Angewandte Algebra

Modul Math Ma ANGALG
Kurzskript

Jens Zumbrägel

TU Dresden · WiSe 2016/17

In dieser Vorlesung für die Masterstudiengänge des Fachbereichs Mathematik werden algebraische und algorithmische Methoden vorgestellt im Hinblick auf ihre Anwendungen, insbesondere in der Codierungstheorie und der Kryptologie.

Inhaltsverzeichnis

1 Grundlagen aus Algebra und Computeralgebra

1.1 Arithmetik großer Zahlen

Sei $B \geq 2$ Basis (z. B. $B = 10$, $B = 2^{64}$). Eine Speichereinheit („Ziffer“, „Wort“) kann ein Element in $\{0, \ldots, B-1\}$ speichern. Darstellung einer (beliebig großen) Zahl $\alpha \in \mathbb{Z}$ in Basis B:

$$\alpha = (-1)^s \sum_{i=0}^{n-1} a_i B_i \qquad (s \in \{0,1\},\ a_i \in \{0, \ldots, B-1\}) .$$

Die Algorithmen für arithmetische Operationen ähneln den aus der Schule bekannten „schriftlichen Rechenverfahren“.

Addition Prozessor hat Befehl ADD:

input $a, b \in \{0, \ldots, B-1\},\ \varepsilon \in \{0,1\}$ (Übertrag, carry)
output $c \in \{0, \ldots, B-1\},\ \varepsilon^* \in \{0,1\}$ so dass $a + b + \varepsilon = \varepsilon^* B + c$

Für beliebige Zahlen hat man damit den Algorithmus „Addition“:

input $\alpha = \sum_{i=0}^{n-1} a_i B^i,\ \beta = \sum_{i=0}^{n-1} b_i B^i \quad (a_i, b_i \in \{0, \ldots, B-1\})$
output $\gamma = \sum_{i=0}^{n} c_i B^i$ so dass $\alpha + \beta = \gamma \quad (c_i \in \{0, \ldots, B-1\})$

1. $\varepsilon_0 := 0$
2. for $i := 0 \ldots n-1$ do
   a. $(c_i, \varepsilon_{i+1}) := \mathrm{ADD}(a_i, b_i, \varepsilon_i)$
3. $c_n := \varepsilon_n$

Der Einfachheit halber sind hier α, β nicht-negativ und von gleicher Länge (ggfs. Nullen ergänzen).

Laufzeit: falls ADD k Zyklen benötigt, dann $k \cdot n +$ Zyklen für Kontrollfluss etc., also in $O(n)$. (Erinnere: „$f(n)$ in $O(g(n))$“ heißt $\exists\, N, C > 0\ \forall n \geq N$: $f(n) \leq C g(n)$.)

(Subtraktion: Übung.)

Multiplikation Prozessor hat Befehl MUL:

input $a, b \in \{0, \ldots, B-1\}$
output $c, d \in \{0, \ldots, B-1\}$ so dass $a \cdot b = dB + c$

Wir benötigen zunächst eine Subroutine „Ziffer mal Zahl“:

input $a,\ \beta = \sum_{i=0}^{n-1} b_i B^i \quad (a, b_i \in \{0, \ldots, B-1\})$
output $\gamma = \sum_{i=0}^{n} c_i B^i$ so dass $\gamma = a \cdot \beta \quad (c_i \in \{0, \ldots, B-1\})$

1. $c_0 := 0$
2. for $i := 0 \ldots n-1$ do
   a. $(x, y) := \mathrm{MUL}(a, b_i)$
   b. $(c_i, \varepsilon) := \mathrm{ADD}(c_i, x, 0)$
   c. $c_{i+1} := y + \varepsilon$

Laufzeit: n MUL, n ADD, sowie Kontrollfluss, Inkremente, etc.

Damit hat man den Algorithmus „Multiplikation“.

input $\alpha = \sum_{i=0}^{n-1} a_i B^i,\ \beta = \sum_{i=0}^{m-1} b_i B^i$
output $\gamma = \sum_{i=0}^{m+n-1} c_i B^i$ so dass $\gamma = \alpha \cdot \beta$

1. for $i := 0 \ldots n-1$ do
   a. $\delta_i := a_i \cdot \beta$ („Ziffer mal Zahl“)
2. $\gamma := \sum_{i=0}^{n-1} \delta_i B^i$ (Shift und „Addition“)

Laufzeit: mn MUL, $mn + (n-1)(m+1)$ ADD, also in $O(mn)$ Operationen.

Division mit Rest Prozessor-Befehl DIV:

input $a, b, d \in \{0, \ldots, B-1\}$ mit $bB + a < dB$
output $c \in \{0, \ldots, B-1\}$ so dass $\lfloor (bB+a)/d \rfloor = c$

Wir haben einen Algorithmus „Division“:

input $\alpha = \sum_{i=0}^{n-1} a_i B^i$, $\delta = \sum_{i=0}^{m-1} d_i B^i$
output $\beta = \sum_{i=0}^{n-m} b_i B^i$, $\rho = \sum_{i=0}^{m-1} r_i B^i$ mit $\alpha = \beta \cdot \delta + \rho$ und $\rho \in [0, \delta)$ (Rest)

1. $\rho := \alpha$
2. wähle $k \in \{n-m, n-m+1\}$ mit $\delta B^{k-1} \le \alpha$, aber $\delta B^k > \alpha$
3. for $i := k-1 \ldots 0$ do
   a. $b_i := \mathrm{DIV}(r_{i+m-1}, r_{i+m}, d_{m-1})$
   b. while $b_i B^i \cdot \delta > \rho$ do („Ziffer mal Zahl“)
      i. $b_i := b_i - 1$
   c. $\rho := \rho - b_i B^i \cdot \delta$

Laufzeit: benötigt $k \le n-m+1$ mal je *etwa* 1 DIV, m MUL, $2m$ ADD; also in $O\big((n-m)m\big)$ Operationen. (Für eine genauere Laufzeitanalyse müsste Schritt *3a.* unter Verwendung der zusätzlichen Ziffern r_{i+m-2} und d_{m-2} verfeinert werden.)

1.2 Monoidring, Polynomring

Sei $(R, +, \cdot)$ (kurz: R) ein Ring (mit Eins), das heißt $(R, +)$ ist abelsche Gruppe, $(R, \cdot)$ ist Monoid und beide Distributivgesetze gelten.

Definition 1.1. Sei $(M, *)$ ein Monoid, dann ist

$$R[M] \;:=\; R[(M, *)] \;:=\; (R^{(M)}, +, *)\,,$$

mit $R^{(M)} := \{f \colon M \to R \mid \operatorname{supp} f \text{ endlich}\}$, sowie

$$(f+g)(m) := f(m) + g(m)\,, \qquad (f * g)(m) := \sum_{\substack{k,\ell \in M \\ k*\ell = m}} f(k) \cdot g(\ell)$$

für $m \in M$, ein Ring, der *Monoidring* von M über R.

Zu $k \in M$ definiere $\delta_k \in R^{(M)}$ durch

$$\delta_k(m) := \begin{cases} 1 & \text{falls } k = m\,, \\ 0 & \text{sonst}\,. \end{cases}$$

Dann gilt $\delta_k * \delta_\ell = \delta_{k*\ell}$ für alle $k, \ell \in M$.

Jedes Element $f \in R[M]$ hat zwei Darstellungen:

i) „dünn (sparse)“: als Summe $f = \sum_{s \in S} f(s)\, \delta_s$, wobei $S := \operatorname{supp} f \subseteq M$;

ii) „dicht (dense)“: als Abbildung $f : M \to R$, $m \mapsto f(m)$; falls $\operatorname{supp} f \subseteq T$ (fix) auch $f|_T : T \to R$, $m \mapsto f(m)$.

Produkt in Darstellung i): sei $f = \sum_{s \in S} f(s)\, \delta_s$ und $g = \sum_{t \in T} g(t)\, \delta_t$, dann ist also $f * g = \sum_{s \in S, t \in T} f(s)\, g(t)\, \delta_{s*t}$.

Polynomring Bezeichnung $\mathbb{N} := \{0, 1, 2, \dots\}$.

Definition 1.2. Sei $n \in \mathbb{N}_{>0}$, dann ist

$$R[X_1, \dots, X_n] \;:=\; R[(\mathbb{N}^n, +)]$$

der *multivariate Polynomring* in n Unbekannten über R.

Speziell ist $R[X] := R[X_1] := R[(\mathbb{N}, +)]$ der (univariate) *Polynomring* über R. In Polynomringen schreiben wir für $f * g$ üblicherweise $f \cdot g$.

Im $R[X]$ sei $X := \delta_1$, dann gilt $X^k = \delta_1 \cdot \ldots \cdot \delta_1 = \delta_{1+\dots+1} = \delta_k$ für alle $k \in \mathbb{N}$. Somit hat jedes *Polynom* $f \in R[X]$ die vertraute Darstellung

$$f \;=\; \sum_{k\in\mathbb{N}} f(k)\,\delta_k \;=\; \sum_{k=0}^{n} a_k X^k\,, \quad \text{wobei } a_k := f(k)\,.$$

Wir haben die Gradfunktion

$$\deg\colon R[X] \to \mathbb{N} \cup \{-\infty\}\,, \quad f \mapsto \deg f = \begin{cases} \max\{k \mid f_k \neq 0\} & \text{falls } f \neq 0\,, \\ -\infty & \text{falls } f = 0\,. \end{cases}$$

Ein Ring R heißt *Integritätsring*, falls $(R, \cdot)$ kommutativ und nullteilerfrei ist, das heißt $rs = 0$ impliziert $r = 0$ oder $s = 0$, für alle $r, s \in R$.

Proposition 1.3. *Sei R Integritätsring.*

1. *Sei $f, g \in R[X]$, dann gilt* $\deg(f \cdot g) = \deg f + \deg g$.
2. *$R[X]$ ist Integritätsring.*

Arithmetik Wir betrachten Addition und Subtraktion, sowie Multiplikation in einem Monoidring $R[M]$ und messen die Anzahl der Operationen in R. Da es keine Überträge gibt, sind die Algorithmen konzeptionell einfacher als bei der Arithmetik großer Zahlen.

Addition bzw. Subtraktion, nach Definition komponentenweise. Im $R[X]$ ist zum Beispiel $\sum_{i=0}^{n} a_i X^i + \sum_{i=0}^{m} b_i X^i = \sum_{i=0}^{\max(m,n)} (a_i + b_i) X^i$, dies benötigt $\min(m, n) + 1$ Operationen in R. Beachte allgemein

$$\operatorname{supp}(f + g) \;\subseteq\; \operatorname{supp} f \cup \operatorname{supp} g\,.$$

Multiplikation. Betrachte die „dünne“ Darstellung; sei $f = \sum_{s\in S} f(s)\,\delta_s$ und $g = \sum_{t\in T} g(t)\,\delta_t$, dann berechne $f * g = \sum_{s\in S, t\in T} f(s)g(t)\,\delta_{s*t}$. Beachte hier

$$\operatorname{supp}(f * g) \;\subseteq\; \operatorname{supp} f * \operatorname{supp} g\,.$$

Die Laufzeit beträgt $|S| \cdot |T|$ Multiplikationen und $|S| \cdot |T| - |S * T|$ Additionen in R. Im $R[X]$ benötigt $(\sum_{i=0}^{n} a_i X^i) \cdot (\sum_{i=0}^{m} b_i X^i)$ also $(n + 1)(m + 1)$ Multiplikationen und nm Additionen, also $O(mn)$ R-Operationen.

Division. Im Polynomring $R[X]$ hat man einen Algorithmus „Division mit Rest“:

input $f = \sum_{i=0}^{n} a_i X^i$, $g = \sum_{i=0}^{m} b_i X^i$, $b_m \in R^*$ Einheit
output $q = \sum_{i=0}^{n-m} q_i X^i$, $r = \sum_{i=0}^{m-1} r_i X^i$ mit $f = q \cdot g + r$ und $\deg r < \deg g$

1. $r := f$
2. for $i := n-m \dots 0$ do
   a. $q_i := r_{i+m}\, b_m^{-1}$
   b. $r := r - q_i X^i \cdot g$

Laufzeit: benötigt $n - m + 1$ mal je 1 Division und $m + 1$ Multiplikationen und Subtraktionen, also in $O\big((n - m)m\big)$ Operationen im Ring R.

1.3 Euklidischer Algorithmus

Teiler, ggT, kgV Sei R ein Ring. Betrachte auf R die Relation „teilt“

$$a \mid b \quad :\Leftrightarrow \quad \exists\, x \in R :\, b = xa \quad \Leftrightarrow \quad b \in Ra \quad (a, b \in R).$$

Diese ist reflexiv und transitiv, d. h. definiert eine Präordnung auf R. Wir definieren

$$a \sim b \quad :\Leftrightarrow \quad a \mid b \;\wedge\; b \mid a \quad (a, b \in R),$$

genannt a *assoziiert* zu b; dies ist eine Äquivalenzrelation. Auf den Äquivalenzklassen $R/\!\sim$ definiere dann $[a] \mid [b]$ falls $a \mid b$; dies ist wohldefiniert und eine Ordnungsrelation.

Bemerkung 1.4. Ist R ein Integritätsring, dann gilt

$$a \sim b \quad \Leftrightarrow \quad \exists\, u \in R^* :\, a = ub\,.$$

Definition 1.5. Sei $a, b, c \in R$, so heißt c ein *ggT* (größter gemeinsamer Teiler) von a und b, falls $[c] = [a] \wedge [b] = \inf\{[a], [b]\}$ gilt in $R/\!\sim$, das heißt

$$c \mid a,\; c \mid b \quad \text{und} \quad d \mid a,\; d \mid a \;\Rightarrow\; d \mid c \;\text{ für }\; d \in R\,.$$

Dual heißt c ein *kgV* (kleinstes gemeinsames Vielfaches) von a, b, falls $[c] = [a] \vee [b]$.

Falls ggT bzw. kgV existieren, dann sind sie eindeutig in $R/\!\sim$, also bis auf Assoziiertheit.

Euklidische Ringe

Definition 1.6. Ein Integritätsring R mit Gradfunktion $d : R \to \mathbb{N} \cup \{-\infty\}$ heißt *euklidischer Ring*, falls

$$\forall f, g \in R,\; g \neq 0 \;\; \exists\, q, r \in R :\; f = qg + r \;\text{ mit }\; d(r) < d(g)\,.$$

Man zeigt leicht, dass euklidische Ringe Hauptidealringe sind, d. h. jedes Ideal wird von einem Element erzeugt. Wichtige Beispiele für euklidische Ringe sind $R = \mathbb{Z}$ mit $d(a) := |a|$, und $R = F[X]$, wobei F Körper, mit $d(a) := \deg a$.

Sei R ein euklidischer Ring mit Gradfunktion d. Für den ggT haben wir nun den „erweiterten euklidischen Algorithmus“:

input $f, g \in R,\; g \neq 0$
output $d, s, t \in R$ mit d ein $\mathrm{ggT}(f, g)$ und $d = sf + tg$

1. $r_0 := f \quad s_0 := 1 \quad t_0 := 0$
   $r_1 := g \quad s_1 := 0 \quad t_1 := 1$
   $i := 1$
2. while $r_i \neq 0$ do
   a. $(q_i, r_{i+1}) := \mathrm{DivMitRest}(r_{i-1}, r_i)$
   b. $s_{i+1} := s_{i-1} - q_i s_i$
   c. $t_{i+1} := t_{i-1} - q_i t_i$
   d. $i := i + 1$
3. $\ell := i - 1$
   $d := r_\ell \quad s := s_\ell \quad t := t_\ell$

Korrektheit: Für alle i gilt

$$\begin{pmatrix} r_i \\ r_{i+1} \end{pmatrix} = Q_i \cdot \ldots \cdot Q_1 \begin{pmatrix} f \\ g \end{pmatrix} \text{ und } \begin{pmatrix} s_i & t_i \\ s_{i+1} & t_{i+1} \end{pmatrix} = Q_i \cdot \ldots \cdot Q_1\,, \text{ mit } Q_j = \begin{pmatrix} 0 & 1 \\ 1 & -q_j \end{pmatrix}.$$

Mit $i = \ell$ ist also $d = sf + tg$ und $\mathrm{ggT}(f,g) \mid d$; umgekehrt ist

$$\begin{pmatrix} f \\ g \end{pmatrix} = Q_1^{-1} \cdot \ldots \cdot Q_\ell^{-1} \begin{pmatrix} d \\ 0 \end{pmatrix} \quad \text{mit} \quad Q_j^{-1} = \begin{pmatrix} q_j & 1 \\ 1 & 0 \end{pmatrix}$$

und daher $d \mid f$, $d \mid g$, also $d \mid \mathrm{ggT}(f,g)$.

Laufzeit: Im Fall $R = F[X]$ sei $n := \deg f \geq \deg g$ und $\delta_i := \deg r_i$, dann gilt

$$n = \delta_0 \geq \delta_1 > \cdots > \delta_\ell \,,$$

also $\ell \leq n+1$. Da Schritt i höchstens $C(\delta_i - \delta_{i-1} + 1)(\delta_i + 1) \leq C(\delta_i - \delta_{i-1} + 1)(n+1)$ F-Operationen benötigt, ist die Gesamtlaufzeit höchstens

$$\sum_{i=1}^{\ell} C(\delta_i - \delta_{i-1} + 1)(n+1) \leq C(n+1)\Big(\sum_{i=1}^{\ell} (\delta_i - \delta_{i-1}) + \sum_{i=1}^{\ell} 1 \Big) \leq C(n+1)(n+n+1) = O(n^2)$$

F-Operationen (diese Laufzeitabschätzung ist korrekt, aber recht grob).

Im Fall $R = \mathbb{Z}$ sei $f \geq g > 0$, dann gilt $f = qg + r \geq g + r > 2r$, somit $r < \frac{1}{2} f$. Also folgt im Algorithmus $r_{i+1} < \frac{1}{2} r_{i-1}$ für alle i. Daraus ergibt sich die Abschätzung $\ell \leq 2 \log_2 f$. Ähnlich wie oben folgt eine Laufzeit, die quadratisch in der Stellenzahl ist.

1.4 Modulare Arithmetik

Restklassenring Sei R ein Ring und $I \leq R$ ein Ideal. Betrachte auf R die Relation

$$a \sim_I b \quad :\Leftrightarrow \quad a - b \in I \,.$$

Dies ist eine Äquivalenzrelation und eine *Kongruenz* auf $(R, +, \cdot)$, das heißt $a \sim_I a'$, $b \sim_I b'$ impliziert $a + b \sim_I a' + b'$ und $a \cdot b \sim_I a' + b'$. Auf der Menge $R/I := \{[r]_I \mid r \in R\}$ der Äquivalenzklassen sind somit Verknüpfungen definiert durch

$$[a] + [b] := [a + b] \,, \qquad [a] \cdot [b] := [a \cdot b] \qquad (a, b \in R) \,,$$

so dass die Ringgesetze gelten, und wir nennen R/I den *Restklassenring* R modulo I.

Arithmetik Wir betrachten $R = \mathbb{Z}$ oder $R = F[X]$ mit F Körper. Dann ist jedes Ideal ein Hauptideal, also ist $I = (m) := mR$ für ein $m \in R$, und es gilt $a \sim_I b \;\Leftrightarrow\; m \mid a - b$. Man kann eine Äquivalenzklasse $[f] \in R/I$ eindeutig repräsentieren durch

- $R = \mathbb{Z}$: kleinstes nicht-negatives Element f' in $f + mR$, also $0 \leq f' < m$,
- $R = F[X]$: Element f' kleinsten Grades in $f + mR$, also $\deg f' < \deg m$.

Bezeichnung $f \bmod m := f'$ (wie oben).

Damit ergibt sich folgender Algorithmus für die Arithmetik in R/I:

input $a, b, m \in R$, wobei $* \in \{+, -, \cdot\}$
output $a * b \bmod m$

1. $f := a * b$
2. $(q, r) := \mathrm{DivMitRest}(f, m)$
3. $a * b \bmod m := r$

Inversion, endliche Körper Sei nun R ein euklidischer Ring.

Satz 1.7. *Sei $m, a \in R$. Es ist $[a]$ genau dann invertierbar in $R/(m)$, wenn $\mathrm{ggT}(a, m) = 1$. Gilt $sa + tm = 1$ mit $s, t \in R$ (erweiterter euklidischer Algorithmus), so ist $[a]^{-1} = [s]$.*

Corollar 1.8. *Ist $m \notin R^*$* irreduzibel, *das heißt $a \mid m$ impliziert $a \sim 1$ oder $a \sim m$, dann ist $R/(m)$ ein Körper.*

Für jede Primzahl $p \in \mathbb{Z}$ erhalten wir so den Körper $\mathbb{F}_p := \mathbb{Z}_p$, und für jedes irreduzible Polynom $f \in \mathbb{F}_p[X]$ vom Grad n haben wir einen Körper $\mathbb{F}_q := \mathbb{F}_p[X]/\langle f \rangle$ mit $q = p^n$ Elementen. (Die Existenz und Eindeutigkeit der endlichen Körper klären wir später.)

Die arithmetischen Operationen in $\mathbb{F}_{p^n}$ benötigen höchstens:

Add., Sub.	n	$\mathbb{F}_p$-Op.
Mul.	$4n^2 + O(n)$	$\mathbb{F}_p$-Op.
Inv.	$6n^2 + O(n)$	$\mathbb{F}_p$-Op.

Also sind alle Verknüpfungen in $O(n^2)$ $\mathbb{F}_p$-Operationen durchführbar.

Analog benötigen die Verknüpfungen in $\mathbb{Z}_p$ höchstens $O\big((\log p)^2\big)$ Wortoperationen. Insgesamt haben somit die Verknüpfungen in jedem endlichen Körper mit q Elementen (höchstens) quadratische Komplexität $O\big((\log q)^2\big)$.

Potenzieren Sei $(M, *)$ Monoid (etwa $(R, \cdot)$, R Ring). Algorithmus „square-and-multiply“:

input $a \in M$, $n \in \mathbb{N}_{>0}$
output $a^n \in M$

1. schreibe $n = \sum_{i=0}^{k} n_i 2^i$ mit $n_k = 1$
   $b_k := a$
2. for $i := k-1 \ldots 0$ do
   if $n_i = 1$
   then $b_i := b_{i+1} * b_{i+1} * a$
   else $b_i := b_{i+1} * b_{i+1}$
3. $a^n := b_0$

Laufzeit: höchstens $k \le \log_2 n$ mal je 1 Quadrierung und 1 Multiplikation.

Hieraus ergibt sich ein alternativer Algorithmus für Inversion in $\mathbb{F}_q$, denn $a^{-1} = a^{q-2}$ für $a \in \mathbb{F}_q^*$. Dieser benötigt jedoch $O\big((\log q)^3\big)$ Operationen.

1.5 Körpererweiterungen

Dieser Abschnitt stellt algebraische Grundlagen für die Theorie endlicher Körper bereit.

Primfaktorzerlegung Sei R ein Integritätsring. Wir wiederholen aus der Algebra:

- Sei $p \in R \setminus (R^* \cup \{0\})$, dann heißt

$$p \textit{ irreduzibel}, \text{ falls } p = ab \Rightarrow a \in R^* \vee b \in R^*,$$
$$p \textit{ prim}, \text{ falls } p \mid ab \Rightarrow p \mid a \vee p \mid b,$$

  und ein Primelement ist stets irreduzibel.

Sei nun R ein Hauptidealring.

- Ist $p \in R$ irreduzibel, dann ist $R/\langle p \rangle$ ein Körper und p ist prim.

- Jedes $r \in R \setminus (R^* \cup \{0\})$ besitzt eindeutige Primfaktorzerlegung, das heißt:
  - Es existieren $p_i \in R$ irreduzibel mit $r = p_1 \cdot \ldots \cdot p_a$.
  - Ist $p_1 \cdot \ldots \cdot p_a = q_1 \cdot \ldots \cdot q_b$ mit $q_j \in R$ irreduzibel, so ist $a = b$ und es gibt $\sigma \in S_a$ mit $q_{\sigma(i)} \sim p_i$ für alle i.

Insbesondere gilt dies für die euklidischen Ringe $\mathbb{Z}$ und $K[X]$, wobei K Körper.

Minimalpolynom Sei $K \subseteq L$ eine Körpererweiterung. Zu $\alpha \in L$ betrachte

$$\mathrm{ev}_\alpha : K[X] \to L\,, \quad f \mapsto f(\alpha)\,;$$

dies ist ein K-Morphismus (d. h. ein Ringmorphismus mit $\mathrm{ev}_\alpha|_K = \mathrm{id}_K$), genannt *Einsetzungsmorphismus*.

Da $K[X]$ ein Hauptidealring ist, gibt es $f \in K[X]$ mit $\ker \mathrm{ev}_\alpha = (f)$. Falls $f \neq 0$ heißt α *algebraisch* über K und wir nennen f (normiert) das *Minimalpolynom* von α über K.

- Ein Minimalpolynom ist stets irreduzibel; ist umgekehrt $g \in K[X]$ normiert, irreduzibel mit $g(\alpha) = 0$, so ist g bereits das Minimalpolynom von α.
- Ist $\alpha \in L$ algebraisch über K, so ist $K[\alpha] := \mathrm{im}\,\mathrm{ev}_\alpha$ ein Körper.

Für $\alpha_1, \ldots, \alpha_m \in L$ sei $K(\alpha_1, \ldots, \alpha_m)$ der kleinste Erweiterungskörper von K, der die Elemente α_i enthält; ist $\alpha \in L$ algebraisch, so gilt also $K(\alpha) = K[\alpha]$.

Zerfällungskörper Sei K ein Körper.

- (Kronecker-Konstruktion) Zu $f \in K[X]$, $\deg f > 0$ existiert eine Körpererweiterung $K \subseteq L$ und $\alpha \in L$ mit $f(\alpha) = 0$ und $L = K(\alpha)$.
- (Fortsetzungslemma) Sei $f_1 \in K_1[X]$ irreduzibel, sei $\varphi : K_1 \to K_2$ ein Isomorphismus und sei $f_2 = \varphi(f_1) \in K_2[X]$, koeffizientenweise angewendet. Sind $K_i \subseteq L_i$ Körpererweiterungen und $\alpha_i \in L_i$ mit $f_i(\alpha_i) = 0$ und $L_i = K(\alpha_i)$, dann existiert ein Isomorphismus

  $$\psi : L_1 \to L_2 \quad \text{mit} \quad \psi|_{K_1} = \varphi \ \text{ und } \ \psi(\alpha_1) = \alpha_2\,.$$

  Mit $K_1 = K_2 = K$ und $\varphi = \mathrm{id}_K$ folgt insbesondere die Eindeutigkeit der Kronecker-Konstruktion für f irreduzibel.

Satz 1.9. *Zu $f \in K[X]$ normiert existiert eine (bis auf K-Isomorphie) eindeutige Körpererweiterung $K \subseteq L$ mit $f = \prod_{i=1}^{n}(X - \alpha_i)$, wobei $\alpha_i \in L$ und $L = K(\alpha_1, \ldots, \alpha_n)$.*

Wir nennen dann L den *Zerfällungskörper* von f über K.

1.6 Endliche Körper

Theorem 1.10 (Existenz und Eindeutigkeit)**.**

1. *Sei K ein endlicher Körper, dann ist $|K| = p^n$ für eine Primzahl p und $n \in \mathbb{N}_{>0}$.*
2. *Zu gegebener Primzahl p und $n \in \mathbb{N}_{>0}$ existiert ein Körper mit p^n Elementen.*
3. *Je zwei Körper mit p^n Elementen sind isomorph.*

Beweisskizze.

1. Sei $p := \operatorname{char} K$ die Charakteristik von K. Da K keine Nullteiler hat, ist p prim und $\mathbb{Z}_p \subseteq K$ eine Körpererweiterung. Also ist $|K| = p^n$ mit $n := \dim_{\mathbb{Z}_p} K$.

2. Wir betrachten zu $f := X^q - X \in \mathbb{Z}_p[X]$, wobei $q = p^n$, den Zerfällungskörper K. Die formale Ableitung ist $f' = -1$, also $\operatorname{ggT}(f, f') = 1$, und somit hat f keine doppelten Nullstellen in K. Sei
$$S := \{\alpha \in K \mid \alpha^q = \alpha\},$$
dann ist also $|S| = q$. Unter Benutzung von $(\alpha + \beta)^q = \alpha^q + \beta^q$ sehen wir, dass S ein Unterkörper von K ist, also ein Körper mit q Elementen.

3. Ist K ein Körper mit $q = p^n$ Elementen, so ist $\mathbb{Z}_p \subseteq K$ Unterkörper, und es gilt $\alpha^{q-1} = 1$ für alle $\alpha \in K^*$, somit $\alpha^q = \alpha$ für alle $\alpha \in K$. Es folgt $X^q - X = \prod_{\alpha \in K}(X - \alpha)$, damit ist K der eindeutige Zerfällungskörper von f über $\mathbb{Z}_p$. □

Der Körper mit $q = p^n$ Elementen wird mit $\mathbb{F}_q$ (oder $\mathrm{GF}(q)$) bezeichnet.

Satz 1.11. *Sei $K \subseteq \mathbb{F}_{p^n}$, dann ist $|K| = p^m$ mit $m \mid n$. Ist umgekehrt $m \mid n$, so hat $\mathbb{F}_{p^m}$ genau einen Unterkörper K mit $|K| = p^m$.*

Beweisskizze. Wegen $\operatorname{char} K = p$ ist $|K| = p^m$, und da $\mathbb{F}_{p^n}$ ein K-Vektorraum ist, folgt $p^n = (p^m)^d = p^{md}$ mit $d = \dim_K \mathbb{F}_{p^n}$, also $n = md$.

Ist umgekehrt $m \mid n$, so gilt $p^m - 1 \mid p^n - 1$ und somit $X^{p^m-1} - 1 \mid X^{p^n-1} - 1$, also $X^{p^m} - X \mid X^{p^n} - X$. Die Nullstellen von $X^{p^m} - X$ bilden $\mathbb{F}_{p^m}$ und liegen in $\mathbb{F}_{p^n}$. Zur Eindeutigkeit, jedes Element eines Körpers K mit p^m Elementen ist Nullstelle von $X^{p^m} - X$, aber es gibt nur p^m Nullstellen. □

Satz 1.12. *Für jeden endlichen Körper $\mathbb{F}_q$ ist die multiplikative Gruppe $\mathbb{F}_q^*$ zyklisch.*

Beweisskizze. Sei $q - 1 = p_1^{e_1} \cdot \ldots \cdot p_r^{e_r}$ die Primfaktorzerlegung. Für alle i hat das Polynom $X^{(q-1)/p_i} - 1$ weniger als $q-1$ Nullstellen, also gibt es eine Nichtnullstelle $\alpha_i \in \mathbb{F}_q^*$; betrachte $\beta_i := \alpha_i^{(q-1)/p_i^{e_i}}$, dann ist $\operatorname{ord} \beta_i = p_i^{e_i}$. Für $\beta := \beta_1 \cdot \ldots \cdot \beta_r$ ist dann $\operatorname{ord} \beta = q - 1$. □

Corollar 1.13. *Für alle endlichen Körper $\mathbb{F}_q$ und $n \in \mathbb{N}_{>0}$ existiert ein irreduzibles Polynom über $\mathbb{F}_q$ vom Grad n.*

Beweis. Sei α ein Erzeuger von $\mathbb{F}_{q^n}^*$. Dann ist das Minimalpolynom von α über $\mathbb{F}_q$ ein Polynom wie gewünscht. □

Körperautomorphismen Die *Frobeniusabbildung*

$$\mathrm{Frob}_q = \varphi : \mathbb{F}_{q^n} \to \mathbb{F}_{q^n}, \quad \alpha \mapsto \alpha^q$$

ist ein $\mathbb{F}_q$-Automorphismus von $\mathbb{F}_{q^n}$ der Ordnung n.

Proposition 1.14. *Sei $f \in \mathbb{F}_q[X]$ irreduzibel, $\deg f = n$. Dann hat f eine Nullstelle α in $\mathbb{F}_{q^n}$ und $\alpha, \varphi(\alpha), \ldots, \varphi^{n-1}(\alpha)$ sind die verschiedenen Nullstellen von f; also ist $\mathbb{F}_{q^n}$ der Zerfällungskörper von f über $\mathbb{F}_q$.*

Beweisskizze. Es ist $\alpha := [X] \in \mathbb{F}_q[X]/(f) \cong \mathbb{F}_{q^n}$ eine Nullstelle von f und $\mathbb{F}_{q^n} = \mathbb{F}_q(\alpha)$. Wegen $f(\sigma(\alpha)) = \sigma(f(\alpha)) = 0$ für jedes $\sigma \in \mathrm{Aut}_{\mathbb{F}_q}(\mathbb{F}_{q^n})$ sind insbesondere die $\varphi^i(\alpha)$ Nullstellen. Gilt schließlich $\varphi^k(\alpha) = \varphi^\ell(\alpha)$ mit $k < \ell$, so folgt $\varphi^m(\alpha) = \alpha$ mit $m = \ell - k$, also $\alpha \in \mathbb{F}_{q^m}$ und somit $\mathbb{F}_{q^n} \subseteq \mathbb{F}_{q^m}$, also $n \mid m$. Somit sind die $\alpha, \varphi(\alpha), \ldots, \varphi^{n-1}(\alpha)$ verschieden. □

Satz 1.15. *Es gilt* $\mathrm{Aut}_{\mathbb{F}_q}(\mathbb{F}_{q^n}) = \langle \mathrm{Frob}_q \rangle \cong \mathbb{Z}_n$.

Beweisskizze. Sei $\sigma \in \mathrm{Aut}_{\mathbb{F}_q}(\mathbb{F}_{q^n})$ und sei $\alpha \in \mathbb{F}_{q^n}^*$ ein Erzeuger, sowie f das Minimalpolynom von α, dann ist auch $\sigma(\alpha)$ eine Nullstelle von f, nach Proposition 1.14 also $\sigma(\alpha) = \varphi^i(\alpha)$ für ein $i \in \{0, \ldots, n-1\}$; da α ein Erzeuger ist, folgt $\sigma = \varphi^i$. □

Anzahl irreduzibler Polynome Bezeichne $I_q(d)$ die Menge der normierten irreduziblen Polynome $f \in \mathbb{F}_q[X]$ vom Grad d und sei $N_q(d) := |I_q(d)|$.

Lemma 1.16. *Sei* $f \in \mathbb{F}_q[X]$ *irreduzibel,* $\deg f = m$. *Dann:* $f \mid X^{q^n} - X \;\Leftrightarrow\; m \mid n$.

Satz 1.17. *Es gilt* $X^{q^n} - X = \prod_{d|n} \prod I_q(d)$, *und somit*

$$q^n = \sum_{d|n} d\, N_q(d)\,.$$

Eine explizitere Formel bekommen wir mit der *Möbiusfunktion*

$$\mu : \mathbb{N}_{>0} \to \{0, 1, -1\}\,, \quad n \mapsto \begin{cases} (-1)^r & \text{falls } n = p_1 \cdot \ldots \cdot p_r,\ p_i \text{ versch. Primz.}, \\ 0 & \text{sonst, d. h. } p^2 \mid n\,. \end{cases}$$

Lemma 1.18 (Möbiusinversion)**.** *Für* $n > 1$ *ist* $\sum_{d|n} \mu(d) = 0$. *Ist* $(G, +)$ *eine abelsche Gruppe und sind* $H, h : \mathbb{N}_{>0} \to G$, *dann:*

$$\forall n:\ H(n) = \sum_{d|n} h(d) \quad \textit{impliziert} \quad \forall n:\ h(n) = \sum_{d|n} \mu(d)\, H(\tfrac{n}{d})\,.$$

Beweis. Sei $n = p_1^{e_1} \cdot \ldots \cdot p_r^{e_r}$ wobei $r > 0$ und sei $D := \{p_2^{f_2} \cdot \ldots \cdot p_r^{f_r} \mid f_i \in \{0, 1\}\}$. Dann

$$\sum_{d|n} \mu(d) = \sum_{d \in D \cup p_1 D} \mu(d) = \sum_{d \in D} \big(\mu(d) + \mu(p_1 d)\big) = 0\,.$$

Nun ist $\sum_{d|n} \mu(d)\, H(\frac{n}{d}) = \sum_{d|n} \mu(d) \sum_{c|\frac{n}{d}} h(c) = \sum_{cd|n} \mu(d)\, h(c) = \sum_{c|n} h(c) \sum_{d|\frac{n}{c}} \mu(d)$ und dies ist $h(n)$, denn die innere Summe ist 0 außer für $c = n$. □

Corollar 1.19. *Es gilt* $N_q(n) = \frac{1}{n} \sum_{d|n} \mu(d)\, q^{n/d}$.

1.7 Faktorisierung von Polynomen über endlichen Körpern

Das Faktorisierungsproblem lautet, zu $f \in \mathbb{F}_q[X]$ normiert die Primfaktorzerlegung $f = p_1^{e_1} \cdot \ldots \cdot p_r^{e_r}$ mit p_i verschieden irreduzibel normiert und $e_i \in \mathbb{N}_{>0}$ zu finden. Dieses Problem ist algorithmisch „leicht“, das heißt polynomiell in der Eingabegröße lösbar. Wir präsentieren einen Algorithmus hierfür in drei Schritten.

I Zurückführung auf quadratfreien Fall Berechne $d := \mathrm{ggT}(f, f')$. Falls $d = 1$ ist, so ist f quadratfrei. Im Fall $d = f$ ist $f' = 0$ und somit $f = g^p$ für ein $g \in \mathbb{F}_q[X]$, fahre mit g fort. Ansonsten ist $0 < \deg d < \deg f$, fahre fort mit f/d (ist quadratfrei) und mit d.

II Faktorisierung nach verschiedenem Grad Erinnere: Nach Satz 1.17 ist $X^{q^i} - X$ das Produkt aller irreduziblen normierten Polynome vom Grad $d \mid i$.

input $f \in \mathbb{F}_q[X]$ quadratfrei
output $f = g_1 \cdot \ldots \cdot g_s$ mit g_i Produkt aller Primfaktoren von f vom Grad i

1. $f_0 := f,\ i := 0$
2. wiederhole:
   a. $i := i + 1$
   b. $g_i := \mathrm{ggT}(X^{q^i} - X, f_{i-1})$
   c. $f_i := f_{i-1}/g_i$
   bis $f_i = 1$

Berechnung der g_i: Bestimme $X^{q^i} \bmod f$ für $i = 1, \ldots, s \leq n = \deg f$ mit „square-and-multiply“, und berechne s mal den ggT. Aufwand: $O(s(\log q + 1)n^2)$ $\mathbb{F}_q$-Operationen.

III Faktoren vom gleichen Grad Gegeben sei $f \in \mathbb{F}_q[X]$ quadratfrei und $d \in \mathbb{N}_{>0}$, so dass alle Primfaktoren von f Grad d haben. Sei $f = f_1 \cdot \ldots \cdot f_r$ die gesuchte Primfaktorzerlegung, also ist $\deg f = n = r \cdot d$. Ohne Einschränkung sei $r > 1$.

Betrachte nach dem Chinesischen Restsatz den Isomorphismus

$$\chi\colon R = \mathbb{F}_q[X]/(f) \longrightarrow \mathbb{F}_q[X]/(f_1) \times \ldots \times \mathbb{F}_q[X]/(f_r) \cong \mathbb{F}_{q^d} \times \ldots \times \mathbb{F}_{q^d}\,.$$

Cantor-Zassenhaus-Algorithmus: Schreibe $\chi = (\chi_1, \ldots, \chi_r)$. Idee: Finde $a \in \mathbb{F}_q[X]$ mit $\varnothing \neq J := \{i \mid \chi_i([a]) = 0\} \neq \{1, \ldots, r\}$. Dann ist $\mathrm{ggT}(a, f) = \prod_{i \in J} f_i =: d$, also fahre mit d und f/d fort. Wir finden nun solches a auf *probabilistische* Weise.

Bemerkung 1.20. Sei q ungerade und sei $\alpha \in \mathbb{F}_q^*$ zufällig gewählt. Dann ist $a^{(q-1)/2} = \pm 1$ mit Wahrscheinlichkeit je $1/2$.

Ist q ungerade, wähle $g \in \mathbb{F}_q[X]$, $\deg g < n$ zufällig. Falls $\mathrm{ggT}(f, g) \neq 1$ ist Faktor gefunden, sonst ist $[g] \in R^*$ und somit $\alpha_i := \chi_i([g]) \in \mathbb{F}_{q^d}^*$ für alle i. Betrachte nun

$$a \;:=\; g^{(q^d-1)/2} - 1 \bmod f\,,$$

dann ist $\chi_i([a]) = \alpha_i^{(q^d-1)/2} - 1 = 0$ mit Wahrscheinlichkeit $1/2$, unabhängig über i. Somit ist dieses a mit Wahrscheinlichkeit $1 - (\frac{1}{2})^{r-1} \geq \frac{1}{2}$ wie gewünscht.

Bemerkung 1.21. Sei $q = 2^m$ gerade und $T_m := X^{2^{m-1}} + X^{2^{m-2}} + \cdots + X^2 + X \in \mathbb{F}_2[X]$ das *Spurpolynom*. Ist $\alpha \in \mathbb{F}_q$ zufällig gewählt, so ist $T_m(\alpha) \in \{0, 1\} = \mathbb{F}_2$ mit Wahrscheinlichkeit je $1/2$.

Sei nun $q = 2^k$ und wähle wieder $g \in \mathbb{F}_q[X]$, $\deg g < n$ zufällig. Betrachte

$$a \;:=\; T_{kd}(g) \bmod f\,.$$

Mit $\alpha_i := \chi_i([g]) \in \mathbb{F}_{q^d}$ ist dann $\chi_i([a]) = T_{kd}(\alpha_i) = 0$ mit Wahrscheinlichkeit $1/2$, unabhängig über i, und wir können wie oben argumentieren.

Erwarteter Aufwand: jeweils $O((d \log q + 1)n^2)$ $\mathbb{F}_q$-Operationen.

1.8 Schnelle Arithmetik

In modernen Computeralgebra-Systemen (wie Sage) hat z. B. die Multiplikation großer Zahlen eine subquadratische Laufzeit. Wir stellen einige Verfahren dieser Art vor.

Karazuba-Methode Sei R ein Ring und seien $f, g \in R[X]$ Polynome vom Grad $< n$. Die Berechnung des Produkts $f \cdot g = \sum_{i,j=0}^{n-1} f_i g_j X^{i+j}$ benötigt nach konventioneller Methode n^2 Multiplikationen und $(n-1)^2$ Additionen. Für das Produkt

$$(aX + b) \cdot (cX + d) \;=\; acX^2 + (ad + bc)X + bd$$

haben wir also 4 Multiplikationen und 1 Addition. Mittels der Beobachtung $ad + bc = (a + b)(c + d) - ac - bd$ kann man alternativ auch 3 Multiplikationen und 4 Additionen verwenden. Eine rekursive Anwendung dieses einfachen Tricks ist sehr nützlich und führt zu folgendem Algorithmus.

input $f, g \in R[X]$ mit $\deg f, g < n = 2^k$
output $f \cdot g \in R[X]$

1. falls $n = 1$ gib $f_0 g_0 \in R$ aus
2. sei $f = aX^m + b$, $g = cX^m + d$ mit $2m = n$ und $\deg a, b, c, d < m$
3. berechne ac, bd, $(a + b)(c + d)$ durch rekursiven Aufruf
4. gib $acX^{2m} + ((a + b)(c + d) - ac - bd)X^m + bd$ aus

Bezeichne $T(n)$ die Laufzeit in R-Operationen für die Berechnung von $f \cdot g$ bei $\deg < n$, so haben wir $T(1) = 1$ und $T(n) \leq 3T(\frac{n}{2}) + 4n$. Die Auflösung dieser Rekursion ergibt:

Proposition 1.22. *Für $n = 2^k$ benötigt die Karazuba-Methode höchstens $9 \cdot 3^k = 9 \cdot n^{\log_2 3}$ R-Operationen, wobei $\log_2 3 \approx 1.58$.*

Der Karazuba-Algorithmus kann auch für Zahlen angewandt werden. Sei $f = aB^m + b$ und $g = cB^m + d$ mit $a, b, c, d < B^m$, so ist $f \cdot g = acB^{2m} + (ad + bc)B^m + bd$. Dies führt zur gleichen Komplexität $O(n^{\log_2 3})$ (mit $f, g < B^n$) in Prozessor-Operationen.

FFT-basierte Verfahren Noch schnellere Verfahren basieren auf der diskreten Fourier-Transformation. Sei zunächst K ein Körper und $u_1, \ldots, u_n \in K$ paarweise verschieden. Betrachte die *Auswertung*

$$\mathrm{ev} := (\mathrm{ev}_{u_1} \ldots \mathrm{ev}_{u_n}) \colon K[X] \to K^n\,, \quad f \mapsto (f(u_1), \ldots, f(u_n))\,.$$

Eingeschränkt auf $P_n := K[X]^{<n} := \{f \in K[X] \mid \deg f < n\}$ ist $\mathrm{ev} \colon P_n \to K^n$ dann ein K-Vektorraum-Isomorphismus. Die Matrix von ev ist die *Vandermonde-Matrix*

$$V \;:=\; \begin{pmatrix} 1 & u_1 & & u_1^{n-1} \\ \vdots & \vdots & \cdots & \vdots \\ 1 & u_n & & u_n^{n-1} \end{pmatrix}$$

mit $\det V = \prod_{j<i}(u_i - u_j)$.

Die Umkehrabbildung $\mathrm{ev}^{-1} \colon K^n \to P_n$ entspricht der *Interpolation*, das heißt, zu gegebenen $v_1, \ldots, v_n \in K$ ein $f \in P_n$ zu finden mit $f(u_i) = v_i$ für alle i. Sei

$$\ell_i \;:=\; \prod_{j=1,\, j\neq i}^{n} \frac{X - u_j}{u_i - u_j} \;\in\; P_n\,,$$

dann gilt $\ell_i(u_j) = \delta_{ij}$; es ist $(\ell_i)_{1\leq i\leq n}$ die *Lagrange*-Interpolationsbasis und $f := \sum_{i=1}^n v_i \ell_i$ löst das Interpolationsproblem. Der Aufwand für eine Berechnung von ev oder ev^{-1} ist $O(n^2)$ Operationen in K.

Sei nun R ein Integritätsring, sei ω eine *primitive n-te Einheitswurzel*, das heißt $\omega \in R^*$, $\operatorname{ord} \omega = n$, und sei $n1_R \in R^*$.

Bemerkung 1.23. Es hat $\mathbb{F}_q$ genau dann eine primitive n-te Einheitswurzel, wenn $n \mid q-1$.

Definition 1.24. Die *diskrete Fourier-Transformation* ist der Ringmorphismus

$$\mathrm{DFT}_\omega := (\mathrm{ev}_1 \,...\, \mathrm{ev}_{\omega^{n-1}})\colon R[X]/(X^n-1) \to R^n\,, \quad f \mapsto (f(1), f(\omega), \ldots, f(\omega^{n-1}))\,.$$

Die Matrix von DFT_ω (als R-lineare Abbildung) ist

$$V_\omega \;:=\; (\omega^{ij})_{0\le i,j<n} = \begin{pmatrix} 1 & 1 & & 1 \\ 1 & \omega & \cdots & \omega^{n-1} \\ \vdots & \vdots & & \vdots \\ 1 & \omega^{n-1} & \cdots & \omega^{(n-1)^2} \end{pmatrix}.$$

Proposition 1.25. *Es gilt $V_\omega V_{\omega^{-1}} = nI$. Somit ist DFT_ω ein Ringisomorphismus und die Matrix von $(\mathrm{DFT}_\omega)^{-1}$ ist $\frac{1}{n}V_{\omega^{-1}}$.*

Die Multiplikation in R^n wird punktweise gebildet und kann somit in linearer Zeit ausgeführt werden. Daher eignet sich die Fourier-Transformierte für eine schnelle Polynom-Multiplikation, sofern man DFT_ω (und damit $(\mathrm{DFT}_\omega)^{-1}$) schnell berechnen kann.

Schnelle Fourier-Transformation: Sei $n = 2m$ und sei $f \in P_n$. Wir schreiben $f = aX^m + b$ mit $a, b \in P_m$. Dann ist

$$f \bmod X^m-1 = a+b =: r_0 \quad \text{und} \quad f \bmod X^m+1 = -a+b =: r_1\,,$$

somit $f(\omega^{2\ell}) = r_0(\omega^{2\ell})$ und $f(\omega^{2\ell+1}) = r_1(\omega^{2\ell+1}) = r_1^*(\omega^{2\ell})$ für $r_1^* = r_1(\omega X)$.

Dies führt zum rekursiven Algorithmus „FFT“:

input $f = \sum_{i=0}^{n-1} f_i X^i \in P_n$, wobei $n = 2^k$
output $\mathrm{DFT}_\omega\, f$

1. falls $n = 1$: gib f_0 aus
2. sei $r_0 := \sum_{0\le j<m}(f_j + f_{m+j})X^j$, $r_1^* := \sum_{0\le j<m}(f_j - f_{m+j})\omega^j X^j$
3. berechne $\mathrm{DFT}_{\omega^2}\, r_0$ und $\mathrm{DFT}_{\omega^2}\, r_1^*$ (ω^2 ist prim. m-te Einheitsw.)
4. gib $\big(r_0(1), r_1^*(1), \ldots, r_0(\omega^{n-2}), r_1^*(\omega^{n-2})\big)$ aus

Bezeichne $T(n)$ die Laufzeit in R-Operationen für $f \in P_n$, so erhalten wir $T(1) = 0$ und $T(n) \le 2T(\frac{n}{2}) + \frac{3}{2}n$. Dies führt zur Abschätzung $T(n) \le \frac{3}{2}n\log_2 n$. Schnelle Fourier-Transformation und somit Polynom-Multiplikation über R können also in $O(n\log n)$ Operationen in R berechnet werden.

Division mit Rest mit Newton-Iteration Für die Arithmetik in einem endlichen Körper $\mathbb{F}_{p^n} = \mathbb{F}_p[X]/(h)$ (mit h irreduzibel und $\deg h = n$) ist neben der Multiplikation in $\mathbb{F}_p[X]$ auch die Division mit Rest wichtig. Hierfür stellen wir ein schnelles Verfahren vor.

Sei K ein Körper und seien $f, g \in K[X]$ mit $\deg f = n$, $\deg g = m \le n$, g normiert gegeben. Gesucht sind $q, r \in K[X]$, $\deg r < m$ mit $f = qg + r$. Aus der Gleichung folgt

$$X^n f(\tfrac{1}{X}) \;\equiv\; X^{n-m} q(\tfrac{1}{X})\, X^m g(\tfrac{1}{X}) \mod X^{n-m+1}\,.$$

Sei zu $a = \sum_{i=0}^{\ell} a_i X^i \in K[X]$ mit $\deg a = \ell$ das „umgedrehte“ Polynom definiert durch $\mathrm{rev}\, a := X^\ell a(\frac{1}{X}) = a_0X^\ell + \cdots + a_{\ell-1}X + a_\ell$, dann erhalten wir

$$\mathrm{rev}\, f \;\equiv\; \mathrm{rev}\, q\ \mathrm{rev}\, g \mod X^{n-m+1}\,.$$

So ist q durch die Gleichung $\operatorname{rev} q \equiv \operatorname{rev} f / \operatorname{rev} g$ bestimmt, und dies liefert $r = f - qg$.

Wir bestimmen nun für $a \in K[X]$ mit $a(0) = 1$ das Inverse $\frac{1}{a} \bmod X^\ell$ via Newton-Iteration (Idee: mit $f(x) := \frac{1}{x} - a$ erhalten wir $x_{i+1} := x_i - \frac{f(x_i)}{f'(x_i)} = 2x_i - ax_i^2$.)

input $a \in K[X]$ mit $a(0) = 1$, sowie $\ell \geq 1$
output $b \in K[X]$ mit $ab \equiv 1 \bmod X^\ell$

1. $b_0 := 1$
2. for $i := 0$ to $\lceil \log_2 \ell \rceil - 1$ do
   a. $b_{i+1} := 2b_i - ab_i^2 \bmod X^{2^{i+1}}$
3. gib $b_{\lceil \log_2 \ell \rceil}$ aus

Proposition 1.26. *Dieser Algorithmus arbeitet korrekt und benötigt für $\ell = 2^k$ eine Laufzeit von $3M(\ell) + \ell$ Operationen in K, wobei $M(n)$ die Laufzeit einer Polynom-Multiplikation mit* $\deg < n$ *bezeichne.*

Beweisskizze. Die Korrektheit folgt leicht aus $1 - ab_{i+1} = (1 - ab_i)^2$ und Induktion über i. Schritt *2a.* benötigt $M(2^i) + M(2^{i+1}) + 2^i \leq \frac{3}{2} M(2^{i+1}) + 2^i$ Operationen, insgesamt also höchstens $\sum_{i=0}^{k-1} \frac{3}{2} M(2^{i+1}) + 2^i \leq 3M(\ell) + \ell$ Operationen. □

2 Codierungstheorie

Wir betrachten in diesem Kapitel fehler-korrigierende Codes, zunächst für die klassische Situation der Kanalcodierung, und später für die neuere Netzwerkcodierung.

2.1 Codes

Sei A eine endliche Menge mit q Elementen (Alphabet).

Definition 2.1. Seien M und I endliche Mengen. Eine injektive Abbildung $\varphi\colon M \to A^I$ heißt *Kodierer* und die Bildmenge $C := \operatorname{im}(\varphi) \subseteq A^I$ heißt *Code*.

Es sei $n := |I|$ die *Codelänge* von C (oft $I = \{1, \dots, n\}$ und $A^I = A^n$).

Bemerkung 2.2. Der *Hamming-Abstand* d_H ist eine Metrik, wobei

$$d_H\colon A^I \times A^I \to \mathbb{N}\,, \quad (x, y) \mapsto |\{i \in I \mid x_i \neq y_i\}|\,.$$

Definition 2.3. Zu einem Code $C \subseteq A^I$ sei der *Minimalabstand* definiert als

$$d(C) := \min_{x \neq y \in C} d_H(x, y)\,.$$

Bemerkung 2.4. Falls $d(C) > 2t$, so kann man bis zu t Übertragungsfehler korrigieren. (Da $B_t(x) \cap B_t(y) = \varnothing$ für $x, y \in C$, $x \neq y$, wobei $B_t(x) := \{y \in A^I \mid d_H(x, y) \leq t\}$.)

Satz 2.5 (Kugelpackungsschranke)**.** *Sei $C \subseteq A^I$ ein Code mit $d(C) > 2t$, dann gilt*

$$\bigsqcup_{x \in C} B_t(x) \subseteq A^I \quad \textit{und somit} \quad |C| \cdot \sum_{i=0}^{t} \tbinom{n}{i}(q-1)^i \leq q^n\,.$$

Satz 2.6 (Singleton-Schranke)**.** *Sei $C \subseteq A^I$ ein Code und $d := d(C)$. Dann gilt*

$$|C| \leq q^{n-d+1}\,.$$

Für $k := \log_q |C|$ folgt also $k + d \leq n + 1$.

Beweisidee. Sei $J \subseteq I$ mit $|J| = n - d + 1$, dann ist $\pi\colon C \to A^J$, $x \mapsto x|_J$ injektiv. □

Falls in Satz 2.5 Gleichheit gilt, so heißt der Code C *perfekt*, und bei Gleichheit in Satz 2.6 sagt man *MDS*-Code (maximum distance separable).

Beispiel 2.7. Sei $A = \mathbb{F}_q$, seien $u_1, \dots, u_n \in \mathbb{F}_q$ verschieden und $k \leq n$. Betrachte die Auswertung

$$\operatorname{ev}|_{P_k}\colon P_k \to A^n\,, \quad f \mapsto (f(u_1), \dots, f(u_n))\,,$$

wobei $P_k = \{f \in \mathbb{F}_q[X] \mid \deg f < k\}$, dann ist $C := \operatorname{im}(\operatorname{ev}_{P_k}) \subseteq A^n$ ein MDS-Code.

Additive Codes Sei nun $(A, +)$ eine endliche abelsche Gruppe.

Definition 2.8. Ist $C \leq A^I$ eine Untergruppe, so heißt C *additiver Code*.

Einen Monomorphismus $\varphi\colon M \to A^I$ mit $\operatorname{im}(\varphi) = C$ nennen wir *Kodierer*, einen Epimorphismus $\psi\colon A^I \to S$ mit $\ker(\psi) = C$ *Syndromabbildung* (hierbei seien $(M, +)$ und $(S, +)$ abelsche Gruppen). Als exakte Sequenz (d. h. $\ker(\psi) = \operatorname{im}(\varphi)$) geschrieben:

$$0 \longrightarrow M \xrightarrow{\varphi} A^I \xrightarrow{\psi} S \longrightarrow 0$$

Bemerkung 2.9. Sei $w_H \colon A^I \to \mathbb{N}$, $x \mapsto |\operatorname{supp} x|$ das *Hamming-Gewicht*, dann gilt

$$d(C) = \min_{c \in C \setminus \{0\}} w_H(c) .$$

Lemma 2.10. *Sei $C \leq A^I$ ein additiver Code mit Syndromabbildung $\psi \colon A^I \to S$. Dann sind äquivalent:*

a) $\psi|_{B_t(0)}$ ist injektiv,

b) C kann bis zu t Fehler korrigieren, das heißt $B_t(c) \cap B_t(c') = \varnothing$ für $c, c' \in C$, $c \neq c'$.

„Syndromdekodierung“: Wird $c \in C$ gesendet und $y = x + e$ empfangen, wobei $e \in B_t(0)$, dann berechne zunächst das *Syndrom* $s := \psi(y) = \psi(e)$ und finde dann $e = \psi^{-1}(s)$, somit kann $c = y - e$ rekonstruiert werden.

Lineare Codes Nun sei $A = \mathbb{F}_q$ ein endlicher Körper.

Definition 2.11. Ein Untervektorraum $C \leq \mathbb{F}_q^I$ heißt *linearer Code*. Mit $k := \dim_{\mathbb{F}_q} C$ und $d := d(C)$ sagt man auch $[n, k]$ Code bzw. $[n, k, d]$ Code.

Ein Kodierer φ und eine Syndromabbildung ψ für C seien analog wie oben, nun als lineare Abbildungen von $\mathbb{F}_q$-Vektorräumen, definiert. Es gilt $M \cong \mathbb{F}_q^K$ und $S \cong \mathbb{F}_q^J$ für endliche Mengen K, J. Somit können diese Abbildungen als lineare Abbildungen

$$\varphi \colon \mathbb{F}_q^K \to \mathbb{F}_q^I , \quad m \mapsto mG \quad \text{und} \quad \psi \colon \mathbb{F}_q^I \to \mathbb{F}_q^J , \quad x \mapsto Hx^t$$

aufgefasst werden (Konvention: Vektoren sind Zeilenvektoren). Die $k \times n$-Matrix G heißt *Erzeugermatrix* und die $(n-k) \times n$-Matrix H *Kontrollmatrix*.

Lemma 2.12. *Sei $0 \neq C \leq \mathbb{F}_q^I$ ein linearer Code. Dann ist $d(C)$ gleich dem minimalen d, so dass d Spalten der Kontrollmatrix H linear abhängig sind.*

Hamming-Codes $C = \operatorname{Ham}_q(r)$. Setze

$$I := \mathbf{P}_{r-1}(q) := \{\ell \leq \mathbb{F}_q^r \mid \dim_{\mathbb{F}_q} \ell = 1\} ,$$

somit $n = |I| = \frac{q^r - 1}{q - 1}$, und sei $R \colon I \to \mathbb{F}_q^r \setminus \{0\}$ so gewählt, dass $R(\ell) \in \ell \setminus \{0\}$ für $\ell \in I$. Bilde mit den $R(\ell)$ als Spalten eine $r \times n$-Kontrollmatrix $H = (R(\ell)_j)_{j,\ell}$ und sei $C := \{x \in \mathbb{F}_q^I \mid Hx^t = 0\}$ (hängt nicht von der Wahl R ab). Dann gilt $d(C) = 3$ und C ist ein perfekter $[n, n - r, 3]$ Code.

Beispiel 2.13. $\operatorname{Ham}_3(2)$ ist ein $[4, 2, 3]$ Code mit Kontrollmatrix

$$H = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 2 \end{pmatrix} ;$$

dieser Code ist perfekt *und* MDS.

2.2 Zyklische Codes

Definition 2.14. Ein linearer Code $C \leq \mathbb{F}_q^n \cong \mathbb{F}_q^{\mathbb{Z}_n}$ heißt *zyklisch*, falls

$$c = (c_0, c_1, \ldots, c_{n-1}) \in C \quad \text{impliziert} \quad (c_{n-1}, c_0, \ldots, c_{n-2}) \in C\,.$$

Es sei R_n der Ring $\mathbb{F}_q[X]/\langle X^n - 1\rangle$. Wir identifizieren $(c_0, c_1, \ldots, c_{n-1}) \in \mathbb{F}_q^n$ mit dem Polynom $\sum_{i=0}^{n-1} c_i X^i$, bzw. mit dem Ringelement $[\sum_{i=0}^{n-1} c_i X^i] \in R_n$.

Bemerkung 2.15. Sei $C \leq \mathbb{F}_q^n \cong R_n$ ein linearer Code. Genau dann ist C zyklisch, also $[X] \cdot C \subseteq C$, wenn $C \leq R_n$ ein Ideal ist. Jedes Ideal in R_n ist ein Hauptideal.

Zu jedem zyklischen Code $C \leq R_n$ existiert eindeutig ein normierter Teiler $g \mid X^n - 1$ mit $C = (g) = gR_n$, das *Erzeugerpolynom*, und es sei $h := \frac{X^n-1}{g}$ das *Kontrollpolynom*. Ist $\deg g = r$ und $k := n - r$, dann ist $k = \deg h$ und C ist ein $[n, k]$ Code.

In diesem Fall ist $C = \mathrm{im}(\varphi) = \ker(\psi)$ für die linearen Abbildungen

$$\varphi\colon P_k \to P_n\,, \quad m \mapsto m \cdot g \quad \text{und} \quad \psi\colon P_n \to (\mathbb{F}_q)^r\,, \quad c \mapsto (c \cdot h)_{k+1,\ldots,n}\,.$$

Die zugehörige Erzeuger- bzw. Kontrollmatrix ist dann

$$G = \begin{pmatrix} g_0 & \ldots & g_r & 0 \\ & \ddots & & \ddots \\ 0 & g_0 & \ldots & g_r \end{pmatrix} \quad \text{und} \quad H = \begin{pmatrix} h_k & \ldots & h_0 & 0 \\ & \ddots & & \ddots \\ 0 & h_k & \ldots & h_0 \end{pmatrix}.$$

Faktorisierung von $X^n - 1$. Es sei $\mathrm{ggT}(n, q) = 1$ angenommen. Dann ist $X^n - 1$ quadratfrei und hat somit verschiedene Primfaktoren $f = f_1 \cdot \ldots \cdot f_r$, $\deg f_i = d_i$. Mit dem Chinesischen Restsatz haben wir dann einen Ringisomorphismus

$$R_n \;\to\; \mathbb{F}_{q^{d_1}} \times \cdots \times \mathbb{F}_{q^{d_r}}\,.$$

Sei ω eine primitive n-te Einheitswurzel in einem Erweiterungskörper $\mathbb{F}_{q^r}$. Ist $f \in \mathbb{F}_q[X]$ ein Teiler von $X^n - 1$, so sind alle Nullstellen n-te Einheitswurzeln, und es gilt $f(\omega^i) = 0$ impliziert $f(\omega^{qi}) = 0$. Die Nullstellen $\{\omega^{i_1}, \ldots, \omega^{i_r}\}$ formen somit eine „zyklotomische Menge“ $I = \{i_1, \ldots, i_r\}$ (das heißt $q \cdot I \subseteq I$), und $\deg f = |I|$.

Codes via Nullstellen. Seien $\alpha_1, \ldots, \alpha_s \in \mathbb{F}_{q^r}$ n-te Einheitswurzeln, dann ist

$$\overline{\mathrm{ev}}_{\alpha_1,\ldots,\alpha_s}\colon R_n \to (\mathbb{F}_{q^r})^s\,, \quad f \mapsto \big(f(\alpha_1), \ldots, f(\alpha_s)\big)$$

wohldefiniert und ein Ringmorphismus, somit ist $C = \ker \overline{\mathrm{ev}}_{\alpha_1,\ldots,\alpha_s} = \{f \in R_n \mid f(\alpha_1) = \ldots = f(\alpha_s) = 0\}$ ein Ideal, also ein zyklischer Code.

Satz 2.16. *Sei* $n = \frac{q^r-1}{q-1}$ *und* $\mathrm{ggT}(r, q-1) = 1$, *sowie* ω *primitive* n*-te Einheitswurzel. Dann ist* $C = \ker \overline{ev}_\omega$ *(bis auf Indizes) der* $[n, n-r, 3]$ *Hamming-Code* $\mathrm{Ham}_q(r)$.

Insbesondere sind alle binären Hamming-Codes zyklisch.

2.3 BCH-Codes und Decodierung

BCH-Codes (Bose und Ray-Chaudhuri ’60 bzw. Hocquenghem ’59) sind spezielle zyklische Codes. Sei $R_n := \mathbb{F}_q[X]/(X^n - 1)$ und stets $\mathrm{ggT}(q, n) = 1$, sowie $\omega \in \mathbb{F}_{q^r}$ eine primitive n-te Einheitswurzel.

Definition 2.17. Ein *BCH-Code* der Länge n und geplantem Abstand δ ist

$$C \;:=\; \ker \overline{\mathrm{ev}}_{\omega^1,\ldots,\omega^{\delta-1}} \;=\; \{f \in R_n \mid f(\omega^1) = \ldots = f(\omega^{\delta-1}) = 0\}\,.$$

Satz 2.18 (BCH-Schranke)**.** *Mit C wie oben ist $d(C) \geq \delta$.*

Beweis. Betrachte die diskrete Fouriertransformation

$$\mathrm{DFT}_{\omega^{-1}} \colon R_n \to (\mathbb{F}_{q^r})^n\,, \quad f \mapsto \big(f(1), f(\omega^{-1}), \ldots, f(\omega^{-(n-1)})\big)\,.$$

Ist $f \in C$, so ist das Bild $\mathrm{DFT}(f)$, interpretiert als Polynom $g = \sum_{i=0}^{n-1} f(\omega^{-i})X^i$, vom Grad höchstens $n-\delta$, hat also höchstens $n-\delta$ Nullstellen falls $f \neq 0$. Mit der inversen Fouriertransformation ist jedoch $f = \frac{1}{n}\sum_{i=0}^{n-1} g(\omega^i)X^i$, und somit $w_H(f) \geq \delta$. □

Beispiel 2.19. Was ist die Dimension der binären ($q=2$) BCH-Codes der Länge $n = 15$? Z. B. für $\delta = 5$ ist $C = \{f \in R_n \mid f(\omega) = f(\omega^3) = 0\} = (g_1 g_3)$, wobei $g_1, g_3 \in \mathbb{F}_2[X]$ die Minimalpolynome von ω bzw. ω^3 seien.

Sind $I_1 := \{1,2,4,8\}$ und $I_3 := \{3,6,12,9\}$ die von 1 bzw. 3 erzeugten zyklotomischen Mengen in $\mathbb{Z}_n$, so gilt $g_j = \prod_{i \in I_j}(X - \omega^i)$ für $j = 1, 3$. Also ist $\dim C = n - \deg(g_1 g_3) = 15 - 8 = 7$; tatsächlich ist $d(C) = 5$ und wir haben einen $[15, 7, 5]$-Code.

Algebraische Decodierung Sei C ein BCH-Code mit $\delta = 2t+1$. Es sei $c \in C$ gesendet und $y = c + e$ empfangen, wobei $w_H(e) \leq t$. Wir setzen

$$\begin{aligned} M &:= \{i \mid e_i \neq 0\} \qquad \text{(Fehlerpositionen)},\ |M| \leq t \\ u &:= \prod_{i \in M} (1 - \omega^i Z) \in \mathbb{F}_q[Z] \quad \text{(Fehlerortungspolynom)} \\ v &:= \sum_{i \in M} e_i \omega^i Z \prod_{j \in M \setminus \{i\}} (1 - \omega^j Z) \in \mathbb{F}_q[Z]\,. \end{aligned}$$

Ist u, v gefunden, dann kann der Fehler e rekonstruiert werden. Denn es ist $i \in M$ genau dann, wenn $u(\omega^{-i}) = 0$; und, falls $q > 2$, so ist $-e_i = \frac{v(\omega^{-i})\omega^i}{u'(\omega^{-i})}$.

Um u und v zu berechnen, betrachten wir

$$w = \frac{v}{u} = \sum_{i \in M} e_i \tfrac{\omega^i Z}{1-\omega^i Z} = \sum_{i \in M} e_i \sum_{k=1}^{\infty} (\omega^i Z)^k = \sum_{k=1}^{\infty} \Big(\sum_{i \in M} e_i \omega^{ki} \Big) Z^k\,,$$

wobei $s_k := \sum_{i \in M} e_i \omega^{ki} = e(\omega^k) = y(\omega^k)$ für $1 \leq k \leq 2t$ bekannt ist.

Wir kennen also $w = \frac{u}{v}$ modulo Z^{2t+1}, wobei $\deg u, \deg v \leq t$. Diese Gleichung ist äquivalent zu $u \cdot w = v \bmod Z^{2t+1}$, wobei der k-te Koeffizient $(u \cdot w)_k = 0$ für $t+1 \leq k \leq 2t$. Mit $u = \sum u_i Z^i$ und $u_0 = 1$ ist dies ein lineares Gleichungssystem in $u_1, \ldots, u_t$.

Alternativ kann man die Zwischenschritte des erweiterten euklidischen Algorithmus mit $f = Z^{2t+1}$ und $g = \sum_{i=1}^{2t} s_i Z^i$ benutzen, nämlich $af + bg = r$ mit $\deg b, \deg r \leq t$.

2.4 Netzwerkcodierung

Wir betrachten nun Informationsübertragung in einem gerichteten Graphen.

Fluss in einem Netzwerk Sei V eine endliche Menge von Knoten und $s, t \in V$ mit $s \neq t$, wobei s „Quelle“, t „Senke“, sowie $r \colon V \times V \to \mathbb{R}_{\geq 0}$ eine „Kapazität“ der Kanten.

Definition 2.20. Ein *Fluss* von s nach t ist eine Abbildung $f \colon V \times V \to \mathbb{R}_{\geq 0}$ derart, dass

$$\sum_{i \in V} f(i,v) \;=\; \sum_{j \in V} f(v,j) \quad \text{für alle } v \in V \setminus \{s,t\}\,,$$

sowie $f(i,j) \leq r(i,j)$ für alle $i, j \in V$ gilt. Der „Wert“ von f ist $|f| := \sum_{j \in V} f(s,j) - \sum_{i \in V} f(i,s) = \sum_{i \in V} f(i,t) - \sum_{j \in V} f(t,j)$.

Definition 2.21. Ein *s-t-Schnitt* $\mathcal{C} = (S, T)$ ist eine Partition $V = S \dot\cup T$ mit $s \in S$ und $t \in T$. Der „Wert“ von $\mathcal{C}$ ist $|\mathcal{C}| := \sum_{i \in S, j \in T} r(i, j)$.

Satz 2.22 (Max-flow-Min-cut, 1956)**.** *Der Wert des maximalen Flusses von s nach t ist gleich dem Wert des minimalen s-t-Schnitts, also* $\max_{\text{Fluss } f} |f| = \min_{\text{Schnitt } \mathcal{C}} |\mathcal{C}|$.

Beweis. Für jeden Fluss f und jeden Schnitt $\mathcal{C} = (S, T)$ ist stets

$$|f| = \sum_{i \in S,\, j \in T} f(i, j) - \sum_{i \in T,\, j \in S} f(i, j) \leq \sum_{i \in S,\, j \in T} r(i, j) = |\mathcal{C}| .$$

Umgekehrt findet man einen Fluss f und einen Schnitt $\mathcal{C}$ mit $|f| = |\mathcal{C}|$ durch den Algorithmus von *Ford-Fulkerson*. Die Idee ist, einen gegebenen Fluss f, falls möglich, durch einen Weg zu augmentieren. Betrachte dazu den „Residualgraphen“ $G_f = (V, E_f)$ mit Kantenmenge $E_f := \{(i, j) \in V \times V \mid f(i, j) < r(i, j) \vee f(j, i) > 0\}$.

Falls ein Weg von s nach t in G_f existiert, kann der Wert des Flusses f erhöht werden. Ansonsten definiert die Menge S aller Knoten, die von s in G_f erreichbar sind, einen Schnitt $\mathcal{C} = (S, T)$. Dann gilt $f(i, j) = r(i, j)$ und $f(j, i) = 0$ für alle $i \in S$, $j \in T$, und somit $|f| = \sum_{i \in S,\, j \in T} f(i, j) - \sum_{i \in T,\, j \in S} f(i, j) = \sum_{i \in S,\, j \in T} r(i, j) = |\mathcal{C}|$.

Man kann ferner zeigen, dass, wenn man stets einen kürzesten Weg in G_f wählt, der Fluss nur endlich oft augmentiert wird. □

Lineare Netzwerkcodierung Wir betrachten nun ein Netzwerk mit ganzzahligen Kapazitäten $r \colon V \times V \to \mathbb{N}$. Dies interpretieren wir als einen *Multigraphen* $G = (V, E, \rho)$, das heißt V und E sind Mengen und $\rho = (\sigma, \tau) \colon E \to V \times V$ ist eine Abbildung, wobei $r(v, w) = |\rho^{-1}(v, w)|$ (Anzahl Kanten von v nach w) für alle $v, w \in V$.

Bemerkung 2.23. Seien $s, t \in V$ Knoten mit $\tau^{-1}(s) = \varnothing$ (Quelle, keine eingehenden Kanten) und $\sigma^{-1}(t) = \varnothing$ (Senke, keine ausgehenden Kanten). Ist $f \colon V \times V \to \mathbb{N}$ ein Fluss von s nach t in G mit Wert $R = |f|$, dann gibt es genau R kantendisjunkte Wege von s nach t. Der Wert eines Schnitts $\mathcal{C} = (S, T)$ ist hier $|\rho^{-1}(S, T)|$, und das Max-flow-Min-cut-Theorem entspricht Mengers Satz: die maximale Anzahl disjunkter s-t-Wege ist gleich dem minimalen Wert eines s-t-Schnitts.

Man kann in einem solchen Netzwerk dann R Symbole simultan übertragen, durch Weiterleitung entlang der kantendisjunkten Wege (wenn jede Kante Einheitskapzität hat). Bei einer Quelle s und mehreren Senken $t_1, \ldots, t_L$ ist jedoch die Methode des direkten Weiterleitens (Routing) suboptimal, denn man kann den „Informationsdurchfluss“ erhöhen und zwar durch Netzwerkcodierung.

Was ist lineare Netzwerkcodierung? Sei $G = (V, E, \rho)$ ein azyklischer Multigraph, $s \in V$ die Quelle, und wir betrachten zunächst eine Senke $t \in V$. Es sei $\text{in}(v) := \tau^{-1}(v) = \{e \in E \mid \tau(e) = v\}$ und $\text{out}(v) := \sigma^{-1}(v) = \{e \in E \mid \sigma(e) = v\}$, für $v \in V$. Sei $R \in \mathbb{N}$ eine Rate und $F := \mathbb{F}_q$ ein endlicher Körper.

Wir möchten einen Vektor $x \in F^R$ von s nach t in G übertragen. Jede Kante $e \in E$ (mit $\sigma(e) = v$) sendet ein Symbol $y_e \in F$, wobei

$$y_e = \sum_{e' \in \text{in}(v)} \beta_{e',e} y_{e'} \quad (+ \sum_{i=1}^{R} \alpha_{i,e} x_i \quad \text{falls } v = s) ;$$

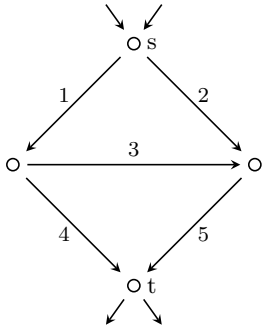
ferner sei $z_i = \sum_{e' \in \text{in}(t)} \delta_{e',i} y_{e'}$ die Ausgabe — hierbei sind die $\beta_{e',e}$, $\alpha_{i,e}$ und $\delta_{e',i}$ Parameter in F. Wir erhalten somit eine lineare *Transferabbildung* $F^R \to F^R$, $x \mapsto z$, und versuchen, die Parameter so zu wählen, dass diese Abbildung die Identität ist.

Bemerkung 2.24. Die *Transfermatrix* M der Transferabbildung $x \mapsto z = xM$ hängt polynomiell von den Parametern $\beta_{..}$, $\alpha_{..}$ und $\delta_{..}$ ab.

Genauer gilt $M = ACD$ mit $C = I + B + B^2 + \dots$ (Transportmatrix), wobei $B = (\beta_{e',e}) \in F^{E\times E}$ nilpotent ist, sowie $A = (\alpha_{i,e}) \in F^{[R]\times E}$ und $D = (\delta_{e',i}) \in F^{E\times[R]}$.

Der Träger von B ist enthalten in $E^{\langle 2\rangle} := \{(e', e) \mid \tau(e') = \sigma(e)\}$ (hierbei ist $LG = (E, E^{\langle 2\rangle})$ der *Kantengraph* von G), der Träger von A ist in $[R] \times \mathrm{out}(s)$ und der von D ist in $\mathrm{in}(t) \times [R]$.

Beispiel 2.25. Sei $R = 2$ und G gegeben durch



Sind $a, b, c, d \in F$ die Parameter $\beta_{..}$, so ist

$$B = \begin{pmatrix} \cdot & \cdot & a & b & \cdot \\ \cdot & \cdot & \cdot & \cdot & c \\ \cdot & \cdot & \cdot & \cdot & d \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix} \quad \text{und} \quad C = \begin{pmatrix} 1 & \cdot & a & b & ad \\ \cdot & 1 & \cdot & \cdot & c \\ \cdot & \cdot & 1 & \cdot & d \\ \cdot & \cdot & \cdot & 1 & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 \end{pmatrix},$$

somit ist $M = ACD = I_2$ genau dann möglich (also das Netzwerkproblem lösbar), wenn

$$\det \begin{pmatrix} b & ad \\ 0 & c \end{pmatrix} \neq 0\,,$$

das heißt $b \neq 0$ und $c \neq 0$, also $\beta_{1,4} \neq 0$ und $\beta_{2,5} \neq 0$.

Bemerkung 2.26. Betrachte $\det M$ als Polynom in $\beta_{..}$, $\alpha_{..}$, $\delta_{..}$. Ist das Netzwerkproblem lösbar, so gilt notwendig $\det M \neq 0$. Ist umgekehrt $\det M$ nicht das Nullpolynom und ist $|F|$ genügend groß, so gibt es eine Lösung des Netzwerkproblems.

Ist (S, T) ein Schnitt, so sind die Ausgabesymbole z_i Linearkombinationen von Symbolen y_e der Schnittkanten $e \in \rho^{-1}(S, T)$. Ist das Netzwerkproblem lösbar, wird also $x \in F^R$ rekonstruiert, so muss für die Rate $R \leq |\rho^{-1}(S, T)|$ gelten (wie bei einem Fluss).

Broadcast-Szenario. Betrachte nun verschiedene Senken $t_1, \dots, t_L \in V$. Unser Ziel ist es, die Information $x \in F^R$ an s zu allen t_ℓ zu übertragen.

Satz 2.27 (Ahlswede, Cai, Li, Yeung 2000)**.** *Das Broadcast-Netzwerkproblem in G mit Rate R ist genau dann lösbar, falls $R \leq \min_\ell \min_{s\text{-}t_\ell\text{-Schnitt } C} |C|$.*

Beweisskizze (Koetter, Médard 2003). Ist das Netzwerkproblem mit Rate R lösbar, so gilt $R \leq |\rho^{-1}(S, T)|$ für jeden s-t_ℓ-Schnitt (S, T), also ist die Bedingung notwendig.

Für jedes ℓ betrachte nun die Transfermatrix $M_\ell = ACD_\ell$ über den multivariaten Polynomring $F[\alpha_{..}, \beta_{..}, \delta_{.._\ell}]$. Ist $R \leq \min_{s\text{-}t_\ell\text{-Schnitt } C} |C|$, so gibt es nach dem Max-flow-Min-cut-Theorem einen Fluss von s nach t_ℓ mit Rate R, also ist $\det M_\ell \neq 0$ (als Polynom). Somit ist auch das Produktpolynom $\prod_{\ell=1}^{L} \det M_\ell \neq 0$. Für $|F|$ groß hat dieses Polynom eine Nichtnullstelle, somit gibt es Parameter $\alpha_{..}$, $\beta_{..}$, $\delta_{.._\ell}$ mit $\det M_\ell \neq 0$ (in F) simultan für alle ℓ. Dann kann D_ℓ so gewählt werden, dass $M_\ell = I_R$ für alle ℓ. □

Zufällige Netzwerkcodierung. Sei $G = (V, E, \rho)$ wieder ein azyklisches Netzwerk. Ein einfacher praktischer Ansatz für Netzwerkcodierung ist, einige oder alle der Koeffizienten $A = (\alpha_{i,e})$, $B = (\beta_{e,e'})$ und $D = (\delta_{e',i})$ zufällig aus $\mathbb{F}_q$ zu wählen.

Wir nennen einen Netzwerkcode (A, B) (wobei $\operatorname{supp} A \subseteq [R] \times E$ und $\operatorname{supp} B \subseteq E^{\langle 2\rangle} \subseteq E \times E$) *gültig* für $t \in V$, falls $AC|_{\operatorname{in}(t)}$ den vollen Rang R hat, das heißt falls eine Matrix D mit $\operatorname{supp} D \subseteq \operatorname{in}(t) \times [R]$ und $ACD = I_R$ existiert (wobei $C = I + B + B^2 + \dots$).

Satz 2.28. *Sei $\alpha_{..}$, $\beta_{..}$, $\delta_{..\ell}$ die Lösung eines Netzwerkproblems mit Senken $t_1, \dots, t_L$. Wähle nun η dieser Koeffizienten zufällig (unabhängig, gleichverteilt) in $\mathbb{F}_q$. Dann ist (A, B) mit Wahrscheinlichkeit $\geq 1 - \frac{L\eta}{q}$ ein gültiger Netzwerkcode für $t_1, \dots, t_L$.*

Für $|F| = q$ groß ist zufällige Netzwerkcodierung also praktisch gut verwendbar.

Beweisskizze. Betrachte die zufälligen Koeffizienten als Variable $\xi_1, ..., \xi_\eta$. Sei $M_\ell := ACD_\ell$ die Transfermatrix für t_ℓ, sei $p_\ell := \det M_\ell$, und betrachte $p := \prod_{\ell=1}^{L} p_\ell \neq 0$ als Polynom in den ξ_i. Gesucht ist dann die Wahrscheinlichkeit $\Pr(p(x_1, ..., x_\eta) \neq 0)$.

Es gilt $\det M_\ell = \pm \det M'_\ell$ mit $M'_\ell := \begin{pmatrix} A & 0 \\ I - B & D_\ell \end{pmatrix}$, und somit folgt $\deg p_\ell \leq \eta$ für alle ℓ. Also ist $\deg p \leq L\eta$ und der Satz ergibt direkt aus folgendem Resultat. □

Lemma 2.29 (Schwartz-Zippel)**.** *Sei $0 \neq f \in \mathbb{F}_q[X_1, \dots, X_n]$ mit Totalgrad $\deg f \leq d$, sowie $x \in \mathbb{F}_q^n$ gleichverteilt, dann gilt $\Pr(f(x) = 0) \leq \frac{d}{q}$.*

Statt einzelner Symbole in F kann man auch Elemente eines F-Vektorraums V im Netzwerk übertragen. Mit der Transfermatrix $M \in F^{R \times R}$ erhalten wir so eine Transferabbildung $V^R \to V^R$ mit $x \mapsto z = xM$. Für den Fall $V = F^N$ erhalten wir somit

$$F^{N \times R} \to F^{N \times R}, \quad X \mapsto Z = XM\,.$$

Sei nun M zufällig und dem Empfänger unbekannt, wie kann man dann X rekonstruieren? Mit einem Header haben wir

$$X = \begin{pmatrix} I_R \\ X' \end{pmatrix} \;\mapsto\; Z = XM = \begin{pmatrix} M \\ X'M \end{pmatrix},$$

also erhält man M, und falls es invertierbar ist, folgt $X' = X'M \cdot M^{-1}$.

2.5 Unterraum-Codes

Eine verallgemeinerte Sichtweise der linearen Netzwerkcodierung führt zu einer neuartigen Theorie von „subspace codes“ (Koetter, Kschischang 2008). Hierbei werden Untervektorräume U eines F-Vektorraums V übertragen (etwa $\operatorname{colsp}(X)$ im Beispiel $V = F^N$ und $X \in F^{N \times R}$). Dieses Modell ist insbesondere für Fehlerkorrektur nützlich.

Definition 2.30. Sei V ein n-dimensionaler F-Vektorraum und sei $L(V)$ die Menge aller Untervektorräume $U \leq V$. Zu $0 \leq k \leq n$ definieren wir die *Graßmann'sche* als

$$G(n, k) := \{U \in L(V) \mid \dim U = k\}\,.$$

Als einen *Unterraum-Code* (von konstanter Dimension) bezeichnet man eine Teilmenge $C \subseteq L(V)$ (bzw. eine Teilmenge $C \subseteq G(n, k)$).

Lemma 2.31. *Für $F = \mathbb{F}_q$ gilt*

$$|G(n, k)| = \frac{(q^n - 1) \cdot \ldots \cdot (q^n - q^{k-1})}{(q^k - 1) \cdot \ldots \cdot (q^k - q^{k-1})} > q^{k(n-k)}\,.$$

Beweisskizze. Betrachte $X := \{x \in V^k \mid \{x_1, \dots, x_k\} \text{ linear unabhängig}\}$, sowie die Abbildung $X \to G(n,k)$, $x \mapsto \operatorname{span}(x)$. Für $x, y \in X$ ist genau dann $\operatorname{span}(x) = \operatorname{span}(y)$, wenn es $S \in \mathrm{GL}_k(F)$ gibt mit $y = xS$, und daher folgt $|G(n,k)| = |X|/|\mathrm{GL}_k(F)|$. □

Wird nun $U \leq V$ gesendet und $W \leq V$ empfangen, so schreibe $W = U' \oplus E$ mit $U' \leq U$ und $E \cap U = \{0\}$. Wir sprechen von $s := \dim U - \dim U'$ Auslöschungen (erasures) und $t := \dim E$ Fehlern (errors).

Definition 2.32. Zu $A, B \in L(V)$ definiere den „Abstand“

$$d(A,B) := \dim(A+B) - \dim(A \cap B)\,.$$

Dies ist eine Metrik auf $L(V)$ und es gilt $d(U,W) = \dim U + \dim W - 2\dim(U \cap W) = s + t$. Für $|C| > 1$ sei der *Minimalabstand* von C dann definiert durch

$$d(C) := \min_{A,B \in C,\, A \neq B} d(A,B)\,.$$

Falls $2(s+t) < d(C)$, so kann man $s+t$ Auslöschungen/Fehler korrigieren.

Proposition 2.33. *Für $k \leq \ell \leq m$ gibt es einen Unterraum-Code $C \subseteq G(\ell+m, \ell)$ mit*

$$|C| = q^{km} \quad \text{und} \quad d(C) \geq 2(\ell - k + 1)\,.$$

Für $k = 2$, $\ell = 3$, $m = 4$ beispielsweise ist C ist eine Familie von 3-dimensionalen Unterräumen im 7-dimensionalen Raum, mit $|C| = q^8$ und $d(C) \geq 4$, d. h. sie schneiden sich paarweise in höchstens einer Dimension.

Beweis. Betrachte den Erweiterungskörper $\mathbb{F}_{q^m}$ als m-dimensionalen Vektorraum über $\mathbb{F}_q$. Sei $U \leq \mathbb{F}_{q^m}$ ein Unterraum der Dimension ℓ und setze $V := U \times \mathbb{F}_{q^m}$, also $\dim V = \ell + m$. Definiere die Menge

$$\mathcal{L}_k := \Big\{ \sum_{i=0}^{k-1} a_i X^{q^i} \mid a_i \in \mathbb{F}_{q^m} \Big\}$$

von *linearisierten Polynomen* bis Grad q^{k-1}, das heißt für jedes Polynom $f \in \mathcal{L}_k$ ist die Abbildung $\mathbb{F}_{q^m} \to \mathbb{F}_{q^m}$, $\alpha \mapsto f(\alpha)$ linear über $\mathbb{F}_q$. Dann sei $C := \operatorname{im} \mathrm{ev} \subseteq \mathrm{Gr}(\ell+m, \ell)$ für

$$\mathrm{ev}\colon \mathcal{L}_k \to \mathrm{Gr}(\ell+m,\ell)\,, \quad f \mapsto A_f := \{(u, f(u)) \mid u \in U\}\,.$$

Sind nun $f, g \in \mathcal{L}_k$ mit $\dim(A_f \cap A_g) \geq k$, so hat $f - g$ mindestens q^k Nullstellen und es folgt $f = g$. Daher ist ev injektiv und es gibt $|C| = q^{km}$ Codewörter, sowie $d(A_f, A_g) = \dim A_f + \dim A_g - 2\dim(A_f \cap A_g) \geq 2(\ell - k + 1)$ für alle $f \neq g$. □

3 Public-Key-Kryptographie

Wir behandeln die wichtigsten mathematischen Grundlagen der Kryptographie mit öffentlichen Schlüsseln, nämlich das Problem des diskreten Logarithmus und das Problem der Faktorisierung ganzer Zahlen.

3.1 Diskretes Logarithmusproblem

Sei $(G, \cdot)$ eine endliche zyklische Gruppe der Ordnung n und sei $\alpha \in G$ ein Erzeuger. Wir nehmen an, dass die Gruppenverknüpfung effizient berechenbar ist.

Definition 3.1. Das *diskrete Logarithmusproblem (DLP)* in G ist das Problem:

gegeben Erzeuger $\alpha \in G$, sowie $\beta \in G$,
finde $x \in \mathbb{Z}_n$ mit $\alpha^x = \beta$.

Dieses x heißt *diskreter Logarithmus* von β zur Basis α, Notation $\log_\alpha \beta := x \in \mathbb{Z}_n$.

Der surjektive Homomorphismus $\mathbb{Z} \to G$, $x \mapsto \alpha^x$ induziert einen Isomorphismus

$$\varphi \colon \mathbb{Z}_n \to G\,, \quad [x] \mapsto \alpha^x\,,$$

mit Umkehrabbildung $\varphi^{-1} = \log_\alpha \colon G \to \mathbb{Z}_n$, $\beta \mapsto \log_\alpha \beta$.

Man kann die Abbildung φ via „square-and-multiply“ schnell auswerten, jedoch ist das Berechnen von φ^{-1} im Allgemeinen schwierig, denn dies ist gerade das DLP – eine solche Funktion heißt *Einbahnfunktion*. Man beachte, dass die Schwierigkeit von der konkreten Darstellung der Gruppe abhängt und auch formal nicht bewiesen ist.

Bemerkung 3.2. Für $\beta, \gamma, \delta \in G$ und $s \in \mathbb{Z}_n$ gelten:

$$\log_\alpha(\beta\gamma) = \log_\alpha \beta + \log_\alpha \gamma\,, \quad \log_\alpha(\beta^s) = s \log_\alpha \beta\,, \quad \log_\alpha \beta = \log_\alpha \delta \cdot \log_\delta \beta\,.$$

Ein wichtiges Beispiel für $(G, \cdot)$ ist die multiplikative Gruppe $\mathbb{F}_q^*$ des endlichen Körpers $\mathbb{F}_q$ (vgl. Satz 1.12), hier ist also $n = q-1$.

Anwendungen des DLP Die Schwierigkeit des diskreten Logarithmusproblems wird bei den folgenden kryptographischen Protokollen benutzt. Hierbei sei stets die Gruppe $(G, \cdot)$ und der Erzeuger α öffentlich.

1. **Schlüsselaustausch** (Diffie, Hellman 1976):

Alice		public		Bob
$a \in \mathbb{Z}_n$	$\to$	α^a		
		α^b	$\leftarrow$	$b \in \mathbb{Z}_n$
$k_A = (\alpha^b)^a$				$k_B = (\alpha^a)^b$

   Alice und Bob haben den gemeinsamen Schlüssel $k_A = k_B = \alpha^{ab}$.

2. **Verschlüsselung** (ElGamal 1985): Alice $\to$ Bob

   Bob wählt $b \in \mathbb{Z}_n$ and veröffentlicht seinen *public key* $\beta := \alpha^b$.
   Alice hat geheime Nachricht $m \in G$, wählt $a \in \mathbb{Z}_n$ und schickt den Geheimtext

$$(\gamma, \delta) \;:=\; (\alpha^a,\, m \oplus \beta^a).$$

   Bob entschlüsselt durch $\delta \ominus \gamma^b = \delta \ominus \alpha^{ab} = \delta \ominus \beta^a = m$.
   Dabei ist $\oplus$ eine beliebige Gruppenverknüpfung (etwa XOR) und $\ominus$ die Inverse.

3. **Digitale Signatur** (ElGamal 1985): Bob → Alice

   Bob wählt $b \in \mathbb{Z}_n$ und veröffentlicht seinen *public key* $\beta := \alpha^b$.
   Weiter sei $G \to \mathbb{Z}_n$, $r \mapsto \bar{r}$ eine öffentliche „Hash"-Abbildung.
   Um eine Nachricht $m \in \mathbb{Z}_n$ zu signieren, wählt Bob $k \in \mathbb{Z}_n^*$ und sendet

$$(r, s) \;:=\; (\alpha^k,\, k^{-1}(m - b\bar{r})) \;\in\; G \times \mathbb{Z}_n\,.$$

   Alice verifiziert die Signatur via $\alpha^m = \beta^{\bar{r}} r^s$; in der Tat ist $\beta^{\bar{r}} r^s = \alpha^{b\bar{r}} \alpha^{m-b\bar{r}} = \alpha^m$.

3.2 Algorithmen für das DLP

Wir unterscheiden zwischen *generischen* Algorithmen, die nur die Gruppenverknüpfungen benötigen und somit für beliebige Gruppen anwendbar sind, und den Indexkalkül-Methoden, die nur für bestimmte Gruppen effizient sind.

Generische Algorithmen Sei G eine zyklische Gruppe der Ordnung n mit Erzeuger α. Zu einem Element $\beta \in G$ sei der diskrete Logarithmus $x = \log_\alpha \beta \in \mathbb{Z}_n$ gesucht.

Die simpelste Methode wäre, $\alpha^0, \alpha^1, \alpha^2, \ldots$ zu berechnen, bis $\beta = \alpha^x$ gefunden ist; Laufzeit $O(n)$ Gruppenoperationen.

Baby-Step-Giant-Step. Sei $m := \lceil \sqrt{n} \rceil$. Berechne Tabelle $\{(j, \alpha^j) \mid j \in \{0..m-1\}\}$ und sortiere nach zweiter Komponente. Berechne $\gamma := \alpha^{-m}$, sowie $\beta, \beta\gamma, \beta\gamma^2, \ldots$, bis eine *Kollision* $\beta\gamma^i = \alpha^j$ gefunden wird; gib dann $\log_\alpha \beta = x = im + j$ aus.

Benötigt $O(\sqrt{n})$ Speicher, sowie $O(\sqrt{n} \log n)$ Gruppenoperationen (bzw. $O(\sqrt{n})$ bei Verwendung einer Hashtabelle).

Pollards rho. Sei $G = S_1 \dot\cup S_2 \dot\cup S_3$ eine „zufällige" Partition. Definiere Folgen (x_i) in G und (a_i), (b_i) in $\mathbb{Z}_n$, so dass $\alpha^{a_i}\beta^{b_i} = x_i$, durch $x_0 := 1_G$ und $a_0 := b_0 := 0$, sowie

$$x_{i+1} := \begin{cases} \beta x_i \\ x_i^2 \\ \alpha x_i \end{cases} \quad \text{und} \quad (a_{i+1}, b_{i+1}) := \begin{cases} (a_i, b_i{+}1) & \text{falls } x_i \in S_1, \\ (2a_i, 2b_i) & \text{falls } x_i \in S_2, \\ (a_i{+}1, b_i) & \text{falls } x_i \in S_3. \end{cases}$$

Ist $x_j = x_{j+p}$ mit $p > 0$, so gibt es i mit $x_i = x_{2i}$, das heißt $\alpha^{a_i}\beta^{b_i} = \alpha^{a_{2i}}\beta^{b_{2i}}$, dann ist $\log_\alpha \beta = x = \frac{a_{2i}-a_i}{b_i-b_{2i}}$ (falls der Nenner in $\mathbb{Z}_n^*$ ist).

Es ist i in $O(\sqrt{n})$ zu erwarten (vgl. Geburtstagsparadoxon), es sind also $O(\sqrt{n})$ Gruppenoperationen und kaum Speicher nötig, die Analyse ist jedoch nur heuristisch.

Indexkalkül-Methode Sei wieder $\log_\alpha \beta$ in der Gruppe G gesucht. Wähle Teilmenge $S \subseteq G$ mit $\langle S \rangle = G$ (oft ist $\alpha \in S$), genannt *Faktorbasis*, und betrachte den surjektiven Gruppenhomomorphismus

$$\varphi\colon \mathbb{Z}_n^S \to G\,, \quad (e_s)_{s\in S} \mapsto \prod_{s\in S} s^{e_s}\,.$$

Das Prinzip ist, zunächst $\log_\alpha s$ für alle $s \in S$ zu bestimmen. Schema:

1. **Erzeuge Relationen**: finde Elemente in $\ker\varphi$, erhalte also Teilmenge $R \subseteq \ker\varphi$.
2. **Lineare Algebra**: berechne $0 \neq (x_s) \in R^\perp$, also $\sum_{s\in S} x_s e_s = 0$ für alle $(e_s) \in R$.
3. **Individueller Log**: finde Urbild $(e_s) \in \varphi^{-1}(\beta)$, dann ist $\log_\alpha \beta = \sum_{s\in S} e_s \log_\alpha s$.

In der Tat, sind genügend Relationen gefunden, so sind die Logarithmen der Faktorbasis-Elemente (bis auf Vielfachheit) bestimmt:

Lemma 3.3. *Sei* $\operatorname{span} R = \ker \varphi$ *und* $(x_s) \in R^\perp$*, dann gibt es* $\lambda \in \mathbb{Z}_n$ *mit* $x_s = \lambda \log_\alpha s$*.*

Basisversion. Sei $\alpha \in S$. Wähle jeweils $k \in \mathbb{Z}_n$ zufällig und versuche, α^k als Produkt $\prod_{s\in S} s^{e_s}$ zu schreiben. Sind dann die Logarithmen der Faktorbasis-Elemente bestimmt, so finde noch eine Darstellung $\beta\alpha^k = \prod_{s\in S} s^{e_s}$, somit ist $\log_\alpha \beta = -k + \sum_{s\in S} e_s \log_\alpha s$.

In dem Fall, dass G die Einheitengruppe eines Restklassenrings R/I mit faktoriellem Ring R ist, kann man bei der Relationenerzeugung die Faktorisierung in R verwenden. Beispiele hierfür sind $G = \mathbb{Z}_p^*$ mit $\mathbb{Z}_p = \mathbb{Z}/(p)$, sowie $G = \mathbb{F}_{q^n}^*$ mit $\mathbb{F}_{q^n} = \mathbb{F}_q[X]/(f)$.

Beispiel 3.4. Sei $G = \mathbb{Z}_{101}^*$, also $n = 100$, sowie $\alpha = 2$ und $S = \{2, 3, 5, 7\}$.

Für $k = 7$, $k = 9$, $k = 33$ etwa ist $2^7 \equiv 3^3$, $2^9 \equiv 7$, $2^{33} \equiv 5 \cdot 7$, somit ist $R = \{(-7, 3, 0, 0), (-9, 0, 0, 1), (-33, 0, 1, 1)\}$. Wir erhalten dann $R^\perp = \mathbb{Z}_n(1, 69, 24, 9)$.

Angenommen, $\beta = 26$ und sei $k = 11$. Dann ist $\beta\alpha^k = 26 \cdot 2^{11} \equiv 3 \cdot 7$, also ist $\log_\alpha \beta = -k + x_3 + x_7 = -11 + 69 + 9 = 67$.

Laufzeitabschätzung. Betrachte die Basisversion des Indexkalküls für das DLP in $\mathbb{Z}_p^*$ mit Faktorbasis $S := \{q \text{ Primzahl} \mid q \le B\}$ für eine Schranke B. Wie groß sollte B gewählt werden und welche Laufzeit ergibt sich? Antworten liefert der folgende Satz, wobei eine Zahl *B-glatt* heißt, falls ihre sämtlichen Primfaktoren $\le B$ sind.

Satz 3.5 (Canfield, Erdős, Pomerance 1983)**.** *Eine zufällige Zahl aus* $\{1, \dots, M\}$ *ist B-glatt mit Wahrscheinlichkeit*

$$P = u^{-u(1+o(1))}, \qquad \text{wobei} \quad u = \tfrac{\log M}{\log B}.$$

In unserem Fall ist $M = n = p-1$, und wir müssen die Schranke B so wählen, dass die inverse Wahrscheinlichkeit $\frac{1}{P}$ in etwa $|S| \approx \frac{B}{\log B}$ ist. Aus dem Ansatz

$$B = \mathcal{L}_n(c) := \exp\big((c + o(1))(\log n \log\log n)^{1/2}\big),$$

für eine Konstante c, erhalten wir $\log\log B = (\frac{1}{2} + o(1)) \log\log n$, und somit folgt

$$\log u^u = u \log u = \tfrac{\log n}{\log B}(\log\log n - \log\log B) = (\tfrac{1}{2c} + o(1))(\log n \log\log n)^{1/2}.$$

Für die Relationen-Erzeugung erhalten wir somit eine (heuristische) Gesamtlaufzeit von

$$|S| \cdot \tfrac{1}{P} = \mathcal{L}_n(c) \cdot \mathcal{L}_n(\tfrac{1}{2c}) = \mathcal{L}_n(c + \tfrac{1}{2c}),$$

und die optimale Wahl $c := \frac{1}{\sqrt{2}}$ resultiert in die Laufzeit $\mathcal{L}_n(c + \frac{1}{2c}) = \mathcal{L}_n(\sqrt{2})$.

Das lineare Gleichungssystem kann (mit speziellen Algorithmen für dünnbesetzte Matrizen) in Komplexität $|S|^2 \cdot \log|S|$ gelöst werden, wegen $|S| = \mathcal{L}_n(c)$ ist die Laufzeit somit ebenfalls $\mathcal{L}_n(\sqrt{2})$. Für individuelle Logarithmen ist hingegen die Laufzeit geringer.

Glatte Polynome. Ein Polynom heißt *b-glatt*, falls sämtliche irreduziblen Teiler vom Grad $\le b$ sind; 1-glatte Polynome sind also jene, die in Linearfaktoren zerfallen.

Satz 3.6. *Ein zufälliges Polynom* $f \in \mathbb{F}_q[X]$*,* $\deg f = m$*, ist b-glatt mit Wahrscheinlichkeit* $P = u^{-u(1+o(1))}$*, wobei* $u = \frac{m}{b}$*.*

Für das DLP in $G = \mathbb{F}_{q^m}^*$, wobei q fix sei, ergibt sich damit ein ganz analoger Indexkalkül-Algorithmus wie für $G = \mathbb{Z}_p^*$, mit der gleichen Analyse und Laufzeit $\mathcal{L}_n(\sqrt{2})$.

3.3 Faktorisierungsproblem

Definition 3.7. Das *Faktorisierungsproblem* lautet:

gegeben zusammengesetzte Zahl $n = p \cdot q$ für zwei Primzahlen p, q mit $\log p \approx \log q$,
finde die Primfaktoren p, q von n.

Anwendung: RSA (Rivest, Shamir, Adleman 1978).
Sei $n = p \cdot q$ wie oben, also $\varphi(n) = |\mathbb{Z}_n^*| = (p-1) \cdot (q-1)$. Wähle $e \in \mathbb{Z}_{\varphi(n)}^*$ und sei $d := e^{-1} \in \mathbb{Z}_{\varphi(n)}^*$. Bobs öffentlicher Schlüssel ist (n, e), sein privater Schlüssel ist d.

1. Verschlüsselung: Alice $\to$ Bob
   Nachricht $m \in \mathbb{Z}_n \rightsquigarrow$ Geheimtext $c := m^e$, Bob entschlüsselt via $c^d = m^{ed} = m$.
2. Signatur: Bob $\to$ Alice
   Nachricht $m \in \mathbb{Z}_n \rightsquigarrow$ Signatur $s := m^d$, Alice verifiziert via $s^e = m^{de} = m$.

Faktorisierungsalgorithmen Um $n = p \cdot q$ (wie oben) zu faktorisieren, verwendet man häufig die Methode der *kongruenten Quadrate*, das heißt, finde x, y mit

$$x^2 \equiv y^2 \mod n\,.$$

Denn dann ist $n \mid x^2 - y^2 = (x-y)(x+y)$ und mit Wahrscheinlichkeit $\approx \frac{1}{2}$ (falls x und y zufällig, unabhängig) ist $x \not\equiv \pm y$, somit liefert $\mathrm{ggT}(x \pm y, n)$ die Primfaktoren.

Man kann solche kongruenten Quadrate mit folgender Indexkalkül-Methode finden.

Quadratisches Sieb. Sei $S := \{p \text{ Primzahl} \mid p \le B\}$ die Faktorbasis.

1. Finde x_i, so dass $x_i^2 \bmod n =: \prod_{p \in S} p^{e_{p,i}}$ über S faktorisiert; sei I die Menge dieser i.
2. Finde $J \subseteq I$ mit $f_p := \sum_{i \in J} e_{p,i}$ gerade für alle $p \in S$.

Dann ist mit $x := \prod_{i \in J} x_i$ und $y := \prod_{p \in S} p^{f_p/2}$ eine gesuchte Kongruenz $x^2 \equiv y^2 \bmod n$ gefunden. Für Schritt 2. ist ein lineares Gleichungssystem über $\mathbb{F}_2$ zu lösen.

Speziell wählt man $x_i := m + a_i$ mit $m := \lceil \sqrt{n} \rceil$ und $|a_i|$ klein (und fügt $-1 \in S$ hinzu), denn dann ist $|x_i^2 - n| = |m^2 - n + 2ma_i + a_i^2|$ in $O(\sqrt{n})$, also relativ klein.

Die Laufzeitabschätzung verläuft ähnlich wie beim DLP in $\mathbb{Z}_p^*$. Wir setzen $B = \mathcal{L}_n(c)$ und betrachten nun

$$u = \frac{\log \sqrt{n}}{\log B} = \frac{1}{2} \cdot \frac{\log n}{\log B}\,.$$

Für die Wahrscheinlichkeit P, dass eine Zahl in $\{1, \ldots, \sqrt{n}\}$ über S faktorisiert, gilt dann $\frac{1}{P} = u^{u(1+o(1))} = \mathcal{L}_n(\frac{1}{4c})$. Dies ergibt (mit $c = \frac{1}{2}$) eine erwartete Gesamtlaufzeit von

$$\mathcal{L}_n(c) \cdot \mathcal{L}_n(\tfrac{1}{4c}) = \mathcal{L}_n(c + \tfrac{1}{4c}) = \mathcal{L}_n(1)\,.$$

3.4 Fortentwicklungen des Indexkalküls

Für $\alpha \in [0,1]$ und $c \in \mathbb{R}_{>0}$ sei

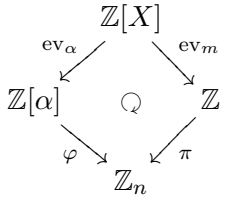
$$L_n(\alpha, c) := \exp\big((c + o(1))(\log n)^\alpha (\log\log n)^{1-\alpha}\big)\,,$$

und bei unbestimmtem c schreiben wir $L_n(\alpha)$. Demnach heißt $\alpha = 0$ gerade polynomiell und $\alpha = 1$ exponentiell in $\log n$. Die Basisversion des Indexkalküls hat eine Laufzeit von $L_n(\frac{1}{2}, c) = \mathcal{L}_n(c)$, welche sich jedoch weiter verbessern lässt.

$L(\frac{1}{3})$-**Algorithmen** Die Grundidee ist dabei, Relationen so zu erzeugen, dass die Elemente auf beiden Seiten zufällig erscheinen und simultan glatt sein müssen, dabei jedoch erheblich „kleiner“ sind.

Zahlkörpersieb (1993) zur Faktorisierung von n.

Sei $f \in \mathbb{Z}[X]$ ein irreduzibles normiertes Polynom mit einer Nullstelle $\alpha \in \mathbb{C}$, sowie einer Nullstelle $m \in \mathbb{Z}$ modulo n, also $n \mid f(m)$. Dann existiert ein Homomorphismus $\varphi\colon \mathbb{Z}[\alpha] \to \mathbb{Z}_n$ im folgenden kommutativen Diagramm:



Speziell sei f so, dass $n = m^d + f_{d-1}m^{d-1} + \cdots + f_1 m + f_0$ mit $m = \lfloor n^{1/d} \rfloor$.

Finde nun $g_i \in \mathbb{Z}[X]$ mit $\prod g_i(\alpha) = \beta^2$ in $\mathbb{Z}[\alpha]$, somit $\varphi(\prod g_i(\alpha)) = \varphi(\beta)^2 = x^2$, sowie $\prod g_i(m) = y^2$, denn dann ist $x^2 \equiv y^2 \bmod n$ ein kongruentes Quadrat.

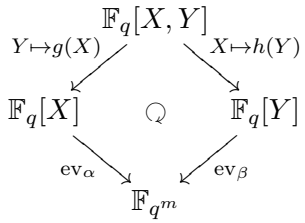
Ein Problem hierbei ist die Faktorisierung in $\mathbb{Z}[\alpha]$, wofür man eine *Normabbildung* $N\colon \mathbb{Q}(\alpha) \to \mathbb{Q}$ mit $N(\mathbb{Z}[\alpha]) \subseteq \mathbb{Z}$ verwendet, um glatte Elemente $g_i(\alpha)$ zu erzeugen.

Speziell wähle $g_i := a_i X + b_i$ mit $|a_i|, |b_i| \le C$, wobei

$$C = L_n(\tfrac{1}{3}) \quad \text{und} \quad d \approx (\tfrac{\log n}{\log\log n})^{1/3},$$

somit $m \approx n^{1/d} = L_n(\frac{1}{3})$. Dann haben wir $N(g_i(\alpha)) = a_i^d f(-\frac{b_i}{a_i}) \le C^d n^{1/d} = L_n(\frac{2}{3})$, und dies resultiert in eine Laufzeit von $L_n(\frac{1}{3})$ (mit $c = (\frac{64}{9})^{1/3} = 1.923$).

Für das DLP in $\mathbb{Z}_p^*$ hat man einen zum Zahlkörpersieb analogen Algorithmus. Diesen hat man für $\mathbb{F}_{q^m}^*$ angepasst, und man spricht vom *Funktionenkörpersieb*. Wir betrachten folgende Variante (Joux, Lercier 2006):



Hierbei seien $g, h \in \mathbb{F}_q[X]$ so, dass $\mathbb{F}_{q^m} = \mathbb{F}_q[X]/\langle f\rangle$ mit $f \mid h(g(X)) - X$, sowie $\alpha := [X]$ und $\beta := g(\alpha)$, so dass $\alpha = h(\beta)$. Für $a, b, c \in \mathbb{F}_q$ betrachte nun $XY + aY + bX + c \in \mathbb{F}_q[X,Y]$, also in $\mathbb{F}_{q^m}$ entlang des Diagramms

$$\alpha g(\alpha) + a g(\alpha) + b\alpha + c \;=\; h(\beta)\beta + a\beta + b h(\beta) + c\,.$$

Sind die entsprechenden Polynome auf beiden Seiten glatt, so ist eine Relation gefunden. Im Fall $q = L_{q^m}(\frac{1}{3})$ beispielsweise genügt es, als Faktorbasis die linearen Polynome zu wählen, und mit $\deg g, \deg h \approx \sqrt{n}$ erhält man wiederum einen $L_n(\frac{1}{3})$-Algorithmus (mit verbesserter Konstante $c = 3^{1/3} = 1.442$).

Körper kleiner Charakteristik Dramatische Weiterentwicklungen seit 2013 beruhen auf dem folgenden einfachen Resultat.

Lemma 3.8. *Seien $u, v \in \mathbb{F}_q[X]$. Dann gilt*

$$v \prod_{\mu \in \mathbb{F}_q} (u - \mu v) \;=\; u^q v - u v^q \,. \qquad (*)$$

Betrachte nun das DLP in $\mathbb{F}^*_{(q^k)^m}$ mit $k \geq 2$ und q fix.

Wählt man $u = \alpha X + \beta$ und $v = \gamma X + \delta$ für $\alpha, \beta, \gamma, \delta \in \mathbb{F}_{q^k}$, so zerfällt die linke Seite von $(*)$ „systematisch“ in Linearfaktoren. Die rechte Seite von $(*)$ ist (bis auf Vielfaches r) von der Form $X^{q+1} + aX^q + bX + c$; sei nun $\mathbb{F}_{(q^k)^m} = \mathbb{F}_{q^k}[X]/(f)$ mit $f \mid h_1 X^q - h_0$ für $h_i \in \mathbb{F}_{q^k}[X]$ mit $\deg h_i \leq 2$, so erhalten wir mit $x := [X]$ dann

$$(\gamma x + \delta) \prod_{\mu \in \mathbb{F}_q} \big((\alpha x + \beta) - \mu(\gamma x + \delta)\big) \;=\; \tfrac{r}{h_1}\big(x h_0(x) + a h_0(x) + b x h_1(x) + c h_1(x)\big)\,,$$

wobei die rechte Seite nur Grad ≤ 3 hat.

Dieser Ansatz resultiert in einen *polynomiellen* Algorithmus für die Phasen 1 und 2 der Indexkalkül-Methode. Weiterhin gibt es ein effizientes Verfahren für die Phase 3, welches ebenfalls auf Lemma 3.8 basiert, und man erhält insgesamt eine *quasi-polynomielle* Laufzeit, nämlich $m^{O(\log m)}$.

Mit diesem Algorithmus wurde 2014 eine Rekordberechnung aufgestellt, für das DLP in $\mathbb{F}^*_{(2^{18})^{513}} = \mathbb{F}^*_{2^{9234}}$, siehe auch `http://en.wikipedia.org/Discrete_Logarithm_Records`.