begin{cases} L & l = 1 \\ (1 + R_F \bar{u})L & l = 2 \\ 2^{R_F \bar{u}} L & l \geq 3 \end{cases}$$

- $\blacktriangleright$   $L$  Anzahl der Monome in  $f$
- $\blacktriangleright$   $l$  Max. Anzahl von Monomen in Polynom aus  $G$
- $\blacktriangleright$   $R_F$  Konstante, die von  $>$  und  $G$  abhängt
- $\blacktriangleright$   $\bar{u}$  Maximum der „Gewichte“ der Monomen in  $f$   
( $\bar{u} = O(\text{grad}(f))$ )
- $\blacktriangleright$  Head-Reduktion
  - $\blacktriangleright$   $\langle_{lex}:$   $\leq (d + 1)^{\frac{n^2+n}{2}} D^n$   $\bullet$   $d$  Max. Grad eines Mon. in  $f$   
 $\bullet$   $D$  Max. Grad eines Mon. in  $g \in G$
  - $\blacktriangleright$   $\langle_{grlex}:$   $\leq (d + 1)^n D^n$  bzw.  $(D + 1)^n$



## 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

- $f \in I$ .
- Jedes Monom von  $f$  liegt in  $I$ .
- $f$  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  $x^\gamma$  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  $A \neq \emptyset.$

## Dickson's Lemma (Forts.)

- ▶ Betrachte  $\leq$  auf  $\mathbb{N}^n$  mit  
 $\alpha \leq \beta$  gdw  $\alpha_i \leq \beta_i, 1 \leq i \leq n$  (d. h.  $x^\alpha \mid x^\beta$ ).  
 Schreibe  $\alpha < \beta$ , falls  $\alpha \leq \beta$  und  $\alpha \neq \beta$ .
- ▶  $<$  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,  $B \subseteq A$ ,

$$(*) \quad \forall \alpha \in A \quad \exists \beta \in B, \beta \leq \alpha$$

- ▶ 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  $\infty$  fallende Kette

$$\alpha^{(1)} > \alpha^{(2)} > \alpha^{(3)} > \dots \text{ in } \mathbb{N}^n$$

- ▶ Insbesondere folgt (\*).

## Dickson's Lemma (Forts.)

- ▶ z. Z.  $B$  ist endlich. **Induktion nach  $n$ .**
  - ▶  $n = 1$ , dann ist  $< \text{total } B = \{\text{kleinstes Element von } A\}$ .
  - ▶  $n \geq 2$ , 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 Induktionvoraussetzung 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:**  $(\alpha_1, \dots, \alpha_n) \in B$ , 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  $\beta \leq (\alpha_1, \dots, \alpha_{n-1})$ .

## Dickson's Lemma (Forts.)

- ▶ Ist  $\alpha_n > b$ , so

$$(\beta_1, \dots, \beta_{n-1}, b) \leq (\beta_1, \dots, \beta_n, \alpha) < \alpha$$

$\alpha$  ist nicht minimal, d. h.  $\alpha_n \leq b$ .

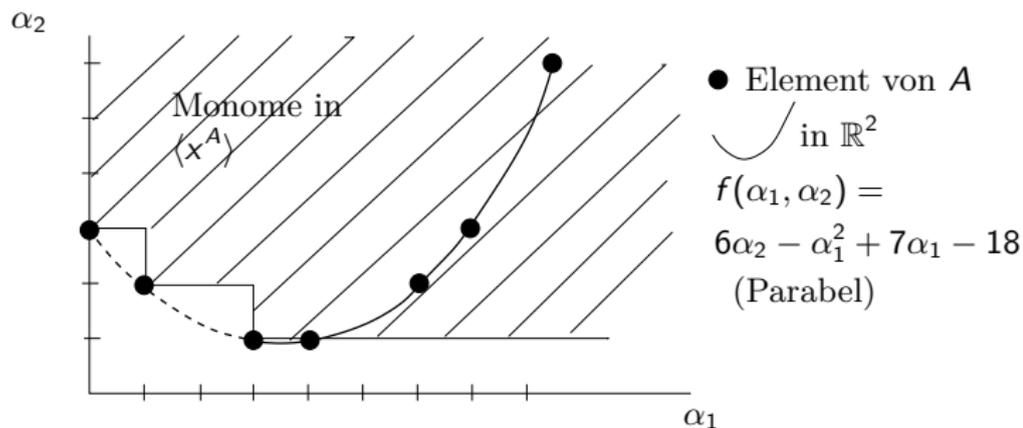
- ▶ Analog zeigt man, dass alle Komponenten beschränkt sind, es gibt nur endlich viele  $(\alpha_1, \dots, \alpha_n) \in B$ .
- ▶  $\alpha \leq \beta$  gdw  $x^\alpha \mid x^\beta \rightsquigarrow x^A \subseteq \langle x^B \rangle$  und somit  $\langle x^A \rangle \subseteq \langle x^B \rangle$ .  
 $\supseteq$  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

$$B = \{(0, 3), (1, 2), (3, 1)\}, \text{ d. h. } \langle x^A \rangle = \langle y^3, xy^2, x^3y \rangle$$



## Folgerung

**8.24 Folgerung** Sei  $\prec$  eine totale Ordnung auf  $\mathbb{N}^n$  mit

$$\forall \alpha, \beta, \gamma \in \mathbb{N}^n, \alpha \prec \beta \Rightarrow \alpha + \gamma \prec \beta + \gamma.$$

- ▶  $\prec$  ist wohlfundiert 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.  $\exists \alpha_1, \dots, \alpha_s \in A$ :

$I = \langle x^{\alpha_1}, \dots, x^{\alpha_s} \rangle$  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.  $\alpha = \alpha_i + \gamma \succeq \alpha_1 + \gamma \succeq \alpha_1 + 0 = \alpha_1 \rightsquigarrow \alpha_1 = \min_{\prec} A$ .

- ▶ D.h. die Bedingung iii) der Termordnungen kann durch (iii)\*  $\forall \alpha \in \mathbb{N}^n, \alpha \succeq 0$  ersetzt werden.

# Notation-Beispiel

**Schreibweise:**  $G \subseteq R = F[x_1, \dots, x_n]$

$LM(G) = \{LM(g) : g \in G\}$ ,  $LT(G) = \{LT(g) : g \in G\}$

- ▶ 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:  $g = x^3 - 2xy$      $h = x^2y - 2y^2 + x$      $\prec = \prec_{\text{grlex}}$   
 $G = \{g, h\}$      $I = \langle G \rangle$      $x^2 = -yg + xh$ , d. h.  
 $x^2 \in \langle LT(I) \rangle$ , aber  $x^2 \notin \langle LT(G) \rangle = \langle x^3, x^2y \rangle$ .

## Hilbert's Basissatz

**8.25 Lemma** Sei  $I$  ideal in  $R = F[x_1, \dots, x_n]$ .

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

- ▶  $f = q_1 g_1 + \dots + q_s g_s + r$  mit  $q_1, \dots, q_s, r \in R$ .  
Wobei  $r = 0$  oder kein Term in  $r$  ist durch  $LT(f_i)$  für ein  $i$  teilbar.
- ▶  $r = f - q_1 g_1 - \dots - q_s g_s \in I \rightsquigarrow LT(r) \in LT(I) \subseteq \langle LT(G) \rangle$ .
- ▶ 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  $\langle LT(I) \rangle$ .



## 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

$$\begin{array}{l}
 f \in I \quad \text{gdw} \quad r = f \operatorname{rem}(G) = 0 \\
 \quad \quad \quad \text{gdw} \quad f \xrightarrow[G]{*} 0
 \end{array}$$

► d. h.  $\xrightarrow[G]{*}$  ist konfluent auf  $I$ .

**8.29 Folgerung** Jedes Ideal  $I$  in  $R = F[x_1, \dots, x_n]$  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  $\langle g, h \rangle$ .

## Konfluenz von $\longrightarrow_G$ 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

1.  $f - r \in I$ .
2. Kein Monom in  $r$  ist teilbar durch ein Element in  $LT(G)$ .

**Beweis:**

- **Existenz** folgt aus Algorithmus multivariate Division mit Rest.
- **Eindeutigkeit:** Angenommen  $f = h_1 + r_1 = h_2 + r_2$   $h_1, h_2 \in I$ .  
Kein Monom in  $r_1, r_2$  ist teilbar durch ein Element in  $LT(G)$ .  
 $r_1 - r_2 = h_2 - h_1 \in I \rightsquigarrow LM(r_1 - r_2)$  ist teilbar durch  $LT(g)$  mit  $g \in G$   
nach Lemma 8.18  $\rightsquigarrow r_1 - r_2 = 0$ .

# Konfluenz von $\xrightarrow{G}$ für Gröbner Basen (Forts.)

- ▶ **Folgerung:** Wir können für Gröbnerbasen  $G$  schreiben  
 $f \operatorname{rem} G = r \in R$   
 $r$  ist die einzige Normalform von  $f$  bzgl.  $\xrightarrow{G}$ .
- ▶  $\rightsquigarrow \xrightarrow{G}$  ist konvergent.

**8.32 Satz** Sei  $G$  Gröbner-Basis für  $I \subseteq R$  bzgl. Termordnung  $\prec$ ,  $f \in R$ .

- $f \in I$  gdw  $f \operatorname{rem} G = 0$  gdw  $f \xrightarrow[*]{G} 0$ .

**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  $A = \{\alpha : x^\alpha \in \langle LT(I) \rangle\}$ .

## S-Polynome: Konfluenztest

### 8.33 Definition

Seien  $g, h \in R$  nicht Null,  $\alpha = (\alpha_1, \dots, \alpha_n) = \text{mdeg}(g)$ ,  
 $\beta = (\beta_1, \dots, \beta_n) = \text{mdeg}(h)$  und  $\gamma = (\max\{\alpha_1, \beta_1\}, \dots, \max\{\alpha_n, \beta_n\})$   
das **S-Polynom** von  $g$  und  $h$  ist

$$S(g, h) = \frac{x^\gamma}{LM(g)}g - \frac{x^\gamma}{LM(h)}h \quad \in R$$

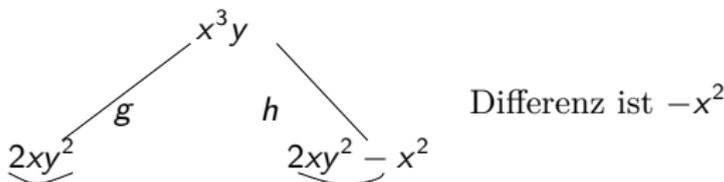
- ▶ Offenbar  $S(g, h) = -S(h, g)$ .
- ▶ Da  $\frac{x^\gamma}{LM(g)}, \frac{x^\gamma}{LM(h)} \in R$  gilt, folgt  $S(g, h) \in \langle g, h \rangle$ .
- ▶ **Beachte:**  $LT(S(g, h)) \prec x^\gamma$ .  
(Wichtig für Noethersche Induktionsbeweise nach  $\prec$ ).

## S-Polynome: Konfluenztest (Forts.)

**8.34 Beispiel**  $g = x^3 - 2xy$      $h = x^2y - 2y^2 + x$      $\in \mathbb{Q}[x, y] \prec_{grlex}$

$\hookrightarrow \alpha = (3, 0), \beta = (2, 1), \gamma = (3, 1)$

$$\bullet S(g, h) = \frac{x^3y}{x^3}g - \frac{x^3y}{x^2y}h = y(x^3 - 2xy) - x(x^2y - 2y^2 + x) = -x^2$$



- ▶ Um Konfluenz auf  $\langle g, h \rangle$  zu erreichen müssen S-Polynome sich nach 0 reduzieren lassen. **Frage:** Folgt aus  $u - v \rightarrow^*_{\langle g, h \rangle} 0$  auch  $u \downarrow^*_{\langle g, h \rangle} v$
- ▶ Die Leitmonome bei Linearkombinationen können sich wegheben, dieses kann durch die S-Polynome charakterisiert werden.

## S-Polynome: Hauptlemma

**8.35 Lemma** Sei  $g_1, \dots, g_s \in R$ ,  $\alpha_1, \dots, \alpha_s \in \mathbb{N}^n$ ,  $c_1 \dots c_s \in F \setminus \{0\}$

►  $f = \sum_{1 \leq i \leq s} c_i x^{\alpha_i} g_i \in R$  und

►  $\delta \in \mathbb{N}^n$  mit  $\alpha_i + \text{mdeg}(g_i) = \delta$  für  $1 \leq i \leq s$  und  $\text{mdeg}(f) < \delta$   
(d. h.  $x^\delta$  ist nicht Leitterm von  $f$ ).

Dann gilt  $x^{\gamma_{ij}}$  teilt  $x^\delta$  für  $1 \leq i < j \leq s$  wobei  
 $x^{\gamma_{ij}} = \text{KGV}(LT(g_i), LT(g_j))$  und es gibt  $c_{ij} \in F$  mit

$$(*) \quad f = \sum_{1 \leq i < j \leq s} c_{ij} x^{\delta - \gamma_{ij}} S(g_i, g_j)$$

und  $\text{mdeg}(x^{\delta - \gamma_{ij}} S(g_i, g_j)) < \delta$  für alle  $1 \leq i < j \leq s$ .



# S-Polynome: Hauptlemma-Beweis

► Sei  $s \geq 2$

$$\begin{aligned}
 g &= f - c_1 x^{\delta - \gamma_{12}} S(g_1, g_2) \\
 &= c_1 x^{\alpha_1} g_1 + c_2 x^{\alpha_2} g_2 + \sum_{3 \leq i \leq s} c_i x^{\alpha_i} g_i - c_1 x^{\delta - \gamma_{12}} \left( \frac{x^{\gamma_{12}}}{LT(g_1)} g_1 - \frac{x^{\gamma_{12}}}{LT(g_2)} g_2 \right) \\
 &= c_1 \underbrace{(x^{\alpha_1} - x^{\delta - mdeg(g_1)})}_{=0} g_1 + (c_2 x^{\alpha_2} + c_1 x^{\delta - mdeg(g_2)}) g_2 + \sum_{3 \leq i \leq s} c_i x^{\alpha_i} g_i \\
 &= (c_1 + c_2) x^{\alpha_2} g_2 + \sum_{3 \leq i \leq s} c_i x^{\alpha_i} g_i
 \end{aligned}$$

► Hierbei verwende  $\alpha_1 + mdeg(g_1) = \delta = \alpha_2 + mdeg(g_2)$ .

# S-Polynome: Hauptlemma-Beweis

- ▶  $\text{mdeg}(g) \preceq \max_{\prec} \{ \text{mdeg}(f), \text{mdeg}(x^{\delta-\gamma_{12}} S(g_1, g_2)) \} \prec \delta$ ,  
 $g$  hat die ursprüngliche Gestalt mit  $s - 1$  Summanden (falls  $c_1 + c_2 \neq 0$ )  
 sonst  $s - 2$  Summanden.

↪ Nach Induktionsvor.  $g = \sum_{2 \leq i < j \leq s} c_{ij} x^{\delta-\gamma_{ij}} S(g_i, g_j)$ .

Für  $c_{ij} \in F$  ( $2 \leq i < j \leq s$ ).  $g = 0$ , falls  $s = 2$ .

Setzt man  $c_{12} = c_1$  und  $c_{1j} = 0$  für  $3 \leq j \leq s$ , so

↪

$$f = g + c_1 x^{\delta-\gamma_{12}} S(g_1, g_2) = \sum_{1 \leq i < j \leq s} c_{ij} x^{\delta-\gamma_{ij}} S(g_i, g_s)$$





# 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. Ordnung  $y \succ z \succ x$ .

- $$S(y - x^2, z - x^3) = z(y - x^2) - y(z - x^3) = yx^3 - zx^2 \xrightarrow[G]{*} 0$$

$$= x^3(y - x^2) + (-x^2)(z - x^3) + 0$$

# Buchberger's Algorithmus

{Eingabe:  $f_1, \dots, f_s \in R = F[x_1, \dots, x_n]$ ,  $\prec$  Termordnung.}  
 {Ausgabe: Gröbner Basis  $G \subseteq R$  für  $I = \langle f_1, \dots, f_s \rangle$  bzgl.  $\prec$ }  
 { mit  $\{f_1, \dots, f_s\} \subseteq G$ . }

**begin**

1  $G := \{f_1, \dots, f_s\}$

2 **repeat**

$S := \emptyset$

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**  $S = \emptyset$  **then** return  $G$  **else**  $G := G \cup S$

**end**

## 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  $G_i$ :

$S(\cdot, \cdot) \text{ rem } (G_i) = 0$ .

- ▶ **Frage:** Platz und Zeitbedarf, Implementierungen.

**8.39 Folgerung** Folgende Probleme sind mit  $G$ -Basen entscheidbar

1. Wortproblem ( $f \in \langle G \rangle$ )
2.  $\langle G \rangle \subseteq \langle H \rangle$
3.  $\langle G \rangle = \langle H \rangle$

## Beispiel

$$f_1 = x^3 - 2xy \quad f_2 = x^2y - 2y^2 + x \in \mathbb{Q}[x, y], y < x \prec_{\text{grlex}}$$

▶  $S(f_1, f_2) = -x^2 \quad LT(S(f_1, f_2)) = -x^2 \notin \langle x^3, x^2y \rangle$

•  $f_3 := S(f_1, f_2) \text{ rem } (f_1, f_2) = -x^2.$

Dann  $S(f_1, f_2) \text{ rem } (f_1, f_2, f_3) = 0$

▶  $S(f_1, f_3) = 1f_1 - (-x)f_3 = -2xy$

$S(f_1, f_3) \text{ rem } (f_1, f_2, f_3) = -2xy =: f_4$  •

$S(f_1, f_3) \xrightarrow{*}_{f_1, f_2, f_3, f_4} 0$

▶  $S(f_1, f_4) = yf_1 - (-\frac{1}{2}x^2)f_4 = -2xy^2 = yf_4 \xrightarrow{*} 0$

▶  $S(f_2, f_3) = 1f_2 - (-y)f_3 = -2y^2 + x$  irred.

•  $f_5 = S(f_2, f_3) \text{ rem } (f_1, \dots, f_4) = -2y^2 + x$

$\rightsquigarrow S(f_i, f_j) \text{ rem } (f_1, \dots, f_5) = 0$  für  $1 \leq i < j \leq 5$ ,

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.

# 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  $g \in G$

1.  $LC(g) = 1$
2.  $LT(g) \notin \langle LT(G \setminus \{g\}) \rangle$  (**minimal**)
3.  $g$  ist irreduzibel bzgl.  $G \setminus \{g\}$ . (**reduzierte**)

**8.42 Satz** Eindeutigkeitsatz

Jedes Ideal hat eine eindeutige reduzierte Gröbner Basis bzgl.  $\prec$ .

**Beweis:** **Existenz:** Anwendung von Lemma 8.40 o.B.d.A.

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

$$\begin{aligned} \text{Sei } h_i &= g_i \text{ rem } \{h_1, \dots, h_{i-1}, g_{i+1}, \dots, g_s\} & i = 1, \dots, s \\ &= \text{NF}(g_i, G \setminus \{g_i\}) \end{aligned}$$

## 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  $\rightsquigarrow LT(g) = LT(g^*) = t \rightsquigarrow LT(G) \subseteq LT(G^*)$ .  
= 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:

$$p - q \xrightarrow{F} h \rightsquigarrow \exists p', q' : p \xrightarrow{*} p', q \xrightarrow{*} q', h = p' - q'$$

$$p - q \xrightarrow{*} 0 \rightsquigarrow \exists g : p \xrightarrow{*} g, q \xrightarrow{*} g$$

$$p \rightarrow q \rightsquigarrow tp \rightarrow tq \wedge p + h \downarrow^* q + h$$

**Frage:** Wieviele Polynome müssen hinzugenommen werden, um eine Gröbner Basis zu bekommen?

## Äquivalente Charakterisierungen für GB

**8.44 Satz** Charakterisierungssatz 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  $\xrightarrow{G}$  Normalform
2. (Standard Basen)  $\langle LT(G) \rangle = \langle LT(I) \rangle$
3. (Buchberger Krit.) für alle  $f, g \in G$  gilt  $S(f, g) \xrightarrow{G}^* 0$
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

$$f = \sum_{1 \leq i \leq s} q_i g_i \quad \text{mit } \text{mdeg}(f) \succeq \text{mdeg}(q_i g_i) \quad i = 1, \dots, s$$

6. (Ideal Membership) für alle  $f \in I$  gilt  $f \xrightarrow{G}^* 0$  oder  $f \text{ rem}(G) = 0$ .

## Beispiele von GB

**8.45 Beispiel**  $\{f_1, \dots, f_5\}$ , Beispiel ist Gröbner Basis

$$x^3 - 2xy, \quad x^2y - 2y^2 + x, \quad -x^2, \quad -2xy, \quad -2y^2 + x$$

↪  $\{x^2, xy, y^2 - \frac{1}{2}x\}$  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.

$$\text{z.B. } x \succ y : \{x - 2y^2, y^3\}$$

$$y \succ x : \{y^2 - \frac{1}{2}x, yx, x^2\}$$

**Beachte:** Termideale, Monomideale und homogene Ideale haben reduzierte Gröbner Basen, die **unabhängig** von der Ordnung sind.

# Homogene Ideale

Ein Polynom  $f = \sum \alpha_e x^e$  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 (Forts.)

**8.46 Lemma** Sei  $I \subseteq R = K[x_1, \dots, x_n]$ ,  $G$  Gröbnerbasis von  $I$ .

1.  $V = R/I$  ist  $K$ -Vektorraum.
2.  $B = \{t \in T[X] : t = \text{rep}_G(t)\}$ . Dann ist  $\bar{B} = \{\bar{t} : t \in B\}$  eine  $K$ -Vektorraumbasis von  $V$ .
3.  $V$  hat endliche  $K$ -Dimension gdw  $I$  nulldimensional.
  - ▶ Eine Teilmenge  $S \subseteq R$  ist nulldimensional, falls  $\text{Zero}(S) \subseteq \bar{K}^n$ , ( $\bar{K}$  alg. Abschluss von  $K$ ) endlich ist
  - ▶ hier  $\text{Var}_{\bar{K}}(I)$  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:**

“ $\Leftarrow$ ” Sei  $0 \neq f_i \in I \cap K[x_i]$   $i = 1, \dots, n$ .

Sei  $\xi \in \text{Var}(I)$ ,  $\xi$  muss Nullstelle von  $f_i$  sein, d. h.  $\xi_i$  muss Nullstelle von  $f_i$  sein und davon gibt es nur endlich viele.  $\rightsquigarrow \text{Var}(I)$  ist endlich.

“ $\Rightarrow$ ” angenommen  $\text{Var}_{\bar{K}}(I) = \{\xi^1, \dots, \xi^m\}$  mit  $\xi^j = (\xi_1^j, \dots, \xi_n^j)$ . Da  $\xi_i^j \in \bar{K}$ , gibt es Polynome  $f_i \in K[x_i]$  mit  $f_i(\xi_i^j) = 0$  für  $j = 1, \dots, m$ , d. h.  $f_i$  ist null auf  $\text{Var}(I)$ .

Nach **Hilbert's Nullstellensatz** gibt es  $m_i \in \mathbb{N}^+$  mit  $f_i^{m_i} \in I$ , d. h.

$I \cap K[x_i] \neq \{0\}$ .



## Anwendungen von Gröbner Basen (Forts.)

**8.48 Folgerung** Sei  $G$  eine Gröbner Basis für  $I$ .

$I$  ist **nulldimensional** gdw für jedes  $i = 1, \dots, n$  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  $i = 1, \dots, k$ :

$$\begin{aligned} I \cap K[x_i] \neq \{0\} & \text{ gdw } \langle LT(I) \rangle \cap K[x_i] \neq \{0\} \\ & \text{ gdw } LT(G) \cap K[x_i] \neq \emptyset \\ & \text{ gdw } B = \{t \in T[x] : t = rep_G(t)\} \text{ ist endlich.} \end{aligned}$$

**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  $x_i^l \in B$  für alle  $l \geq 0$ , d. h.  $B$  ist unendlich und  $V$  ist  $\infty$ -dimensional als  $K$ -Vektorraum.





## Beispiele (Forts.)

**8.50 Beispiel** Folgende Polynomrelationen seien gegeben.

$$\begin{aligned}
 f_1 &= 4z - 4xy^2 - 16x^2 - 1 = 0 \\
 I :: f_2 &= 2y^2z + 4x + 1 = 0 \quad f_i \in \mathbb{R}[x, y, z] \\
 f_3 &= 2x^2z + 2y^2 + x = 0
 \end{aligned}$$

► Folgt hieraus

$$g(x, y) = 4xy^2 + 16x^2y^2 + y^2 + 8x + 2 = 0$$

Für  $(x, y)$  mit  $\exists z : (x, y, z) \in \text{Var}(I)$ ?

► Ist  $g \in I = \langle f_1, f_2, f_3 \rangle$ ?  $g$  ist irreduzibel bzgl.  $\{f_1, f_2, f_3\}$

## Beispiele (Forts.)

- ▶ Berechne Gröbner Basis bzgl. lex. Ordnung  $x < y < z$  liefert:

$$g_1 = 32x^7 - 216x^6 + 34x^4 - 12x^3 - x^2 + 30x + 8$$

$$g_2 = 2745y^2 - 112x^6 - 812x^5 + 10592x^4 - 61x^3 - 812x^2 + 988x + 2$$

$$g_3 = 4z - 4xy^2 - 16x^2 - 1$$

- $\rightsquigarrow \text{Var}(I)$  ist endlich.

$$g \xrightarrow[G]{*} 0 \quad , \text{ d. h. } \quad g(x, y) = 0 \quad \text{ auf } \text{Var. } I.$$

- ▶ Lösung von Systemen von Polynomgleichungen

$$V = \{a \in K^n : f_1(a) = \dots = f_m(a) = 0\}$$

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

**8.51 Definition** Sei  $I \subseteq K[x_1, \dots, x_n] = R$ , **radikal**( $I$ ) ist ideal in  $R$  mit:

$$f \in \text{radikal}(I) \text{ gdw } f^n \in I \text{ für ein } n \in \mathbb{N}^+$$

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!**.
- ▶  $I = \langle f_1, \dots, f_n \rangle$ ,  $f_i \in R_0[x_1, \dots, x_d]$   
 $\text{Zero}_{R_1}(I) = \text{Var}_{R_1}(I) = \{a \in \mathbb{A}^d(R_1) : f_i(a) = 0, i = 1, \dots, n\}$

## Eliminationseigenschaften - Polynomgleichungen (Forts.)

- $U \mapsto \text{Ideal}(U)$  für  $U \subseteq \mathbb{A}^d(R_1)$
- $I \mapsto \text{Zero}_{R_1}(I)$  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  $I \subseteq R_0[x_1, \dots, x_d]$  und  $U \subseteq \mathbb{A}^d(R_1)$ .
  - ▶  $I \subseteq \text{Ideal}[\text{Zero}_{R_1}(I)]$
  - ▶  $U \subseteq \text{Zero}_{R_1}[\text{Ideal}(U)]$
  - ▶  $\text{Zero}[\text{Ideal}[\text{Zero}(I)]] = \text{Zero}(I)$   $I \subseteq R_0[x_1, \dots, x_d]$
  - ▶  $\text{Ideal}[\text{Zero}[\text{Ideal}(U)]] = \text{Ideal}(U)$   $U \subseteq \mathbb{A}^d(R_1)$
- ▶ Schränkt man die Abbildungen auf ideale und alg. Mengen, sind sie dann **invers zueinander**?  
**Nur für ideale, die radikal sind**, d. h.  $f^n \in I, n \geq 1 \rightsquigarrow f \in I$ .  
 Da  $\text{Ideal}(U)$  stets radikal ist.

## Hilbert's Nullstellensatz

### ▶ Hilberts Nullstellensatz (schwache Form)

Sei  $D$  noetherscher ZPE-Ring,  $\bar{D}$  alg. Abschluss.

Ein Ideal  $I \subseteq D[x_1, \dots, x_d]$  hat keine Nullstellen in  $\mathbb{A}^d(\bar{D})$  gdw  $I$  enthält nichtriviales Element von  $D$ .

### ▶ Hilberts Nullstellensatz (starke Form)

Sei  $D$  wie oben.  $I \subseteq D[x_1, \dots, x_d]$  Ideal und  $f \in D[x_1, \dots, x_d]$ .

$f$  verschwindet auf  $\text{Var}_{\bar{D}}(I)$  gdw es gibt  $m \geq 0$ ,  $0 \neq a \in D$  mit  $a \cdot f^m \in I$ .

$\hookrightarrow$  d.h.  $f \in \sqrt{I}$  (Körperfall).

$\hookrightarrow f_1(\bar{x}) = 0 \wedge \dots \wedge f_m(\bar{x}) = 0 \Rightarrow f(\bar{x}) = 0$

$\rightsquigarrow f \in \sqrt{\langle f_1, \dots, f_m \rangle}$

## Hilbert's Nullstellensatz: Motivation

► Offenbar gilt:

- Starke Form  $\rightsquigarrow$  schwache Form.

- Schwache Form  $\rightsquigarrow$  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  $f_1, \dots, f_m \rightsquigarrow$  es gibt  $0 \neq a \in D \cap \langle f_1, \dots, f_m, g \rangle$

$$a = \sum_{i=1}^m \alpha_i f_i + \beta(1 - zf) \quad \alpha_i, \beta \in D[x_1, \dots, x_d, z]$$

- Setze  $z = 1/f \rightsquigarrow a = \sum_{i=1}^m \alpha'_i f_i$  mit  $\alpha'_i \in D(x_1, \dots, x_d)$ .

Rationale Funktionen mit Nenner Potenz von  $f \rightsquigarrow$

- $a \cdot f^n = \sum (\alpha'_i f^{m_i}) f_i$  mit  $\alpha'_i f^{m_i} \in D[x_1, \dots, x_d]$ .

## Anwendungen von Hilbert's Nullstellensatz

### Anwendung: Radikal-Membership-Problem

- Seien  $f_1, \dots, f_m \in K[x_1, \dots, x_d]$ ,  $f \in K[x_1, \dots, x_d]$

Frage: Gilt  $f \in \sqrt{\langle f_1, \dots, f_m \rangle}$ ?

$$f \in \sqrt{\langle f_1, \dots, f_m \rangle} \quad \text{gdw} \quad \begin{array}{l} f \text{ verschwindet auf } \text{Var}_K(\langle f_1, \dots, f_m \rangle) \\ f_1(\bar{x}) = f_2(\bar{x}) = \dots = (1 - z \cdot f)(\bar{x}) = 0 \\ \text{hat keine Lösungen in } \bar{K}^d \end{array}$$

$$\text{d.h. } f \in \sqrt{\langle f_1, \dots, f_m \rangle} \quad \text{gdw} \quad \begin{array}{l} 1 \in \langle f_1, \dots, f_m, 1 - zf \rangle_{K[x_1, \dots, x_d, z]} \\ 1 \in GB(f_1, \dots, f_m, 1 - zf) \end{array}$$

- Frage: Gilt  $\text{Var}_{\bar{K}}(\langle f_1, \dots, f_m \rangle) = \emptyset$

Lösung: Ja, falls  $1 \in GB(f_1, \dots, f_m)$ .

# Anwendungen: Eliminationseigenschaften, Varietäten

- **Frage:** Gilt  $\text{Var}_{\bar{K}}(\langle f_1, \dots, f_m \rangle)$  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  $\text{Var}_{\bar{K}}(\langle f_1, \dots, f_m \rangle)$  sei endlich:

- **Frage:** Kann man die Lösungen explizit darstellen?

## 8.52 Satz Eliminationseigenschaft für Gröbner Basen.

Sei  $G$  Gröbner Basis von  $I = \langle f_1, \dots, f_m \rangle$  bzgl. der Lex-Ordnung mit  $x_1 < x_2 < \dots < x_n$ . Dann gilt

$$J = I \cap K[x_1, \dots, x_j] = \langle G \cap K[x_1, \dots, x_j] \rangle_{K[x_1, \dots, x_j]}$$

Insbesondere ist  $G \cap K[x_1, \dots, x_j]$  eine  $G$ -Basis für  $J$ .

## Eliminationseigenschaften: Beweis

Beweis:

“ $\supseteq$ ” klar.

“ $\subseteq$ ”  $LT(f) \in T[x_1, \dots, x_i]$  gdw  $f \in K[x_1, \dots, x_i]$ ,  
 $f \in I \cap K[x_1, \dots, x_i] \rightsquigarrow f \xrightarrow[G]{*} 0$  und die Leiterterme der Basispolynome bei  
 der Reduktion liegen in  $T[x_1, \dots, x_i]$ , d. h. diese Basispolynome liegen in  
 $G \cap K[x_1, \dots, x_i]$ .

**8.53 Folgerung** Sei  $G$   $G$ -Basis von  $I$  bzgl. der Lex Ordnung mit  
 $x_1 < x_2 < \dots < x_n$ .

$I$  ist nulldimensional gdw für jedes  $i = 1, \dots, n$  gibt es ein  $g_i \in G$  mit  
 $HT(g_i) \in K[x_1, \dots, x_i] \setminus K[x_1, \dots, x_{i-1}]$ .

## Eliminationseigenschaften: Beweis (Forts.)

Beweis:

“ $\hookrightarrow$ ” klar

“ $\Leftarrow$ ”  $\xi = (\xi_1, \dots, \xi_n) \in \text{Var}_{\bar{K}}(I)$ . So  $g_i(\xi_1, \dots, \xi_i) = 0 \quad i = 1, \dots, n$ .

- $i = 1$   $g_1$  Polynom in einer Var.  $\rightsquigarrow$  endlich viele Nullstellen, d.h. nur endlich viele Möglichkeiten für  $\xi_1$ .
- **Induktiv**  $\rightsquigarrow g_i(\xi_1, \dots, \xi_{i-1}, \xi_i) = 0$  für  $\xi_1, \dots, \xi_{i-1}$  nur endlich viele Möglichkeiten  $\rightsquigarrow$  für  $\xi_i$  nur endlich viele Möglichkeiten.

## Anwendung: Polynomgleichungen

### 8.54 Beispiel Pol. Gleichungssystem

$$\begin{array}{rcllcl}
 f_1 & :: & 4xz & -4xy^2 & -16x^2 & -1 & = 0 \\
 f_2 & :: & & 2y^2z & +4x & +1 & = 0 \\
 f_3 & :: & & 2x^2z & +2y^2 & +x & = 0
 \end{array}
 \quad \mathbb{Q}[x, y, z]$$

$x < y < z$

► Gröbner Basis bzgl. lex. Ordnung:

$$\begin{array}{rcllclcl}
 g_1 & = & 65z & +64x^4 & -423x^3 & +168x^2 & -354x & +104 \\
 g_2 & = & 26y^2 & -16x^4 & +108x^3 & -16x^2 & +17x & \\
 g_3 & = & 32x^5 & -216x^4 & +64x^3 & -42x^2 & +32x & +5
 \end{array}$$

## Anwendung: Polynomgleichungen (Forts.)

↪  $\text{Var}_{\mathbb{C}}(\langle f_1, f_2, f_3 \rangle)$  ist endlich,  $\text{DIM}_K(\mathbb{Q}[x, y, z]/I) = 10$ ,  
d. h.  $|\text{Zero}(f_1, f_2, f_3)| = 10$  (Nullstellen mit Vielfachheit zählen).

▶ Lösungen von  $g_3$  als Parameter:  $\alpha$  (5-Nullstellen)

$$\left(\alpha, \pm \frac{1}{\sqrt{26}} \sqrt{\alpha} \sqrt{16\alpha^3 - 108\alpha^2 + 16\alpha - 17}, \frac{1}{65} (64\alpha^4 - 423\alpha^3 + \dots)\right)$$

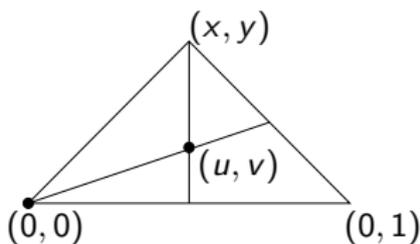
↪ Darstellung von  $\text{Var}_{\mathbb{C}}(\langle f_1, f_2, f_3 \rangle)$ .



## Automatisches Beweisen in der Geometrie

**8.56 Beispiel** Dreieck, Schnitt der Medianen, Formulierung:

$$f_1, f_2 \in \mathbb{R}[u, v, x, y]$$



- ▶  $f_1 = 0 \wedge f_2 = 0 \Rightarrow g_1 = 0 \wedge g_2 = 0 \wedge g_3 = 0,$
- falls  $g_1, g_2, g_3 \in \langle f_1, f_2 \rangle,$  so ok.  $g_1$  bereits gezeigt.

▶  $GB(\langle f_1, f_2 \rangle) \quad u \succ v \succ x \succ y$  lex Ordnung.

$$f_1 = uy - vx - v \quad f_2 = uy - vx + 2v - y$$

$$\hookrightarrow S(f_1, f_2) = f_1 - f_2 = -3v + y = -g_3$$

$\{f_1, f_2, g_3\}$  ist Gröbner Basis.





## Implizitierung : Lösungsansatz

Lösung mit  $G$ -Basen:

- ▶ 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  $t_1 \succ \dots \succ t_m \succ x_1 \succ \dots \succ x_n \prec_{lex}$ .

↪ Einige der  $g$  in  $GB(J)$  hängen nur von  $x_1, \dots, x_n$  ab, dies sind Kandidaten für die Implizitierung.

### 8.58 Beispiel

- $t \succ z \succ y \succ x$
- ▶  $GB\{x - t, y - t^2, z - t^3\}$  ist  $\{t - x, z - x^3, y - x^2\}$
- ▶  $GB\langle x - t^2, y - t^3, z - t^4 \rangle$  ist  $\{t^2 - x, ty - x^2, tx - y, z - x^2, y^2 - x^3\}$
- ▶ Die Varietät, die von  $G \cap K[x_1, \dots, x_n]$  definiert wird, ist die kleinste Varietät (Alg-Menge), die das Bild der Parametrisierung enthält.





## Basis für Syzygienmodul für Gröbner-Basen (Forts.)

### Beweis:

- Jedes Element in  $S$  ist eine Syzygie von  $G$ .  
Sei  $s_{ij} = p_{ij}e_i - q_{ij}e_j - (k_{ij}^1 \cdots k_{ij}^s)$  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  $z = (z_1, \dots, z_s) \in R^s$  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  $\text{syz}(G)$ .

## Basis für Syzygienmodul für Gröbner-Basen (Forts.)

**8.60 Satz** Sei  $F = (f_1, \dots, f_s)^T$  mit  $f_i \in K[\bar{x}]$  und  $G = (g_1, \dots, g_m)^T$  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  $\text{syz}(G)$  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.)

$$\text{Sei } Q := \begin{pmatrix} I_s - B \cdot A \\ R \cdot A \end{pmatrix}_{s+r, s}$$

Dann bilden die Zeilen von  $Q$  eine Basis für  $\text{syz}(F)$ .

## Beweis

**Beweis** Seien  $b_1, \dots, b_{s+r}$  Polynome  $b = (b_1 \dots b_{s+r})$ .

$$\begin{aligned}
 (b \cdot Q)F &= ((b_1, \dots, b_s)(I_s - BA) + (b_{s+1} \dots b_{s+r})RA)F \\
 &= (b_1 \dots b_s)(F - \underbrace{BAF}_{=F}) + (b_{s+1} \dots b_{s+r})R \underbrace{AF}_{=G} \\
 &= 0
 \end{aligned}$$

↪ 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

- ▶  $f_1 z_1 + \dots + f_s z_s = f$  Existenz gdw  $f \in \langle F \rangle$  Gröbner Basis für  $F$ ,  
 $G = A \cdot F$ ,  $f \xrightarrow[G]{*} f' \neq 0$  nicht lösbar, sonst  $f \xrightarrow[G]{*} 0$ ,  
 $g_1 h'_1 + \dots + g_m h'_m = f \iff H = (h'_1 \dots h'_m)A$  ist **partikuläre Lösung**.



# 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  $\mathbb{Q}[x_1, \dots, x_s]$ .)
- Mayr & Mayer 82:  $IM$  ist  $EXPSPACE$ -hart für allg. Ideale.
- Mayr 89,92:  $IM$  ist in  $EXPSPACE$ , d h.  $IM$  ist **vollständig**.

## Zur Komplexität der Berechnung von $G$ -Basen (Forts.)

- ▶  $f \in \langle f_1, \dots, f_m \rangle$  gdw  $f \xrightarrow[GB(f_1, \dots, f_m)]{*} 0$
- $f \xrightarrow[GB]{*} 0$  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( $2^{\text{lin}}$ ) wird **nur Platz auf Arbeitsband** gemessen).

- ▶ Gleiche Ergebnisse gelten für binomial-ideale, d. h. Ideale werden durch Binome  $x^\alpha - x^\beta$  erzeugt.

## Zur Komplexität der Berechnung von $G$ -Basen (Forts.)

- ▶ **Bürgisser** (98)  $K$   $\infty$ -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  $q_i$  doppeltextponentiell. Siehe auch Mayr & Mayer 82.
- ▶ Die Grade der Polynome in einer reduzierten Gröbner Basis für  $\langle f_1, \dots, f_s \rangle \subseteq F[x_1, \dots, x_n]$  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  $2^{2^{cn}}$  Elemente enthält mit Elementen vom Grad mindestens  $2^{2^{c^*n}}$  ( $c, c^* \in \mathbb{R}^+$ ).
- ▶ Varietäten:  $V(f_1, \dots, f_s)$  ist leer gdw  $1 \in \langle f_1, \dots, f_s \rangle$  in PSPACE.
- ▶ Für nulldimensionale Ideale einfache Exp-Schranken.  
Lakshman (90), Mayr (97) Übersicht.
- ▶ Systeme: Macaulay, Singular. Projekte: Posso Frisco.