Diskrete Mathematik

Götz Kersting, SS 2005

Die Diskrete Mathematik behandelt ‚diskrete', insbesondere endliche Objekte und Strukturen. Der Name betont den Gegensatz zu den kontinuierlichen Ansätzen und Methoden in der Mathematik. Die Grenzwertbetrachtungen der Analysis treten in den Hintergrund, wichtig sind Prozeduren, die nach endlich vielen Schritten zu einem Resultat führen, also **Algorithmen**. Die Diskrete Mathematik verdankt ihren Aufschwung auch den Computerwissenschaften, für die sie eine wichtige Rolle spielt. Die Theoretische Informatik ist besonders an effizienten Algorithmen interessiert, während für die Kryptographie algorithmisch schwere Probleme von Bedeutung sind.

Literatur

M. Aigner (1994): Diskrete Mathematik, Vieweg

L. Childs (1979): A Concrete Introduction to Higher Algebra, Springer

T. Ihringer (1993): Diskrete Mathematik, Teubner

D.E. Knuth (1973,1981): The Art of Computer Programming, Vol. I,II, Addison-Wesley

N. Koblitz (1994): A Course in Number Theory and Cryptography, Springer

J.H. van Lint (1992): Introduction to Coding Theory, Springer

C. Papadimitriou (1994): Computational Complexity, Addison Wesley

Inhaltsverzeichnis

1 Der Euklidische Algorithmus **4**
- 1.1 Größte gemeinsame Teiler und der Euklidische Algorithmus . . 4
- 1.2 Laufzeit des Euklidischen Algorithmus 8
- 1.3 Ein binärer Algorithmus 9
- 1.4 Der Euklidische Algorithmus für Polynome 10
- 1.5 Euklidische Ringe 12
- 1.6 Eine geometrische Sicht 15

2 Kongruenzen und modulares Rechnen **22**
- 2.1 Der Restklassenring $\mathbb{Z}_m$ 22
- 2.2 Der Chinesische Restsatz 24
- 2.3 Ein probabilistischer Gleichheitstest 26
- 2.4 Exakte Lösung ganzzahliger Gleichungssysteme 27
- 2.5 Ein allgemeiner Chinesischer Restsatz 29
- 2.6 Prime Restklassen 33
- 2.7 Ein probabilistischer Primzahltest 35
- 2.8 Öffentliche Chiffriersysteme 36
- 2.9 Zero-Knowledge Beweise 39
- 2.10 Faktorzerlegung 45
- 2.11 Gruppen 53

3 Fehlerkorrigierende Kodes **55**
- 3.1 Der Hamming–Kode 55
- 3.2 Die mittlere Fehlerzahl des Hamming-Kodes 58
- 3.3 Lineare Kodes 61
- 3.4 Zyklische Kodes 66

4 Endliche Körper **70**
- 4.1 Eine algebraische Version des Hamming-Kodes 70
- 4.2 Die Struktur endlicher Körper 72
- 4.3 Konstruktion von endlichen Körpern 76

4.4 BCH-Kodes 78
4.5 Spezielle Fälle von BCH-Kodes 82
4.6 Elliptische Kurven über endlichen Körpern 84

5 Lineares Programmieren **90**
5.1 Grundbegriffe 90
5.2 Dualität 92
5.3 Eckpunkte 94
5.4 Ganzzahliges Programmieren 96
5.5 Der Simplex-Algorithmus 102

Kapitel 1

Der Euklidische Algorithmus

1.1 Größte gemeinsame Teiler und der Euklidische Algorithmus

Der Euklidische Algorithmus gehört zu den ältesten Rechenverfahren, er war schon Eudoxus (375 v.Chr.) bekannt. Er ist grundlegend und kommt in vielen Rechenprozeduren zur Anwendung. Der Algorithmus dient zur Bestimmung von größten gemeinsamen Teilern. Wir betrachten zunächst den Ring $\mathbb{Z}$ der ganzen Zahlen. Zur Notation: Man schreibt $a \mid b$ für ganze Zahlen a, b, falls a Teiler von b ist, d.h., falls es ein ganze Zahl c gibt, so dass $ac = b$.

Definition 1.1. *$d \in \mathbb{Z}$ heißt **größter gemeinsamer Teiler**, kurz* ggT *der ganzen Zahlen $a_1, \ldots, a_n \neq 0$, falls gilt:*

i) $d \mid a_1, \ldots, d \mid a_n$,

ii) Gilt $z \mid a_1, \ldots, z \mid a_n$ für eine ganze Zahl z, so folgt $z \mid d$.

*Wir schreiben dann $d = \mathrm{ggT}(a_1, \ldots, a_n)$. Gilt $1 = \mathrm{ggT}(a_1, \ldots, a_n)$, so sagt man, $a_1, \ldots, a_n$ sind **relativ prim** oder **teilerfremd**.*

Aus $z \mid d$ folgt $|z| \leq |d|$ (wegen $a_1, \ldots, a_n \neq 0$ ist $d = 0$ ausgeschlossen). In diesem Sinne ist d am größten unter allen Teilern von $a_1, \ldots, a_n$. Ist d' ein weiterer ggT für $a_1, \ldots, a_n$, so teilen sich d und d' gegenseitig, so dass $|d'| = |d|$, also $d' = \pm d$ folgt. Der ggT von $a_1, \ldots, a_n$ ist daher bis auf das Vorzeichen eindeutig bestimmt.

Etwas weniger evident ist, dass immer ein größter gemeinsamer Teiler existiert – etwa für zwei natürliche Zahlen a und b. Man kann Gebrauch machen davon, dass sich ganze Zahlen in eindeutiger Weise in Primfaktoren zerlegen lassen. Seien $p_1, \ldots, p_r$ die Primzahlen, die in a oder b als Teiler

enthalten sind. Dann gilt $a = p_1^{e_1} \cdots p_r^{e_r}$ und $b = p_1^{f_1} \cdots p_r^{f_r}$ mit ganze Zahlen $e_1, \ldots, e_r, f_1, \ldots, f_r \geq 0$ ($e_i = 0$ bedeutet, dass p_i kein Teiler von a ist), und die gemeinsamen Teiler von a und b haben die Gestalt $z = \pm p_1^{h_1} \cdots p_r^{h_r}$ mit $h_i \in \{0, 1, \ldots, g_i\}$, $g_i := \min(e_i, f_i)$. Die beiden größten gemeinsamen Teiler von a und b sind folglich $d = \pm p_1^{g_1} \cdots p_r^{g_r}$. – Für explizite Rechnungen ist dieses Vorgehen nicht geeignet, denn die Zerlegung einer Zahl in ihre Primfaktoren ist sehr rechenaufwendig.

Wir gehen hier anders vor und klären die Existenzfrage, indem wir ein Rechenverfahren angeben, dass größte gemeinsame Teiler liefert. Es beruht auf einer grundlegenden Eigenschaft ganzer Zahlen, der **Division mit Rest**: Zu ganzen Zahlen $a, b \neq 0$ gibt es ganze Zahlen m, r, so dass

$$a \;=\; mb + r \;\text{ und }\; 0 \;\leq\; r \;<\; |b|.$$

Die Idee des Euklidischen Algorithmus ist es, aus zwei Zahlen den ggT schrittweise herauszudividieren (durch ‚Wechselwegnahme' zu gewinnen).

Euklidischer Algorithmus.

Eingabe: ganze Zahlen $a, b \neq 0$.

Ausgabe: $r_{j-1} = \mathrm{ggT}(a, b)$.

Verfahren: Setze $r_{-1} = a, r_0 = b$. Bestimme durch Division mit Rest sukzessive ganze Zahlen $r_1, \ldots, r_{j-1}$ mit $|b| > r_1 > \cdots > r_{j-1} > r_j = 0$, bis kein Rest mehr bleibt. r_i sei also der Rest, der bei Division von r_{i-2} durch r_{i-1} entsteht:

$$\begin{aligned}
r_{-1} &= m_1 r_0 + r_1, \\
r_0 &= m_2 r_1 + r_2, \\
&\vdots \\
r_{i-2} &= m_i r_{i-1} + r_i, \\
&\vdots \\
r_{j-3} &= m_{j-1} r_{j-2} + r_{j-1}, \\
r_{j-2} &= m_j r_{j-1},
\end{aligned}$$

mit ganzen Zahlen $m_1, \ldots, m_j$. □

Da die Divisionsreste r_i strikt fallen, bricht das Verfahren nach endlich vielen Schritten ab. Es ist korrekt: r_{j-1} teilt der Reihe nach $r_{j-2}, r_{j-3}, \ldots, r_0 = b$ und $r_{-1} = a$, wie sich sukzessive aus den Gleichungen $r_{i-2} = m_i r_{i-1} + r_i$ ergibt. Teilt umgekehrt z sowohl a wie b, so teilt z der Reihe nach $r_1, \ldots, r_{j-1}$, wie aus den Gleichungen $r_i = r_{i-2} - m_i r_{i-1}$ folgt.

Beispiel. Für $a = 1736$, $b = 1484$ ergibt das Verfahren

$$\begin{aligned} 1736 &= 1 \cdot 1484 + 252 \\ 1484 &= 5 \cdot 252 + 224 \\ 252 &= 1 \cdot 224 + 28 \\ 224 &= 8 \cdot 28 \ (+0). \end{aligned}$$

Also

$$28 = \mathrm{ggT}(1736, 1484).$$

Liest man die Gleichungen von unten nach oben, so erkennt man: $28 \mid 224$, $28 \mid 252$, $28 \mid 1484$, $28 \mid 1736$. Liest man von oben nach unten, so ist klar, dass jeder Teiler von 1736 und 1484 auch 28 teilt. □

Weiter liefert der Algorithmus eine Darstellung des ggT als Linearkombination: Aus der Gleichung $r_{j-3} = m_{j-1}r_{j-2} + r_{j-1}$ läßt sich r_{j-2} mittels $r_{j-2} = r_{j-4} - m_{j-2}r_{j-3}$ eliminieren. Genauso lassen sich der Reihe nach $r_{j-3}, \ldots, r_1$ eliminieren, und wir erhalten r_{j-1} als ganzzahlige Linearkombination von a und b. Im Beispiel sieht das so aus:

$$28 = 252 - 224 = 6 \cdot 252 - 1484 = 6 \cdot 1736 - 7 \cdot 1484.$$

Damit haben wir im Fall $n = 2$ den folgenden **Satz von Bézout** bewiesen (Den Fall $n > 2$ folgt per Induktion, Übung).

Satz 1.2. *Für $d = \mathrm{ggT}(a_1, \ldots, a_n)$ gibt es ganze Zahlen $z_1, \ldots, z_n$, so dass*

$$d = z_1a_1 + \cdots + z_na_n.$$

Durch Erweiterung des Euklidischen Algorithmus kann man mit dem ggT zweier Zahlen a und b gleichzeitig seine Darstellung nach dem Satz von Bézout gewinnen. Man bestimme dazu ganze Zahlen $s_{-1}, \ldots, s_{j-1}, t_{-1}, \ldots, t_{j-1}$ rekursiv aus den Gleichungen

$$\begin{aligned} s_{-1} &= 1,\ s_0 = 0, \quad t_{-1} = 0,\ t_0 = 1, \\ s_{i-2} &= m_is_{i-1} + s_i, \quad t_{i-2} = m_it_{i-1} + t_i, \end{aligned}$$

unter Benutzung der vom Euklidischen Algorithmus gewonnenen ganzen Zahlen $m_1, \ldots, m_j$. Dann gilt

$$\mathrm{ggT}(a, b) = r_{j-1} = as_{j-1} + bt_{j-1}.$$

Beweis. Es gilt sogar $r_i = as_i + bt_i$ für alle $-1 \le i < j$, wie sich per Induktion nach i ergibt: Für $i = -1, 0$ folgt dies nach Wahl von s_{-1}, s_0, t_{-1} und t_0, und der Induktionsschritt folgt aus

$$\begin{aligned} r_i &= r_{i-2} - m_ir_{i-1} \\ &= as_{i-2} + bt_{i-2} - m_i(as_{i-1} + bt_{i-1}) \\ &= as_i + bt_i. \end{aligned}$$

Beispiel. a=1736, b=1484.

i	r_{i-1}	r_i	m_{i+1}	s_{i-1}	s_i	t_{i-1}	t_i
0	1736	1484	1	1	0	0	1
1	1484	252	5	0	1	1	-1
2	252	224	1	1	-5	-1	6
3	224	28	8	-5	6	6	-7

Also

$$28 \;=\; 6 \cdot 1736 - 7 \cdot 1484. \qquad \square$$

Eine andere Verwendung des Euklidischen Algorithmus besteht darin, rationale Zahlen als Kettenbrüche auszudrücken. Dazu formen wir die Divisionen $r_{i-2} = m_i r_{i-1} + r_i$ des Algorithmus um zu den Gleichungen

$$\frac{r_{i-2}}{r_{i-1}} \;=\; m_i + \left(\frac{r_{i-1}}{r_i}\right)^{-1}. \tag{1.1}$$

Diese Ausdrücke lassen sich ineinander einsetzen, und es entsteht ausgehend von $r_{-1}/r_0 = a/b$ die Kettenbruchdarstellung

$$\frac{a}{b} \;=\; m_1 + \left(\frac{r_0}{r_1}\right)^{-1} \;=\; m_1 + \cfrac{1}{m_2 + \left(\cfrac{r_1}{r_2}\right)^{-1}} \;=\; \dots \tag{1.2}$$

Zum Beispiel ist

$$\frac{1736}{1484} \;=\; 1 + \cfrac{1}{5 + \cfrac{1}{1 + \cfrac{1}{8}}}$$

Allgemein schreiben wir

$$\frac{a}{b} \;=\; m_1 + \frac{1}{m_2+}\,\frac{1}{m_3+}\cdots\frac{1}{m_{j-1}+}\,\frac{1}{m_j} \tag{1.3}$$

unter Verwendung der Notation

$$m_1 + \frac{1}{m_2+}\,\frac{1}{m_3+}\cdots\frac{1}{m_{j-1}+}\,\frac{1}{m_j} \;:=\; m_1 + \cfrac{1}{m_2 + \cfrac{1}{m_3 + \cfrac{1}{\ddots\cfrac{}{m_{j-1} + \cfrac{1}{m_j}}}}}$$

Für die Berechnung der Kettenbrüche braucht man nicht auf den Euklidischen Algorithmus zurückzugreifen. Wir formen (1.1) um zu der Gleichung

$$\alpha_i \;=\; m_i + \alpha_{i+1}^{-1},\ i = 1, \ldots, j-1, \quad \text{mit } \alpha_i := \frac{r_{i-2}}{r_{i-1}}.$$

Da die Reste r_i strikt fallen, gilt $\alpha_{i+1} > 1$ für $i \geq 1$, und es folgt

$$m_i \;=\; [\alpha_i], \quad \alpha_{i+1} \;=\; (\alpha_i - m_i)^{-1}.$$

Ausgehend vom Startwert $\alpha_1 = a/b$ kann man so $m_1, m_2, \ldots$ rekursiv berechnen. Diese Prozedur heißt **Kettenbruchalgorithmus**.

1.2 Laufzeit des Euklidischen Algorithmus

Der Euklidische Algorithmus führt sehr schnell zum Resultat, die Anzahl der benötigten Divisionen hat eine Größe, die nur logarithmisch von den Eingabedaten a und b abhängt. Dies zeigt eine *worst-case Analyse* des Algorithmus. Besonders ungünstige Fälle erhält man mit den Fibonacci–Zahlen 0,1,1,2,3,5,8,13,21,34,55,...

Definition 1.3. *Die* ***Fibonacci–Zahlen*** $F_0, F_1, \ldots$ *sind rekursiv definiert als* $F_0 := 0, F_1 := 1$ *und* $F_i := F_{i-1} + F_{i-2}$ *für* $i \geq 2$.

Bei der Wahl $a = F_{j+2}$, $b = F_{j+1}$ ergeben sich für den Euklidischen Algorithmus folgende j Divisionen

$$\begin{aligned} a &= b + F_j, \\ b &= F_j + F_{j-1}, \\ &\vdots \\ F_{j-i+3} &= F_{j-i+2} + F_{j-i+1}, \\ &\vdots \\ F_4 &= F_3 + F_2, \\ F_3 &= 2F_2, \end{aligned}$$

denn abgesehen von $F_1 = F_2 = 1$ sind die Fibonacci–Zahlen strikt wachsend, so dass $F_i = F_{i-1} + F_{i-2}$ die Division von F_i durch F_{i-1} mit Rest ist. Es folgt $1 = \mathrm{ggT}(F_{j+2}, F_{j+1})$. Im vorliegenden Fall gilt $m_1 = \cdots = m_{j-1} = 1$, $m_j = 2$ und $r_{j-1} = 1$, und dies sind die minimal möglichen Werte, denn $m_j = 1$ ist beim Euklidischen Algorithmus in der letzten Division ausgeschlossen.

Für die Fibonacci-Zahlen gilt die bemerkenswerte Formel

$$F_i \;=\; \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^i - \left(\frac{1-\sqrt{5}}{2}\right)^i\right].$$

Der Beweis ergibt sich leicht per Induktion. Die Fälle $i = 0$ und 1 prüft man direkt nach. Außerdem erfüllt der angegebene Ausdruck für F_i die rekursiven Gleichungen der Fibonacci-Zahlen. Dies liegt daran, dass die Zahlen

$$\phi_1 \;:=\; \frac{1+\sqrt{5}}{2}, \quad \phi_2 \;:=\; \frac{1-\sqrt{5}}{2}$$

die Gleichung $\phi^2 = \phi + 1$ und folglich $\phi^i = \phi^{i-1} + \phi^{i-2}$ erfüllen.

Satz 1.4. *Der Euklidische Algorithmus benötigt bei der Eingabe $a > b > 0$ höchstens $c \ln(b\sqrt{5})$ Divisionen, mit $c = \left(\ln \frac{1+\sqrt{5}}{2}\right)^{-1} \approx 2,08$.*

Beweis. Für die Divisionsreste gilt $r_i \geq F_{j-i+1}$, $i \leq j-1$, wie man per Induktion von $i = j-1$ bis $i = -1$ zeigt: Es gilt $r_{j-1} \geq 1 = F_2$, $r_{j-2} \geq 2 = F_3$, sowie im Induktionsschritt

$$r_{i-1} \;=\; m_{i+1} r_i + r_{i+1} \;\geq\; F_{j-i+1} + F_{j-i} \;=\; F_{j-i+2}.$$

Insbesondere folgt $b = r_0 \geq F_{j+1}$. Aus obiger Formel für F_j erhält man

$$\frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^j \;=\; F_j + \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^j \;\leq\; F_j + 1 \;\leq\; b$$

(es gilt $\phi_2 \approx -0,618$) bzw.

$$j \;\ln \frac{1+\sqrt{5}}{2} \;\leq\; \ln(b\sqrt{5}).$$

□

Der Beweis zeigt: Für jedes Paar $a > b > 0$ von Zahlen, für das der Euklidische Algorithmus genau j Divisionen braucht, gilt $b \geq F_{j+1}$ und $a \geq F_{j+2}$.

1.3 Ein binärer Algorithmus

Für den Computer ist Euklids Algorithmus in den ganzen Zahlen nicht das günstigste Verfahren. Divisionen mit Rest sind vergleichsweise rechenaufwendig. Der folgende Algorithmus benötigt nur Divisionen durch 2, sie sind auf

einem Rechner besonders schnell zu realisieren, da er Zahlen in Dualdarstellung verarbeitet. Die Idee ist, a und b schrittweise zu verkleinern auf eine Art, dass sich der ggT in kontrollierbarer Weise verändert. Im ersten Schritt nimmt man aus a und b den gemeinsamen geraden Anteil hinaus, gemäß der Regel

$$a, b \text{ gerade}, d = \mathrm{ggT}(a, b) \quad \Rightarrow \quad d = 2 \cdot \mathrm{ggT}(a/2, b/2).$$

Anschließend verkleinert man a und b schrittweise, ohne den ggT noch zu verändern. Ist a gerade und b ungerade, so ersetzt man a durch $a/2$ und läßt b unverändert. Der ggT ändert sich nicht, denn

$$a \text{ gerade}, b \text{ ungerade}, d = \mathrm{ggT}(a, b) \quad \Rightarrow \quad d = \mathrm{ggT}(a/2, b).$$

Der Fall a ungerade, b gerade ist analog. Sind a und b beide ungerade, so benutzen wir die Regel

$$d = \mathrm{ggT}(a, b) \quad \Rightarrow \quad d = \mathrm{ggT}(a - b, b).$$

Wir ziehen dann die kleinere der Zahlen von der größeren ab. Diese Regeln kommen abwechselnd zu Zuge, wie das folgende Beispiel zeigt.

$$\begin{aligned}
\mathrm{ggT}(1736, 1484) &= 4 \cdot \mathrm{ggT}(434, 371) = 4 \cdot \mathrm{ggT}(217, 371) \\
&= 4 \cdot \mathrm{ggT}(217, 154) = 4 \cdot \mathrm{ggT}(217, 77) = 4 \cdot \mathrm{ggT}(140, 77) \\
&= 4 \cdot \mathrm{ggT}(35, 77) = 4 \cdot \mathrm{ggT}(35, 42) = 4 \cdot \mathrm{ggT}(35, 21) \\
&= 4 \cdot \mathrm{ggT}(14, 21) = 4 \cdot \mathrm{ggT}(7, 21) = 4 \cdot \mathrm{ggT}(7, 14) \\
&= 4 \cdot \mathrm{ggT}(7, 7) = 28.
\end{aligned}$$

1.4 Der Euklidische Algorithmus für Polynome

Der binäre Algorithmus ist an die Computerarithmetik angepaßt und benutzt spezielle Eigenschaften der ganzen Zahlen. Die Stärke des Euklidischen Algorithmus besteht darin, dass er sich auch in anderen Rechenbereichen als dem der ganzen Zahlen anwenden läßt. Wir betrachten nun **Polynome** in der Variablen x mit rationalen Koeffizienten (oder allgemeiner mit Koeffizienten in einem Körper K). Ein rationales Polynom ist gegeben durch einen Ausdruck der Gestalt

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

mit $n \in \mathbb{N}$ und $a_i \in \mathbb{Q}$, $i = 0, \ldots, n$. Glieder $a_i x^i$ mit dem Koeffizienten $a_i = 0$ dürfen aus der Summe weggelassen bzw. zur Summe beliebig hinzugefügt werden. Wir betrachten also zwei Polynome als identisch, falls sie dieselben Glieder haben, abgesehen von Summanden mit dem Koeffizienten 0.

Bemerkung. Es ist hier weder nötig noch angemessen, sich Polynome wie in der Analysis als Funktionen vorzustellen. Wir fassen Polynome als formale Ausdrücke auf. Dieser Unterschied in der Auffassung macht sich zwar für rationale Polynome nicht bemerkbar, sehr wohl jedoch für Polynome mit Koeffizienten in anderen Körpern. So gilt $f(0) = f(1) = 0$ für das Polynom $f(x) = x(x+1) = x^2 + x$ mit Koeffizienten aus dem Körper $\mathbb{Z}_2 = \{0, 1\}$, es nimmt nur den Wert 0 an. Dennoch ist dieses Polynom vom Nullpolynom verschieden. □

Mit Polynomen kann man rechnen wie mit ganzen Zahlen. Zwei Polynome $f(x) = a_n x^n + \cdots + a_0$ und $g(x) = b_m x^m + \cdots + b_0$ lassen sich addieren und multiplizieren:

$$(f+g)(x) \;:=\; (a_n + b_n)x^n + \cdots + (a_0 + b_0)$$

(o.E.d.A. kann man $m = n$ annehmen) und

$$(fg)(x) \;:=\; c_{m+n}x^{m+n} + \cdots + c_0, \qquad \text{mit } c_i \;:=\; \sum_{j+k=i} a_j b_k.$$

Es bereitet keine Mühe, die bekannten Rechenregeln für Zahlen auf Polynome zu übertragen, Assoziativität, Kommutativität, Distributivität. Die Rolle der Zahl 0 übernimmt das **Nullpolynom**, dessen Koeffizienten alle verschwinden. Insgesamt haben wir den Ring $\mathbb{Q}[x]$ der rationalen Polynome konstruiert (zur Definition eines Ringes vergleiche den folgenden Abschnitt).

Um eine Division mit Rest für Polynome einzuführen, definieren wir den **Grad** $\deg(f)$ eines vom Nullpolynom verschiedenen Polynoms $f(x) = a_n x^n + \cdots + a_0$. Er ist die größte Zahl i, so dass $a_i \neq 0$. Das zugehörige a_i heißt **Anfangskoeffizient** von f.

Seien nun $f, g \neq 0$ Polynome vom Grade n und m, mit Anfangskoeffizienten a_n und b_m. Falls $n \geq m$, können wir g aus f hinausdividieren, also das Polynom

$$h(x) \;=\; f(x) - a_n b_m^{-1} x^{n-m} g(x)$$

bilden. Offenbar hat h einen kleineren Grad als f. Gilt $\deg(h) \geq m$, so kann g aus h ein weiteres Mal hinausdividiert werden. Dies läßt sich fortsetzen, bis ein Polynom r vom Grade kleiner m oder aber das Nullpolynom übrigbleibt. Wir erhalten wie für die ganzen Zahlen eine **Division mit Rest**: Zu rationalen Polynomen $f(x), g(x) \neq 0$ existieren rationale Polynome $m(x), r(x)$, so dass

$$f(x) \;=\; m(x)g(x) + r(x), \;\text{ mit }\; \deg(r) < \deg(g) \text{ oder } r(x) = 0.$$

Der Euklidische Algorithmus läßt sich nun auch auf rationale Polynome anwenden. Da sich der Grad der Restpolynome bei jeder Division verkleinert, bricht er nach endlich vielen Schritten ab, seine Laufzeit ist durch den Grad des Polynoms g beschränkt. Wie für ganze Zahlen können wir also feststellen: Zwei rationale Polynome $f(x), g(x) \neq 0$ besitzen einen ggT $d(x)$, und es gibt rationale Polynome $s(x)$ und $t(x)$, so dass

$$d(x) \;=\; s(x)f(x) + t(x)g(x).$$

Beispiel. Seien $a > b > 0$ natürliche Zahlen. Wir wollen den ggT von $x^a - 1$ und $x^b - 1$ berechnen. Sei $a = mb + r$ die Division von a durch b mit Rest r. Für die Division von $x^a - 1$ durch $x^b - 1$ erhalten wir

$$x^a - 1 \;=\; \left(x^{a-b} + x^{a-2b} + \cdots + x^{a-mb}\right)\left(x^b - 1\right) + x^r - 1.$$

Der Rest ist $x^r - 1$, er verschwindet genau dann, wenn $r = 0$ ist, und ist sonst von demselben Typ wie die beiden Polynome, von denen wir ausgegangen sind. Der Euklidische Algorithmus, angewandt auf $x^a - 1$ und $x^b - 1$, läuft daher parallel ab zum Euklidischen Algorithmus, angewandt auf a und b. Insbesondere folgt

$$x^{\mathrm{ggT}(a,b)} - 1 \;=\; \mathrm{ggT}(x^a - 1, x^b - 1).$$

□

Diese Überlegungen gelten nicht nur für rationale Polynome, die Koeffizienten können genauso gut reelle oder komplexe Zahlen sein. Für die Division mit Rest langt es, dass die Koeffizienten einem Körper K angehören, denn in Körpern kann man wie in den reellen Zahlen addieren, multiplizieren und dividieren (die Definition eines Körpers wiederholen wir im nächsten Abschnitt). Später werden insbesondere endliche Körper wichtig.

Dagegen hat man im Ring $\mathbb{Z}[x]$ aller Polynome $f(x) = a_n x^n + \cdots + a_0$, deren Koeffizienten a_i ganze Zahlen sind, im allgemeinen keine Division mit Rest, denn der Kehrwert b_m^{-1} des Anfangskoeffizienten von $g(x)$ läßt sich in den ganzen Zahlen nur bilden, falls $b_m = \pm 1$ ist.

1.5 Euklidische Ringe

Eine Division mit Rest hat man nicht nur für die ganzen Zahlen und für Polynome. Allgemein nennt man die Bereiche, in denen eine Division mit Rest möglich ist, Euklidische Ringe. Wir rekapitulieren kurz diese Terminologie aus der Algebra.

Sei G eine Menge mit einer binären Verknüpfung. Je zwei Elementen a, b aus G seien also ein Element ab (ihr ‚Produkt') zugeordnet. Gilt Assoziativität $(ab)c = a(bc)$, so heißt G (zusammen mit der Verknüpfung) eine **Halbgruppe**. Ein **neutrales Element** $e \in G$ ist durch die Eigenschaft $a = ea = ae$ für alle $a \in G$ gekennzeichnet. G kann höchstens ein neutrales Element besitzen, denn für neutrale Elemente e, e' folgt $e = ee' = e'$. Gibt es zu $a \in G$ ein $b \in G$, so dass $ab = ba = e$, so heißt b das **inverse Element** von a. Ist auch b' invers zu a, so folgt $b' = b'(ab) = (b'a)b = b$, das inverse Element ist also eindeutig. Man schreibt $b = a^{-1}$. Enthält eine Halbgruppe G ein neutrales Element, und besitzt jedes $a \in G$ ein inverses Element, so nennt man G eine **Gruppe**. Es gilt dann $(ab)^{-1} = b^{-1}a^{-1}$. Ist in G zusätzlich noch Kommutativität $ab = ba$ für alle a, b gegeben, so heißt G eine **abelsche Gruppe**.

Sei nun R eine Menge, die wie die ganzen Zahlen mit zwei binären Operationen $a + b$ und ab (‚Addition' und ‚Multiplikation') versehen ist. Dann heißt R (zusammen mit $+$ und $\cdot$) ein **Ring**, falls gilt:

i) R ist in Bezug auf die Addition eine abelsche Gruppe, mit neutralem Element 0. Jedes $a \in R$ besitzt also ein ‚entgegengesetztes' Element $-a$, so dass $a + (-a) = 0$ (das inverse Element bzgl. der Addition).

ii) R ist in Bezug auf die Multiplikation eine Halbgruppe.

iii) Es gilt Distributivität: $a(b + c) = ab + ac$, $(a + b)c = ac + bc$.

Man schreiben $a - b$ für $a + (-b)$ und $a \mid b$, wenn es ein $c \in R$ mit $ac = b$ gibt. Es gilt

$$a \cdot 0 = 0 \cdot a = 0$$

(beachte $a \cdot 0 + a \cdot a = a(0 + a) = a \cdot a$), ein Produkt ist also 0, falls ein Faktor 0 ist. Die Umkehrung ist im Allgemeinen nicht richtig, wir werden Ringe kennenlernen, in denen $ab = 0$ gelten kann für zwei Elemente $a, b \neq 0$. a und b heißen dann **Nullteiler**. Ringe ohne Nullteiler haben den Vorteil, dass man in ihnen kürzen darf: Aus $ac = bc$ folgt $(a - b)c = 0$ und damit $a = b$, sofern c kein Nullteiler ist.

Wir werden uns nur mit Ringen befassen, die ein Einselement enthalten. Ein **Einselement** $1 \in R - \{0\}$ in einem Ring R ist ein neutrales Element bzgl. der Multiplikation in $R - \{0\}$, es erfüllt also $1a = a1 = a$ für alle $a \neq 0$. Diejenigen $a \in R - \{0\}$, die bzgl. der Multiplikation ein Inverses a^{-1} besitzen, heißen die **Einheiten** des Ringes. Die Menge R^* aller Einheiten ist eine Gruppe: Für $a, b \in R^*$ ist $b^{-1}a^{-1}$ invers zu ab.

In manchen Ringen ist das Kommutativitätsgesetz verletzt. Ist die Multiplikation kommutativ, $ab = ba$, so spricht man von einem **kommutativen**

Ring. Ein kommutativer Ring ohne Nullteiler mit Einselement heißt **Integritätsbereich**. Besitzt jedes Element $a \neq 0$ eines Integritätsbereiches R ein Inverses a^{-1}, gilt also $R^* = R - \{0\}$, so nennt man R einen **Körper**.

Beispiel. Sei R ein Ring. Dann ist auch die Menge aller Polynome $f(x) = a_n x^n + \cdots + a_1 x + a_0$ mit $a_i \in R$ ein Ring, der **Polynomring** $R[x]$. Zwei Polynome gelten als gleich, wenn sie sich nur um Summanden unterscheiden, deren Koeffizienten 0 sind. Addition und Multiplikation werden wie in Abschnitt 1.4 definiert. R ist in $R[x]$ durch die Polynome vom Grad 0 eingebettet. Ein Einselement in R ist auch Einselement in $R[x]$, und es gilt $R[x]^* = R^*$. Man sagt, $R[x]$ entsteht aus R *durch Adjunktion der Unbestimmten* x. □

Definition 1.5. *Ein Integritätsbereich R heißt **Euklidischer Ring**, falls jedem $a \in R - \{0\}$ eine nicht-negative ganze Zahl $g(a)$ zugeordnet ist, und falls zu beliebigen $a, b \in R - \{0\}$ Ringelemente m, r existieren, so dass gilt: $a = mb + r$ und entweder $r = 0$ oder $g(r) < g(b)$.*

Unsere Überlegungen über größte gemeinsame Teiler übertragen sich vollständig auf Euklidische Ringe. Man hat erneut den Euklidischen Algorithmus zur Verfügung, daher existieren größte gemeinsame Teiler im Sinne der Definition 1.1, und es gilt der Satz von Bézout: Für $a, b \neq 0$ und $d = \mathrm{ggT}(a, b)$ gibt es $s, t \in R$, so dass

$$d \;=\; sa + tb.$$

Man überzeugt sich leicht, dass größte gemeinsame Teiler bis auf Einheiten eindeutig bestimmt ist.

Beispiele.

1) $\mathbb{Z}$ ist ein Euklidischer Ring, mit $g(a) := |a|$.

2) Sei K ein Körper (etwa $\mathbb{Q}, \mathbb{R}, \mathbb{C}$ oder $\mathbb{Z}_2 = \{0, 1\}$). Dann ist der Polynomring $K[x]$ ein Euklidischer Ring, mit $g(f) := \deg(f)$ (vergleiche Abschnitt 1.4).

3) Die *ganzen Gaußschen Zahlen*, also die komplexen Zahlen $z = x + iy$ mit $x, y \in \mathbb{Z}$, bilden bzgl. der komplexen Addition und Multiplikation einen Euklidischen Ring, mit $g(z) := |z|^2 = x^2 + y^2$ (Übung). □

1.6 Eine geometrische Sicht

Seien a, b teilerfremde ganze Zahlen. Nach dem Satz von Bézout gibt es dann ganze Zahlen s, t, so dass $as+bt = 1$ gilt, bzw. (nach Wechsel von Vorzeichen) ganze Zahlen a', b', so dass

$$ab' - ba' = \pm 1$$

gilt. In diesem Abschnitt liefern wir einen geometrischen Kontext.

Wir betrachten das 2-dimensionale Gitter

$$G \;=\; \{(x, y) : x, y \in \mathbb{Z}\}$$

aller Tupel (x, y) mit ganzzahligen Komponenten. Jeder Gitterpunkt $\overline{v} = (x, y)$ hat die Darstellung $\overline{v} = x\overline{e} + y\overline{e}'$ mit

$$\overline{e} \;:=\; (1, 0)\;, \quad \overline{e}' \;:=\; (0, 1)$$

und $x, y \in \mathbb{Z}$. Man sagt, $\overline{e}, \overline{e}'$ bilden eine *Gitterbasis*. Allgemeiner heißen Gitterpunkte

$$\overline{g} \;=\; (a, b)\;, \quad \overline{g}' \;=\; (a', b')$$

eine Gitterbasis, falls für alle $\overline{v} \in G$ ganze Zahlen x, y gibt, so dass

$$\overline{v} \;=\; x\overline{g} + y\overline{g}'$$

gilt. Diese Darstellung ist dann nach der Linearen Algebra notwendigerweise eindeutig. Wir fragen, wann $\overline{g}, \overline{g}'$ eine Gitterbasis bilden. Dazu ist es offenbar notwendig wie hinreichend, dass es ganze Zahlen x, y, x', y' gibt, so dass

$$\overline{e} \;=\; x\overline{g} + y\overline{g}'\;, \quad \overline{e}' \;=\; x'\overline{g} + y'\overline{g}'$$

gilt, oder in Matrix–Schreibweise

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \;=\; \begin{pmatrix} x & y \\ x' & y' \end{pmatrix} \cdot \begin{pmatrix} a & b \\ a' & b' \end{pmatrix}.$$

Mit anderen Worten: Notwendig und hinreichend ist, dass die aus a, b, a', b' gebildete Matrix eine Inverse besitzt, deren Einträge zudem ganzzahlig sind. Für die Determinanten folgt $1 = (xy' - yx')(ab' - ba')$, so dass wir die Bedingung $ab' - ba' = \pm 1$ als notwendig erkennen. Sie ist auch hinreichend, wie man der Gleichung

$$\begin{pmatrix} a & b \\ a' & b' \end{pmatrix}^{-1} \;=\; \frac{1}{ab' - ba'} \begin{pmatrix} b' & -b \\ -a' & a \end{pmatrix}$$

entnimmt. Wir halten fest: $\overline{g}$ und $\overline{g}'$ bilden genau dann eine Gitterbasis, wenn

$$ab' - ba' \;=\; \pm 1$$

gilt.

Bemerkung. Bekanntlich ist $ab' - ba'$ der (orientierte) Flächeninhalt des von $\overline{g}$ und $\overline{g}'$ in der Ebene aufgespannten Parallelogramms. $\overline{g}$ und $\overline{g}'$ bilden also genau dann eine Gitterbasis, wenn dieser Flächeninhalt gleich ± 1 ist. □

Eine notwendige Bedingung, $\overline{g} \in G$ zu einer Gitterbasis ergänzt zu können, ist offenbar, dass a und b relativ prim sind. In diesem Fall nennt man $\overline{g}$ *sichtbar*, denn dann befindet sich auf der Verbindungsstrecke zwischen $\overline{g}$ und dem Gitterursprung $\overline{0} := (0,0)$ kein weiterer Gitterpunkt (und ein Beobachter im Ursprung kann $\overline{g}$ sozusagen sehen).

Der Satz von Bézout besagt, dass die Bedingung auch hinreichend ist: *Jeder sichtbare Gitterpunkt $\overline{g}$ lässt sich zu einer Gitterbasis $\overline{g}, \overline{g}'$ ergänzen* (die Aussage gilt analog für höherdimensionale Gitter). Wir können den Satz nun auch auf geometrische Art beweisen.

Dazu stellen wir ein weiteres Kriterium auf dafür, dass Gitterpunkte eine Gitterbasis bilden und definieren das von $\overline{g}, \overline{g}'$ aufgespannte Dreieck als

$$\Delta[\overline{g}, \overline{g}'] \ := \ \{\lambda\overline{g} + \lambda'\overline{g}' \ : \ \lambda \geq 0 \ , \ \lambda' \geq 0 \ , \ \lambda + \lambda' \leq 1\} \ .$$

Wir behaupten: *$\overline{g}, \overline{g}'$ bilden eine Gitterbasis, falls $\Delta[\overline{g}, \overline{g}']$ nicht entartet ist und von den Gitterpunkten nur $\overline{0}$, $\overline{g}$ und $\overline{g}'$ (also seine Eckpunkte) enthält.* Ist nämlich $\overline{g}, \overline{g}'$ Gitterbasis, so durchläuft $x\overline{g}+y\overline{g}'$, $x, y \in \mathbb{Z}$ alle Gitterpunkte und in $\Delta[\overline{g}, \overline{g}']$ genau die Eckpunkte. Ist andererseits $\overline{g}, \overline{g}'$ keine Gitterbasis, so ist entweder $\overline{g}, \overline{g}'$ überhaupt keine Basis im gewöhnlichen Sinne der Linearen Algebra - dann entartet $\Delta[\overline{g}, \overline{g}']$ zu einer Linie - oder es gibt einen Gitterpunkt $\overline{v} = (x + \lambda)\overline{g} + (y + \lambda')\overline{g}'$ mit $x, y \in \mathbb{Z}$, $\lambda, \lambda' \in [0, 1)$, so dass λ und λ' nicht gleichzeitig 0 sind. Dann sind auch $\lambda\overline{g} + \lambda'\overline{g}' = \overline{v} - x\overline{g} - y\overline{g}'$ und $(1 - \lambda)\overline{g} + (1 - \lambda')\overline{g}' = (x + 1)\overline{g} + (y + 1)\overline{g}' - \overline{v}$ Gitterpunkte, und einer von beiden liegt in $\Delta[\overline{g}, \overline{g}']$, ohne Eckpunkt zu sein.

Man kann den Satz von Bézout nun folgendermaßen begründen: Ist $\overline{g}$ sichtbarer Gitterpunkt und $\overline{v}$ weiterer Gitterpunkt. Wähle in $\Delta[\overline{g}, \overline{v}]$ einen Gitterpunkt $\overline{g}' \neq \overline{g}, \overline{0}$, der möglichst nahe an der Strecke von $\overline{0}$ nach $\overline{g}$ liegt. Dann enthält $\Delta[\overline{g}, \overline{g}']$ keinen zusätzlichen Gitterpunkt, und $\overline{g}, \overline{g}'$ ist Gitterbasis.

Beispiel. Die Farey–Reihe. Wir betrachten zu vorgegebener natürlicher Zahl n alle Brüche a/b in gekürzter Form mit

$$0 \leq a \leq b \leq n \ .$$

In aufsteigender Folge $0 = a_1/b_1 < a_2/b_2 < \cdots < a_k/b_k = 1$ heißen sie die Farey–Reihe F_n. Zum Beispiel ist F_5

$$\frac{0}{1} \ , \ \frac{1}{5} \ , \ \frac{1}{4} \ , \ \frac{1}{3} \ , \ \frac{2}{5} \ , \ \frac{1}{2} \ , \ \frac{3}{5} \ , \ \frac{2}{3} \ , \ \frac{3}{4} \ , \ \frac{4}{5} \ , \ \frac{1}{1} \ .$$

Die zugehörigen Gitterpunkte (b_i, a_i) durchlaufen alle sichtbaren Gitterpunkte im Dreieck $\Delta[(n,0),(n,n)]$, mit wachsender Steigung geordnet. Für zwei benachbarte Brüche a/b und a'/b' in einer Farey–Reihe enthalten damit die Dreiecke $\Delta[(b,a),(b',a')]$ keine weiteren Gitterpunkte, und es folgt

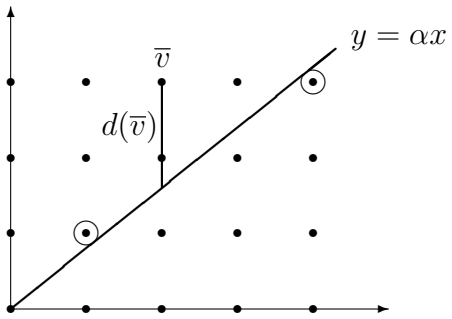
$$ab' - ba' = \pm 1 \ .$$

□

Ein Euklidischer Algorithmus in der Ebene. Wir betrachten nun die Frage, wie man eine Zahl $\alpha > 0$ möglichst genau durch Brüche a/b annähern kann. Der Ansatz ist geometrisch: Gesucht werden Gitterpunkte $\overline{v} = (b,a)$, die besonders nahe an der durch die Gleichung $y = \alpha x$ gegebenen Geraden liegen, so dass also

$$d(\overline{v}) \ := \ |a - \alpha b|$$

besonders klein wird (bzw. $d(\overline{v})/\sqrt{1+\alpha^2}$, der Abstand zwischen $\overline{v}$ und der Geraden).



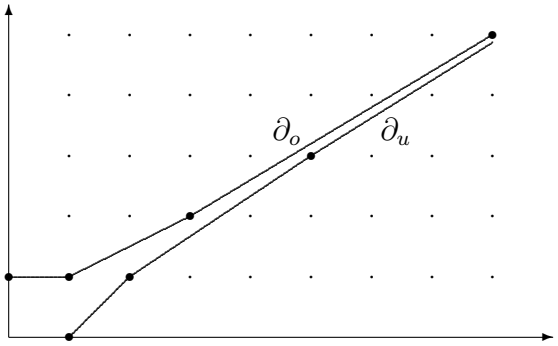
Dazu betrachten wir einerseits die konvexe Hülle $K_o \subset \mathbb{R}^2_+$ aller Gitterpunkte ungleich $\overline{0}$ oberhalb der Geraden im positiven Quadranten (also die kleinste konvexe Menge, die all diese Gitterpunkte enthält),

$$K_o \ := \ kH\{(b,a) \in G \ : \ a \geq \alpha b \ , \ a > 0\}$$

sowie die analoge konvexe Menge unterhalb der Geraden,

$$K_u \ := \ kH\{(b,a) \in G \ : \ \alpha b \geq a \ , \ b > 0\} \ .$$

Der Rand von K_o oberhalb der Geraden ist ein konvexer Streckenzug ∂_o mit Eckpunkten, den Extremalpunkten von K_o, die wir mit $\overline{g}_0 = (0,1), \overline{g}_2 = (b_2,a_2), \overline{g}_4 = (b_4,a_4), \ldots$ bezeichnen, und der Rand von K_u ist ein konkaver Streckenzug ∂_u mit den Eckpunkten $\overline{g}_{-1} = (1,0), \overline{g}_1 = (b_1,a_1), \overline{g}_3 = (b_3,a_3), \ldots$



Diese Folge der Eckpunkte bieten sich an als Kandidaten für Brüche a_i/b_i, die α besonders gut approximieren. Es ist geometrisch klar, dass die Eckpunkte sichtbare Gitterpunkte sind, dass es sich also um gekürzte Brüche handelt.

Bemerkenswerterweise lassen sich die Eckpunkte nach Art des Euklidischen Algorithmus bestimmen. Wir betrachten dazu folgende geometrische Konstruktion.

Konstruktion. *$\overline{v}_1 = (b_1, a_1)$, $\overline{v}_2 = (b_2, a_2)$ seien zwei Punkte in $\mathbb{R}^2_+$, die nicht auf derselben Seite der Geraden $y = \alpha x$ liegen, also*

$$\frac{a_1}{b_1} < \alpha < \frac{a_2}{b_2} \quad \textit{oder} \quad \frac{a_1}{b_1} > \alpha > \frac{a_2}{b_2}.$$

Setze $\overline{v}_3 = \overline{v}_1 + m\overline{v}_2$, wobei m die größte ganze Zahl sei, so daß $\overline{v}_3$ noch auf derselben Seite der Geraden liegt wie $\overline{v}_1$. Dann gilt

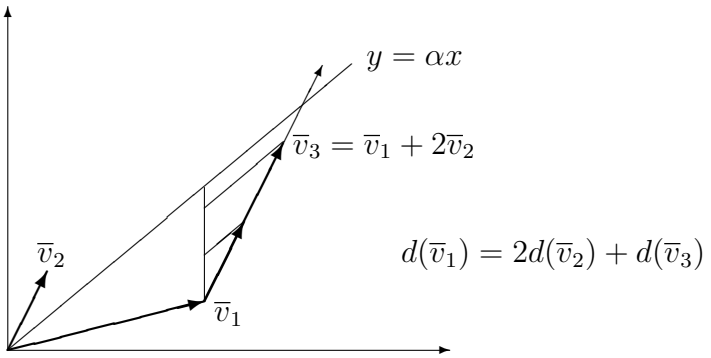
$$d(\overline{v}_1) \;=\; md(\overline{v}_2) + d(\overline{v}_3), \quad 0 \le d(\overline{v}_3) < d(\overline{v}_2).$$

Im Falle $d(\overline{v}_1) \ge d(\overline{v}_2)$, also $m \ge 1$, gilt

$$\frac{a_1}{b_1} < \frac{a_3}{b_3} \le \alpha \quad \textit{bzw.} \quad \frac{a_1}{b_1} > \frac{a_3}{b_3} \ge \alpha.$$

□

$d(\overline{v}_3)$ berechnet sich also aus $d(\overline{v}_1)$ und $d(\overline{v}_2)$ nach demselben Schema wie bei der Division mit Rest für die ganzen Zahlen. Die beiden letzten Ungleichungen ergeben sich daraus, daß die Steigung des Vektors $\overline{v}_3$ zwischen α und der Steigung von $\overline{v}_1$ liegt. Das folgende Bild verdeutlicht die Situation im Fall $a_1/b_1 < \alpha < a_2/b_2$.



Durch wiederholte Anwendung der Konstruktion entsteht aus den Startwerten $(1,0)$ und $(0,1)$ eine ganze Folge von Gitterpunkten.

Euklidischer Algorithmus in der Ebene.

Eingabe: $\alpha > 0$

Ausgabe: $\overline{v}_1 = (b_1, a_1)$, $\overline{v}_2 = (b_2, a_2)$, ...

Verfahren: Setze $\overline{v}_{-1} = (1,0)$, $\overline{v}_0 = (0,1)$. Bestimme $m_1, m_2, \ldots \in \mathbb{N}_0$, $r_1, r_2, \ldots \geq 0$ und $\overline{v}_1 = (b_1, a_1)$, $\overline{v}_2 = (b_2, a_2) \ldots \in \mathbb{N}_0^2$ rekursiv aus den Gleichungen

$$\begin{aligned} d(\overline{v}_{i-2}) &= m_i d(\overline{v}_{i-1}) + r_i, \quad 0 \leq r_i < d(\overline{v}_{i-1}) \quad \text{und} \\ \overline{v}_i &= m_i \overline{v}_{i-1} + \overline{v}_{i-2}, \end{aligned}$$

$i = 1, 2 \ldots$ Gilt $r_j = 0$, so beende die Prozedur. □

Wegen $d(\overline{v}_{-1}) = \alpha$ und $d(\overline{v}_0) = 1$ ist $m_1 = [\alpha]$ und $\overline{v}_1 = (1, [\alpha])$. Für ganzzahliges α bricht das Verfahren bereits ab. Andernfalls ergibt sich die Ungleichung $d(\overline{v}_0) = 1 > d(\overline{v}_1) = \alpha - [\alpha]$, und unsere Konstruktion kommt zum Zuge. Es folgt schrittweise $d(\overline{v}_{i+1}) = r_{i+1} < d(\overline{v}_i)$, und folglich nach unserer Konstruktionsvorschrift $m_2, m_3, \ldots \geq 1$. Die zugehörigen Näherungsbrüche nähern sich α und liegen abwechselnd oberhalb und unterhalb von α,

$$\frac{a_1}{b_1} < \frac{a_3}{b_3} < \frac{a_5}{b_5} < \cdots \leq \alpha \leq \cdots < \frac{a_4}{b_4} < \frac{a_2}{b_2}.$$

Das Verfahren bricht ab, falls $r_j = d(\overline{v}_j) = 0$ für ein j, also $a_j/b_j = \alpha$. Für irrationales α ist dies nicht möglich, dann entstehen unendlich viele Näherungsbrüche. Im rationalen Fall $\alpha = a/b$ mit $a, b \in \mathbb{N}$ sind $bd(\overline{v}_i) = |ba_i - ab_i|$

ganzzahlig, und die Gleichungen $bd(\overline{v}_{i-2}) = m_i bd(\overline{v}_{i-1}) + bd(\overline{v}_i)$ sind Divisionen mit Rest, wie sie beim Euklidischen Algorithmus in den ganzen Zahlen auftreten. Daher bricht im rationalen Fall unser Verfahren genau wie der Euklidische Algorithmus ab.

Wir zeigen nun, dass der Algorithmus genau die Eckpunkte von ∂_o und ∂_u erzeugt.

Behauptung. Es gilt $\overline{v}_i = \overline{g}_i$ für alle $i = -1, 0, \ldots$

Beweis. Die Verbindungsstrecken zwischen $\overline{v}_0, \overline{v}_2, \ldots$ entstehen durch Abtragen von Vielfachen von $\overline{v}_{-1}, \overline{v}_1, \ldots$ und haben daher wachsende Steigung. Deswegen ist der Streckenzug durch $\overline{v}_0, \overline{v}_2, \ldots$ konvex, und analog der Streckenzug durch $\overline{v}_{-1}, \overline{v}_1, \ldots$ konkav. Dazwischen liegt die Gerade $y = \alpha x$, daher langt es zu zeigen, dass zwischen den beiden Streckenzügen keine Gitterpunkte liegen. Wir zerlegen den Bereich in die Dreiecke Δ_i mit Eckpunkten $\overline{v}_{i-1}, \overline{v}_i, \overline{v}_{i+1}$ und zeigen, dass in Δ_i höchstens auf der Verbindungslinie von $\overline{v}_{i-1}$ nach $\overline{v}_{i+1}$ Gitterpunkte liegen.

Dazu stellen wir fest, dass $\overline{v}_{i-1}, \overline{v}_i$ für alle $i \geq 0$ Gitterbasis ist: Für $i = 0$ liegt das Einheitsgitter vor, und mit $\overline{v}_{i-1}, \overline{v}_i$ ist offenbar auch $\overline{v}_i, \overline{v}_{i+1} = \overline{v}_{i-1} + m_{i+1}\overline{v}_i$ Gitterbasis. Betrachten wir also auf Δ_i die lineare Funktion

$$(a, b) \mapsto \ell(a, b) := ab_i - ba_i,$$

so gilt $\ell(\overline{v}_{i-1}) = \ell(\overline{v}_{i+1}) = \pm 1$ sowie $\ell(\overline{v}_i) = 0$. Daher hat ℓ auch auf der Strecke zwischen $\overline{v}_{i-1}$ und $\overline{v}_{i+1}$ den Wert ± 1. Sonst kann ℓ auf Δ_i keine ganzzahligen Werte annehmen, deswegen liegen dort keine Punkte mit ganzzahligen Koordinaten. Dies ergibt unsere Behauptung. □

Ergänzungen.

1) Da $\overline{v}_i$, $\overline{v}_{i+1}$ Gitterbasis ist, gilt

$$a_{i+1}b_i - b_{i+1}a_i \;=\; \pm 1.$$

2) **Die Approximationsgüte.** Da α zwischen a_i/b_i und a_{i+1}/b_{i+1} liegt, folgt wegen $a_{i+1}b_i - b_{i+1}a_i = \pm 1$ und $b_{i+1} > m_{i+1}b_i$

$$\left|\alpha - \frac{a_i}{b_i}\right| \;\leq\; \left|\frac{a_{i+1}}{b_{i+1}} - \frac{a_i}{b_i}\right| = \frac{1}{b_{i+1}b_i} \;<\; \frac{1}{m_{i+1}b_i^2} \tag{1.4}$$

und damit der **Satz von Lagrange**

$$\left|\alpha - \frac{a_i}{b_i}\right| \;<\; \frac{1}{b_i^2}\,.$$

3) **Eine Umformung des Euklidischen Algorithmus.** Algorithmisch gesehen ist folgende Vorgehensweise günstig: Wir formen die Gleichungen $d(\overline{v}_{i-2}) = m_i d(\overline{v}_{i-1}) + d(\overline{v}_i)$ um zu

$$\alpha_i \;=\; m_i + \alpha_{i+1}^{-1}, \quad \text{mit} \;\; \alpha_i := \frac{d(\overline{v}_{i-2})}{d(\overline{v}_{i-1})}.$$

Da $d(\overline{v}_j)$ für $j \geq 0$ strikt fällt , ist $\alpha_{i+1} > 1$, also

$$m_i = [\alpha_i], \quad \alpha_{i+1} = (\alpha_i - m_i)^{-1}.$$

Unter Beachtung von $\alpha_1 = \alpha$ lassen sich aus diesen Gleichungen die $m_1, m_2, \ldots$ rekursiv bestimmen.

4) **Kettenbruchdarstellung der Näherungsbrüche.** Es gilt

$$\frac{a_j}{b_j} \;=\; m_1 + \frac{1}{m_2+}\frac{1}{m_3+}\cdots\frac{1}{m_{j-1}+}\frac{1}{m_j}.$$

Zum Beweis betrachten wir auch die Gerade $y = \beta x$ mit $\beta = a_j/b_j$. Sie verläuft zwischen ∂_o und ∂_u, bis sie $\overline{v}_j$ erreicht. Deswegen ergibt der Euklidische Algorithmus, nun auf β angewandt, bis dahin dieselben Vielfachen $m_1, \ldots, m_j$, dann bricht er ab. Mit den entsprechenden Abstände $d'(\overline{v}_i)$ folgt wie eben

$$\beta_i \;=\; m_i + \beta_{i+1}^{-1} \quad \text{mit} \;\; \beta_i := \frac{d'(\overline{v}_{i-2})}{d'(\overline{v}_{i-1})}.$$

Nun gilt $\beta_1 = \beta = a_j/b_j$ und $\beta_{j+1}^{-1} = 0$. Die Behauptung folgt also durch sukzessives Einsetzen wie früher bei (1.2).

Diese Darstellung der Näherungsbrüche als Kettenbrüche motiviert für irrationale Zahlen α die Schreibweise

$$\alpha \;=\; m_1 + \frac{1}{m_2+}\frac{1}{m_3+}\cdots\ .$$

Beispielsweise gilt (Übung)

$$\sqrt{2} \;=\; 1 + \frac{1}{2+}\frac{1}{2+}\frac{1}{2+}\cdots\ .$$

Die Kettenbruchentwicklung von $\pi = 3,14159265\ldots$ ergibt sich als

$$\pi \;=\; 3 + \frac{1}{7+}\,\frac{1}{15+}\,\frac{1}{1+}\,\frac{1}{292+}\,\frac{1}{1+}\cdots$$

Wegen $m_5 = 292$ ist die Näherung $3 + \frac{1}{7+}\,\frac{1}{15+}\,\frac{1}{1} = 3 + \frac{16}{113} = 3,14159292\ldots$ für π besonders gut (vgl. die Abschätzung (1.4)). □

Kapitel 2

Kongruenzen und modulares Rechnen

2.1 Der Restklassenring $\mathbb{Z}_m$

Beim Rechnen mit ganzen Zahlen ist es häufig von Vorteil, nicht mit den Zahlen selbst zu operieren, sondern mit den Resten, die beim Teilen der Zahlen durch ein fest vorgegebenes $m \in \mathbb{Z}$ übrigbleiben. m wird dann als **Modul** bezeichnet. Entscheidend ist, daß man mit den Resten genauso rechnen kann wie mit den Zahlen selbst, die Rechenregeln bleiben zu einem wesentlichen Teil bestehen. Der Vorteil des Rechnens mit Resten liegt auf der Hand: Die Reste sind im allgemeinen viel kleiner als die Zahlen selbst.

Definition 2.1. *Sei $m \in \mathbb{N}$ und $a, b \in \mathbb{Z}$.*

*i) a und b heißen **kongruent modulo m**, falls $m \mid b - a$, falls also a und b denselben Rest bei Division durch m haben. Man schreibt dann*

$$a \equiv b \bmod m.$$

*ii) Die Menge $\overline{a} := \{a + zm \ : \ z \in \mathbb{Z}\} = a + m\mathbb{Z}$ heißt **Restklasse** von a modulo m. Eine andere Schreibweise für die Restklasse ist $a \bmod m$. Die Menge aller Restklassen wird mit $\mathbb{Z}/m\mathbb{Z}$ oder $\mathbb{Z}_m$ bezeichnet.*

Es gilt $a \equiv b \bmod m \Leftrightarrow \overline{a} = \overline{b}$. Die Kongruenz ist also eine Äquivalenzrelation, und $\mathbb{Z}$ zerfällt modulo m in m disjunkte Restklassen. Der folgende Satz präzisiert die Aussage, daß sich das Rechnen mit ganzen Zahlen im wesentlichen auf die Restklassen überträgt.

Satz 2.2. *Es gilt*

$$a \equiv a',\ b \equiv b' \bmod m \quad \Rightarrow \quad a+b \equiv a'+b',\ ab \equiv a'b' \bmod m,$$

so daß $\overline{a}+\overline{b} := \overline{a+b}, \overline{a}\cdot\overline{b} := \overline{ab}$ *wohldefinierte Verknüpfungen in* $\mathbb{Z}_m$ *sind. Damit wird* $\mathbb{Z}_m$ *zu einem kommutativen Ring mit Einselement.*

Beweis. $m \mid (a-a')$ und $m \mid (b-b')$ implizieren $m \mid (a+b-(a'+b'))$ und, wegen $ab - a'b' = a(b-b') + (a-a')b'$, auch $m \mid (ab-a'b')$. Dies ergibt die erste Behauptung. Die Rechenregeln übertragen sich von $\mathbb{Z}$ unmittelbar auf $\mathbb{Z}_m$, z.B.

$$\begin{aligned}\overline{a}+(\overline{b}+\overline{c}) &= \overline{a}+\overline{b+c} = \overline{a+(b+c)}\\ &= \overline{(a+b)+c} = \overline{a+b}+\overline{c} = (\overline{a}+\overline{b})+\overline{c}.\end{aligned}$$

Das Nullelement ist $\overline{0}$, das Einselement $\overline{1}$. □

Beispiel. Seien $a = \sum a_i 10^i$, $b = \sum b_i 10^i$ ganze Zahlen in Dezimaldarstellung und $c = \sum c_i 10^i$ ihr Produkt. Dann gilt wegen $10 \equiv 1 \bmod 9$

$$\sum a_i \sum b_i \equiv ab \equiv c \equiv \sum c_i \bmod 9.$$

Die **Neunerprobe** zur Kontrolle von Multiplikationen (bei der eine Zahl durch die Summe der Ziffern aus ihrer Dezimaldarstellung ersetzt wird) beruht auf dieser Feststellung. □

$\mathbb{Z}_m$ heißt **Restklassenring modulo m**. Diese Ringe haben Besonderheiten, wie man sie von $\mathbb{Z}$ nicht kennt:

Nullteiler. Haben a und m einen gemeinsamen Teiler d, also $a = cd$ und $m = bd$, so folgt $\overline{a}\cdot\overline{b} = \overline{cdb} = \overline{cm} = \overline{0}$. Ist $\overline{a}, \overline{b} \neq \overline{0}$, so heißen $\overline{a}, \overline{b}$ Nullteiler. Zum Beispiel gilt $\overline{2}\cdot\overline{3} = \overline{0}$ in $\mathbb{Z}_6$, jedoch $\overline{2}, \overline{3} \neq \overline{0}$. Anders als $\mathbb{Z}$ enthält $\mathbb{Z}_m$ Nullteiler, abgesehen von dem Fall, daß m eine Primzahl ist.

In Ringen mit Nullteilern kann man im allgemeinen nicht kürzen. So gilt $\overline{4}\cdot\overline{2} = \overline{4}\cdot\overline{5}$, aber $\overline{2} \neq \overline{5}$ in $\mathbb{Z}_6$.

Einheiten. Seien nun a und m teilerfremd. Dann gibt es Zahlen s und t, so daß $as + mt = 1$, also $\overline{1} = \overline{a}\cdot\overline{s} + \overline{mt} = \overline{a}\cdot\overline{s} + \overline{0} = \overline{a}\cdot\overline{s}$. Es ist also $\overline{a}$ eine Einheit in $\mathbb{Z}_m$ und $\overline{s} = \overline{a}^{-1}$. Inverse Elemente lassen sich mit dem (erweiterten) Euklidischen Algorithmus berechnen. In $\mathbb{Z}_7$ gilt $\overline{2}^{-1} = \overline{4}$, $\overline{3}^{-1} = \overline{5}$ und $\overline{6}^{-1} = \overline{6}$.

Eine besondere Rolle spielen die Restklassenringe modulo einer Primzahl p. In $\mathbb{Z}_p$ besitzt jede Restklasse ungleich $\overline{0}$ eine inverse Restklasse, es gilt also der folgende Satz.

Satz 2.3. *Ist p eine Primzahl, so ist $\mathbb{Z}_p$ ein Körper.*

2.2 Der Chinesische Restsatz

Es ist eine allgemeine Strategie zur Bewältigung von umfangreichen Rechnungen, das Problem in kleinere, vom Rechenaufwand her überschaubare Teile zu zerlegen (‚divide et impera'). Dazu hat sich das Rechnen mit Restklassen als wirksam erwiesen. In der **modularen Arithmetik** wird eine ganze Zahl a durch ein Zahlentupel $(a_1, \ldots, a_k)$ ersetzt. Man gibt sich paarweise teilerfremde Moduln $m_1, \ldots, m_k$ vor und bestimmt die a_i als die Reste, die bei Division von a durch m_i übrigbleiben. Im letzten Abschnitt haben wir gesehen, daß sich das Rechnen mit a in kanonischer Weise auf die a_i überträgt. Gerechnet wird daher soweit wie möglich mit den Resten modulo m_i.

Diese Strategie setzt voraus, daß man a aus $(a_1, \ldots, a_k)$ wieder zurückgewinnen kann. Sei also a' eine weitere ganze Zahl, die bei Division durch m_i die Reste a_i ergibt. Dann gilt $m_i \mid (a - a')$ für alle i. Wegen der Eindeutigkeit der Primfaktorzerlegung in den ganzen Zahlen folgt $m \mid (a - a')$ bzw. $a \equiv a' \bmod m$, mit $m = m_1 \cdots m_k$, denn die m_i sind als paarweise teilerfremd angenommen. Ist daher $-m/2 < a \leq m/2$, so ist a durch $(a_1, \ldots, a_k)$ eindeutig festgelegt. Man wird also die m_i so wählen, daß m ausreichend groß ist.

Beispiel. Modulares Multiplizieren. Wir bilden $21 \cdot 45$, indem wir erst modulo 9,10 und 11 multiplizieren, und dann die Resultate zusammenfassen. Es gilt $23 \cdot 45 \equiv 0 \bmod 9$, $21 \cdot 45 \equiv 5 \bmod 10$, $21 \cdot 45 \equiv -1 \bmod 11$. Zusammensetzen ergibt $23 \cdot 45 \equiv 945 \bmod 990$. □

Es ergibt sich also die Frage, wie man am Ende der Rechnung a aus den Resten a_i wieder rekonstruiert. Dazu muß man das System von Kongruenzen

$$x \equiv a_1 \bmod m_1, \quad x \equiv a_2 \bmod m_2, \quad \ldots, \quad x \equiv a_k \bmod m_k$$

in der Unbekannten x lösen. Unsere bisherigen Überlegungen zeigen, daß dieses Gleichungssystem modulo m eindeutig lösbar ist, d.h. zwei Lösungen a und a' zu derselben Restklasse modulo m gehören.

Um eine Lösung zu finden, konstruieren wir Basislösungen $e_i, i = 1, \ldots, k$. e_i soll das Gleichungssystem in dem speziellen Fall $a_i = 1,\ a_j = 0$ für $j \neq i$

lösen. Setze

$$m_i' \ := \ \prod_{j \neq i} m_j \ = \ m/m_i.$$

Da die Modulen $m_1, \ldots, m_k$ paarweise teilerfremd sind, gilt $1 = \mathrm{ggT}(m_i, m_i')$. Mit dem erweiterten Euklidischen Algorithmus gewinnt man ganze Zahlen s_i, t_i, so daß

$$1 \ = \ m_i s_i + m_i' t_i.$$

Wir setzen nun

$$e_i \ := \ 1 - m_i s_i \ = \ m_i' t_i.$$

Dann gilt nach Definition von m_i' wie gewünscht

$$e_i \ \equiv \ \begin{cases} 1 \bmod m_i \\ 0 \bmod m_j \quad \text{für } j \neq i. \end{cases}$$

Aus den Basislösungen können wir eine Lösung

$$a \ = \ \sum_i a_i e_i$$

für das ursprüngliche Kongruenzensystem zusammensetzen. Insgesamt erhalten wir den **Chinesischen Restsatz** (der in einem speziellen Fall bereits Sun Tsu etwa 300 n.Chr. bekannt war).

Satz 2.4. *Seien $m_1, \ldots, m_k$ paarweise teilerfremde natürliche Zahlen und seien $a_1, \ldots, a_k$ ganze Zahlen. Dann existiert eine Lösung a des Systems von Kongruenzen*

$$x \equiv a_1 \bmod m_1, \ x \equiv a_2 \bmod m_2, \ \ldots \ , x \equiv a_k \bmod m_k,$$

und sie ist modulo $m = m_1 \cdots m_k$ eindeutig. Mit anderen Worten: Es gibt eine ganze Zahl a, so daß das Gleichungssystem äquivalent ist zu der Gleichung

$$x \equiv a \bmod m.$$

In Restklassen ausgedrückt bedeutet der Satz, daß die Zuordnung

$$a \bmod m \ \mapsto \ (a \bmod m_1, \ldots, a \bmod m_k)$$

eine Bijektion zwischen $\mathbb{Z}_m$ und $\mathbb{Z}_{m_1} \times \cdots \times \mathbb{Z}_{m_k}$ definiert. Die Eindeutigkeitsaussage bedeutet Injektivität der Abbildung, die Existenzaussage Surjektivität. Da beide Mengen $m = m_1 \cdots m_k$ Elemente enthalten, bedingen sich Injektivität und Surjektivität gegenseitig. So gesehen ist die Existenzaussage des Chinesischen Restsatzes eine Folgerung aus seiner Eindeutigkeitsaussage, und umgekehrt.

Beispiel. Gesucht ist eine Lösung des Systems

$$x \equiv 2 \bmod 3,\ x \equiv 3 \bmod 5,\ x \equiv 4 \bmod 7.$$

Es gilt $m_1' = 5 \cdot 7 = 35$, $m_2' = 3 \cdot 7 = 21$,$m_3' = 3 \cdot 5 = 15$. Aus

$$\begin{array}{rcrcl} 1 & = & \mathrm{ggT}(3,35) & = & 12 \cdot 3 - 1 \cdot 35 \\ 1 & = & \mathrm{ggT}(5,21) & = & -4 \cdot 5 + 1 \cdot 21 \\ 1 & = & \mathrm{ggT}(7,15) & = & -2 \cdot 7 + 1 \cdot 15 \end{array}$$

erhalten wir das Basissystem $e_1 = -35, e_2 = 21, e_3 = 15$. Es ist also

$$-2 \cdot 35 + 3 \cdot 21 + 4 \cdot 15 \ = \ 53,$$

Lösung, und das Gleichungssystem ist äquivalent zu $x \equiv 53 \bmod 105$. □

2.3 Ein probabilistischer Gleichheitstest

Zwei Personen an den Enden eines Nachrichtenkanals wollen zwei natürliche Zahlen $a, b < 2^{10.000}$ auf Gleichheit hin überprüfen. Um Übertragungsfehler zu vermeiden, möchten sie die Zahlen nicht vollständig übermitteln. Das folgende Verfahren erlaubt einen Vergleich der beiden Zahlen, dabei werden anstelle der 10.000 Bits einer Zahl nur $k \cdot 101$ Bits gesendet. Wir werden sehen, daß schon für $k = 1$ das Verfahren höchste Sicherheit garantiert.

Probabilistischer Gleichheitstest.

Ausgabe: ‚$a = b$‘ oder ‚$a \neq b$‘

Verfahren: Wähle zufällig Primzahlen $p_1, \ldots, p_k$ zwischen 2^{100} und 2^{101}. Übertrage a modulo p_i für alle $i = 1, \ldots, k$. Falls $a \not\equiv b \bmod p_i$ für ein i, gilt ‚$a \neq b$‘. Andernfalls treffe die Entscheidung ‚$a = b$‘. □

Das Verfahren setzt voraus, daß man sich die benötigten Primzahlen leicht verschaffen kann. Darauf kommen wir später zurück.

Wir schätzen nun die Wahrscheinlichkeit ab, daß das Verfahren zu einer Fehlentscheidung führt. Nehmen wir dazu zunächst an, daß es 100 verschiedene Primzahlen $q_1, \ldots, q_{100} > 2^{100}$ gibt, für die alle $a \equiv b \bmod q_i$ gilt. Nach dem chinesischen Restsatz folgt $a \equiv b \bmod m$, mit $m = q_1 \cdots q_{100}$. Es gilt $m > (2^{100})^{100} = 2^{10.000}$, nach Annahme folgt daher $a = b$. In diesem Fall ist ein Übertragungsfehler ausgeschlossen.

Eine Fehlentscheidung ist also nur möglich, falls es weniger als 100 Primzahlen $q > 2^{100}$ mit der Eigenschaft $a \equiv b \bmod q$ gibt, und eine Fehlentscheidung tritt nur dann ein, wenn das Verfahren zufälligerweise nur derartige Primzahlen auswählt. Nach dem berühmten Primzahlsatz gilt für die Anzahl $\pi(x)$ aller Primzahlen $q < x$ die asymptotische Formel $\pi(x) \sim x/\ln x$. Zwischen 2^{100} und 2^{101} gibt es daher approximativ

$$\pi(2^{101}) - \pi(2^{100}) \;\approx\; \frac{2^{101}}{\ln 2^{101}} - \frac{2^{100}}{\ln 2^{100}} \;\approx\; \frac{2^{100}}{100 \ln 2} \;\approx\; 9,90 \cdot 10^{27}$$

Primzahlen. Bei k-facher unabhängiger Wahl einer Primzahl ist die Fehlerwahrscheinlichkeit also höchstens

$$\left(\frac{99}{9,90 \cdot 10^{27}}\right)^k \;=\; 10^{-26 \cdot k}.$$

Schon für $k = 1$ ist dies verschwindend klein.

2.4 Exakte Lösung ganzzahliger Gleichungssysteme

Wir lösen nun lineare Gleichungsysteme mit modularer Arithmetik. Sei $A = (a_{ij})$ eine ganzzahlige $n \times n$ Matrix mit $\det A \neq 0$, und sei $b = (b_1, \ldots, b_n)^t$ ein ganzzahliger Vektor. Dann ist das lineare Gleichungssystem

$$Ax \;=\; b$$

mit $x = (x_1, \ldots, x_n)^t$ eindeutig in den rationalen Zahlen lösbar. Das bekannteste Lösungsverfahren ist das Gaußsche Eliminationsverfahren. Es kann in den Zwischenrechnungen zu so großen Werten führen, daß es nicht mehr in exakter Weise gelingt. Bei Verwendung von Gleitkomma-Arithmetik kann es sehr ungenaue Ergebnisse liefern.

Es liegt daher nahe, das Gaußsche Verfahren in modularer Arithmetik durchzuführen. Der Hauptanteil der arithmetischen Operationen kann so mit kleinen Zahlen vollzogen werden. Unser Ausgangspunkt ist die Cramersche Regel, nach der die Lösung durch

$$x_j \;=\; (\det A)^{-1} \sum_i b_i \det A_{ij}$$

gegeben ist; dabei bezeichnet A_{ij} die Matrix, die aus A durch Streichen der i-ten Zeile und der j-ten Spalte entsteht. Es genügt daher, $\det A$ und den

ganzzahligen Vektor $y = \det A \cdot x$ zu bestimmen. Man erhält y als eindeutige ganzzahlige Lösung des Gleichungssystems

$$Ay \;=\; \det A \cdot b.$$

Das Verfahren besteht aus mehreren Schritten.

1) Wähle Primzahlen $p_1, \ldots, p_k$ (und zwar an der Grenze der verfügbaren Zahlen: Hat der Computer Wortlänge 64, so wähle man die p_i zwischen 2^{63} und 2^{64}).

2) Löse des Gleichungssystems modulo p_i. Bringe dazu die erweiterte Koeffizientenmatrix (A, b) modulo p_i auf obere Dreiecksform:

$$(A, b) \;\mapsto\; (A'_i, b'_i) = \begin{pmatrix} a'_{11} & a'_{12} & \ldots & a'_{1n} & b'_1 \\ 0 & a'_{22} & \ldots & a'_{2n} & b'_2 \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & \ldots & 0 & a'_{nn} & b'_n \end{pmatrix}$$

Dies läßt sich erreichen, indem man das Gaußsche Eliminationsverfahren modulo p_i durchführt. Das Verfahren besteht bekanntlich aus der Addition von Vielfachen einer Matrixzeile zu einer anderen Zeile und dem Austausch von Zeilen. Diese Operationen lassen sich in Vektorräumen mit beliebigem Skalarbereich durchführen. In unserem Fall ist der Skalarbereich der Körper $\mathbb{Z}_{p_i}$. Die Addition einer Zeile zu einer anderen läßt $\det A$ unverändert, während Zeilenvertauschungen das Vorzeichen von $\det A$ wechseln. Für den Rest d_i von $a'_{11}a'_{22}\cdots a'_{nn}$ modulo p_i folgt also

$$\pm d_i \;\equiv\; \det A \mod p_i$$

(+ bei einer geraden, – bei einer ungeraden Anzahl von Zeilenvertauschungen). Eine Lösung $y^{(i)} = (y_1^{(i)}, \ldots, y_n^{(i)})$ des Gleichungssystems modulo p_i erhält man nun schrittweise aus den Gleichungen

$$\begin{aligned} y_n^{(i)} &\equiv \pm d_i b'_n / a'_{nn} \bmod p_i \\ &\vdots \\ y_1^{(i)} &\equiv (\pm d_i b'_1 - \sum_{j=2}^{n} a'_{1j} y_j^{(i)}) / a'_{11} \bmod p_i \end{aligned}$$

Für die Divisionen werden die inversen Elemente von $a'_{nn}, \ldots, a'_{11}$ in $\mathbb{Z}_{p_i}$ benötigt. Sie existieren, falls $d_i \not\equiv 0 \bmod p_i$. Der Ausnahmefall $d_i \equiv 0 \bmod p_i$ tritt nur für Primteiler p_i von $\det A$ ein, man verwerfe dann p_i.

3) Nur in diesem Schritt sind Rechnungen mit großen Zahlen erforderlich. Setze $m = p_1 \cdots p_k$. Bestimme nach dem Chinesischen Restsatz ganze Zahlen d und $y'_1, \ldots, y'_n$ zwischen $-m/2$ und $m/2$, so daß für alle i

$$d \equiv d_i \bmod p_i \text{ und } y'_j \equiv y_j^{(i)} \bmod p_i.$$

Dann folgt

$$\pm d \equiv \det A \bmod m \text{ und } y' \equiv \det A \cdot b \bmod m,$$

4) Die Zahl $\pm d$ und der Vektor y' sind Kandidaten für $\det A$ und y. Mit dem probabilistischen Gleichheitstest aus Abschnitt 3.3 kann man dies überprüfen. Man wählt dazu zufällig eine weitere Primzahl p_{k+1} und testet, ob $Ay' \equiv \pm d \cdot b \bmod p_{r+1}$ gilt. Bestehen $\pm d$ und y' diesen Test, so wird die Prozedur mit der

$$\textit{Ausgabe}: \quad x = \pm d^{-1} \cdot y'$$

beendet. Andernfalls wird die Prozedur mit der erweiterten Primzahlfolge $p_1, \ldots, p_{k+1}$ wiederholt. Die Fehlerwahrscheinlichkeit dieses Tests ist vernachlässigbar klein, ähnlich wie die des Gleichheitstests im letzten Abschnitt.

2.5 Ein allgemeiner Chinesischer Restsatz

Die Grundidee der letzten Abschnitte war es, die ganzen Zahlen so in Klassen aufzuteilen, daß man mit diesen Klassen in ähnlicher Weise wie mit den Zahlen rechnen kann. Man muß nur darauf achten, daß sich die Klassenbildung mit der Addition und Multiplikation ganzer Zahlen verträgt, im Sinne von Satz 2.2. Diese einfache Idee bewährt sich auch in anderen Rechenbereichen. Für Polynomringe wird sie sich später als nützlich erweisen, in diesem Abschnitt betrachten wir Restklassenbildung in beliebigen Ringen.

Sei $\equiv$ eine Äquivalenzrelation in einem Ring R. Die Äquivalenzklasse, in der a liegt, bezeichnen wir wieder mit $\overline{a}$. Wir setzen voraus, daß sich die Relation mit Addition und Multiplikation verträgt, daß also

$$a_1 \equiv a_2,\ b_1 \equiv b_2 \quad \Rightarrow \quad a_1 + b_1 \equiv a_2 + b_2,\ a_1 b_1 \equiv a_2 b_2 \tag{2.1}$$

gilt. Dann können wir erneut durch $\overline{a} + \overline{b} := \overline{a+b}$, $\overline{a} \cdot \overline{b} := \overline{ab}$ in wohldefinierter Weise Addition und Multiplikation von Äquivalenzklassen einführen. Die Äquivalenzklassen bilden einen Ring $\overline{R}$. Es ist offensichtlich, daß sich Assoziativ-, Distributiv- und Kommutativgesetze von R auf $\overline{R}$ übertragen,

wir haben dies bereits für den Restklassenring $\mathbb{Z}_m$ festgestellt. Das Nullelement in $\overline{R}$ ist die Restklasse $\overline{0}$, denn $\overline{a} + \overline{0} = \overline{a}$, und aus $a \equiv b$ folgt $-a \equiv (-a) + b + (-b) \equiv (-a) + a + (-b) \equiv -b$, so daß $-\overline{a} := \overline{-a}$ wohldefiniert ist. Besitzt R ein Einselement 1, so ist $\overline{1}$ Einselement in $\overline{R}$.

Wir wollen nun die Äquivalenzrelationen charakterisieren, die (2.1) erfüllen. Es gilt $a \equiv b$ genau dann, wenn $a - b \equiv 0$, daher ist die Äquivalenzrelation bereits durch die Teilmenge $I := \overline{0}$ von R eindeutig gekennzeichnet. Als Nullelement von $\overline{R}$ hat I folgende Eigenschaften:

$$\begin{array}{ll} a, b \in I \;\Rightarrow\; a + b \in I, & \text{denn } a \equiv 0, b \equiv 0 \;\Rightarrow\; a + b \equiv 0 + 0 = 0, \\ a \in I \;\Rightarrow\; -a \in I, & \text{denn } a \equiv 0 \;\Rightarrow\; -a \equiv -0 = 0, \\ a \in I, b \in R \;\Rightarrow\; ab, ba \in I, & \text{denn } a \equiv 0 \;\Rightarrow\; ab \equiv 0b = 0. \end{array}$$

Definition 2.5. *Eine nichtleere Teilmenge I eines Ringes R heißt **Ideal**, falls gilt:*

$$\begin{array}{lcl} a, b \in I & \Rightarrow & a + b \in I, \\ a \in I & \Rightarrow & -a \in I \\ a \in I, b \in R & \Rightarrow & ab,\ ba \in I. \end{array}$$

Ist umgekehrt I ein Ideal, so wird, wie man leicht nachprüft, in R durch $a \equiv b \;:\Leftrightarrow\; a - b \in I$ eine Äquivalenzrelation erklärt. Wir schreiben dann

$$a \equiv b \bmod I$$

und nennen a und b **kongruent modulo I**. Die Relation ist mit Addition und Multiplikation verträglich, denn aus $a_1 - b_1,\ a_2 - b_2 \in I$ folgt $a_1 + a_2 - (b_1 + b_2) \in I$ und $a_1 a_2 - b_1 b_2 = a_1(a_2 - b_2) + (a_1 - b_1)b_2 \in I$. Insgesamt erkennen wir, daß eine eineindeutige Beziehung besteht zwischen Idealen und denjenigen Äquivalenzrelationen, die (2.1) erfüllen. Die Äquivalenzklassen nennt man **Restklassen**. Sie sind gegeben durch $\overline{a} = a + I$, $a \in R$, wir schreiben sie auch als $a \bmod I$. Der Ring der Restklassen wird als **Faktorring** R/I bezeichnet. Speziell erhalten wir bei der Wahl $R = \mathbb{Z}$ und $I = m\mathbb{Z}$ den Restklassenring $\mathbb{Z}_m$.

Die Abbildung $a \mapsto \overline{a}$ von R nach R/I ist, wie wir gesehen haben, mit Addition und Multiplikation verträglich. Man spricht von einem Homomorphismus.

Definition 2.6. *Seien R, R' Ringe. Eine Abbildung $\phi : R \to R'$ heißt **Ring-Homomorphismus**, falls $\phi(a+b) = \phi(a) + \phi(b)$ und $\phi(ab) = \phi(a) \cdot \phi(b)$ für alle $a, b \in R$ gilt. Ist ϕ bijektiv, so spricht man von einem **Isomorphismus** und schreibt $R \cong R'$.*

Ein Homomorphismus ϕ von R nach R' induziert seinerseits in R eine Äquivalenzrelation: $a \equiv b :\Leftrightarrow \phi(a) = \phi(b)$. Hier entsprechen Restklassen Elementen aus R', das zugehörige Ideal ist $\ker(\phi) := \{a \in R \,:\, \phi(a) = 0\}$, der Kern der Abbildung. Wir erhalten so den **Homomorphiesatz für Ringe**.

Satz 2.7. *Sei $\phi : R \to R'$ ein Homomorphismus, dann ist* $\ker(\phi)$ *ein Ideal und* $R/_{\ker(\phi)} \cong \text{bild}(\phi) := \{\phi(a) \,:\, a \in R\}$.

Wir beweisen nun eine allgemeine Version des Chinesischen Restsatzes. Dazu vereinbaren wir die folgende Sprechweise: Zwei Ideale I_1, I_2 heißen **teilerfremd**, falls

$$R \;=\; I_1 + I_2$$

gilt, mit $I_1 + I_2 \;:=\; \{r + s \,:\, r \in I_1, s \in I_2\}$.

Satz 2.8. *Sei R ein Ring mit Einselement 1 und seien $I_1, \ldots, I_k \subset R$ paarweise teilerfremde Ideale. Dann gibt es zu $a_1, \ldots, a_k \in R$ ein $a \in R$, so daß das Kongruenzensystem*

$$x \equiv a_1 \bmod I_1, \ldots, x \equiv a_k \bmod I_k$$

äquivalent ist zu der Kongruenz

$$x \equiv a \bmod I.$$

Dabei bezeichnet I das Ideal $I_1 \cap \ldots \cap I_k$.

Beweis. Wir gehen wie im ganzzahligen Fall vor. Da nach Voraussetzung $I_i + I_j = R$ für $i \neq j$, gibt es $b_{ij} \in I_i, c_{ij} \in I_j$, so daß $b_{ij} + c_{ij} = 1$. Setze

$$e_i \;:=\; \prod_{j \neq i} c_{ij} \;=\; \prod_{j \neq i} (1 - b_{ij}).$$

Dann gilt $e_i \equiv 0 \mod I_j$ für $i \neq j$ und $e_i \equiv 1 \mod I_i$. Folglich löst $a = \sum_i a_i e_i$ das Gleichungssystem. Ist a' eine weitere Lösung des Systems, so gilt $a' - a \in I_i$ für alle i, d.h. $a' - a \in I$ bzw. $a' \equiv a \mod I$. □

Der Satz läßt sich zu einer Isomorphieaussage von Ringen umformulieren. Wir benötigen dazu den Begriff des direkten Produktes von Ringen. Seien $R_1, \ldots, R_k$ Ringe und $R = R_1 \times \cdots \times R_k$ ihr kartesisches Produkt. Durch die Vereinbarung

$$\begin{aligned} (a_1, \ldots, a_k) + (b_1, \ldots, b_k) &:= (a_1 + b_1, \ldots, a_k + b_k), \\ (a_1, \ldots, a_k) \cdot (b_1, \ldots, b_k) &:= (a_1 b_1, \ldots, a_k b_k) \end{aligned}$$

wird R zum Ring. R heißt das **direkte Produkt** von $R_1, \ldots, R_k$.

Wir betrachten nun die Abbildung

$$a \bmod I \;\mapsto\; (a \bmod I_1, \ldots, a \bmod I_k)$$

von R/I nach $R/I_1 \times \cdots \times R/I_k$. Sie ist wohldefiniert, denn $a \bmod I = b \bmod I$ bzw. $b - a \in I$ impliziert $b - a \in I_i$ bzw. $a \bmod I_i = b \bmod I_i$ für alle i. Umgekehrt folgt $a \bmod I = b \bmod I$ aus $a \bmod I_i = b \bmod I_i$ für alle i, die Abbildung ist also injektiv. Nach dem letzten Satz gibt es zu $a_1, \ldots, a_k$ ein a, so daß $(a_1 \bmod I_1, \ldots, a_k \bmod I_k) = (a \bmod I_1, \ldots, a \bmod I_k)$. Daher ist die Abbildung surjektiv. Schließlich ist sie offenbar mit Addition und Multiplikation verträglich. Wir können daher den Chinesischen Restsatz auch als Aussage über die Isomorphie von Ringen ansehen,

$$R/\bigcap_{i=1}^{k} I_i \;\cong\; R/I_1 \times \cdots \times R/I_k.$$

Zum Abschluß gehen wir auf den Begriff Hauptideal ein.

Definition 2.9. *Sei R ein Integritätsbereich. Ein Ideal I heißt **Hauptideal**, wenn es ein $m \in R$ gibt, so daß $I = mR := \{mr \;:\; r \in R\}$. Dieses von m erzeugte Ideal wird mit (m) bezeichnet. Ist jedes Ideal in R ein Hauptideal, so heißt R **Hauptidealring**.*

Für ein Hauptideal $I = mR$ gilt $a \equiv b \mod I$ genau dann, wenn $m \mid b - a$, dies entspricht völlig der Situation in $\mathbb{Z}$.

Proposition 2.10. *Jeder Euklidische Ring R ist ein Hauptidealring.*

Beweis. Das Nullideal $\{0\} = 0 \cdot R$ ist ein Hauptideal. Sei also $I \subset R$ ein Ideal ungleich $\{0\}$. Wähle $m \in I - \{0\}$ derart, daß $g(m)$ auf $I - \{0\}$ minimal ist. Für jedes $b \in I$ gibt es dann $s, r \in R$, so daß $b = sm + r$, mit $r = 0$ oder $g(r) < g(m)$. Es folgt $r \in I$, nach Wahl von m kann also nur $r = 0$ gelten. Daher gilt $b = sm$ und $I = mR$. □

Von einem algebraischen Standpunkt unterscheiden sich Hauptidealringe nicht wesentlich von Euklidischen Ringen. So existieren in einem Hauptidealring R zu Elementen a und b immer ein größter gemeinsamer Teiler. Zum Beweis bilden wir das Ideal

$$I \;:=\; \{as + bt \;:\; s, t \in R\}.$$

Nach Annahme ist es von der Gestalt $I = dR$ für ein $d \in R$. Wegen $a, b \in I$ ist d Teiler von a und b. Andererseits gibt es $s, t \in R$, so daß $d = as + bt$. Jeder Teiler von a und b teilt daher auch d. d ist also ein größter gemeinsamer

Teiler von a und b. Dieser Beweis bestätigt auch die Gültigkeit des Satzes von Bézout in Hauptidealringen. Ist d' ein weiterer ggT von a und b, so gibt es $e, e' \in R$, so daß $d = e'd'$ und $d' = ed$. Es folgt $d = e'ed$ und durch Kürzen $1 = e'e$. Daher sind e und e' Einheiten. Ein ggT ist also bis auf Einheiten eindeutig festgelegt. Schließlich stellen wir noch fest, daß $R = aR + bR$ genau dann gilt, wenn $1 = \mathrm{ggT}(a, b)$, wenn also a und b teilerfremd sind. Teilerfremdheit von Idealen, wie oben eingeführt, hat in Hauptidealringen daher die übliche Bedeutung.

Von einem algorithmischen Standpunkt sind dagegen Euklidische Ringe und Hauptidealringe wesentlich verschieden. Der Euklidische Algorithmus steht in Hauptidealringen nicht zur Verfügung.

2.6 Prime Restklassen

Wie wir früher festgestellt haben, ist eine Restklassen $\overline{a}$ in $\mathbb{Z}_m$ entweder Einheit oder Nullteiler. Wir untersuchen nun die Gruppe der Einheiten.

Definition 2.11. *Sei $a \in \mathbb{Z}$. Die Restklasse $\overline{a} \in \mathbb{Z}_m - \{0\}$ heißt* ***prime Restklasse*** *modulo m, falls $1 = \mathrm{ggT}(a, m)$. Die Menge der primen Restklassen modulo m wird mit $\mathbb{Z}_m^*$ bezeichnet, ihre Anzahl mit $\phi(m)$. ϕ heißt* ***Eulersche ϕ-Funktion****.*

Es gilt $\mathbb{Z}_m^* = \mathbb{Z}_m - \{0\}$ und $\phi(m) = m - 1$ genau dann, wenn m prim ist. Wir zeigen nun, daß $\mathbb{Z}_m^*$ bzgl. der Restklassenmultiplikation eine Gruppe ist.

Satz 2.12.

i) $\overline{a}, \overline{b} \in \mathbb{Z}_m^* \quad \Rightarrow \quad \overline{a} \cdot \overline{b} \in \mathbb{Z}_m^*$.

ii) Jedes $\overline{a} \in \mathbb{Z}_m^*$ besitzt ein inverses Element $\overline{a}^{-1} \in \mathbb{Z}_m^*$.

iii) Die Gleichung $\overline{a} \cdot \overline{x} = \overline{b}$ ist in $\mathbb{Z}_m^*$ eindeutig lösbar.

Beweis. Zu *i)*: Aus $1 = \mathrm{ggT}(a, m) = \mathrm{ggT}(b, m)$ folgt $1 = \mathrm{ggT}(ab, m)$. Zu *ii)*: Wegen $1 = \mathrm{ggT}(a, m)$ gibt es nach dem Satz von Bézout ganze Zahlen r, s, so daß $ar + ms = 1$. Es folgt $1 = \mathrm{ggT}(r, m)$ und $ar \equiv 1 \bmod m$, also $\overline{r} = \overline{a}^{-1}$. Zu *iii)*: $\overline{x} = \overline{a}^{-1} \cdot \overline{b}$. □

Von fundamentaler Bedeutung ist das **Theorem von Euler**. Wir schreiben wie üblich

$$\overline{a}^{\,n} \;=\; \underbrace{\overline{a} \cdots \overline{a}}_{n\text{-mal}}.$$

Satz 2.13. *Für alle $\overline{a} \in \mathbb{Z}_m^*$ gilt*

$$\overline{a}^{\,\phi(m)} \;=\; \overline{1}.$$

Anders ausgedrückt: $\overline{a}^{\,-1} = \overline{a}^{\,\phi(m)-1}$.

Beweis. Seien $\overline{a}_1, \ldots, \overline{a}_{\phi(m)}$ alle Restklassen in $\mathbb{Z}_m^*$. Dann durchlaufen auch $\overline{a} \cdot \overline{a}_1, \ldots, \overline{a} \cdot \overline{a}_{\phi(m)}$ alle Restklassen in $\mathbb{Z}_m^*$, denn aus $\overline{a} \cdot \overline{a}_i = \overline{a} \cdot \overline{a}_j$ folgt durch Multiplikation mit $\overline{a}^{\,-1}$ die Gleichung $\overline{a}_i = \overline{a}_j$ und damit $i = j$. Daher gilt

$$\overline{a}_1 \cdots \overline{a}_{\phi(m)} = \overline{a}^{\,\phi(m)} \cdot \overline{a}_1 \cdots \overline{a}_{\phi(m)}$$

und, indem wir mit $\overline{a}_i^{-1}$, $i = 1, \ldots, \phi(m)$, multiplizieren,

$$\overline{a}^{\,\phi(m)} = \overline{1}.$$

□

Im Fall eines primen Moduls geht diese Aussage auf Fermat zurück.

Korollar 2.14. (Theorem von Fermat) *Sei p eine Primzahl. Dann gilt für jede ganze Zahl a*

$$a^p \equiv a \bmod p.$$

Beweis. Für $a \equiv 0 \bmod p$ ist die Aussage offensichtlich, und für $a \not\equiv 0$ gilt wegen $\phi(p) = p - 1$

$$a^{p-1} \equiv 1 \bmod p.$$

□

Die Eulersche ϕ-Funktion läßt sich leicht berechnen, vorausgesetzt wir kennen die Primfaktorzerlegung von m.

Proposition 2.15. *Seien $p_1, \ldots, p_k$ die unterschiedlichen Primteiler von m. Dann gilt*

$$\phi(m) \;= m \prod_{i=1}^{k} \left(1 - \frac{1}{p_i}\right).$$

Beweis. Ist $m = p^r$ mit primem p, so sind genau die Zahlen $0, p, 2p, \ldots, (p^{r-1} - 1)p$ Vertreter von Restklassen, die nicht zu $\mathbb{Z}_m^*$ gehören. Also:

$$\phi(p^r) \;=\; p^r - p^{r-1} \;=\; p^r \left(1 - \frac{1}{p}\right).$$

Sei nun $m = m_1 \cdots m_k$ mit paarweise teilerfremden Zahlen $m_1, \ldots, m_k$. Dann gilt $1 = \mathrm{ggT}(a, m)$ genau dann, wenn $1 = \mathrm{ggT}(a, m_i)$ für alle i. Nach dem Chinesischen Restsatz ist daher

$$a \bmod m \;\mapsto\; (a \bmod m_1, \ldots, a \bmod m_k)$$

eine Bijektion von $\mathbb{Z}_m^*$ nach $\mathbb{Z}_{m_1}^* \times \cdots \times \mathbb{Z}_{m_k}^*$. Es folgt

$$\phi(m) \;=\; \phi(m_1) \cdots \phi(m_k).$$

□

2.7 Ein probabilistischer Primzahltest

Für den probabilistischen Gleichheitstest in Abschnitt 2.3 wie für Anwendungen der modularen Arithmetik in den folgenden Abschnitten benötigt man ein effektives Verfahren zur Gewinnung großer Primzahlen. Nach dem Primzahlsatz ist in dem Intervall $[n, 2n]$ im Mittel jede $(\ln n)$-te Zahl eine Primzahl. Durch zufällige Wahl von Zahlen aus dem Intervall wird man also ausreichend oft auf Primzahlen treffen. Um sie zu erkennen, benötigt man einen effektiven Primzahl-Test. Das folgende probabilistische Verfahren testet, ob die Zahl m die Fermatsche Identität $a^{m-1} \equiv 1 \bmod m$ erfüllt. Nach dem Fermatschen Theorem genügen Primzahlen dieser Identität, sie sind aber nicht die einzigen Zahlen.

Definition 2.16. *Die ganze Zahl m heißt **Carmichael-Zahl**, falls m nicht prim ist, und falls für alle $a \in \mathbb{Z}_m^*$ gilt*

$$a^{m-1} \equiv 1 \bmod m.$$

Die kleinste Carmichael-Zahl ist $561 = 3 \cdot 11 \cdot 17$. Computerberechnungen haben gezeigt, daß sie sehr viel seltener sind als Primzahlen. Lehmer und Poulet haben festgestellt, daß es unter den ersten 100.000.000 Zahlen 5.761.455 Primzahlen gibt, aber nur 252 Carmichael-Zahlen. – Zahlen die weder prim noch Carmichael-Zahlen sind, lassen sich mit ausreichender Sicherheit erkennen.

Proposition 2.17. *Ist m weder prim noch Carmichael-Zahl, so gilt*

$$\mathrm{card}\,\{a \in \mathbb{Z}_m^* \;:\; a^{m-1} \equiv 1 \bmod m\} \leq \phi(m)/2.$$

Beweis. Seien $a_1, \ldots, a_k$ alle Elemente aus $\mathbb{Z}_m^*$ mit $a_i^{m-1} \equiv 1 \bmod m$. Nach Voraussetzung gibt es ein $b \in \mathbb{Z}_m^*$ mit $b^{m-1} \not\equiv 1 \bmod m$. Dann gilt für $b_i = ba_i$

$$b_i^{m-1} = a_i^{m-1} b^{m-1} \not\equiv 1 \bmod m,$$

es gibt daher mindestens ebenso viele Elemente in $\mathbb{Z}_m^*$, die die Fermatsche Identität nicht erfüllen. Also gilt $k \leq \phi(m)/2$. □

Ein probabilistischer Primzahltest.

Eingabe: Eine natürliche Zahl m.

Ausgabe: ‚m ist prim' oder ‚m ist nicht prim'.

Verfahren: Wähle rein zufällig Zahlen $a_1, \ldots, a_k < m$, die teilerfremd zu m sind. Gilt $a_i^{m-1} \equiv 1 \bmod m$ für alle $i = 1, \ldots, k$, so entscheide ‚m ist prim'. Andernfalls gilt ‚m ist nicht prim'. □

Dieser Test erkennt sicher Primzahlen. Ein Zahl, die weder prim noch Carmichal-Zahl ist wird mit ausreichender Sicherheit erkannt, die Fehlerwahrscheinlichkeit ist kleiner als 2^{-k}. Nur Carmichael-Zahlen bleiben unerkannt. Praktisch ist dies für das Verfahren bedeutungslos, weil Carmichael-Zahlen so überaus selten auftreten. – Ein raffinierteres Verfahren von Rabin beseitigt diese letzte Unsicherheit (vgl. die Monographie von Knuth Vol. II, 3. Ausgabe, S. 395).

2.8 Öffentliche Chiffriersysteme

Die **Kryptographie** ist die Lehre von den Chiffriersystemen. Stellen wir uns vor, daß eine Person A eine geheime Nachricht an die Person B übermitteln möchte. A kodiert sie deswegen mittels einer bijektiven Kodierabbildung

$$\kappa : \mathcal{N} \to \mathcal{K}.$$

Anstelle der Nachricht $a \in \mathcal{N}$ im Klartext sendet A die chiffrierte Nachricht $\kappa(a)$. Der Empfänger dekodiert mittels der inversen Abbildung

$$\kappa^{-1} : \mathcal{K} \to \mathcal{N}.$$

Ein bekanntes Verfahren beruht auf der Addition modulo 2. Wir nehmen an, daß die Nachrichten als 01-Folgen der Länge n vorliegen, wählen also $\mathcal{N} = \mathcal{K} = \{0,1\}^n$. Wir fassen die Folgen als Vektoren der Läne n über dem Körper $\mathbb{Z}_2$ auf. Zum Kodieren wird ein 01-String $k = k_1 k_2 \ldots k_n$ verwendet. Wir setzen

$$\kappa(a) \;:=\; a + k,$$

die beiden 01-Folgen a und k werden also komponentenweise modulo 2 addiert. Es gilt $\kappa^{-1} = \kappa$, denn $k + k$ ist die nur aus Nullen bestehende Folge. Nachrichten werden daher nach demselben Verfahren dekodiert. Dieses klassische Verfahren hat den Namen ‚One-time-pad'. Es gilt $a + \kappa(a) = k$, gelingt es daher, eine kodierte Nachricht $k(a)$ zu entschlüsseln, so kennt man k und

damit bereits die vollständige Kodier- und Dekodiervorschrift. Dies bedeutet, daß das Verfahren sicher ist, man hat keine Chance, eine Nachricht zu dechiffrieren, wenn man auf k keinen Zugriff hat. Ist k rein zufällig aus $\{0,1\}^n$ gewählt, so gilt dies auch für $\kappa(a)$.

Das RSA-Schema

Für die klassischen Chiffrierverfahren besteht ein Sicherheitsproblem darin, daß man nicht nur die Dekodiervorschrift κ^{-1} geheimhalten muß, sondern auch das Kodierverfahren κ. Wie das Beispiel des One-time-pad zeigt, lassen sich die Nachrichten mit Hilfe der Chiffrier- wie der Dechiffriervorschrift leicht entschlüsseln. Diffie und Hellman haben daher 1976 einen (damals beherzten) Vorschlag gemacht: Man solle Kodierabbildungen κ benutzen, für die $b = \kappa(a)$ aus a leicht berechnet werden kann, umgekehrt aber die Berechnung von $a = \kappa^{-1}(b)$ aus b typischerweise mit einem immensen Rechenaufwand verbunden ist, der praktisch nicht bewältigt werden kann (selbst wenn einem die Abbildung κ bekannt ist!). In dieser asymmetrischen Situation darf man das Chiffrierverfahren öffentlich bekannt machen, ohne die Sicherheit zu gefährden. Dies ist die Grundidee der modernen, computergestützten **öffentlichen Kodiersysteme**. Theoretisch ist es schwer zu begründen, daß solche Abbildungen κ, man spricht von **Einweg- (one-way) Abbildungen**, wirklich existieren. Für praktische Zwecke haben sich jedoch verschiedene Abbildungen als brauchbar erwiesen. Wir werden Abbildungen betrachten, bei denen das Dekodieren erst dann praktisch durchführbar ist, wenn man über einen zusätzlichen ‚Schlüssel' verfügt. Dann spricht man von **Falltür- (trapdoor) Abbildungen**.

Zum Beispiel kann man schnell modulo m potenzieren:

$$a^{20} = a^{16+4} = (((a^2)^2)^2)^2 \cdot (a^2)^2.$$

Allgemein berechnet man a^c modulo m, indem man den Exponenten als binäre Zahl darstellt, $c = d_0 + d_1 \cdot 2 + \cdots + d_k \cdot 2^k$ mit $d_i \in \{0,1\}$, in k Schritten rekursiv die Potenzen $a^{2^i} = (a^{2^{i-1}})^2$ modulo m berechnet und schließlich diejenigen Potenzen modulo m multipliziert, für die $d_i = 1$ gilt. Die Anzahl der Operationen ist von der Ordnung $O(k) = O(\log c)$. – Eine allgemeine Methode, mit der man schnell (schneller als Durchprobieren) die Gleichung $b \equiv a^x \bmod m$ nach x auflöst (‚diskreter Logarithmus'), kennt man dagegen nicht.

Wir beschreiben nun das bekannteste öffentliche Chiffriersystem, das von Rivest, Shamir und Adleman 1978 vorgeschlagene **RSA-System**. Es baut darauf auf, daß es schwer ist, eine Zahl m in ihre Primfaktoren zu zerlegen.

- Als Nachrichtenmenge $\mathcal{N} = \mathcal{K}$ wird $\mathbb{Z}_m$ gewählt. Dabei sei $m = pq$ das Produkt zweier sehr großer Primzahlen p und q. Wir gehen also davon aus, daß die Nachricht als natürliche Zahl $a < m$ dargestellt ist.

- Zum Kodieren wird eine natürliche Zahl $s < \phi(m)$ gewählt, die teilerfremd zu $\phi(m)$ ist. Die Kodierung der Nachricht $a \in \mathcal{N}$ ist dasjenige $\kappa(a) \in \mathcal{N}$, so daß
$$\kappa(a) \equiv a^s \bmod m.$$

- Zum Dekodieren benötigt man einen Schlüssel, eine natürliche Zahl $t < \phi(m)$ mit der Eigenschaft
$$s \cdot t \equiv 1 \bmod \phi(m).$$
bzw. $st = 1 + k\phi(m) = 1 + k(p-1)(q-1)$ für eine natürliche Zahl k. Eine solche Zahl t existiert, denn nach Wahl von s gilt $\overline{s} \in \mathbb{Z}^*_{\phi(m)}$. Gilt nun $a \not\equiv 0 \bmod p$, so folgt nach dem Satz von Euler $a^{p-1} \equiv 1 \bmod p$, also
$$(a^s)^t \equiv a \cdot (a^{p-1})^{k(q-1)} \equiv a \bmod p.$$
Offenbar gilt die Behauptung auch für $a \equiv 0 \bmod p$. Genauso folgt $(a^s)^t \equiv a \bmod q$ für alle a. Nach dem chinesischen Restsatz folgt
$$(a^s)^t \equiv a \bmod m$$
für alle a, und es folgt
$$\kappa^{-1}(b) \equiv b^t \bmod m. \qquad \square$$

Man darf dieses Verfahren so einrichten, daß die Nachrichtenmenge $\mathcal{N}$, die Zahl s und auch die kodierte Nachricht $\kappa(a)$ öffentlich zugänglich sind. Nur der Schlüssel t muß geheim bleiben. Bei Kenntnis von $\phi(m)$ läßt sich t natürlich mühelos mit Euklids Algorithmus aus s errechnen. Daher muß dafür gesorgt sein, daß $\phi(m)$ praktisch nicht berechenbar ist. Diese Aufgabe ist aber im wesentlichen so aufwendig wie das Zerlegen von m in seine beiden Primfaktoren. Einerseits gilt nämlich
$$\phi(m) = (p-1)(q-1),$$
andererseits kann man m bei Kenntnis von $\phi(m)$ sofort faktorisieren: p und q erhält man aus den Gleichungen
$$\begin{aligned} p+q &= m - \phi(m) + 1, \\ p-q &= \pm\sqrt{(p+q)^2 - 4pq} \\ &= \pm\sqrt{(m-\phi(m)+1)^2 - 4m}. \end{aligned}$$

Das RSA-System steht und fällt damit, daß die Primfaktoren p und q geheim bleiben. Die Erfahrung zeigt, daß die Faktorisierung von m einen hohen Rechenaufwand erfordert und praktisch nicht mehr gelingt, wenn p und q ausreichend groß sind. Dies ist ein *Grundpostulat der Kryptographie*: Man geht davon aus, daß es keinen schnellen Algorithmus zum Zerlegen einer Zahl in seine Primfaktoren gibt. Man nennt deshalb in der Kryptographie ein Chiffriersystem ‚beweisbar sicher', wenn es letztlich auf die Faktorisierung großer Zahlen zurückgeführt werden kann.

Signatur von Nachrichten

Das RSA-Schema eignet sich gut zum geheimen Austausch von Nachrichten innerhalb einer Gruppe von Teilnehmern. Jeder Teilnehmer A erhält von einer Zentrale einen öffentlichen Schlüssel $(m(A), s(A))$ und seinen persönlichen geheimen Schlüssel $t(A)$. A teilt B eine geheime Nachricht a als $b := \kappa_B(a) \equiv a^{s(B)} \bmod m(B)$ mit, B dekodiert mittels $\kappa_B^{-1}(b) \equiv b^{t(B)} \bmod m(B)$. Ein vorheriger Kontakt zwischen A und B ist nicht erforderlich, A braucht lediglich den öffentlich zugänglichen Schlüssel $s(B)$ von B. Ein wichtiger Vorteil des Systems ist, daß es den Teilnehmern erlaubt, Mitteilungen fälschungssicher zu signieren (‚Unterschriftensystem'). A beglaubigt eine öffentliche Nachricht a durch die öffentliche Mitteilung von $c = \kappa_A^{-1}(a) \equiv a^{t(A)} \bmod m(A)$. Alle anderen Teilnehmer können verifizieren, daß $a \equiv c^{s(A)} \bmod m(A)$ gilt.

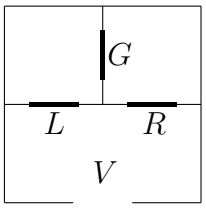
Randomisiertes RSA

Um ein einzelnes bit $a = 0$ oder 1 geheim zu übertragen, muß man das RSA-Schema modifizieren. Dann gilt $a^s = a$, und die Nachricht bleibt unverschlüsselt. Stattdessen kann man so vorgehen: Man wähle rein zufällig eine gerade Zahl $2u$ aus $\{0, 1, \ldots, (m-1)\}$ und übertragt anstelle von a mit dem RSA-Schema die Zahl $2u + a$, die verschlüsselte Nachricht ist also $(2u + a)^s$. Der Empfänger kann mit seinem geheimen Schlüssel die Nachricht als $2u + a$ dekodieren und a als das letzte bit dieser Nachricht zurückgewinnen. Alexi, Chor, Goldreich und Schnorr haben gezeigt, daß dieses Verfahren genauso sicher wie das RSA-Schema ist.

2.9 Zero-Knowledge Beweise

Eine Person möchte beweisen (und sich dadurch ausweisen), daß sie ein Geheimnis (eine Geheimzahl) kennt, ohne von ihrem Wissen irgend etwas preiszugeben. Wie ist das möglich?

Um die Idee des Verfahrens zu erläutern, betrachten wir das folgende Szenario: Die Person A möchte einer zweiten Person B demonstrieren, daß sie den Schlüssel zu einer Geheimtür G besitzt, die zwei Räume verbindet. Den Schlüssel möchte A nicht vorzeigen, ja nicht einmal einen Blick in die Räume gestatten. Deswegen einigen sich A und B auf folgendes Vorgehen: A geht durch den Vorraum V und dann rein zufällig entweder durch die Tür R in den rechten Raum, oder die Tür L in den linken Raum.



Nun betritt B den Vorraum. Sie weiß nicht, in welchem der beiden Räume sich A befindet. Sie wählt zufällig eine der Türen R oder L aus und fordert A auf, durch diese Tür wieder herauszukommen. Dies ist A immer möglich, wenn sie die Geheimtür öffnen kann. Andernfalls besteht sie den Test nur mit Wahrscheinlichkeit 1/2, und bei r-facher unabhängiger Wiederholung sogar nur mit Wahrscheinlichkeit 2^{-r}. Auch für kleines r bietet dieses Verfahren schon hohe Sicherheit.

Der Fiat-Shamir-Algorithmus

Nach einem verwandten Muster funktioniert der Algorithmus, den Fiat und Shamir 1988 vorgeschlagen haben, und bei dem der Schlüssel eine geheime Zahl ist. Der Algorithmus beruht darauf, daß es sehr aufwendig ist, die Wurzel eines quadratischen Restes modulo einem Primzahlprodukt $m = pq$ zu berechnen. Zuerst erläutern wir diesen Sachverhalt. a heißt **quadratischer Rest modulo m**, falls die Kongruenz

$$x^2 \equiv a \bmod m \tag{2.2}$$

eine Lösung b besitzt, wenn also die Restklassengleichung $\overline{b}^2 = \overline{a}$ gilt. $\overline{b}$ nennen wir dann **Wurzel** von $\overline{a}$.

Beispiel. $\mathbb{Z}_{15}^* = \{\overline{1}, \overline{2}, \overline{4}, \overline{7}, \overline{8}, \overline{11}, \overline{13}, \overline{14}\}$ enthält die quadradischen Reste $\overline{1}$ und $\overline{4}$:

$$\begin{array}{ccccccccc} \overline{1}^2 & = & \overline{4}^2 & = & \overline{11}^2 & = & \overline{14}^2 & = & \overline{1}, \\ \overline{2}^2 & = & \overline{7}^2 & = & \overline{8}^2 & = & \overline{13}^2 & = & \overline{4}. \end{array}$$

Die quadratischen Reste sind $\overline{1}$ und $\overline{4}$, sie besitzen beide vier verschiedene Wurzeln. □

Wir zeigen nun, daß allgemein quadratische Reste vier verschiedene Wurzeln haben für Moduln $m = pq$ mit zwei Primzahlen $p, q > 2$. Die Gleichung $\overline{x}^2 = \overline{a} = \overline{b}^2$ in $\mathbb{Z}_m^*$ ist dann äquivalent zu der Aussage $pq \mid (x-b)(x+b)$. Daher teilt p entweder $x-b$ oder $x+b$, und ebenso q entweder $x-b$ oder $x+b$. Es sind vier Fälle möglich:

$$\begin{array}{lll} x \equiv b \bmod p, & x \equiv b \bmod q, & \text{oder} \\ x \equiv -b \bmod p, & x \equiv -b \bmod q, & \text{oder} \\ x \equiv b \bmod p, & x \equiv -b \bmod q, & \text{oder} \\ x \equiv -b \bmod p, & x \equiv b \bmod q. & \end{array}$$

Nach dem Chinesischen Restsatz sind alle Fälle eindeutig lösbar, und die Lösungen sind voneinander verschieden. Die beiden ersten Kongruenzen haben offenbar die Lösungen b und $-b$, die beiden anderen Kongruenzen führen zu weiteren Lösungen c und $-c$. b und c unterscheiden sich nicht nur durch das Vorzeichen, sie heißen **wesentlich verschiedene Lösungen**.

Wir begründen nun, daß es genauso rechenintensiv ist, (2.2) im Fall $m = pq$ zu lösen (selbst wenn bekannt ist, daß a quadratischer Rest ist), wie m in seine Primfaktoren zu zerlegen. Nehmen wir an, daß wir über einen Algorithmus verfügen, der zu jedem quadratischen Rest modulo m eine Wurzel berechnet. Wir können dann auch leicht zwei wesentlich verschiedene Wurzeln eines quadratischen Restes finden. Dazu wähle man rein zufällig ein $\overline{b}$ aus $\mathbb{Z}_m^*$ und bilde daraus den quadratischen Rest $\overline{a} = \overline{b}^2$. Mit dem Algorithmus erhalten wir eine Wurzel $\overline{c}$ von $\overline{a}$, die mit Wahrscheinlichkeit $1/2$ wesentlich verschieden von $\overline{b}$ ist (im gegenteiligen Fall wähle zufällig ein neues $\overline{b}$). In $\mathbb{Z}_m^*$ gilt dann $(\overline{b}+\overline{c})(\overline{b}-\overline{c}) = \overline{0}$, d.h. $pq \mid (b+c)(b-c)$. Sind $\overline{b}$ und $\overline{c}$ wesentlich verschieden, so können p und q nicht gleichzeitig $b+c$ teilen, und auch nicht gleichzeitig $b-c$ teilen. Es gilt also $\mathrm{ggT}(m, b+c) = p$ oder $\mathrm{ggT}(m, b+c) = q$. Daher läßt sich p oder q mit dem Euklidischen Algorithmus berechnen und m problemlos faktorisieren.

Das Fazit ist: Das Ziehen von Quadratwurzeln in $\mathbb{Z}_m^*$ ist genauso rechenintensiv wie das Faktorisieren von m. Ist es ausgeschlossen, m in Primfaktoren zu zerlegen, so kann man auch keine Quadratwurzeln modulo m berechnen.

Dies macht sich der Fiat-Shamir Algorithmus zu Nutzen. Verschiedene Personen können sich beteiligen. Wir stellen uns vor, daß eine Zentrale geheime Schlüssel verteilt. Sie wählt zwei Primzahlen p und q und veröffentlicht $m = pq$. Die Faktoren p und q bleiben geheim und müssen deswegen so

groß sein, daß sich m praktisch nicht in seine Primteiler zerlegen läßt. Die Zentrale weist jedem Teilnehmer eine geheime Zahl b zu, die zu einer primen Restklasse modulo m gehört. Gleichzeitig wird die Restklasse $\overline{a} = \overline{b}^{\,2}$ veröffentlicht.

Die Person A möchte nun sein Gegenüber B davon überzeugen, daß sie die Geheimzahl b kennt. Dazu führen A und B den folgenden ‚Dialog‘:

- A wählt rein zufällig $\overline{r} \in \mathbb{Z}_m^*$ und bildet $\overline{s} = \overline{r}^{\,2}$. A übermittelt $\overline{s}$ an B.
- B wählt rein zufällig die Zahl $t = 0$ oder 1. Falls $t = 0$, fragt er A nach einer Wurzel $\overline{w}$ von $\overline{s}$, andernfalls fragt er nach einer Wurzel $\overline{w}$ von $\overline{a} \cdot \overline{s}$.
- Ist $t = 0$, so gibt A an B die Antwort $\overline{w} = \overline{r}$, ist dagegen $t = 1$, so gibt A als Antwort $\overline{w} = \overline{b} \cdot \overline{r}$.
- B überprüft die Antwort: $\overline{r}^{\,2} = \overline{s}$ im Falle $t = 0$, und $(\overline{b} \cdot \overline{r})^{\,2} = \overline{a} \cdot \overline{s}$ im Falle $t = 1$.

Kennt A die Geheimzahl, so hat sie mit ihrem Teil des Dialogs kein Problem. Kennt A die Geheimzahl dagegen nicht, so wird sie höchstens eine der Fragen von B beantworten können. Sonst hätte A nämlich (wie immer sie sich auch $\overline{s}$ verschafft haben mag) sowohl eine Wurzel $\overline{x}$ von $\overline{s}$ als auch eine Wurzel $\overline{y}$ von $\overline{a} \cdot \overline{s}$ zur Verfügung, die sich zu einer Wurzel $\overline{x}^{\,-1}\overline{y}$ von $\overline{a}$ zusammensetzen ließen. Das Ziehen von Wurzeln quadratischer Reste modulo m ist aber, wie wir gesehen haben, genau so schwer wie das Faktorisieren von m. Ein Schwindler A wird daher r Dialogrunden mit dem Verifizierer B höchstens mit Wahrscheinlichkeit 2^{-r} bestehen.

Schießlich wird in dem Dialog keine Information über $\overline{b}$ preisgegeben. Wie kann man dies einsehen? Protokolliert B den Dialog, so kann das Protokoll nur die Zahlen s, t, w enthalten. Sie haben die folgenden Eigenschaften:

- $\overline{s}$ ist Quadrat eines rein zufälligen Elements aus $\mathbb{Z}_m^*$, ein rein zufälliger quadratischer Rest modulo m.
- t ist rein zufällig aus $\{0, 1\}$ gewählt.
- $\overline{w}^{\,2} = \overline{s}$ oder $\overline{a} \cdot \overline{s}$, je nachdem, ob $t = 0$ oder 1 ist.

Solch ein Protokoll kann nun B aber auch ohne die Hilfe von A simulieren, nämlich so:

- B wählt rein zufällig $t \in \{0, 1\}$ und ein $\overline{w} \in \mathbb{Z}_m^*$. B bildet daraus $\overline{s} = \overline{w}^{\,2}$ bzw. $\overline{s} = \overline{w}^{\,2} \cdot \overline{a}^{\,-1}$, je nachdem, ob $t = 0$ oder 1 ist.
- B erstellt das Protokoll (s, t, w).

Dieses Protokoll ist ohne Kenntnis von $\overline{b}$ erstellt und hat dieselben Eigenschaften wie das Protokoll aus dem Dialog zwischen A und B. Deswegen kann auch der Dialog keine Informationen über $\overline{b}$ tragen. Man sagt, der Dialog besitzt die **Zero-Knowledge Eigenschaft**.

Auf den ersten Blick könnte man meinen, dass auch ein verkürzter Dialog, in dem nur nach der Wurzel von $\overline{a} \cdot \overline{s}$ gefragt wird und demzufolge kein zufälliges bit $\overline{t}$ generiert wird, brauchbar ist. Er ist jedoch ungeeignet: Wohl hat er dann die Zero-Knowledge Eigenschaft, jedoch kann einen solchen Dialog auch ein Schwindler A bestehen, der $\overline{b}$ gar nicht kennt. A könnte dann etwa so vorgehen:

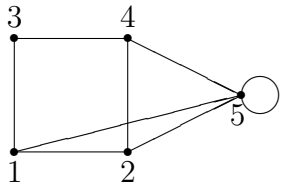
- A wählt rein zufällig $\overline{w}$ und berechnet $\overline{s} = \overline{w}^2\overline{a}^{-1}$. A übermittelt $\overline{s}$ an B.
- A teilt $\overline{w}$ mit. B überprüft: $\overline{w}^2 = \overline{a} \cdot \overline{s}$.

B kann diesen Dialog nicht von dem vorigen im Fall $\overline{t} = 1$ unterscheiden.

Zero-Knowledge Beweis für ein NP-vollständiges Problem

Wir betrachten nun ein Entscheidungsproblem, von dem man gute Gründe hat anzunehmen, daß es zu den Problemen zählt, die den höchsten Rechenaufwand erfordern. Es handelt sich darum zu entscheiden, ob ein gegebener endlicher Graphen G 3-färbbar ist, d.h. ob man seine Ecken so mit drei verschiedenen Farben anmalen kann, daß benachbarte Ecken immer verschieden gefärbt sind.

Ein **Graph** $G = (E, K)$ besteht bekanntlich aus einer Menge E, den **Ecken**, und einer Menge $K \subset \{\{a, b\} \ : \ a, b \in E\}$, den **Kanten**, die jeweils zwei Ecken verbinden. Ecken $a \neq b$ heißen **benachbart**, falls sie durch eine Kante verbunden sind, falls also $\{a, b\} \in K$ gilt. Für $E = \{1, 2, 3, 4, 5\}$ und $K = \{\{1, 2\}, \{1, 3\}, \{2, 4\}, \{2, 5\}, \{3, 4\}, \{4, 5\}, \{5, 5\}\}$ ergibt sich das folgende Bild



Ob ein Graph 2-färbbar ist, läßt sich schnell feststellen, man probiert es, ausgehend von einer Ecke, einfach aus. Nur ganz am Anfang steht einem die Wahl der Farbe offen. Bei 3 Farben ist dies anders, man muß beim Färben der Ecken immer wieder zwischen Alternativen entscheiden, und führt ein

Versuch nicht zum Erfolg, so bedeutet dies noch nicht, daß der Graph nicht 3-färbbar ist. Es stellt sich heraus, daß die 3-Färbbarkeit von Graphen ein NP-vollständiges Entscheidungsproblem ist, also zur Klasse der Entscheidungsprobleme gehört, die nach heutigem Kenntnisstand den größten Rechenaufwand erfordern.

Wir beschreiben nun, wie eine Person A eine andere Person B mit beliebig hoher Sicherheit davon überzeugen kann, daß sie eine 3-Färbung eines Graphen G kennt, ohne von ihrem Wissen irgend etwas zu verraten. Dieser Dialog erfordert von A eine sorgfältige Vorbereitung.

- Für jede Ecke $e \in E$ des Graphen erzeugt A eine RSA-Kodierung $m_e = p_e q_e, s_e, t_e$. Sie gibt ihrem Gegenüber m_e und s_e für alle e bekannt.
- Weiter generiert A eine zufällige Permutation der drei Farben und damit eine neue zufällige 3-Färbung des Graphen. Die permutierte Farbe jeder Ecke e läßt sich durch 2 bits $b_e b'_e$ beschreiben. A kodiert diese bits mit dem randomisierten RSA-Schema und gibt sie kodiert als $y_e \equiv (2u + b_e)^{s_e}$ bzw. $y'_e \equiv (2u' + b'_e)^{s_e}$ modulo m_e bekannt.
- Nun wählt B zufällig zwei benachbarte Ecken $e(1), e(2)$, zu denen A ihm die Schlüssel $t_{e(1)}$ und $t_{e(2)}$ überläßt. B kann dann die Farben $b_{e(1)} b'_{e(1)}$ und $b_{e(2)} b'_{e(2)}$ dekodieren und feststellen, ob sie, wie von A behauptet, verschieden sind.

Hat A eine 3-Färbung des Graphen, so kann sie die Frage von B immer beantworten. Dagegen wird ein Schwindler, der keine 3-Färbung kennt, mit einer Wahrscheinlichkeit von mindestens k^{-1} überführt, wobei k die Anzahl der Kanten bezeichne. r unabhängige Runden übersteht er nur mit Wahrscheinlichkeit $(1 - k^{-1})^r \leq e^{-r/k}$.

B lernt aus diesem Dialog nichts. Er erfährt von A ein zufälliges Paar von verschiedenen Farben, das zudem Runde für Runde unabhängig ist. Solche Paare kann er sich auch verschaffen, ohne eine 3-Färbung des Graphen zu kennen. Folglich trägt der Dialog keine Information über die 3-Färbung.

2.10 Faktorzerlegung

Wir kommen nun auf Anwendungen des modularen Rechnens auf die Zerlegung von ganzen Zahlen und Polynomen zu sprechen. Ein Integritätsbereich, in dem sich jedes Element (bis auf Einheiten) eindeutig in irreduzible Elemente (Primelemente) zerlegen läst, heißt *faktorieller Ring*. In der Algebra wird gezeigt, dass jeder Hauptidealring R faktoriell ist und auch jeder Polynomring $R[x]$ mit Koeffizienten in einem faktoriellen Ring R. Die Beweise geben jedoch keinen Hinweis, wie man im Einzelnen effizient faktorisiert. Wir betrachten hier zwei Methoden, Fermats Methode zur Zerlegung ganzer Zahlen und den Berlekamp-Algorithmus zum Zerlegen von ganzzahligen Polynomen modulo einer Primzahl p.

Fermats Methode

Die Methode von Fermat führt das Faktorisieren einer Zahl n zurück auf ihre Darstellung als Differenz zweier Quadratzahlen. Sei $n = a \cdot b$ eine ungerade Zahl. Dann sind auch a und b ungerade, und wir können die Zahlen

$$s \;=\; \frac{a-b}{2}, \quad t \;=\; \frac{a+b}{2}$$

bilden. Es folgt

$$n \;=\; t^2 - s^2.$$

Umgekehrt erhalten wir aus dieser Darstellung die Zerlegung

$$n \;=\; (t+s)(t-s).$$

Dies führt zu folgendem Algorithmus: Bilde die Zahlen $t_0 = [\sqrt{n}] + 1, t_1 = [\sqrt{n}]+2, \ldots$ und überprüfe der Reihe nach, ob $t_0^2 - n, t_1^2 - n, \ldots$ Quadratzahlen sind.

Beispiel. Für $n = 1767$ ist $t_0^2 - n = 43^2 - 1767 = 82$ keine Quadratzahl und $t_1^2 - n = 44^2 - 1767 = 169 = 13^2$. Wir erhalten die Zerlegung $1767 = (44 + 13) \cdot (44 - 13) = 57 \cdot 31$. □

Das Verfahren findet zuerst die großen Teiler von n und führt schnell zum Ziel, wenn sich n in zwei ähnlich große Faktoren zerlegen läßt.

Schon Fermat gestaltete seine Methode effizienter, indem er die Überprüfung auf Quadratzahl modular durchführte. Man wählt dazu teilerfremde Moduln $m_1, \ldots, m_k$ und betrachtet zunächst $t_i^2 - n$ als Rest modulo m_j. Liegt kein quadratischer Rest vor, so kann $t_i^2 - n$ keine Quadratzahl sein. Nur in

dem Fall, daß es sich für jeden Modul $m_1, \ldots, m_k$ um einen quadratischen Rest handelt, muß man nachprüfen, ob man auf eine Quadratzahl gestoßen ist.

In der praktischen Durchführung stellt man eine Liste (ein ‚Sieb') der quadratischen Reste auf. Knuth faktorisiert in seiner Monographie auf diese Weise die Zahl $n = 8.616.460.799$ (von der der englische Ökonom und Logiker W.S. Jevons 1874 unvorsichtigerweise behauptet hat, daß man sie niemals wird zerlegen können, vgl. Knuth Vol. II, S. 388). Er stellt dazu die folgende Tabelle auf:

m_i	$t \bmod m_i$	$t^2 \bmod m_i$	$(t^2 - n) \bmod m_i$
3	$0, 1, 2$	$0, 1, 1$	$1, 2, 2$
5	$0, 1, 2, 3, 4$	$0, 1, 4, 4, 1$	$1, 2, 0, 0, 2$
7	$0, 1, 2, 3, 4, 5, 6$	$0, 1, 4, 2, 2, 4, 1$	$5, 6, 2, 0, 0, 2, 6$
8	$0, 1, 2, 3, 4, 5, 6, 7$	$0, 1, 4, 1, 0, 1, 4, 1$	$1, 2, 5, 2, 1, 2, 5, 2$
11	$0, 1, 2, 3, 4, 5, 6, 7,$ $8, 9, 10$	$0, 1, 4, 9, 5, 3, 3, 5,$ $9, 4, 1$	$10, 0, 3, 8, 4, 2, 2, 4,$ $8, 3, 0$

Ein Vergleich der beiden letzten Spalten zeigt, daß nur dann $t^2 - n$ eine Quadratzahl sein kann, falls gilt

$$\begin{array}{lcll} t & \equiv & 0 & \bmod 3 \\ t & \equiv & 0, 2 \text{ oder } 3 & \bmod 5 \\ t & \equiv & 2, 3, 4 \text{ oder } 5 & \bmod 7 \\ t & \equiv & 0 \text{ oder } 4 & \bmod 8 \\ t & \equiv & 1, 2, 4, 7, 9 \text{ oder } 10 & \bmod 11. \end{array}$$

Die Suche nach einem passenden t wird so erheblich eingeschränkt. Nach den Bedingungen modulo 3 und modulo 8 muß t ein Vielfaches von $3 \cdot 4 = 12$ sein. Das kleinste t größer als $[\sqrt{n}] + 1 = 92825$ mit dieser Eigenschaft ist 92832. Es hat modulo 5 den Rest 2, modulo 7 den Rest 5 und modulo 11 den Rest 3. Die Liste zeigt, daß dieses t kein Kandidat zum Faktorisieren ist. Erhöhen wir t um den Betrag 12, so ändert sich sein Residuum jeweils um 2 modulo 5, um 5 modulo 7 und um 1 modulo 11. Wie eine kurze Rechnung zeigt, ist das erste t, daß alle aufgestellten Bedingungen erfüllt, gleich 92880. Es folgt $92880^2 - n = 10233601 = 3199^2$. Wir haben also eine Lösung $s = 3199, t = 92880$ gefunden und können n zerlegen:

$$8.616.460.799 \;=\; (t - s)(t + s) \;=\; 89681 \cdot 96079.$$

Allgemein gesprochen wird zum Faktorisieren von n mit den Moduln $m_1, \ldots, m_k$ eine *Siebtabelle* $s(i, t), 1 \leq i \leq k, 0 \leq t < m_i$ aufgestellt. Es ist $s(i, t) = 1$ oder 0, je nachdem ob $q = t^2 - n$ quadratischer Rest modulo m_i ist oder nicht.

Faktorisieren mit Sieben.

Eingabe: n und Moduln $m_1, \ldots, m_k$.

Ausgabe: Ein Teiler m von n.

Verfahren: Stelle die Siebtafel auf. Berechne sukzessive modulo $m_1, \ldots, m_k$ die Zahlen $q_t = t^2 - n, t > [\sqrt{n}]$ (also $q_{t+1} = q_t + 2t + 1$). Bestimme mit der Siebtafel das kleinste t, so das q_t für alle j quadratischer Rest modulo m_j ist. Teste, ob q_t eine Quadratzahl s^2 ist. In diesem Fall ist $m = t - s$ Teiler von n. Andernfalls setze die Suche nach einem geeigneten t fort. □

Faktorisieren in $\mathbb{Z}_p[x]$

Will man ein ganzzahliges Polynom in seine irreduziblen Anteile zerlegen, so ist es im Allgemeinen günstig, das Polynom zunächst modulo m, d.h. in $\mathbb{Z}_m[x]$ zu zerlegen. Häufig gelingt es, dann auf die Zerlegung in $\mathbb{Z}[x]$ zurückzuschließen. Sei etwa $f(x) \in \mathbb{Z}[x]$ ein **normiertes Polynom**, ein Polynom mit Anfangskoeffizient 1,

$$f(x) \;=\; x^n + a_{n-1}x^{n-1} + \cdots + a_0.$$

Dann hat $f(x)$ auch als Polynom in $\mathbb{Z}_m[x]$ (a_i wird zur Restklasse a_i mod m) unverändert den Grad n. Gibt es eine Zerlegung $f(x) = g(x)h(x)$, so können wir auch $g(x)$ und $h(x)$ als normiert annehmen, und die Zerlegung bleibt in $\mathbb{Z}_m[x]$ gültig. Anders ausgedrückt: Ist das normierte Polynom $f(x)$ in $\mathbb{Z}_m[x]$ irreduzibel, so ist $f(x)$ auch in $\mathbb{Z}[x]$ irreduzibel. Die Erfahrung zeigt, daß sich auf diese Weise Irreduzibilität häufig feststellen läßt.

Beispiel. Das Polynom $f(x) \;=\; x^8 + x^6 - 3x^4 - 3x^3 + 8x^2 + 2x - 5$ zerfällt in $\mathbb{Z}_2[x]$ und $\mathbb{Z}_{13}[x]$ in die irreduziblen Faktoren

$$\begin{aligned} f(x) &\equiv (x^6 + x^5 + x^4 + x + 1)(x^2 + x + 1) \bmod 2 \\ f(x) &\equiv (x^4 + 2x^3 + 3x^2 + 4x + 6)(x^3 + 8x^2 + 4x + 12)(x + 3) \bmod 13, \end{aligned}$$

wie man mit der gleich beschriebenen Berlekamp-Methode findet. Eine Faktorisierung von $f(x)$ in $\mathbb{Z}[x]$ müßte mit diesen beiden Zerlegungen vereinbar sein, was offenbar nicht möglich ist. Zerfiele $f(x)$ etwa in irreduzible Faktoren vom Grad 6 und 2, so müßten diese Faktoren modulo 13 weiter zerfallen. Ein irreduzibler Teiler vom Grad 4 modulo 13 kann dabei nicht entstehen. Folglich ist $f(x)$ in $\mathbb{Z}[x]$ irreduzibel. □

Wir behandeln nun die Faktorisierung von Polynomen in $\mathbb{Z}_p[x]$ für eine Primzahl p. Die Situation unterscheidet sich deutlich vom Zerlegen ganzer Zahlen. Das Faktorisieren ganzer Zahlen ist mit den heutigen Methoden im wesentlichen ein mehr oder weniger geschicktes Suchen nach Teilern, am Anfang ist schwer abzusehen, wie schnell man fündig wird. Für Polynome gibt es andersartige Verfahren.

Irreduzible Faktoren vom Grade 1 erhält man mittels Nullstellen.

Proposition 2.18. *Sei $f(x)$ ein Polynom mit Koeffizienten aus einem beliebigen Körper K. Dann teilt das Polynom $x - s$ genau dann $f(x)$, wenn s Nullstelle von $f(x)$ ist.*

Beweis. Wir teilen $f(x)$ durch $x - s$ mit Rest,

$$f(x) \;=\; m(x)(x-s) + r$$

mit $r \in R$. Durch Einsetzen von s folgt $f(s) = r$. Es teilt also $x - s$ genau dann $f(x)$, wenn $r = f(s) = 0$ gilt. □

Polynome in $\mathbb{Z}_p[x]$ lassen sich mit der **Methode von Berlekamp** (1967) in ganz systematischer Weise zerlegen. Wir benötigen dazu folgende Polynom-Identitäten.

Proposition 2.19. *Sei p eine Primzahl und seien $h(x)$ und $k(x)$ Polynome in $\mathbb{Z}_p[x]$. Dann gilt*

$$\begin{aligned}(h(x)+k(x))^p &= h(x)^p + k(x)^p,\\ h(x^p) &= h(x)^p,\\ h(x)^p - h(x) &= h(x)(h(x)-1)(h(x)-2)\cdots(h(x)-p+1).\end{aligned}$$

Beweis. Der Binomialkoeffizient $\binom{p}{k} = \frac{p(p-1)\cdots(p-k+1)}{1\cdot 2\cdots k}$ ist für primes p und $0 < k < p$ ein Vielfaches von p, daher gilt

$$\begin{aligned}(h(x)+k(x))^p &= h(x)^p + \binom{p}{1} h(x)^{p-1}k(x) + \cdots + k(x)^p\\ &\equiv h(x)^p + k(x)^p \bmod p.\end{aligned}$$

Dies ist die erste Behauptung. Zusammen mit dem Fermatschen Theorem impliziert sie die zweite Behauptung,

$$h(x)^p \;=\; (\sum_i a_i x^i)^p \;=\; \sum_i a_i^p x^{ip} \;=\; \sum_i a_i (x^p)^i \;=\; h(x^p).$$

Zum Beweis der letzten Aussage verwenden wir die Identität

$$x^p - x \;\equiv\; x(x-1)\cdots(x-p+1) \bmod p\ .$$

Sie ergibt sich aus Proposition 2.18 und der Beobachtung, dass nach dem Satz von Fermat $x^p - x$ die Nullstellen $0, 1, \ldots, p-1$ hat. Die Behauptung folgt, indem wir in der Identität x durch $h(x)$ ersetzen. □

Quadratische Anteile eines Polynoms lassen sich mithilfe seiner (formalen) Ableitung identifizieren. Die **Ableitung** eines Polynoms $f(x) = a_n x^n + \cdots + a_1 x + a_0$ ist definiert als

$$f'(x) \;:=\; n \cdot a_n\, x^{n-1} + (n-1) \cdot a_{n-1}\, x^{n-2} + \cdots + 2 \cdot a_2\, x + a_1.$$

Dabei benutzen wir für ein Element a aus dem Koeffizientenbereich und eine natürliche Zahl n die Schreibweise

$$n \cdot a \;:=\; \underbrace{a + a + \ldots + a}_{n-\text{mal}}.$$

Diese Definition einer Ableitung macht Sinn in beliebigen Polynomringen. Die aus der Analysis bekannte Interpretation einer Ableitung als Steigung läßt sich natürlich nicht mehr aufrechterhalten. Wesentlich ist, daß die bekannten Rechenregeln für Ableitungen erhalten bleiben. Es gilt (Übung)

$$\begin{aligned} f(x) = h(x) + k(x) \quad &\Rightarrow \quad f'(x) = h'(x) + k'(x), \\ f(x) = h(x)k(x) \quad &\Rightarrow \quad f'(x) = h'(x)k(x) + h(x)k'(x). \end{aligned}$$

Proposition 2.20. *Sei p prim und sei $f(x) \in \mathbb{Z}_p[x]$. Dann gilt:*

i) $f'(x) = 0 \;\Rightarrow\; f(x) = h(x)^p$ *für ein* $h(x) \in \mathbb{Z}_p[x]$,

ii) $f'(x) \neq 0 \;\Rightarrow\; f(x) = [\mathrm{ggT}(f(x), f'(x))]^2 h(x)$, *mit einem* $h(x) \in \mathbb{Z}_p[x]$, *das keine quadratischen Teiler mehr enthält.*

Beweis. Zu *i)*: Die Koeffizienten $i \cdot a_i$ von $f'(x)$ verschwinden genau dann für alle i, falls $a_i = 0$ für alle $i \not\equiv 0 \bmod p$. Dann gibt es offenbar ein Polynom $h(x)$, so daß $f(x) = h(x^p)$. Die Behauptung folgt nun aus Proposition 2.19.

Zu *ii)*: Gilt $f(x) = s(x)^2 t(x)$, so teilt $s(x)$ auch

$$f'(x) \;=\; 2s(x)s'(x)t(x) + s(x)t'(x).$$

Sei umgekehrt $s(x)$ ein Teiler von $f(x)$ und $f'(x)$. Wir nehmen an, daß $s(x)$ irreduzibel in $\mathbb{Z}_p$ ist, so daß nach *i)* $s'(x) \neq 0$ gilt. Aus $f(x) = s(x)k(x)$ folgt

$f'(x) = s'(x)k(x) + s(x)k'(x)$, folglich teilt $s(x)$ auch $s'(x)k(x)$. Da $s'(x) \neq 0$ einen Grad kleiner als $s(x)$ besitzt, muß $s(x)$ bereits $k(x)$ teilen. Folglich ist sogar $s(x)^2$ Teiler von $f(x)$. Insgesamt können wir feststellen, daß $s(x)$ genau dann $f(x)$ und $f'(x)$ teilt, falls $s(x)^2$ ein Teiler von $f(x)$ ist. Dies ergibt die zweite Behauptung. □

Da sich größte gemeinsame Teiler problemlos mit dem Euklidischen Algorithmus berechnen lassen, können quadratische Anteile eines Polynoms mit Koeffizienten in $\mathbb{Z}_p$ leicht ermittelt werden. Es bleibt die Aufgabe, ein Polynom zu zerlegen, dessen irreduzible Teiler alle verschieden sind. Dazu ist die folgende Aussage nützlich.

Proposition 2.21. *Sei p Primzahl und sei $f(x) \in \mathbb{Z}_p[x]$. Dann sind äquivalent:*

i) Es gilt $f(x) = f_1(x)f_2(x)$ mit teilerfremden Polynomen $f_i(x)$, $i = 1, 2$, so daß $0 < \deg(f_i) < \deg(f)$.

ii) Es gibt ein Polynom $g(x) \in \mathbb{Z}_p[x]$ mit $0 < \deg(g) < \deg(f)$, so daß $f(x)$ ein Teiler von $g(x)^p - g(x)$ ist.

Es ist dann

$$f(x) \;=\; \prod_{i=0}^{p-1} \mathrm{ggT}(f(x), g(x) - i)$$

eine nicht-triviale Faktorisierung von $f(x)$.

Beweis. Hat $g(x)$ die geforderten Eigenschaften, so ist wegen $\deg(g) \geq 1$ zunächst $g(x)^p - g(x) \neq 0$. Die Polynome $g(x), g(x)-1, \ldots, g(x)-p+1$ sind paarweise teilerfremd. Es folgt daher aus $f(x) \mid g(x)^p - g(x)$

$$\begin{aligned} f(x) &= \mathrm{ggT}(f(x), g(x)^p - g(x)) \\ &= \prod_{i=0}^{p-1} \mathrm{ggT}(f(x), g(x) - i). \end{aligned}$$

Mit $g(x)-i$ sind auch die Polynome $\mathrm{ggT}(f(x), g(x)-i)$ paarweise teilerfremd und haben einen Grad kleiner als $f(x)$. Mindestens zwei haben einen Grad größer 0, die man noch zu zwei teilerfremden Faktoren zusammenfassen kann. Dies ergibt die gewünschte Zerlegung von $f(x)$.

Sei umgekehrt $f(x) = f_1(x)f_2(x)$ eine Zerlegung von $f(x)$ in teilerfremde Faktoren. Wir betrachten das Kongruenzensystem

$$g(x) \;\equiv\; i \bmod f_i(x), \quad i = 1, 2$$

im Polynomring $\mathbb{Z}_p[x]$. Nach dem Chinesischen Restsatz gibt es ein solches Polynom $g(x)$ von einem Grad kleiner als $\deg(f)$. Da $\deg(f_i) > 0$, kann $g(x)$ kein Polynom vom Grade 0 sein. Es folgt, daß $f_i(x)$ ein Teiler von $g(x) - i$ ist. Nach Proposition 2.19 teilt $f_i(x)$ auch $g(x)^p - g(x)$, und da die $f_i(x)$ teilerfremd sind, ist schließlich $f(x)$ ein Teiler von $g(x)^p - g(x)$. □

Damit man die letzte Proposition zum Faktorisieren eines Polynoms nutzen kann, muß man passende Polynome $g(x)$ bestimmen können. Es stellt sich heraus, daß diese Aufgabe auf das Lösen eines linearen Gleichungssystems führt. Wir machen den Ansatz

$$g(x) \;=\; b_{n-1}x^{n-1} + \cdots + b_1 x + b_0$$

mit $n = \deg(f)$. Dann folgt

$$g(x)^p \;=\; b_{n-1}x^{p(n-1)} + \cdots + b_1 x^p + b_0.$$

Division von x^{pi} durch $f(x)$ ergibt

$$x^{ip} \;=\; f(x)m_i(x) + r_i(x)$$

mit einem Rest $r_i(x)$ von einem Grad kleiner als n. Es folgt

$$g(x)^p \;\equiv\; b_{n-1}r_{n-1}(x) + \cdots + b_0 r_0(x) \bmod f(x).$$

$f(x)$ teilt also genau dann $g(x)^p - g(x)$, wenn es

$$b_{n-1}r_{n-1}(x) + \cdots + b_0 r_0(x) - b_{n-1}x^{n-1} - \cdots - b_0$$

teilt, falls dieses Polynom also verschwindet. Koeffizientenvergleich führt zu einem linearen Gleichungssystem für die b_i: Sind $r_{j,i}$ die Koeffizienten von $r_i(x)$,

$$r_i(x) \;=\; r_{n-1,i}x^{n-1} + \cdots + r_{0,i},$$

so ergibt sich in $\mathbb{Z}_p$ das lineare Gleichungssystem

$$b_j \;=\; \sum_{i=0}^{n-1} r_{j,i}b_i \;,$$

oder in Matrix-Schreibweise

$$(R - Id) \cdot b \;=\; 0$$

mit

$$R \;:=\; \begin{pmatrix} r_{00} & \dots & r_{0,n-1} \\ \vdots & & \vdots \\ r_{n-1,0} & \dots & r_{n-1,n-1} \end{pmatrix}, \quad b \;:=\; \begin{pmatrix} b_0 \\ \vdots \\ b_{n-1} \end{pmatrix}.$$

Beispiel. Wir wollen das Polynom

$$f(x) \;=\; x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$$

in $\mathbb{Z}_2[x]$ zerlegen. Die Ableitung ist

$$f'(x) \;=\; 6x^5 + 5x^4 + 4x^3 + 3x^2 + 2x + 1 \;=\; x^4 + x^2 + 1.$$

Division von $f(x)$ durch $f'(x)$ mit Rest ergibt

$$f(x) \;=\; (x^2 + x)f'(x) + 1.$$

Der ggT von $f(x)$ und $f'(x)$ ist 1, $f(x)$ enthält daher keine quadratischen Anteile. Nun berechnen wir die Restpolynome $r_i(x)$:

$$\begin{array}{lcllcl}
x^0 &=& f(x)\cdot 0 + 1 & r_0(x) &=& 1 \\
x^2 &=& f(x)\cdot 0 + x^2 & r_1 &=& x^2 \\
x^4 &=& f(x)\cdot 0 + x^4 & r_2 &=& x^4 \\
x^6 &=& f(x) + (x^5 + \cdots + 1) & r_3(x) &=& x^5 + x^4 + x^3 + x^2 + x + 1 \\
x^8 &=& f(x)(x^2 + x) + x & r_4(x) &=& x \\
x^{10} &=& f(x)(x^4 + x^3) + x^3 & r_5(x) &=& x^3,
\end{array}$$

also

$$R \;=\; \begin{pmatrix}
1 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0
\end{pmatrix}.$$

Für b erhalten wir das Gleichungssystem

$$\begin{array}{lcllcllcl}
b_0 &=& b_0 + b_3, & b_2 &=& b_1 + b_3, & b_4 &=& b_2 + b_3, \\
b_1 &=& b_3 + b_4, & b_3 &=& b_3 + b_5, & b_5 &=& b_3,
\end{array}$$

das sich zu

$$b_1 \;=\; b_2 \;=\; b_4, \quad b_3 \;=\; b_5 \;=\; 0$$

zusammenfassen läßt. b_0 ist beliebig. Aus der Bedingung $\deg(g) > 0$ in Proposition 2.21 folgt, daß nur die Wahl $b_1 = 1$ zu einer Faktorisierung von $f(x)$ führt. Wir erhalten also

$$g(x) \;=\; x^4 + x^2 + x + b_0,$$

und die Faktorisierung

$$\begin{array}{rcl}
f(x) &=& \mathrm{ggT}(f(x), x^4 + x^2 + x) \cdot \mathrm{ggT}(f(x), x^4 + x^2 + x + 1) \\
&=& (x^3 + x + 1)(x^3 + x^2 + 1).
\end{array}$$

Beide Faktoren haben in $\mathbb{Z}_2[x]$ keine Nullstellen und sind damit irreduzibel.

2.11 Gruppen

Die Sätze von Fermat und Euler sind Spezialfälle eines allgemeinen Satzes der Gruppentheorie, den wir nun behandeln wollen. Im folgenden sei G eine Gruppe mit neutralem Element e.

Definition 2.22. *$H \subset G$ heißt **Untergruppe** von G, falls $a \cdot b \in H$ und $a^{-1} \in H$ für alle $a, b \in H$. Als **Links-** und **Rechtsnebenklassen** von H bezeichnet man die Mengen*

$$aH \;:=\; \{ab \,:\, b \in H\} \qquad bzw. \qquad Ha \;:=\; \{ba \,:\, b \in H\},$$

wobei a durch G läuft.

Beispiel. Die Menge H der quadratischen Reste bilden eine Untergruppe von $\mathbb{Z}_m^*$. Gilt nämlich $\overline{a} = \overline{b}^{\,2}$ und $\overline{a'} = \overline{b'}^{\,2}$, so folgt $\overline{a} \cdot \overline{a'} = (\overline{b} \cdot \overline{b'})^2$ und $\overline{a}^{\,-1} = (\overline{b}^{\,-1})^2$. Wegen $\overline{1}^{\,2} = \overline{-1}^{\,2}$ ist $H \neq \mathbb{Z}_m^*$, sofern $m > 2$. □

Zwei Linksnebenklassen a_1H und a_2H sind entweder disjunkt oder gleich: Nehmen wir an, daß $a_1H \cap a_2H \neq \emptyset$. Dann gilt $a_1b_1 = a_2b_2$ für geeignete $b_1, b_2 \in H$. Es folgt $a_1b = a_2b_2b_1^{-1}b \in a_2H$ für alle $b \in H$. Deswegen gilt $a_1H \subset a_2H$, und analog $a_2H \subset a_1H$.

Es ist also G Vereinigung von disjunkten Nebenklassen. Außerdem sind alle Nebenklassen gleichmächtig,

$$\mathrm{card}\, aH \;=\; \mathrm{card} H,$$

denn die Abbildung $b \mapsto ab$ ist eine Bijektion von H nach aH (die Umkehrabbildung ist $b \mapsto a^{-1}b$).

Dies führt unmittelbar zum **Satz von Lagrange**. Sei dazu $|G|$ die **Ordnung** von G, d.h. die Anzahl ihrer Elemente, und sei $[G : H]$ der **Index** der Untergruppe H, d.h. die Anzahl der verschiedenen (disjunkten) Nebenklassen von H in G.

Satz 2.23. *Sei G endliche Gruppe und H Untergruppe. Dann gilt*

$$|G| \;=\; [G : H] \cdot |H|.$$

Insbesondere ist $|G|$ ein Vielfaches von $|H|$.

Beispiel. Die Anzahl der quadratischen Reste ist für $m > 2$ ein echter Teiler der Euler-Funktion $\phi(m)$. □

Wir betrachten nun spezielle Untergruppen. Sei $a \in G$. Dann ist die kleinste Untergruppe von G, die a enthält, gegeben durch

$$H_a \ := \ \{\ldots, a^{-2}, a^{-1}, a^0 = e, a, a^2, \ldots\}.$$

Wir nehmen an, daß H_a endlich ist. Dann gibt es ganze Zahlen $s < t$, so daß $a^s = a^t$, also $a^{t-s} = e$. Ist k die kleinste natürliche Zahl mit der Eigenschaft

$$a^k \ = \ e,$$

so folgt

$$H_a \ = \ \{a, a^2, \ldots, a^{k-1}, e\},$$

insbesondere $(a^j)^{-1} = a^{k-j}$ und $a^s = e \Leftrightarrow k \mid s$. k heißt die **Ordnung** von a und wird mit $\mathrm{ord}(a)$ bezeichnet. $\mathrm{ord}(a)$ ist offenbar die Ordnung von H_a. Nach dem Satz von Lagrange ist die Ordnung von a ein Teiler der Ordnung von G. Daher folgt

$$a^{|G|} \ = \ e. \tag{2.3}$$

Im Fall $G = \mathbb{Z}_m^*$ ergibt sich erneut den **Satz von Euler**:

$$\overline{a}^{\,\phi(m)} \ = \ \overline{1}.$$

Ist $\mathrm{ord}(a) = |G|$, also $H_a = G$, so heißt G **zyklische Gruppe** und a ein **erzeugendes Element** von G.

Kapitel 3

Fehlerkorrigierende Kodes

3.1 Der Hamming–Kode

Bei der Übertragung von Nachrichten durch einen gestörten Kanal entstehen Fehler, die der Empfänger gern erkennen würde. Man sendet daher die Nachricht in redundanter Form, die Information wird teilweise wiederholt oder in eine spezielle Form gebracht, die es erlaubt, Fehler zu entdecken. Sei zum Beispiel $a \dots d$ eine binäre, aus den Bits 0 und 1 zusammengesetzte Nachricht. Man kann ihr das Prüfbit $u \equiv a + \cdots + d \bmod 2$ anhängen und $a \dots du$ senden. Der Empfänger kann dann überprüfen, ob $a + \cdots + d + u \equiv 0 \bmod 2$ gilt (parity check), und so einen Fehler erkennen. Solche Prüfzeichen sind in den Strichkodes auf Warenpackungen, den Internationalen Standard-Buchnummern (ISBN) und den Nummern der deutschen Geldscheine enthalten.

Werden Daten nur einmal übertragen (Satellitendaten, Ablesen von Kompaktdisketten in CD-Spielern), so ist es wichtig, Übertragungsfehler auch beheben zu können. Zum Beispiel könnte man jedes bit a 3-fach senden: aaa. Der Empfänger entscheidet sich für das Bit, das in der Nachricht überwiegt. Dieser Kode korrigiert auf drei Buchstaben einen Fehler, allerdings trägt nur jedes dritte bit Information.

Wir beschreiben nun einen binären Kode, der in 7 bits einen Fehler korrigiert, dabei 4 bits Information überträgt. Die Worte sind Tupel aus 1en und 0en, jedes Tupel der Länge 4 wird in ein Tupel der Länge 7 verwandelt. Die Buchstaben werden als Elemente von $\mathbb{Z}_2$ aufgefaßt. Setze

$$H := \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

H ist spaltenweise aus allen 01-Tripeln aufgebaut, abgesehen vom Tripel aus

drei Nullen.

Kodieren. Soll die Botschaft *abcd* übertragen werden, so wird der Vektor

$$k \;=\; (abcduvw)$$

gesendet, für den

$$H \cdot k^t \;=\; 0,$$

gilt. In Gleichungen in $\mathbb{Z}_2$ ausgedrückt bedeutet dies

$$\begin{aligned} u &= a + c + d, \\ v &= a + b + d \\ w &= a + b + c. \end{aligned}$$

Dekodieren. Wird $\widetilde{k} = (ABCDUVW)$ empfangen, so bestimmt man $H \cdot \widetilde{k}^t$. Gilt $H \cdot \widetilde{k}^t = 0$, so wird an $\widetilde{k}$ keine Änderung vorgenommen. Andernfalls ist $H \cdot \widetilde{k}^t$ eine der sieben Spalten von H. Dann wird $\widetilde{k}$ an der entsprechenden Stelle korrigiert.

Tritt ein einziger Fehler auf, so enthält der Vektor $f = \widetilde{k} - k$ genau eine 1 und sechs 0en, und

$$H \cdot \widetilde{k}^t = H \cdot f^t$$

taucht als Spalte in H an der Position auf, an der es in $\widetilde{k}$ zu dem Übertragungsfehler gekommen ist. Dies ermöglicht es dem Empfänger, den Fehler zu korrigieren. Für $\widetilde{k} = (1011101)$ etwa gilt $H \cdot \widetilde{k}^t = (001)^t$, der Fehler ist (vorausgesetzt es liegt nur ein Fehler vor!) an der letzten Stelle aufgetreten, und der Empfänger dekodiert $k = (1011100)$.

Dieser von Hamming 1950 vorgeschlagene Kode ist in der Korrektur von Fehlern recht wirksam. Ist $p = 0,01$ die Wahrscheinlichkeit, daß ein Bit falsch empfangen wird, und treten Fehler unabhängig voneinander auf, so ist die Wahrscheinlichkeit, daß die dekodierte Nachricht Fehler enthält (daß also mindestens zwei Übertragungsfehler vorliegen), gleich

$$1 - (1-p)^7 - 7p(1-p)^6 = 0,0020.$$

Im Kontrast dazu ist die Wahrscheinlichkeit, daß bei Übertragung von 4 bits mindestens ein Fehler auftritt, fast 20-mal so groß, nämlich $1 - (1-p)^4 = 0,039$.

Durch Modifikation läßt sich erreichen, daß der Hamming-Kode zwei Fehler erkennen und einen korrigieren kann. Sei

$$\overline{H} \;:=\; \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \;=\; \begin{pmatrix} 1 & \dots & 1 \\ & & 0 \\ & H & 0 \\ & & 0 \end{pmatrix}.$$

Die Botschaft *abcd* wird als der Vektor $k = (abcduvwx)$ kodiert, der die Gleichung

$$\overline{H} \cdot k^t \;=\; 0$$

erfüllt. u, v, w ergeben sich wie oben, ferner gilt $a+b+c+d+u+v+w+x = 0$, was zu der weiteren Gleichung

$$x \;=\; b + c + d$$

in $\mathbb{Z}_2$ führt. Der Empfänger bestimmt bei Empfang von $\widetilde{k} = (ABCDUVWX)$ wieder $\overline{H} \cdot \widetilde{k}^t$. Wir unterscheiden 3 Fälle.

Fall 1. Tritt kein Übertragungsfehler ein, so ist $\overline{H} \cdot \widetilde{k}^t = 0$.

Fall 2. Bei einem Fehler enthält der Vektor $f = \widetilde{k} - k$ genau eine 1, und $\overline{H} \cdot \widetilde{k}^t = \overline{H} \cdot f^t$ ist wieder die Spalte in $\overline{H}$ an der Stelle, an der in $\widetilde{k}$ der Fehler steckt.

Fall 3. Bei zwei Fehlern enthält f genau zwei 1en. Dann ist der oberste Eintrag von $\overline{H} \cdot \widetilde{k}^t$ eine 0, und $\overline{H} \cdot \widetilde{k}^t \neq 0$, nämlich die Summe zweier Spalten aus $\overline{H}$. Man erkennt so, daß es zwei Übertragungsfehler gibt, kann ihre Position jedoch nicht mehr erkennen.

Hammings Konstruktion läßt sich verallgemeinern.

Definition 3.1. *Sei H eine binäre $l \times (2^l - 1)$ Matrix, die als Spalten alle 01-Tupel der Länge l besitzt, abgesehen vom Tupel aus lauter Nullen. Der zugehörige binäre Kode heißt* ***(n, m)-Hamming-Kode****, mit $n = 2^l - 1$ und $m = n - l$.*

Diese Kodes, die Worte der Länge m als Worte der Länge $n = m+l$ kodieren, können alle einen Fehler korrigieren. Man kann sie erneut so erweitern, so daß sie 2 Fehler erkennen. Von praktischer Bedeutung ist allerdings nur der (7,4)-Hamming-Kode. Das Problem besteht darin, daß Hamming-Kodes nicht nur Fehler korrigieren, sondern auch zusätzliche Fehler erzeugen können, wenn es nämlich pro Wort zu mehr als einem Fehler kommt. Diese Tendenz verstärkt sich mit wachsendem l. Die folgende Tabelle belegt dies. Sie enthält den Erwartungswert E der Fehler, die ein Kodewort eines (n, m)-Hamming-Kode enthält, unter der Annahme, daß Fehler pro bit mit Wahrscheinlichkeit $p = 0,02$ und unabhängig voneinander auftreten.

	l	E	$(2^l - 1)p$
$(7, 4) -$ Kode	3	$0,024$	$0,14$
$(31, 26) -$ Kode	5	$0,41$	$0,62$
$(127, 120) -$ Kode	7	$3,06$	$2,54$

Zum Vergleich ist die erwartete Anzahl von Fehlern in einer unkodierten Nachricht der Länge $2^l - 1$ angegeben. Das Resultat ist ernüchternd: Für $l = 7$ erzeugt der Kode mehr Fehler, als er beseitigt. Genaueres dazu findet sich im nächsten Abschnitt.

3.2 Die mittlere Fehlerzahl des Hamming-Kodes

In diesem Anhang bestimmen wir die erwartete Fehlerzahl pro Kodewort des (n, m)-Hamming-Kodes. Wir nehmen an, daß die bits unabhängig voneinander jeweils mit Wahrscheinlichkeit p falsch übertragen werden.

Die zulässigen Kodewörter k sind durch die Eigenschaft $H \cdot k^t = 0$ charakterisiert. Besitzt die empfangene Nachricht $\widetilde{k}$ diese Eigenschaft nicht, so verwandelt der Hamming-Kode die Nachricht in ein zulässiges Kodewort $\widehat{k}$, das im Allgemeinen nicht mit der gesendeten Nachricht k übereinstimmen wird. Die Anzahl der Fehler zwischen gesendeter und dekodierter Nachricht ist die Anzahl der Einsen in der Differenz $\delta = k - \widehat{k}$. Es gilt $H \cdot \delta^t = H \cdot k^t - H \cdot \widehat{k}^t = 0$, δ ist also ebenfalls ein zulässiges Kodewort.

Sei umgekehrt δ ein fest vorgegebener Vektor aus i Einsen und $n - i$ Nullen mit $H \cdot \delta^t = 0$. Wir fragen, wie δ bei der Übertragung als Differenz von k und $\widehat{k}$ entstehen kann. Dies ist auf drei Weisen möglich: Entweder gibt δ wirklich die Stellen an, an denen es zu Übertragungsfehlern gekommen ist. Wegen $H \cdot \delta^t = 0$ gilt dann $H \cdot \widetilde{k}^t = 0$. Der Hamming-Kode erkennt also keinen Fehler und nimmt in der Tat keine Korrekturen vor. Dieser Fall tritt mit Wahrscheinlichkeit $p^i q^{n-i}$, $q = 1 - p$ ein. Oder aber es liegt ein weiterer Übertragungsfehler vor, an einer Stelle s, an der in δ eine 0 steht. Dann ist

$$H \cdot \widetilde{k}^t \;=\; H \cdot (\widetilde{k} - k - \delta)^t$$

die s-te Spalte von H, denn $\widetilde{k} - k - \delta$ besteht an der s-ten Stelle aus einer 1 und sonst aus Nullen. Der Hamming-Kode korrigiert also diesen Fehler. Der zusätzliche Fehler kann an den $n - i$ Positionen stehen, an denen δ eine 0 aufweist, dieser Fall tritt also mit Wahrscheinlichkeit $(n-i)p^{i+1}q^{n-i-1}$ ein. Schließlich ist es möglich, daß nur $i - 1$ Fehler aufgetreten sind, einer der i Stellen, die in δ eine 1 aufweisen, ist nicht falsch beim Empfänger angekommen. Auch in diesem Fall ist

$$H \cdot \widetilde{k}^t \;=\; H \cdot (\widetilde{k} - k - \delta)^t$$

die entsprechende Spalte in H. In diesem Fall fügt also der Hamming-Kode an der entsprechenden Stelle einen weiteren Fehler ein. Dieser letzte Fall tritt mit Wahrscheinlichkeit $ip^{i-1}q^{n-i+1}$ ein.

Setzen wir

$$d_i \;:=\; \text{Anzahl der zulässigen Kodewörter mit } i \text{ Einsen},$$

so ergibt sich für die erwartete Anzahl der Fehler pro Kodewort, die der Hamming-Kode produziert, die Formel

$$E_l \;=\; \sum_{i=0}^{n} i d_i \Big(p^i q^{n-i} + (n-i) p^{i+1} q^{n-i-1} + i p^{i-1} q^{n-i+1} \Big).$$

Wir stellen nun eine Rekursionsformel für die d_i auf. Sei $u = (u_1 u_2 \dots u_n)$ ein Vektor aus i Einsen und $n-i$ Nullen. Wir betrachten den Vektor $v = H \cdot u^t$. Es bestehen verschiedene Möglichkeiten. Ist $v = 0$, so ist u ein zulässiges Kodewort. Ist dagegen $v \neq 0$ so taucht v als Spalte in H auf, etwa als s-te Spalte. Dann ändern wir u an der s-ten Position ab und erhalten einen Vektor u' mit der Eigenschaft $H \cdot u' = v \pm v = 0$. Nun ist also u' ein zulässiges Kodewort mit $i+1$ oder $i-1$ Einsen.

Geht man nach diesem Schema durch alle Vektoren u mit i Einsen, so erhält man offenbar jedes Kodeworte mit i Einsen einmal. Die Kodeworte mit $i+1$ Einsen entstehen dagegen $(i+1)$-mal, denn an jeder der $i+1$ Positionen, in der in u' eine Eins steht, kann beim Übergang von u nach u' eine 0 in eine 1 verwandelt werden. Analog erhält man die Kodeworte mit $i-1$ Einsen $(n-i+1)$-mal. Es folgt

$$\binom{n}{i} \;=\; d_i + (i+1) d_{i+1} + (n-i+1) d_{i-1}, \quad i = 0, \dots, n,$$

mit der Vereinbarung $d_{-1} = d_{n+1} = 0$. Diese rekursiven Gleichungen legen die d_i eindeutig fest. Um sie nach d_i aufzulösen, benutzen wir die Methode der erzeugenden Funktionen und betrachten die **erzeugende Funktion**

$$d(t) \;:=\; d_0 + d_1 t + \dots + d_n t^n, t \in \mathbb{R},$$

der Zahlen d_i. Multiplizieren wir die Rekursionsgleichungen mit t^i und summieren über $i = 0, \dots, n$, so ergibt sich nach einer kleinen Rechnung

$$\begin{aligned} (1+t)^n &= \sum_{i=0}^{n} d_i t^i + \sum_{i=0}^{n} (i+1) d_{i+1} t^i + \sum_{i=0}^{n} (n-i+1) d_{i-1} t^i \\ &= d(t) + d'(t) + n t d(t) - t^2 d'(t). \end{aligned}$$

Dies ist eine Differentialgleichung erster Ordnung, ihre Lösung ist eindeutig durch $d(0) = d_0 = 1$ bestimmt. Man rechnet nach, daß die Lösung gegeben ist durch

$$d(t) \;=\; \frac{1}{n+1}(1+t)^n + \frac{n}{n+1}(1+t)^{\frac{n-1}{2}}(1-t)^{\frac{n+1}{2}}.$$

Mit dieser Formel können wir nun E_l explizit bestimmen. Eine einfache Rechnung zeigt

$$E_l \;=\; q^n\{(1+t+(n-1)t^2)d'(t)+(t-t^3)d''(t)\} \quad \text{mit } t=\frac{p}{q}.$$

Mit der Formel für $d(t)$ erhält man nach Ausführung der Differentiationen schließlich

$$\begin{aligned} E_l \;=\; & \left(1+(n-1)\frac{p^2}{q}\right)\left(\frac{n}{n+1}[1-(1+(n-1)p)(1-2p)^{\frac{n-1}{2}}]\right) \\ & +\frac{p(1-2p)}{q}\frac{n(n-1)}{n+1}\left(1+(np^2-3p^2+4p-1)(1-2p)^{\frac{n-3}{2}}\right). \end{aligned}$$

Um einen übersichtlicheren Ausdruck zu erhalten, setzen wir $p=\lambda/n$. Dann ist λ die mittlere Fehlerzahl in einer Nachricht der Länge n. Lassen wir nun l gegen ∞ gehen, so folgt

$$E_l \;\to\; \lambda+1-(1+2\lambda)\exp(-\lambda).$$

Dieser Formel kann man entnehmen, daß ein Hamming-Kode für großes l im Mittel mehr als λ Fehler produziert, falls $(1+2\lambda)\exp(-\lambda)<1$ gilt. Für $\lambda > 1,26$ ist dies bereits der Fall. Dies ist ein wahrhaft enttäuschendes Resultat: Falls pro Kodewort im Mittel mehr als 1,26 Fehler auftreten, erzeugt ein Hamming-Kode für großes l im Durchschnitt mehr Fehler, als er korrigiert. Dies unterstreicht, daß die (n,m)-Hamming-Kodes für großes l unbrauchbar sind.

Abschließend leiten wir noch eine explizite Formel für die Zahlen d_i ab. Da $n=2^l-1$ eine ungerade Zahl ist, gilt

$$
\begin{aligned}
d(t) &= \frac{1}{n+1}(1+t)^n + \frac{n}{n+1}(1-t)(1-t^2)^{\frac{n-1}{2}} \\
&= \frac{1}{n+1}\sum_{k=0}^{n}\binom{n}{k}t^k + \frac{n}{n+1}(1-t)\sum_{j=0}^{\frac{n-1}{2}}\binom{\frac{n-1}{2}}{j}(-1)^j t^{2j} \\
&= \frac{1}{n+1}\sum_{j=0}^{\frac{n-1}{2}}\left(\binom{n}{2j}t^{2j} + \binom{n}{2j+1}t^{2j+1}\right) \\
&\quad + \frac{n}{n+1}\sum_{j=0}^{\frac{n-1}{2}}\left((-1)^j\binom{\frac{n-1}{2}}{j}t^{2j} - (-1)^j\binom{\frac{n-1}{2}}{j}t^{2j+1}\right).
\end{aligned}
$$

Ein Koeffizientenvergleich ergibt

$$
\begin{aligned}
d_{2j} &= \frac{1}{n+1}\binom{n}{2j} + \frac{n}{n+1}(-1)^j\binom{\frac{n-1}{2}}{j}, \\
d_{2j+1} &= \frac{1}{n+1}\binom{n}{2j+1} - \frac{n}{n+1}(-1)^j\binom{\frac{n-1}{2}}{j}.
\end{aligned}
$$

Literatur

J.H. van Lint, Coding Theory. Springer Lecture Notes, Vol. 201, Springer, 1971.

3.3 Lineare Kodes

Wir betrachten Kodes, die Wörter der Länge m in Wörter der Länge n verwandeln. Als Alphabeth, in dem die Nachrichten geschrieben werden, wählen wir einen endlichen Körper $\mathbb{F}$. Im einfachsten Fall eines binären Kodes ist das der Körper $\mathbb{Z}_2$. Die unchiffrierte Nachricht $w = (w_0 w_1 \dots w_{m-1})$ und seine Kodierung $k = (k_0 k_1 \dots k_{n-1})$ können wir dann als Vektoren aus $\mathbb{F}^m$ bzw. $\mathbb{F}^n$ auffassen.

Ein **(n, m)-Blockkode** ist eine injektive Abbildung

$$\kappa : \mathbb{F}^m \to \mathbb{F}^n.$$

Man identifiziert häufig den Kode mit der Menge $K = \kappa(\mathbb{F}^m)$ seiner Kodewörter (und lässt damit die Frage nach dem Kodieren der Information beiseite). Ein Kode heißt **linear**, falls K linearer Teilraum von $\mathbb{F}^n$ ist. m ist dann die Dimension von K.

Lineare Kodes lassen sich mit Matrizen beschreiben. Eine $m \times n$ Matrix G heißt **erzeugende Matrix** von K, falls ihre m Zeilen eine Vektorraumbasis von K bilden. Dann kann man (wie die lineare Algebra lehrt) mit der linearen Abbildung

$$w = (w_0 w_1 \dots w_{m-1}) \;\mapsto\; \kappa(w) := w \cdot G$$

kodieren.

Aus Sicht des Empfängers ist es günstig, K mit einer Matrix H zu beschreiben, so daß

$$k \in K \;\Leftrightarrow\; H \cdot k^t = 0$$

gilt. Wie beim Hamming-Kode kann er dann nachprüfen, ob es Übertragungsfehler gibt. H heißt **Kontrollmatrix** von K. Da bei einem (n, m)-Kode H einen Kern von Dimension m in dem n-dimensionalen Raum $\mathbb{F}^n$ besitzt, hat H Rang $n - m$. Man wird also H als $(n - m) \times n$-Matrix wählen, mit $n - m$ linear unabhängigen Zeilen.

Wichtig für die Güte eines Kodes ist der gegenseitige Abstand seiner Kodewörter. Seien $k = (k_0 k_1 \dots k_{n-1})$, $k' = (k'_0 k_1 \dots k'_{n-1})$ zwei Vektoren der Länge n. Dann wird der **Hamming-Abstand** von k und k' definiert als

$$d(k, k') \;:=\; \text{Anzahl der } i \text{ mit } k_i \neq k'_i.$$

Als **Gewicht** von k bezeichnet man

$$g(k) \;:=\; d(k, 0),$$

die Anzahl der Buchstaben $k_i \neq 0$ in k. Der **Minimalabstand** des Kodes K ist definiert als

$$d_{\min}(K) \;:=\; \min\{d(k, k') \;:\; k, k' \in K, k \neq k'\}.$$

Für lineare Kodes gilt wegen $d(k, k') = g(k - k')$ offenbar

$$d_{\min}(K) \;=\; g_{\min}(K) \;:=\; \min\{g(k) \;:\; k \in K, k \neq 0\}.$$

Proposition 3.2. *Sei K linearer Kode mit Kontrollmatrix H. Dann ist $d_{\min}(K)$ gleich der minimalen Zahl g, für die es g linear abhängige Spalten in H gibt.*

Beweis. Seien $s_0, \dots, s_{n-1}$ die Spalten von H. Dann ist $k = (k_0 \dots k_{n-1})$ genau dann Kodewort, wenn $k_0 s_0 + \dots + k_{n-1} s_{n-1}$ Nullvektor ist. Hat k das Gewicht $g > 0$, so sind die entsprechenden g Spalten linear abhängig. Umgekehrt ergibt sich aus g linear abhängigen Spalten ein Kodewort k vom Gewicht g. □

Damit erhalten wir eine erste Abschätzung des Minimalabstands linearer Kodes: Die Kontrollmatrix hat Rang $n - m$, also gibt es unter den n der Kontrollmatrix $n - m + 1$ linear abhängige, und es folgt

$$d_{\min} \ \leq \ n - m + 1 \ . \tag{3.1}$$

Lineare Kodes mit der Eigenschaft $d_{\min} = n - m + 1$ heißen **optimal**. Wir werden später die Reed-Solomon Kodes als optimale Kodes kennenlernen.

Kommt es bei der Übertragung der Nachricht zu Fehlern, so gehört das empfangene Wort $\widetilde{k}$ möglicherweise nicht mehr zu K. Man dekodiert dann üblicherweise nach dem **Maximum-Likelihood-Prinzip**: Man entscheidet sich für ein Kodewort k , bei dem es zu dem Übertragungsfehler $f = \widetilde{k} - k$ mit möglichst großer Wahrscheinlichkeit kommt. Typischerweise läuft das darauf hinaus, dasjenige Kodewort zu wählen, daß von dem empfangenen Wort minimalen Hamming-Abstand hat.

Ein **Dekodierschema** für einen (nicht notwendigerweise linearen) Kode K ist eine Familie

$$D = (D_k)_{k \in K}$$

mit den Eigenschaften

$$\begin{array}{ll} D_k \subset \mathbb{F}^n & \text{für alle } k \in K, \\ k \in D_k & \text{für alle } k \in K, \\ D_k \cap D_{k'} = \emptyset & \text{für alle } k \neq k'. \end{array}$$

Liegt ein empfangenes Wort in D_k, so wird es als die Nachricht k dekodiert. Man sagt, D **entdeckt den Übertragungsfehler** $f \in \mathbb{F}^n, f \neq 0$, falls für alle $k \in K$

$$k + f \notin \bigcup_{k'} D_{k'},$$

und D **korrigiert den Fehler** $f \neq 0$, falls für alle $k \in K$

$$k + f \in D_k.$$

Proposition 3.3. *Sei $K \subset \mathbb{F}^n$ ein beliebiger Kode und $s \in \mathbb{N}$.*

i) Genau dann existiert ein Dekodierungsschema, das alle Fehler vom Gewicht höchstens s entdeckt, wenn $s < d_{\min}(K)$ ist.

ii) Genau dann existiert ein Dekodierungsschema, das alle Fehler vom Gewicht höchstens s korrigiert, wenn $2s < d_{\min}(K)$ gilt.

Beweis. i) Einen Fehler $f \neq 0$, für den zwei Kodeworte k, k' mit $k + f = k'$ existieren, bleibt beim Empfang von k' unerkannt. Nur diejenigen Fehler f, deren Gewicht $g(f) < d_{\min}(K)$ ist, können sicher entdeckt werden. Das Dekodierschema $D_k = \{k\}$ leistet dies.

ii) Ein Dekodierschema (D_k), daß alle Fehler vom Gewicht höchstens s korrigiert, hat die Eigenschaft, daß für alle $k \in K$ die ‚Kugeln'

$$B_s(k) \ := \ \{v \in \mathbb{F}^n \ : \ d(v,k) \leq s\}$$

in D_k enthalten sind. Es folgt $B_s(k) \cap B_s(k') = \emptyset$, daher muß $d(k,k') > 2s$ für beliebige Kodeworte $k \neq k'$ gelten. Umgekehrt ist unter der angegebene Bedingung $D_k := B_s(k)$ ein Dekodierschema, das s Fehler korrigiert. □

Beispiel. Hadamard-Kodes. Sei n geradzahlig. Eine $n \times n$ Matrix $M = (m_{ij})$ mit den Einträgen 1 oder -1 heißt **Hadamard-Matrix**, falls gilt

$$M \cdot M^t = n \cdot Id.$$

Geometrisch besagt diese Bedingung, daß zwei Zeilen $v \neq v'$ von M zueinander orthogonal sind. Für Vektoren aus 1en und -1en bedeutet dies, daß die Zahl der übereinstimmenden und der unterschiedlichen Komponenten in v und v' gleich sind, nämlich $n/2$. Die $2n$ Zeilen der Matrix, die aus

$$\begin{pmatrix} M \\ -M \end{pmatrix}$$

durch Ersetzen jeder -1 durch eine 0 entsteht, haben daher in $\mathbb{Z}_2^n$ alle mindestens einen Hamming-Abstand $n/2$ voneinander. Sie bilden einen (nicht linearen) Kode aus $2n$ Kodewörtern der Länge n, der bis zu $[\frac{n-2}{4}]$ Fehler korrigieren kann.

Geeignete Matrizen erhält man durch folgende Beobachtung (Übung): Ist M' eine weitere Hadamard-Matrix, so gilt dies auch für das ‚Kronecker-Produkt'

$$M \otimes M' \ := \ \begin{pmatrix} m_{11}M' & \cdots & m_{1n}M' \\ \vdots & & \vdots \\ m_{n1}M' & \cdots & m_{nn}M' \end{pmatrix}.$$

Ausgehend von

$$M_2 \ := \ \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

kann man so Hadamard Matrizen mit $n = 2^r$ Zeilen konstruieren. – Die Mariner-Sonde benutzte 1962 für die Bildübertragung vom Mars einen Hadamard-Kode. Er bestand aus 64 Kodewörtern der Länge 32 und konnte pro

Wort bis zu 7 Fehlern korrigieren. □

Kodes, bei denen die Kodewörter völlig gleichmäßig in $\mathbb{F}^n$ verteilt sind, nennt man perfekt.

Definition 3.4. *Ein Kode $K \subset \mathbb{F}^n$ heißt **perfekt**, falls es eine natürliche Zahl s gibt, so daß für die Kugeln $B_s(k) = \{v \in \mathbb{F}^n \,:\, d(v,k) \leq s\}$, $k \in K$ gilt:*

$$B_s(k) \cap B_s(k') = \emptyset \text{ für alle } k \neq k', \text{ und } \bigcup_{k \in K} B_s(k) = \mathbb{F}^n.$$

Beispiele.

1. Alle (n,m)-Hamming-Kodes sind perfekt: Jede Kugel vom Radius 1 um ein Kodewort enthält genau $n+1 = 2^l$ Worte der Länge n. Diese Kugeln sind disjunkt, denn die Hamming-Kodes korrigieren einen Fehler. Die Dimension des Kodes ist $m = n - l$; dies bedeutet, daß es 2^{n-l} verschiedene Kodeworte gibt. Wegen
$$2^l \cdot 2^{n-l} = 2^n$$
schöpfen die Kugeln vom Radius 1 um die Kodeworte bereits den gesamten $\mathbb{Z}_2^n$ aus.

2. Sonst treten perfekte Kodes nur sporadisch auf. Zum Beispiel gibt es nur einen nicht-trivialen perfekten binären Kode, der 3 Fehler korrigiert. Es handelt sich um den erstaunlichen **Golay-Kode** $K \subset \mathbb{Z}_2^{23}$, einen (23,12)-Blockkode. Seine Kontrollmatrix ist von der Gestalt $H = (H'E)$ mit der 11×11 Einheitsmatrix E und der 12×11 Matrix
$$H' := \begin{pmatrix} 1&0&0&1&1&1&0&0&0&1&1&1 \\ 1&0&1&0&1&1&0&1&1&0&0&1 \\ 1&0&1&1&0&1&1&0&1&0&1&0 \\ 1&0&1&1&1&0&1&1&0&1&0&0 \\ 1&1&0&0&1&1&1&0&1&1&0&0 \\ 1&1&0&1&0&1&1&1&0&0&0&1 \\ 1&1&0&1&1&0&0&1&1&0&1&0 \\ 1&1&1&0&0&1&0&1&0&1&1&0 \\ 1&1&1&0&1&0&1&0&0&0&1&1 \\ 1&1&1&1&0&0&0&0&1&1&0&1 \\ 0&1&1&1&1&1&1&1&1&1&1&1 \end{pmatrix}.$$
Jeweils 6 Spalten von H sind, wie eine langwierige Rechnung zeigt, linear unabhängig, deswegen gilt $d_{\min}(K) \geq 7$ nach Proposition 3.2. In einer

Kugel vom Radius 3 befinden sich

$$1 + 23 + \binom{23}{2} + \binom{23}{3} \;=\; 2048 \;= 2^{11}$$

01-Strings, und es gibt 2^{12} Kodewörter. In den disjunkten Kugeln befinden sich daher 2^{23} 01-Strings der Länge 23, und dies sind bereits alle. □

Literatur

J.H. van Lint (1992): Introduction to Coding Theory, Springer

3.4 Zyklische Kodes

Ein zyklischer Kode ist ein Kode, bei dem Kodewörter bei zyklischer Vertauschung der Buchstaben in Kodewörter übergehen.

Definition 3.5. *Ein linearer Kode $K \subset \mathbb{F}^n$ heißt **zyklisch**, falls gilt*

$$(k_0 k_1 \dots k_{n-1}) \in K \;\Rightarrow\; (k_{n-1} k_0 \dots k_{n-2}) \in K.$$

Die Struktur zyklischer Kodes läßt sich übersichtlich mit Polynomen beschreiben. Wir ordnen dazu jedem Kodewort $k = (k_0 k_1 \dots k_{n-1})$ das Polynom

$$k(x) = k_0 + k_1 x + \cdots + k_{n-1} x^{n-1}$$

zu. Diese Polynome vom Grade kleiner als n nennen wir die zu K gehörigen **Kodepolynome**. Wir können nun die zyklische Vertauschung von Kodewörtern als Multiplikation mit x modulo $x^n - 1$ ausdrücken,

$$\begin{aligned} & k_{n-1} + k_0 x + \cdots + k_{n-2} x^{n-1} \\ & \quad \equiv \;\; x(k_0 + k_1 x + \cdots + k_{n-1} x^{n-1}) \;=\; x k(x) \bmod (x^n - 1). \end{aligned}$$

Genauso entspricht der r-fache zyklische Vertauschung von Buchstaben die Multiplikation der Kodepolynome mit dem Monom x^r modulo $x^n - 1$. Da sich jedes Polynom linear aus Monomen zusammensetzt, erhalten wir die folgende Eigenschaft zyklischer Kodes: Gilt für $k'(x) = k'_0 + k'_1 x + \cdots + k'_{n-1} x^{n-1} \in \mathbb{F}[x]$

$$k'(x) \;\equiv\; k(x) w(x) \bmod (x^n - 1) \tag{3.2}$$

mit einem Kodepolynom $k(x)$ und ein beliebiges Polynom $w(x) \in \mathbb{F}[x]$, so ist auch $k'(x)$ Kodepolynom.

Sei nun $g(x)$ ein Kodepolynom von K von minimalem Grad (ungleich dem Nullpolynom). Dividiert man das Kodepolynom $k(x)$ durch $g(x)$ mit Rest

$$r(x) \;=\; k(x) - w(x)g(x),$$

so folgt nach (3.2), daß auch $r(x)$ Kodepolynom ist. Nach Wahl von $g(x)$ ist $r(x)$ das Nullpolynom. Die Kodepolynome sind also gerade die Vielfachen von $g(x)$, die einen Grad kleiner als n haben. Insbesondere ist $g(x)$ bis auf Einheiten eindeutig bestimmt. Vereinbaren wir, daß $g(x)$ ein normiertes Polynom ist, so ist $g(x)$ eindeutig festgelegt. $g(x)$ heißt das **Basispolynom** oder **erzeugende Polynom** des Kodes K.

Dividieren wir $x^n - 1$ durch $g(x)$ mit Rest

$$q(x) \;=\; x^n - 1 - m(x)g(x),$$

so folgt nach (3.2), daß $q(x)$ ebenfalls ein Kodepolynom ist. Nach Wahl von $g(x)$ ist $q(x)$ das Nullpolynom, das Basispolynom ist also ein Teiler von x^n-1.

Umgekehrt induziert jeder normierte Teiler $g(x)$ von $x^n - 1$ einen zyklischen Kode. Seine Kodepolynome sind alle Vielfachen von $g(x)$ vom Grade kleiner n. Für die Polynome $k(x) = k_0 + k_1x + \cdots + k_{n-1}x^{n-1}$ und $k'(x) = k_{n-1} + k_0x + \cdots + k_{n-2}x^{n-1}$ mit zyklisch vertauschten Koeffizienten gilt

$$k'(x) \;=\; xk(x) - k_{n-1}(x^n - 1),$$

mit $k(x)$ ist daher auch $k'(x)$ Vielfaches von $g(x)$, und der Kode ist zyklisch. Der Vektorraum der Kodepolynome wird durch die linear unabhängigen Polynome $g(x)$, $xg(x)$,...,$x^{m-1}g(x)$ aufgespannt, wobei m so zu wählen ist, daß $x^{m-1}g(x)$ den Grad $n-1$ hat. Es gilt also $m = n-\deg(\mathrm{g})$. Insgesamt erhalten wir das folgende Resultat.

Satz 3.6. *Die zyklischen Kodes $K \subset \mathbb{F}^n$ entsprechen eineindeutig den normierten Teilern $g(x)$ des Polynoms $x^n - 1$ in $\mathbb{F}[x]$. $k = (k_0k_1 \ldots k_{n-1})$ ist genau dann ein Kodewort in K, wenn $k(x) = k_0+\cdots+k_{n-1}x^{n-1}$ ein Vielfaches von $g(x)$ ist. Unter diesen Polynomen ist $g(x)$ ausgezeichnet als normiertes Polynom minimalen Grades. Ist l der Grad von $g(x)$, so handelt es sich bei K um einen (n, m)-Blockkode, mit $m = n - l$.*

Die Kodewörter entstehen aus $g(x)$ durch Multiplikation mit Polynomen $w(x)$. Wir drücken dies nun in Matrixschreibweise aus. Dazu bilden wir aus dem erzeugenden Polynom

$$g(x) \;=\; g_0 + \cdots + g_lx^l \qquad (l < n)$$

die **erzeugende Matrix**

$$G \;=\; \begin{pmatrix} g_0 & g_1 & \cdots & g_l & & \\ & g_0 & g_1 & \cdots & g_l & \\ & & \ddots & & & \ddots \\ & & & g_0 & g_1 & \cdots & g_l \end{pmatrix}$$

mit $m = n-l$ Zeilen und n Spalten. Sie enthält in jeder Zeile die Koeffizienten von $g(x)$, die restlichen Eintragungen sind 0. Es ist unmittelbar einsichtig, daß der Polynommultiplikation $k(x) = w(x)g(x)$ die Matrixmultiplikation

$$w = (w_0 w_1 \dots w_{m-1}) \;\mapsto\; k = (k_0 k_1 \dots k_{n-1}) := w \cdot G$$

entspricht, mit den Koeffizienten $w_0, w_1, \dots, w_{m-1}$ des Polynoms $w(x)$. Damit erhält man eine Kodiervorschrift, bei der die Zeilen von G die Kodierungen der Wörter $(10\dots0), (01\dots0), \dots, (00\dots1)$ sind. Da die Matrix G linear unabhängige Zeilen hat, ist die Abbildung injektiv, und es handelt sich um einen (n, m)-Blockkode.

Eine duale Beschreibung des Kodes K, vorteilhaft für das Dekodieren, gelingt mit dem normierten Polynom $h(x)$ vom Grad $\deg(h) = m = n - l$, gegeben durch die Gleichung

$$g(x)h(x) \;=\; x^n - 1.$$

Aus $k(x) = w(x)g(x)$ folgt $k(x)h(x) = w(x)(x^n - 1)$ und umgekehrt. $k(x)$ ist also genau dann Vielfaches von $g(x)$, wenn $k(x)h(x)$ Vielfaches von $x^n - 1$ ist. Dies ergibt eine alternative Charakterisierung von Kodewörtern.

Proposition 3.7. *Für jeden zyklischen Kode K gibt es ein eindeutig bestimmtes normiertes Polynom $h(x)$, einen Teiler von $x^n - 1$, so daß gilt:*

$$k = (k_0 k_1 \dots k_{n-1}) \in K \;\Leftrightarrow\; k(x)h(x) \equiv 0 \bmod (x^n - 1).$$

$h(x)$ heißt das **Kontrollpolynom** des Kodes. Auch die Kontrollbedingung $k(x)h(x) \equiv 0 \bmod (x^n - 1)$ läßt sich in Matrixschreibweise umsetzen. Sie bedeutet, daß sich im Polynom $k(x)h(x)$ die Koeffizienten von x^i und x^{n+i} in der Summe aufheben,

$$(k_0 h_i + \dots + k_i h_0) + (k_{i+1} h_{n-1} + \dots + k_{n-1} h_{i+1}) \;=\; 0$$

für $i < n$, dabei ist $h_{m+1} = \dots = h_{n-1} = 0$ gesetzt. Bilden wir die $(n-m)\times n$-Matrix

$$H \;=\; \begin{pmatrix} h_m & h_{m-1} & \dots & h_0 & & \\ & h_m & h_{m-1} & \dots & h_0 & \\ & & \ddots & & & \ddots \\ & & & h_m & h_{m-1} & \dots & h_0 \end{pmatrix},$$

die zeilenweise neben den Koeffizienten von $h(x)$ nur aus Nullen besteht, so folgt aus den angegebenen Kontrollgleichungen

$$H \cdot k^t \;=\; 0 \quad \text{für alle } k \in K.$$

Da H den Rang $n - m$ hat, folgt nach der bekannten Dimensionsformel der linearen Algebra, daß die Lösungen des homogenen Gleichungssystems $H \cdot y^t = 0$ einen Raum der Dimension m bilden. Da K von der Dimension m ist, gibt es außerhalb K keine weiteren Vektoren k, für die $H \cdot k^t = 0$ gilt. H ist also Kontrollmatrix des Kodes K.

Beispiel. In $\mathbb{Z}_2[x]$ gilt (vgl. Abschnitt 2.10)

$$x^7 - 1 = (x-1)(x^3 + x + 1)(x^3 + x^2 + 1).$$

Wählen wir $g(x) = x - 1$ und $h(x) = x^6 + x^5 + x^4 + x^3 + x^2 + x + 1$, so gilt $m = 6$ und H besteht aus der einzigen Zeile (1111111). Es ergibt sich der einfache Parity-Check-Kode $k_0 + \cdots + k_6 = 0 \bmod 2$.

Wählen wir $g(x) = x^3 + x + 1, h(x) = x^4 + x^2 + x + 1$, so erhalten wir als Kontrollmatrix

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

Dies ist bis auf die Reihenfolge der Spalten die Kontrollmatrix des (7,4)-Hamming-Kode. Man sagt, die beiden Kodes sind äquivalent. □

Zur Konstruktion weiterer zyklischer Kodes benötigt man die Theorie der endlichen Körper, die Thema des nächsten Kapitels ist.

Kapitel 4

Endliche Körper

Einen Körper mit q Elementen gibt es genau dann, wenn $q = p^n$ eine Primzahlpotenz ist. Man gewinnt ihn durch Restklassenbildung aus geeigneten Polynomringen, so wie wir früher die Körper $\mathbb{Z}_p$ durch Restklassenbildung in $\mathbb{Z}$ erhielten. Er ist eindeutig bis auf Isomorphie und wird als **Galois-Feld** $\mathbb{F}_q$ bezeichnet.

Als Anwendungen konstruieren wir die BCH-Kodes und beschreiben elliptische Kurven über endlichen Körpern zusammen mit ihrem Verwendung in der Kryptographie.

4.1 Eine algebraische Version des Hamming-Kodes

$\mathbb{Z}_2$ ist nicht der einzige Körper, der für binäre Kodes nützlich ist. Wir betrachten das irreduzible Polynom

$$g(x) \;=\; x^3 + x + 1$$

in dem Euklidischen Ring $\mathbb{Z}_2[x]$ und das zugehörige Hauptideal

$$I \;=\; I_g \;:=\; \{f(x)g(x) \;:\; f(x) \in \mathbb{Z}_2[x]\}.$$

Wie früher beim Übergang von $\mathbb{Z}$ zu $\mathbb{Z}_p$ gehen wir jetzt von $\mathbb{Z}_2[x]$ zu dem Restklassenring

$$\mathbb{F} \;:=\; \mathbb{Z}_2[x]/I$$

über. $g(x)$ irreduzibel, denn es hat keine Nullstellen. Daher ist $\mathbb{F}$ (wie im Fall $\mathbb{Z}_p$) ein Körper. Inverse Elemente von Restklassen verschafft man sich wie schon in $\mathbb{Z}_p^*$ mit dem Satz von Bézout.

$\mathbb{F}$ ist ein endlicher Körper. Seine Elemente werden von den Polynomen in $\mathbb{Z}_2[x]$ vom Grade höchstens 2 repräsentiert, von denen es 8 gibt. Ist α die Restklasse, in der das Monom x in $\mathbb{Z}_2[x]$ liegt, so besteht $\mathbb{F}$ aus den Elementen

$$a + b\alpha + c\alpha^2 \quad \text{mit } a, b, c \in \{0, 1\}.$$

Diese Ausdrücke werden in der üblichen Weise addiert und multipliziert, unter Berücksichtigung der Gleichung $g(x) \equiv 0 \bmod I$, also

$$g(\alpha) \;=\; 0 \quad \text{bzw.} \quad \alpha^3 \;=\; \alpha + 1.$$

Die Gruppe $\mathbb{F}^* = \mathbb{F} - \{0\}$ ist zyklisch mit Erzeuger α. Wir können nämlich die Elemente von $\mathbb{F}^*$ darstellen als:

α	$\alpha^5 = \alpha^3 + \alpha^2 = \alpha^2 + \alpha + 1$
α^2	$\alpha^6 = \alpha^3 + \alpha^2 + \alpha = \alpha^2 + 1$
$\alpha^3 = \alpha + 1$	$\alpha^7 = \alpha^3 + \alpha = 1$
$\alpha^4 = \alpha^2 + \alpha$	

Man sagt, α ist **primitives Element** von $\mathbb{F}$. Nach Proposition 2.19 gilt $g(\alpha^2) = g(\alpha)^2 = 0$ und $g(\alpha^4) = g(\alpha^2)^2 = 0$. Es sind also auch α^2 und α^4 Nullstellen von $g(x)$, was man leicht direkt nachrechnet. $g(x)$ zerfällt damit in $\mathbb{F}[x]$ vollständig in lineare Polynome. Man bezeichnet K deswegen als den Zerfällungskörper von $g(x)$. $\mathbb{F}$ ist zugleich Zerfällungskörper von $x^3 + x^2 + 1$, dem zweiten irreduziblen Polynoms dritten Grades in $\mathbb{Z}_2[x]$. Dieses Polynom hat die Nullstellen $\alpha^6 = \alpha^{-1}$, $\alpha^5 = \alpha^{-2}$ und $\alpha^3 = \alpha^{-4}$.

Wir benutzen nun den Körper $\mathbb{F}$ zum Kodieren einer binären Nachricht.

Kodieren. Wir wollen das binäre Wort $w = abcd$ kodieren. Dazu bilden wir das Polynom

$$k_1(x) \;=\; ax^6 + bx^5 + cx^4 + dx^3$$

und teilen es in $\mathbb{Z}_2[x]$ mit Rest durch $g(x)$:

$$k_1(x) \;=\; m(x)g(x) + k_2(x) \;=\; m(x)g(x) + ux^2 + vx + w$$

mit $u, v, w \in \mathbb{Z}_2[x]$. Da $1 = -1$ in $\mathbb{Z}_2$, gilt

$$\begin{aligned} k(x) \;&=\; ax^6 + bx^5 + cx^4 + dx^3 + ux^2 + vx + w \\ &=\; k_1(x) + k_2(x) \;=\; m(x)g(x). \end{aligned}$$

Die Nachricht $abcd$ kodieren wir als $k = (abcduvw)$.

Dekodieren. Zum Dekodieren fassen wir die Polynome als Elemente von $\mathbb{F}[x]$ auf. Dies ist möglich, da $\mathbb{Z}_2$ in $\mathbb{F}$ enthalten ist. Wir können also die Polynome in α auswerten. Es gilt $g(\alpha) = 0$ und daher auch

$$k(\alpha) \;=\; 0.$$

Dies ist die Kontrollgleichung zum Dekodieren. Erhält der Empfänger also die Nachricht $\widetilde{k} = (ABCDUVW)$, so bildet er das Polynom

$$\widetilde{k}(x) = Ax^6 + Bx^5 + Cx^4 + Dx^3 + Ux^2 + Vx + W$$

und überprüft, ob $\widetilde{k}(\alpha) = 0$ gilt. Wir betrachten zwei Fälle:

Fall 1. Wenn kein Fehler aufgetreten ist, ist $k(x) - \widetilde{k}(x) = 0$ und $\widetilde{k}(\alpha) = 0$.

Fall 2. Wenn es einen Übertragungsfehler gibt, dann gilt $k(x) - \widetilde{k}(x) = x^e$. e ist die fehlerhafte Stelle in der Nachricht. Es folgt

$$\widetilde{k}(\alpha) \;=\; \alpha^e,$$

und diese Größe bestimmt e eindeutig, da α ein Erzeuger von $\mathbb{F}^*$ ist. Der Empfänger kann also e rekonstruieren.

Dieses Kodierungsschema ist offenbar zum (7,4)-Hamming-Kode äquivalent, wir werden dies später weiter präzisieren.

Beispiel. Um 1101 zu kodieren, teilt man $x^6 + x^5 + x^3$ in $\mathbb{Z}_2[x]$ durch das Polynom $x^3 + x + 1$:

$$(x^6 + x^5 + x^3) \;=\; (x^3 + x^2 + x + 1)(x^3 + x + 1) + 1.$$

Es gilt

$$k(x) \;=\; x^6 + x^5 + x^3 + 1,$$

die kodierte Nachricht ist also 1101001. – Bei Empfang von 1$\underline{0}$01001 berechnet der Empfänger

$$\alpha^6 + \alpha^3 + 1 \;=\; (\alpha^2 + 1) + (\alpha + 1) + 1 \;=\; \alpha^5.$$

Der Fehler befindet sich an der 2. Stelle, und die Nachricht heißt korrigiert 1$\underline{1}$01001. □

4.2 Die Struktur endlicher Körper

Im nächsten Abschnitt konstruieren wir alle endlichen Körper als Oberkörper der Körper $\mathbb{Z}_p$. Vorbereitend betrachten wir zunächst Unterkörper eines endlichen Körpers $\mathbb{F}$. Den kleinsten Unterkörper, der $\alpha \in \mathbb{F}$ enthält, bezeichnen wir mit $\mathbb{F}\{\alpha\}$. Man erhält ihn als Durchschnitt aller Unterkörper, die α umfassen. Wir wollen diese Körper genauer beschreiben.

Sei zunächst $\alpha = 1$ das Einselement in $\mathbb{F}$. Da 1 in jedem Unterkörper von $\mathbb{F}$ enthalten ist, ist $\mathbb{F}\{1\}$ der kleinste Unterkörper, er heißt **Primkörper** von $\mathbb{F}$. Er enthält alle Elemente

$$i \cdot 1 \;:=\; \underbrace{1 + \cdots + 1}_{i \text{ Summanden}}, \quad i \in \mathbb{N}.$$

Da $\mathbb{F}$ endlich ist, gibt es $i < j$, so daß $i{\cdot}1 = j{\cdot}1$. Für $p = j - i$ folgt $p{\cdot}1 = 0$. Wir wählen $p > 0$ minimal mit dieser Eigenschaft. Dann sind die Körperelemente $i \cdot 1$ für $i = 1, 2, \ldots, p$ voneinander verschieden, und es liegt nahe, $\mathbb{Z}_p$ in $\mathbb{F}$ einzubetten, indem wir der Restklasse $i \bmod p$ das Körperelement $i \cdot 1$ zuordnen. Wegen $p{\cdot}1 = 0$ ist diese Bijektion mit Addition und Multiplikation verträglich. Da $\mathbb{F}$ keine Nullteiler enthält, ist auch $\mathbb{Z}_p$ nullteilerfrei. Folglich ist p eine Primzahl und $\mathbb{Z}_p$ ein Körper. $\mathbb{F}\{1\}$ ist zu diesem Körper isomorph und besteht genau aus den Elementen $i{\cdot}1$, $i = 1, \ldots, p$. Im folgenden identifizieren wir $\mathbb{F}\{1\}$ mit $\mathbb{Z}_p$, also

$$\mathbb{Z}_p \;\subset\; \mathbb{F}\,.$$

p heißt die **Charakteristik** von $\mathbb{F}$.

Wir können nun $\mathbb{F}$ (und jeden seiner Unterkörper) als Vektorraum über $\mathbb{Z}_p$ auffassen. Die Addition von Elementen in $\mathbb{F}$ ist uns vorgegeben und die Skalarmultiplikation $b\beta$ mit $b \in \mathbb{Z}_p$ und $\beta \in \mathbb{F}$ erhalten wir aus der Multiplikation in $\mathbb{F}$ und der Einbettung von $\mathbb{Z}_p$ in $\mathbb{F}$. Da $\mathbb{F}$ endlich ist, ist auch seine Dimension n über $\mathbb{Z}_p$ endlich. Nach den Resultaten der Linearen Algebra läßt sich also jedes $\beta \in \mathbb{F}$ eindeutig in der Gestalt

$$\beta \;=\; b_1\alpha_1 + b_2\alpha_2 + \cdots + b_n\alpha_n, \quad b_i \in \mathbb{Z}_p$$

darstellen, mit einer Vektorraumbasis $\alpha_1, \ldots, \alpha_n \in \mathbb{F}$. Insbesondere können wir eine Aussage über die Anzahl q der Elemente von $\mathbb{F}$ machen.

Proposition 4.1. *Sei $\mathbb{F}$ ein endlicher Körper der Charakteristik p. Dann ist die Anzahl seiner Elemente gleich $q = p^n$, mit einer natürliche Zahl n. Dabei ist n die Dimension von $\mathbb{F}$ über $\mathbb{Z}_p$.*

Für den Körper $\mathbb{F}\{\alpha\}$ liegt es nahe, eine Vektorraumbasis aus Potenzen von α zu bilden.

Proposition 4.2. *$1, \alpha, \alpha^2, \ldots, \alpha^{m-1}$ ist eine Basis von $\mathbb{F}\{\alpha\}$, wobei m die Dimension von $\mathbb{F}\{\alpha\}$ über $\mathbb{Z}_p$ sei.*

Beweis. Sei m die größte natürliche Zahl, so daß $1, \alpha, \ldots, \alpha^{m-1}$ linear unabhängig sind. Wir zeigen, daß dann die Menge $\mathbb{F}'$ aller Körperelemente der Gestalt

$$f(\alpha) \;=\; a_0 + a_1\alpha + \cdots + a_{m-1}\alpha^{m-1}, \quad a_i \in \mathbb{Z}_p$$

bereits ganz $\mathbb{F}\{\alpha\}$ ist, mit $f(x) := a_0 + a_1 x + \cdots + a_{m-1}x^{m-1}$. Offenbar ist $\mathbb{F}'$ unter Addition abgeschlossen und enthält 0 und 1. Nach Definition von m gibt es $b_0, b_1, \ldots, b_{m-1} \in \mathbb{Z}_p$, so daß

$$\alpha^m \;=\; -b_0 - b_1\alpha - \cdots - b_{m-1}\alpha^{m-1}.$$

Wenn wir daher zwei Elemente aus $\mathbb{F}'$ multiplizieren, können wir mit dieser Gleichung Potenzen α^r mit $r \geq m$ durch kleinere Potenzen ersetzen. Dies zeigt, daß $\mathbb{F}'$ unter Multiplikation abgeschlossen ist.

Um zu zeigen zeigen, daß jedes Element $f(\alpha)$ in $\mathbb{F}'$ ein Inverses besitzt, benutzen wir (ähnlich wie früher für $\mathbb{Z}_p$) den Satz von Bézout. Wir bilden das normierte Polynom $g(x) := b_0 + b_1 x + \cdots + b_{m-1}x^{m-1} + x^m$, also

$$g(\alpha) = 0\ .$$

$g(x)$ ist irreduzibel: Aus einer Zerlegung $g(x) = g_1(x)g_2(x)$ folgt nämlich wegen $g(\alpha) = 0$ die Gleichung $g_1(\alpha) = 0$ (oder $g_2(\alpha) = 0$). Nach Wahl von m muß daher $g_1(x)$ vom Grade m und $g_2(x)$ vom Grade 0 sein (oder umgekehrt). $g(x)$ hat daher keine nicht-trivialen Teiler. Da $f(x)$ einen kleineren Grad als $g(x)$ hat, sind $f(x)$ und $g(x)$ teilerfremde Polynome. Daher gibt es Polynome $s(x)$ und $t(x) \in \mathbb{Z}_p[x]$, so daß $1 = s(x)f(x)+t(x)g(x)$. Es folgt $1 = s(\alpha)f(\alpha)$, $s(\alpha)$ ist also das inverse Element von $f(\alpha)$. Andererseits gehört $s(\alpha)$ zu $\mathbb{F}'$, denn Potenzen α^r mit $r \geq m$ können wir erneut durch kleinere Potenzen ersetzen.

Es gilt also $\mathbb{F}' = \mathbb{F}\{\alpha\}$, und m ist die Dimension von $\mathbb{F}\{\alpha\}$ über $\mathbb{Z}_p$. $\square$

Allgemein kann man jeden Körper $\mathbb{F}$ als Vektorraum über jeden Unterkörper $\tilde{\mathbb{F}} \subset \mathbb{F}$ auffassen. Ist $\mathbb{F}$ endlich, so kann man für jedes $\alpha \in \mathbb{F}$ wie eben ein normiertes Polynom $g(x) \in \tilde{\mathbb{F}}[x]$ von minimalem Grade (ungleich dem Nullpolynom) bilden, so daß $g(\alpha) = 0$ gilt. Wie gesehen ist es irreduzibel. $g(x)$ heißt das **Minimalpolynom** von α über $\tilde{\mathbb{F}}$. Für ein Polynom $h(x) \in \tilde{\mathbb{F}}[x]$ gilt dann

$$h(\alpha) = 0 \quad \Leftrightarrow \quad g(x) \mid h(x).$$

Eine Division mit Rest $h(x) = m(x)g(x)+r(x)$ zeigt nämlich, daß aus $h(\alpha) = 0$ die Gleichung $r(\alpha) = 0$ folgt, daher ist $r(x)$ das Nullpolynom, und $g(x)$ teilt $h(x)$. Insbesondere folgt

$$g(x) \mid x^q - x,$$

denn $q - 1$ ist die Ordnung (die Anzahl der Elemente) der Gruppe $\mathbb{F}^* = \mathbb{F} - \{0\}$, so daß nach (2.3) $\alpha^{q-1} = 1$ für alle $\alpha \in \mathbb{F}^*$ und damit

$$\alpha^q \;=\; \alpha \tag{4.1}$$

für alle $\alpha \in \mathbb{F}$ gilt. Nach Korollar 2.18 zerfällt $x^q - x$ in $\mathbb{F}[x]$ vollständig in Linearfaktoren,

$$x^q - x \;=\; \prod_{a \in \mathbb{F}} (x - a). \tag{4.2}$$

Es stellt sich heraus, daß wir uns bei der Betrachtung von $\mathbb{F}\{\alpha\}$ mit keinen speziellen Unterkörpern von $\mathbb{F}$ befaßt haben, jeder Unterkörper von $\mathbb{F}$ ist von dieser Gestalt. Dies folgt aus dem nächsten Resultat.

Proposition 4.3. *Ist $\mathbb{F}$ ein endlicher Körper, so ist $\mathbb{F}^*$ bzgl. der Multiplikation eine zyklische Gruppe.*

Es gibt also ein $\alpha \in \mathbb{F}^*$, so daß α^k für $k = 1, \ldots, q-1$ alle Elemente von $\mathbb{F}^*$ durchläuft. α heißt dann **primitives Element** des Körpers. Der Beweis beruht auf dem folgenden gruppentheoretischen Sachverhalt.

Lemma 4.4. *Sei G eine endliche Gruppe mit neutralem Element e, und sei m das Maximum der Ordnungen aller Gruppenelemente. Dann gilt $a^m = e$ für alle $a \in G$.*

Beweis. Sei $a \in G$ und sei r seine Ordnung, also die kleinste natürliche Zahl, so daß $a^r = e$. Zu zeigen ist, daß r ein Teiler von m ist. Wir führen einen Widerspruchsbeweis und nehmen an, daß $r = st$ gilt, mit $s > 1$ und $1 = \mathrm{ggT}(s, m)$. Dann hat $b := a^t$ offenbar die Ordnung s. Sei weiter $c \in G$ ein Element der Ordnung m und sei u die Ordnung von bc. Dann folgt $b^{um} = (bc)^{um} = e$ und folglich $s \mid um$ bzw. $s \mid u$, da m und s teilerfremd sind. Genauso folgt $c^{us} = (bc)^{us} = e$ und $m \mid u$. Insgesamt erhalten wir $u \geq sm$ im Widerspruch dazu, daß m die maximale Ordnung ist. □

Beweis von Proposition 4.3. Für die maximale Ordnung m der Elemente aus $\mathbb{F}^*$ gilt nach dem letzten Lemma $a^m = 1$ für alle $a \in \mathbb{F}^*$. Das Polynom $x^m - 1 \in \mathbb{F}[x]$ hat daher $q-1$ verschiedene Nullstellen, wobei q die Anzahl der Elemente von $\mathbb{F}$ bezeichne. Nach Korollar 2.18 ist dies nur möglich, falls $m \geq q-1$ gilt. Andererseits ist die Ordnung eines Elementes höchstens $q-1$. Daher muß es Elemente der Ordnung $q-1$ geben. □

Eine wichtige Konsequenz ist, daß zwei endliche Körper gleicher Mächtigkeit sich strukturell nicht unterscheiden.

Proposition 4.5. *Zwei endliche Körper $\mathbb{F}$ und $\bar{\mathbb{F}}$ gleicher Mächtigkeit sind isomorph.*

Beweis. Nach Proposition 4.1 haben $\mathbb{F}$ und $\bar{\mathbb{F}}$ dieselbe Charakteristik. Sei $g(x)$ das Minimalpolynom von $\alpha \in \mathbb{F}$ über $\mathbb{Z}_p$. Es teilt $x^q - x$. Da $x^q - x$ über

$\bar{\mathbb{F}}$ nach (4.2) vollständig in Linearfaktoren zerfällt, hat $g(x)$ auch in $\bar{\mathbb{F}}$ eine Nullstelle $\bar{\alpha}$, und ist wegen seiner Irreduzibilität das Minimalpolynom von $\bar{\alpha}$. Wählen wir nun α als primitives Element von $\mathbb{F}$, so hat $g(x)$ den Grad n, und wir erhalten eine Bijektion

$$a_0 + a_1\alpha + \cdots + a_{n-1}\alpha^{n-1} \;\mapsto\; a_0 + a_1\bar{\alpha} + \cdots + a_{n-1}\bar{\alpha}^{n-1}, \quad a_i \in \mathbb{Z}_p,$$

von $\mathbb{F}$ nach $\bar{\mathbb{F}}$. Diese Abbildung ist ein Isomorphismus, sie verträgt sich mit der Addition wie der Multiplikation, die in den beiden Körpern durch die sich analogen Gleichungen $g(\alpha) = 0$ bzw. $g(\bar{\alpha}) = 0$ gegeben ist. □

4.3 Konstruktion von endlichen Körpern

Wir kehren nun die Überlegungen des letzten Abschnitts um in eine Konstruktion endlicher Körper als Oberkörper von $\mathbb{Z}_p$. Nullstellen von irreduziblen Polynomen stehen nicht mehr zur Verfügung, deswegen muss man sie sich verschaffen.

Allgemein sieht diese Konstruktion der Algebra so aus: Sei $g(x) = x^n + b_{n-1}x^{n-1} + \cdots + b_0$ ein normiertes Polynom vom Grade n, mit Koeffizienten in einem Körper $\tilde{\mathbb{F}}$. Wir nehmen $g(x)$ als irreduzibel an, insbesondere besitzt es keine Nullstellen in $\tilde{\mathbb{F}}$. Daher ‚erfinden' wir eine Nullstelle α von $g(x)$ außerhalb von $\tilde{\mathbb{F}}$ und konstruieren einen Erweiterungskörper $\mathbb{F}$ von $\tilde{\mathbb{F}}$ formal aus den Elementen

$$a_0 + a_1\alpha + a_2\alpha^2 + \cdots + a_{n-1}\alpha^{n-1}, \quad a_i \in \tilde{\mathbb{F}},$$

als Vektorraum über $\tilde{\mathbb{F}}$ der Dimension n mit der Basis $1, \alpha, \ldots, \alpha^{n-1}$. Zusätzlich postulieren wir die Gleichung

$$g(\alpha) \;=\; 0 \quad \text{bzw.} \quad \alpha^n \;=\; -b_{n-1}\alpha^{n-1} - \cdots - b_0.$$

Die Addition und Multiplikation zweier Körperelementen geschieht nach den üblichen Regeln. Bei der Multiplikation können dabei Potenzen α^r entstehen mit einem Exponenten $r \geq n$. Diese Potenzen werden mit Hilfe der Gleichung für α^n eliminiert, bis nur noch Potenzen von α mit Exponenten kleiner als n vorhanden sind. So entsteht, wir man leicht nachprüft, ein Ring. Aus der Irreduzibilität von $g(x)$ folgt wie im Beweis von Proposition 4.2, daß jedes von 0 verschiedene Element ein Inverses besitzt. $\mathbb{F}$ ist also ein Körper. Man spricht von der **Adjunktion** von α an den Körper $\tilde{\mathbb{F}}$ und schreibt für den Erweiterungskörper $\mathbb{F} = \tilde{\mathbb{F}}(\alpha)$.

Beispiel. Die komplexen Zahlen $\mathbb{C}$ entstehen aus den reellen Zahlen $\mathbb{R}$ durch Adjunktion einer Nullstelle i des irreduziblen reellen Polynoms x^2+1. $\mathbb{C}$ besteht aus den Elementen $a+bi,\ a,b\in\mathbb{R}$. i erfüllt die für die imaginäre Zahl charakteristische Gleichung $i^2=-1$. □

Diese Konstruktion läßt sich auch als Restklassenbildung in dem Polynomring $\tilde{\mathbb{F}}[x]$ nach dem von $g(x)$ erzeugten Hauptideal I begreifen. Dann ist $\alpha=\overline{x}$, die von dem Monom x erzeugte Restklasse. Die Gleichung $g(\alpha)=0$ wird nun nicht postuliert, sondern ergibt sich durch Restklassenbildung, denn die Aussagen $g(x)\equiv 0 \bmod I$ bzw. $\overline{g(x)}=0$ und $g(\overline{x})=0$ sind äquivalent. Der konstruierte Körper ist $\tilde{\mathbb{F}}[x]/_I$. – Insgesamt erhalten wir die folgende Aussage.

Proposition 4.6. *Sei $\tilde{\mathbb{F}}$ ein Körper und $g(x)$ ein irreduzibles Polynom mit Koeffizienten in $\tilde{\mathbb{F}}$. Dann gibt es einen $\tilde{\mathbb{F}}$ umfassenden Körper $\mathbb{F}$, in dem $g(x)$ eine Nullstelle besitzt.*

Mit diesem Resultat konstruieren wir nun alle endlichen Körper. Dazu benutzen wir die folgende Aussage.

Proposition 4.7. *In einem Körper der Charakteristik p gilt für alle Elemente a,b*

$$(ab)^p \;=\; a^pb^p, \quad (a\pm b)^p \;=\; a^p\pm b^p.$$

Beweis. Die erste Aussage ist offenbar, die zweite ergibt sich aus der Binomialformel

$$(a\pm b)^p \;=\; \sum_{i=0}^{p}\binom{p}{i}\cdot 1\ a^i(\pm b)^{p-i},$$

denn $\binom{p}{i}$ ist für $i\neq 0,p$ ein Vielfaches von p, und die entsprechenden Summanden verschwinden. □

Proposition 4.8. *Sei p eine Primzahl und n eine natürliche Zahl. Dann gibt es einen Körper $\mathbb{F}$ mit $q=p^n$ Elementen.*

Beweis. Wir konstruieren $\mathbb{F}$ als Körper, in dem das Polynoms $f(x)=x^q-x$ vollständig in Linearfaktoren zerfällt. Dazu zerlegen wir $f(x)$ in $\mathbb{Z}_p[x]$ in irreduzible Faktoren $g_1(x),\ldots,g_r(x)$. Nach der letzten Proposition gibt es einen Erweiterungskörper $\mathbb{F}_1$, in dem $g_1(x)$, und damit $f(x)$ eine Nullstelle hat. Daher besitzt $f(x)$ in $\mathbb{F}_1[x]$ einen Linearfaktor, möglicherweise mehrere, die wir von $f(x)$ abspalten. (Ist $g_1(x)$ bereits linear, so erübrigt sich dieser Schritt.) Den Rest von $f(x)$ zerlegen wir erneut in irreduzible Faktoren und konstruieren einen Körper $\mathbb{F}_2\supset\mathbb{F}_1$ in dem $f(x)$ weitere lineare Faktoren abspaltet.

Nach endlich vielen Erweiterungen zerfällt $f(x)$ schließlich in einem Körper $\mathbb{F}'$ vollständig in Linearfaktoren.

Wir definieren nun $\mathbb{F}$ als die Menge aller Nullstellen von $f(x)$, also derjenigen $a \in \mathbb{F}'$ mit $a^q = a$. Wegen $q \cdot 1 = 0$ gilt $f'(x) = q \cdot 1\ x^{q-1} - 1 = -1$, daher sind alle Nullstellen von $f(x)$ voneinander verschieden (vgl. Proposition 2.20), so daß $\mathbb{F}$ insgesamt q Elemente enthält. $\mathbb{F}$ enthält 0 und 1, und für $a, b \in K$ gilt

$$(ab)^q \ = \ a^q b^q \ = \ ab,$$

daher folgt $ab \in \mathbb{F}$ und $a^{-1} \in \mathbb{F}$. Weiter folgt aus der vorangehenden Proposition

$$(a \pm b)^{p^n} \ = \ ((a \pm b)^p)^{p^{n-1}} \ = \ (a^p \pm b^p)^{p^{n-1}} \ = \ \cdots = a^{p^n} \pm b^{p^n} \ = \ a \pm b.$$

Es gehört also auch $a \pm b$ zu $\mathbb{F}$. Damit ist $\mathbb{F}$ ein Körper mit q Elementen. □

Der konstruierte Körper heißt **Zerfällungskörper** von $x^q - x$. Allgemeiner bezeichnet man als Zerfällungkörper eines Polynoms $f(x) \in \tilde{\mathbb{F}}[x]$ jeden minimalen Körper $\mathbb{F} \supset \tilde{\mathbb{F}}$, so daß $f(x)$ in $\mathbb{F}[x]$ vollständig in Linearfaktoren zerfällt. In der Algebra wird gezeigt, daß Zerfällungkörper immer existieren und bis auf Isomorphie eindeutig sind. – Wir fassen unsere Resultate in dem folgenden Satz zusammen.

Satz 4.9. *Es gibt genau dann einen endlichen Körper $\mathbb{F}$ mit q Elementen, wenn q Potenz einer Primzahl ist. $\mathbb{F}$ ist dann bis auf Isomorphie eindeutig bestimmt.*

Diesen Körper nennt man den **Galois-Feld** $\mathbb{F}_q$.

4.4 BCH-Kodes

In Fortsetzung von Abschnitt 3.4 nutzen wir nun endliche Körper zur Konstruktion von zyklischen Kodes. Vorbereitend betrachten wir eine weitere Möglichkeit zur Konstruktion einer Kontrollmatrix für einen zyklischen Kode K. Sie ist theoretisch wichtig, für das praktische Dekodieren aber weniger geeignet. Sei $g(x) = g_1(x)g_2(x)\cdots g_k(x)$ das erzeugende Polynom von K, zerlegt in seine irreduziblen Teiler $g_1(x), \ldots, g_k(x) \in \mathbb{F}_q[x]$. Wir können dann $g_1(x), \ldots, g_k(x)$ als Minimalpolynome von Elementen $\alpha_1, \ldots, \alpha_l$ in einem Erweiterungskörper von $\mathbb{F}_q$ auffassen. Wir setzen voraus, daß $g_1(x), \ldots, g_k(x)$ alle voneinander verschieden sind. Nach den Eigenschaften von Minimalpolynomen ist dann $k(x) = k_0 + k_1 x + \cdots + k_{n-1}x^{n-1}$ genau dann Vielfaches von $g(x)$, wenn

$$k(\alpha_1) \ = \ k(\alpha_2) \ = \ \cdots \ = \ k(\alpha_l) \ = \ 0$$

gilt. In Matrizenschreibweise lautet die Bedingung

$$H \cdot k^t \;=\; 0,$$

mit $k = (k_0 k_1 \dots k_{n-1})$ und der Matrix

$$H \;:=\; \begin{pmatrix} 1 & \alpha_1 & \alpha_1^2 & \cdots & \alpha_1^{n-1} \\ 1 & \alpha_2 & \alpha_2^2 & \cdots & \alpha_2^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \alpha_l & \alpha_l^2 & \cdots & \alpha_l^{n-1} \end{pmatrix}.$$

Nach Satz 3.6 ist H Kontrollmatrix des Kodes.

Bose, Chaudhuri und Hocquenghem haben nun vorgeschlagen, solche zyklischen Kodes zu betrachten, für die $\alpha_1 = \alpha, \alpha_2 = \alpha^2, \dots, \alpha_l = \alpha^l$ für ein geeignetes α gilt.

Definition 4.10. *Ein zyklischer Kode $K \subset \mathbb{F}^n$ mit erzeugendem Polynom $g(x)$ heißt **BCH-Kode**, falls ein α in einem Erweiterungskörper von $\mathbb{F}$ und eine natürliche Zahl $l < n$ existieren, so daß gilt:*

i) Die Ordnung von α ist n: $\alpha^n = 1$ und $\alpha^j \neq 1$ für $j < n$.

ii) $g(x)$ ist das Produkt der Minimalpolynome von $\alpha, \alpha^2, \dots, \alpha^l$, wobei jedes Minimalpolynom in $g(x)$ als Faktor nur einmal erscheint.

Für einen BCH-Kode ist also

$$H \;:=\; \begin{pmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1} \\ 1 & \alpha^2 & \alpha^4 & \cdots & \alpha^{2(n-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \alpha^l & \alpha^{2l} & \cdots & \alpha^{(n-1)l} \end{pmatrix}.$$

Kontrollmatrix. Dieser Sachverhalt erlaubt den Beweis des folgenden Satzes.

Satz 4.11. *Der Minimalabstand eines BCH-Kodes ist mindestens $l + 1$.*

Für den Beweis benötigen wir die **Vandermonde-Determinante**.

Proposition 4.12. *Für Elemente $\beta_1, \dots, \beta_l$ in einem Körper gilt*

$$\det \begin{pmatrix} \beta_1 & \beta_2 & \cdots & \beta_l \\ \beta_1^2 & \beta_2^2 & \cdots & \beta_l^2 \\ \vdots & \vdots & & \vdots \\ \beta_1^l & \beta_2^l & \cdots & \beta_l^l \end{pmatrix} \;=\; \prod_i \beta_i \cdot \prod_{i<j} (\beta_j - \beta_i).$$

Beweis. Sei A_{ij} die $(l-1)\times(l-1)$-Matrix, die aus einer $l\times l$ Matrix $A=(a_{ij})$ durch Streichen der i-ten Zeile und der j-ten Spalte entsteht. Dann gilt bekanntlich der Laplacesche Entwicklungssatz

$$\det(A) \;=\; \sum_{i=1}^{l}(-1)^{i+l}a_{il}\det(A_{il}).$$

Wir betrachten das Polynom

$$f(x) \;:=\; \sum_{i=1}^{l}(-1)^{i+l}\det(A_{il})\,x^i \;=\; \det\begin{pmatrix} a_{11} & \cdots & a_{1,l-1} & x \\ a_{21} & \cdots & a_{2,l-1} & x^2 \\ \vdots & & \vdots & \vdots \\ a_{l1} & \cdots & a_{l,l-1} & x^l \end{pmatrix}.$$

Es hat im Fall der Vandermonde-Determinante ($a_{ij}=\beta_i^j$) die Nullstellen $0,\beta_1,\ldots,\beta_{l-1}$, daher gilt

$$f(x) \;=\; \det(A_{ll})x(x-\beta_1)\cdots(x-\beta_{l-1}).$$

Es folgt

$$\det\begin{pmatrix} \beta_1 & \cdots & \beta_l \\ \vdots & & \vdots \\ \beta_1^l & \cdots & \beta_l^l \end{pmatrix} \;=\; f(\beta_l) \;=\; \det(A_{ll})\beta_l(\beta_l-\beta_1)\ldots(\beta_l-\beta_{l-1}).$$

Die Behauptung folgt nun per Induktion über l, denn $\det(A_{ll})$ ist wieder eine Vandermonde-Determinante. □

Beweis von Satz 4.11. Nach Voraussetzung des Satzes sind die Elemente der ersten Zeile von H alle voneinander verschieden, damit sind je l Spalten von H linear unabhängig (Vandermonde-Determinante). Die Behauptung folgt daher aus Proposition 3.2. □

Beispiel. Ein Kode, der zwei Fehler korrigiert. Wir konstruieren einen binären BCH-Kode. Ausgangspunkt ist die Zerlegung in irreduzible Faktoren

$$x^{16}-x \;=\; x(x+1)(x^2+x+1)l(x)m(x)n(x)$$

in $\mathbb{Z}_2[x]$ mit

$$l(x):=x^4+x^3+1\ ,\ m(x):=x^4+x+1\ ,\ n(x):=x^4+x^3+x^2+x+1\ .$$

x^2+x+1 ist das einzige quadratische Polynom in $\mathbb{Z}_2[x]$ ohne Nullstellen, und damit das einzige irreduzible quadratische Polynom. Unter den Polynomen *4ten* Grades ohne Nullstellen ist daher nur $x^4+x^2+1=(x^2+x+1)^2$ nicht irreduzibel, und folglich sind $l(x)$, $m(x)$ und $n(x)$ irreduzibel.

Da $x^{16}-x$ über $\mathbb{F}_{16}$ vollständig in Linearfaktoren zerfällt, besitzt auch $m(x)$ in $\mathbb{F}_{16}$ eine Nullstelle α. Nach Proposition 4.2 läßt sich jedes Element aus $\mathbb{F}_{16}$ in eindeutiger Weise als Linearkombination von $1, \alpha, \alpha^2$ und α^3 mit Koeffizienten 0 oder 1 darstellen. Es ist $\alpha^{15}=1$, jedoch $\alpha^3 \neq 1$ und $\alpha^5=\alpha^2+\alpha \neq 0$, daher ist α primitives Element in $\mathbb{K}_{16}$. Als irreduzibles Polynom ist $m(x)$ das Minimalpolynom von α. Gleichzeitig ist $m(x)$ auch das Minimalpolynom von α^2 und α^4, denn es gilt (vgl. Proposition 2.19) $m(\alpha^2)=m(\alpha)^2=0$ und $m(\alpha^4)=m(\alpha^2)^2=0$. Weiter gilt $n(\alpha^3)=0$, $n(x)$ ist also das Minimalpolynom von α^3.

$$g(x) \;:=\; m(x)n(x) \;=\; x^8+x^7+x^6+x^4+1$$

ist damit das erzeugende Polynom eine BCH-Kodes mit $l=4$ und einer Minimaldistanz von mindestens 5, der Kode kann folglich 2 Fehler korrigieren. Die Ordnung von α ist 15, und die Dimension von K ist $m=n-\deg(g)=7$. Es liegt also ein (15,7)-Kode vor, der Kode sendet Wörter der Länge 15, die 7 Informationsstellen haben. k ist genau dann Kodewort, wenn für das zugehörige Polynom $k(x)$

$$k(\alpha) \;=\; k(\alpha^3) \;=\; 0$$

gilt. Genau dann wird $k(x)$ von den Minimalpolynomen $m(x)$ und $n(x)$, und damit von $g(x)$, geteilt.

Kodieren. Für das Kodieren von Wörtern der Länge 7 bestehen verschiedene Möglichkeiten. Dies kann mit der oben angegebenen erzeugenden Matrix geschehen. Übersichtlich ist die folgende Methode. Dem Wort $w=(w_0 \ldots w_6)$ ordnen wir das Polynom

$$k_1(x) \;=\; w_0x^8+w_1x^9+\cdots+w_6x^{14}$$

zu. Wir ergänzen es mit einem Polynom $k_2(x)$ vom Grad höchstens 7 zu

$$k(x) \;=\; k_1(x)+k_2(x),$$

so daß $k(x)$ ein Vielfaches von $g(x)$ wird. Dann gilt für ein Polynom $q(x)$

$$k_1(x) \;=\; q(x)g(x)-k_2(x) \;=\; q(x)g(x)+k_2(x).$$

$k_2(x)$ ist also eindeutig bestimmt als der Rest, der bei Division von $k_1(x)$ durch $g(x)$ übrig bleibt.

Dekodieren. Sei $f = (f_0 \dots f_{14})$ der Vektor der Fehler, es wird anstelle von k also die Nachricht $\widetilde{k} = k + f$ empfangen. Für die zugehörigen Polynome $f(x)$ und $\widetilde{k}(x)$ gilt dann

$$f(\alpha) \;=\; \widetilde{k}(\alpha),\; f(\alpha^2) \;=\; \widetilde{k}(\alpha^2),\; f(\alpha^3) \;=\; \widetilde{k}(\alpha^3)$$

Der Empfänger kann nun mithilfe des Polynoms

$$p(x) \;=\; \widetilde{k}(\alpha)x^2 + \widetilde{k}(\alpha^2)x + \widetilde{k}(\alpha^3) + \widetilde{k}(\alpha)\widetilde{k}(\alpha^2)$$

Fehler korrigieren. Wir unterscheiden drei Fälle.

Fall 1. Gibt es keine Übertragungsfehler, so ist $f(x) = 0$ und $p(x) = 0$.

Fall 2. Bei einem Übertragungsfehler ist $f(x)$ von der Gestalt x^r und damit

$$\begin{aligned} p(x) &= \alpha^r x^2 + \alpha^{2r} x + \alpha^{3r} + \alpha^r \alpha^{2r} \\ &= \alpha^r x(x + \alpha^r). \end{aligned}$$

Fall 3. Bei zwei Übertragungsfehlern gilt $f(x) = x^r + x^s$ mit $r \neq s$. In diesem Fall ergibt sich

$$\begin{aligned} p(x) &= (\alpha^r + \alpha^s)x^2 + (\alpha^{2r} + \alpha^{2s})x \\ &\quad + \alpha^{3r} + \alpha^{3s} + (\alpha^r + \alpha^s)(\alpha^{2r} + \alpha^{2s}) \\ &= (\alpha^r + \alpha^s)(x + \alpha^r)(x + \alpha^s). \end{aligned}$$

Der Empfänger kann diese Fälle unterscheiden, indem er die Nullstellen von $p(x)$ bestimmt. Gleichzeitig kann er in den beiden letzten Fällen r bzw. r, s aus den Nullstellen bestimmen, da α ein primitives Element ist. Er kann also die falsch übertragenen Bits lokalisieren und korrigieren. □

4.5 Spezielle Fälle von BCH-Kodes

Hamming-Kodes

Die Hamming-Kodes ergeben sich bei der Wahl $q = 2$, also $\mathbb{F}_q = \mathbb{Z}_2$, und $n = 2^l - 1$. Wähle α als primitives Element von $\mathbb{F}_{2^l}$, mit Minimalpolynom $g(x) \in \mathbb{Z}_2[x]$. Die Elemente von $\mathbb{F}_{2^l}$ lassen sich eindeutig als Linearkombinationen von $1, \alpha, \alpha^2, \dots, \alpha^{l-1}$ mit Koeffizienten 0 oder 1 darstellen. Daher gibt es eindeutige $h_{ij} \in \mathbb{Z}_2$, so daß die Gleichungen

$$\alpha^j \;=\; \sum_{i=0}^{l-1} \alpha^i h_{ij},$$

$0 \leq j \leq n-1$ gelten. Kodeworte sind durch $k(\alpha) = 0$ charakterisiert, und es gilt

$$k(\alpha) \;=\; \sum_{j=0}^{n-1} k_j \alpha^j \;=\; \sum_{i=0}^{l-1} \left(\sum_{j=0}^{n-1} h_{ij} k_j \right) \alpha^i \;=\; 0$$

genau dann, wenn

$$H \cdot k^t \;=\; \begin{pmatrix} h_{00} & \cdots & h_{0,n-1} \\ \vdots & & \vdots \\ h_{l-1,0} & \cdots & h_{l-1,n-1} \end{pmatrix} \cdot \begin{pmatrix} k_0 \\ \vdots \\ k_{n-1} \end{pmatrix} \;=\; 0.$$

H ist also die Kontrollmatrix des BCH-Kodes mit erzeugendem Polynom $g(x)$. Da α primitives Element ist, durchlaufen die Spalten von H alle möglichen Vektoren aus Nullen und Einsen der Länge l, abgesehen vom Nullvektor. H ist daher bis auf Reihenfolge der Spalten die Kontrollmatrix des $(2^l - 1, 2^l - l - 1)$-Hamming-Kodes, die beiden Kodes sind äquivalent.

Neben $g(\alpha) = 0$ gilt $g(\alpha^2) = g(\alpha)^2 = 0$. Nach Satz 4.11 ist der Mindestabstand des Kodes mindestens $d = 3$, ein Sachverhalt, der uns bereits bekannt ist.

Reed-Solomon-Kodes

BCH-Kodes, für die $n = q - 1$ gilt, heißen Reed-Solomon-Kodes. Nun ist α primitives Element von $\mathbb{F}_q$, und es gilt

$$g(x) \;=\; (x - \alpha)(x - \alpha^2) \ldots (x - \alpha^l).$$

Proposition 4.13. *Reed-Solomon-Kodes sind (n, m)-Blockkodes mit $m = n - l$. Für ihren Mindestabstand gilt*

$$d_{\min} \;=\; l + 1 \;=\; n - m + 1,$$

Reed-Solomon Kodes sind also optimal.

Beweis. Die erste Behauptung folgt aus $l = \deg(g) = n - m$ nach Satz 3.6. Weiter gilt $d_{\min} \geq l + 1$ nach Satz 4.11, und $d_{\min} \leq l + 1$ nach (3.1). □

Ein Reed-Solomon-Kode mit $d_{\min} = 2s + 1$ kann s Fehler korrigieren. Er hat Wörter der Länge $n = q - 1$ und eine Dimension $m = n - d_{\min} + 1 = n - 2s - 2$. Wählen wir $q = 2^k$, so lässt er sich in einen binären Kode umwandeln. Jedes Element aus $\mathbb{F}_q$ kann man dann als k-Vektor mit Symbolen aus $\mathbb{Z}_2$ schreiben. Es entsteht so ein binärer $(kn, k(n-2s))$-Blockkode. Dieser

Kode hat die günstige Eigenschaft, lange Serien aufeinanderfolgender Fehler (‚bursts', z.B. Kratzer auf einer CD) korrigieren zu können, und zwar pro Wort einen Serienfehler, dessen Länge kleiner als $(s-1)k$ ist. Eine solche Serie beeinflussen im Originalkode höchstens s aufeinanderfolgende Buchstaben und wird daher korrigiert. Sind die Fehler dagegen über das Wort verstreut, so ist nur garantiert, daß s Fehler korrigiert werden. Reed-Solomon-Kodes haben z.B. in CD-Spielern Anwendung gefunden.

4.6 Elliptische Kurven über endlichen Körpern

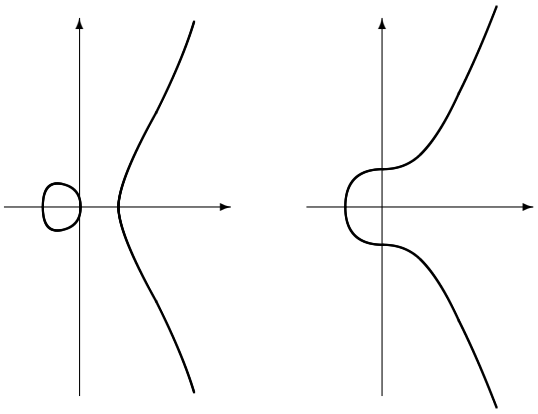
Die wesentliche Eigenschaft einer elliptischen Kurve ist, daß sie eine Gruppenstruktur trägt, was auch für die Kryptographie von Interesse ist. Wir geben eine kurze Einführung in elliptische Kurven über endlichen Körpern.

Definition 4.14. *Sei $\mathbb{F}$ ein Körper von einer Charakteristik $p \neq 2, 3$, und sei $x^3 + rx + s \in K[x]$ ein kubisches Polynom ohne mehrfache Nullstelle. Die zu diesem Polynom gehörige* ***elliptische Kurve über*** *$\mathbb{F}$ besteht aus allen Punkten $(x, y) \in \mathbb{F} \times \mathbb{F}$, die die Gleichung*

$$y^2 = x^3 + rx + s$$

erfüllen, zusammen mit einem weiteren Element O, dem ‚unendlich fernen Punkt'.

Die elliptischen Kurven $y^2 = x^3 - x$ und $y^2 = x^3 + 1$ über den reellen Zahlen sind in der folgenden Zeichnung skizziert.



Bemerkungen.

1. Die Bedingung, daß keine mehrfachen Nullstellen vorkommen, läßt sich in r und s ausdrücken. Sind $\alpha_1, \alpha_2, \alpha_3$ die Nullstellen, so folgt durch Koeffizientenvergleich aus $x^3 + rx + s = (x - \alpha_1)(x - \alpha_2)(x - \alpha_3)$

$$\alpha_1 + \alpha_2 + \alpha_3 \;=\; 0, \quad \alpha_1\alpha_2 + \alpha_2\alpha_3 + \alpha_3\alpha_1 \;=\; r, \quad \alpha_1\alpha_2\alpha_3 \;=\; -s.$$

   Eine Rechnung zeigt, daß sich die *Diskriminante* $D = \prod_{i<j}(\alpha_i - \alpha_j)^2$ aus diesen Ausdrücken zusammensetzen läßt (vgl. van der Waerden, Algebra 1, §33),

$$-D = 4r^3 + 27s^2.$$

   $x^3 + rx + s$ hat also genau dann keine mehrfachen Nullstellen, falls $4r^3 + 27s^2 \neq 0$ gilt.

2. In Körpern der Charakteristik 2 oder 3 ist die Bedingung an eine elliptische Kurve passend abzuwandeln. □

Die fundamentale Eigenschaft von elliptischen Kurven ist, daß man sie als abelschen Gruppe betrachten kann. Das Additionsgesetz läßt sich kurz zu der folgenden Aussage zusammenfassen: *Die 2 oder 3 Punkte einer elliptischen Kurve, die auf einer vorgegebenen Geraden ℓ liegen, summieren sich zu O auf, dem unendlich fernen Punkt*, dabei sind die Punkte in ihrer Vielfachheit zu berücksichtigen. Wir präzisieren dies nun. Unter einer Geraden ℓ verstehen wir alle Punkte (x, y), die einer Gleichung der Gestalt $ax+by+c = 0$ genügen mit $a \neq 0$ oder $b \neq 0$ ($a, b, c \in \mathbb{F}$). Es gibt verschiedene Fälle zu unterscheiden.

1. Sei $b = 0$. Dann können wir ohne Einschränkung $a = -1$ wählen, ℓ ist also durch die Gleichung $x = c$ gegeben. Der Schnitt von ℓ mit der elliptischen Kurve enthält die Punkte (x, y) mit $y^2 = c^3 + rc + s$. Wenn der Schnitt nicht leer ist, enthält er genau zwei Punkte $P = (x, y)$ und $Q = (c, -y)$. Wir setzen dann $P + Q = O$. Im Fall $y = 0$ fallen P und Q zusammen, dann lautet die Gleichung $P + P = O$.

2. Sei nun $b \neq 0$. Dann können wir ohne Einschränkung $b = -1$ setzen, also die Gerade $y = ax+c$ betrachten. Seien weiter $P = (x_1, y_1) \neq Q = (x_2, y_2)$ zwei Punkte der elliptischen Kurve, die auf der Geraden liegen. Es folgt $x_1 \neq x_2$ und

$$a \;=\; \frac{y_2 - y_1}{x_2 - x_1}, \quad c \;=\; y_1 - \frac{y_2 - y_1}{x_2 - x_1}\, x_1.$$

   In diesem Fall gibt es auf der Geraden einen weiteren Punkt der elliptischen Kurve. Alle Schnittpunkte (x, y) erfüllen nämlich die Gleichung

$$f(x) \;:=\; x^3 + rx + s - (ax + c)^2 \;=\; 0.$$

Dieses kubische Polynom hat die beiden Nullstellen x_1, x_2. Es zerfällt daher vollständig in Linearfaktoren und besitzt eine weitere Nullstelle x_3, und auf ℓ liegt genau ein weiterer Punkt $R = (x_3, -y_3)$ der elliptischen Kurve (für spätere Zwecke erhält y_3 ein Minuszeichen). Seine Koordinaten lassen sich aus der Gleichung $x_1 + x_2 + x_3 = a^2$ (dem Koeffizient von x^2 in dem Polynom $f(x)$) bestimmen:

$$\begin{aligned} x_3 &= \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_1 - x_2, \\ y_3 &= -ax_3 - c \\ &= \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x_1 - x_3) - y_1. \end{aligned} \tag{4.3}$$

Insgesamt befinden sich auf der Geraden drei Punkte P, Q, R der elliptischen Kurve, die sich zu O addieren: $P + Q + R = O$. Es ist möglich, daß sich $R = P$ oder $R = Q$ ergibt, dann wird der entsprechende Punkt in der Gleichung $R + P + Q = O$ zweimal aufgeführt.

3. Die soeben betrachtete Gerade ist durch die beiden Punkten $P \neq Q$ bestimmt. Jetzt behandeln wir den Fall einer Geraden $y = ax + c$, deren Lage durch einen Punkt $P = (x_1, y_1)$ gegeben ist, dem eine mehrfache Nullstelle x_1 des kubischen Polynoms $f(x)$ entspricht. Dann gilt zusätzlich zu $f(x_1) = 0$ die Gleichung $0 = f'(x_1) = 3x_1^2 + r - 2(ax_1 + c)a$, also
$$a = \frac{3x_1^2 + r}{2y_1}, \quad c = y_1 - \frac{3x_1^2 + r}{2y_1}\, x_1.$$
(Im Fall $\mathbb{F} = \mathbb{R}$ bedeutet die Bedingung, daß die Gerade eine Tangente der elliptischen Kurve ist.) Wie oben hat nun $f(x)$ neben der doppelten Nullstelle x_1 eine weitere Nullstelle x_3. Wir können sie aus der Gleichung $2x_1 + x_3 = a^2$ bestimmen,
$$\begin{aligned} x_3 &= \Big(\frac{3x_1^2 + r}{2y_1}\Big)^2 - 2x_1, \\ y_3 &= \Big(\frac{3x_1^2 + r}{2y_1}\Big)(x_1 - x_3) - y_1. \end{aligned} \tag{4.4}$$
Die Gleichung für die Punkte $P = (x_1,y_1)$ und $R = (x_3, -y_3)$ der elliptischen Kurve lautet nun $P+P+R = O$, P wird also doppelt berücksichtigt. □

Die Sprechweise vom unendlich fernen Punkt begründet sich aus der Vorstellung, daß O unendlich fern auf allen senkrechten (durch $b = 0$ gegebenen) Geraden liegt. Das hat den Vorteil, daß nun auf *allen* Geraden genau

drei Punkte liegen (gezählt in der jeweiligen Vielfachheit). Dieser Vorstellung kann man einen mathematisch präzisen Sinn geben, das Stichwort ist ‚projektiver Abschluß' der Kurve (vgl. das Buch von Koblitz).

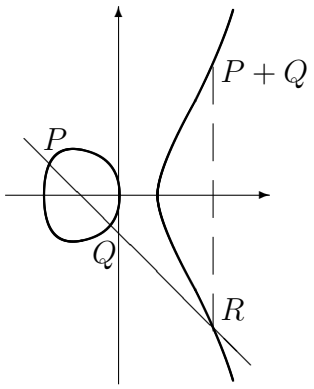
Wir wollen nun Punkte auf einer elliptischen Kurve so addieren, daß die angegebenen Gleichungen gültig bleiben, dabei soll O die Rolle des neutralen Elements übernehmen. Aus 1. ergibt sich der entgegengesetzte Punkt eines Punktes. Wir setzen

$$-P \;:=\; (x, -y), \quad \text{falls } P \;=\; (x, y).$$

$-P$ entsteht aus P durch Spiegelung an der x-Achse. Die Summe von P und Q erhalten wir durch Umformung der Gleichung $P + Q + R = O$ zu $P + Q = -R$. Wir setzen also

$$P + Q \;:=\; (x_3, y_3), \quad \text{falls } P = (x_1, y_1),\; Q = (x_2, y_2),$$

dabei sind x_3 und y_3 durch die Formeln (4.3) bzw. (4.4) gegeben (je nachdem ob $P \neq Q$ oder $P = Q$ gilt).



Beispiel. Die Punkte $P = (-3, 9)$ und $Q = (-2, 8)$ liegen auf der elliptischen Kurve $y^2 = x^3 - 36x$ über den rationalen Zahlen. Es gilt $P + Q = (6, 0)$ und $2 \cdot P = (25/4, -35/8)$. □

Daß man eine elliptische Kurve auf diese Weise zu einer abelschen Gruppe macht, ist nicht offensichtlich, insbesondere ist nicht evident, daß das Assoziativitätsgesetz $(P + Q) + R = P + (Q + R)$ gilt. Man kann es aus den angegebenen Formeln in etwas mühsamer Rechnung bestätigen. (Für den Körper der reellen oder komplexen Zahlen gibt es andere Beweise. Sie benutzen Argumente aus der projektiven Geometrie oder der Funktionentheorie und gewähren einen tieferen Einblick in die Gruppenstruktur elliptischer Kurven.) Wir fassen die Diskussion in einem Satz zusammen.

Satz 4.15. *Jede elliptische Kurve wird durch die in (4.3) bzw. (4.4) gegebene Addition eine Gruppe. Das neutrale Element ist der unendlich ferne Punkt O, und das inverse Element von (x, y) ist $(x, -y)$.*

Elliptische Kurven über den komplexen wie über den rationalen Zahlen sind ein klassischer Untersuchungsgegenstand der Mathematik. Wir betrachten im folgenden elliptische Kurven über den Galois-Feldern $\mathbb{F}_q$. Die Anzahl N der Punkte einer elliptischen Kurve über $\mathbb{F}_q$ ist offensichtlich höchstens $2q + 1$. Neben O gibt es nämlich für jedes $x \in \mathbb{F}_q$ höchstens zwei Punkte (x, y) und $(x, -y)$ auf der elliptischen Kurve. Dazu muß $x^3 + rx + s$ ein quadriertes Element von $\mathbb{F}_q$ sein. Da in dem Fall, daß q ungerade ist, genau die Hälfte der Elemente von $\mathbb{F}_q^*$ Quadrate sind, wird man erwarten, daß eine elliptische Kurve über $\mathbb{F}_q$ etwa q Elemente enthält. Genauer gilt der **Satz von Hasse**:

$$|N - (q + 1)| \ \leq \ 2\sqrt{q}.$$

Eine elliptische Kurve zusammen mit einem Punkt $P = (x, y)$ kann man sich leicht verschaffen. Man wähle zufällige Elemente x, y, r aus $\mathbb{F}_q$ und setze $s = y^2 - (x^3 + rx)$. Dann ist offenbar $P = (x, y)$ ein Punkt auf der Kurve, die durch die Gleichung $y^2 = x^3 + rx + s$ gegeben ist. Es besteht eine gute Chance, daß die Diskriminante $4r^3 + 27s^2 \neq 0$ ist, andernfalls treffe man eine neue Wahl.

Das Schlüsselsystem von Diffie-Hellman

Elliptischen Gruppen über endlichen Körpern sind in der Kryptographie auf Interesse gestoßen. Ein Grund ist, daß man mit ihrer Hilfe Abbildungen konstruieren kann, die leicht zu berechnen sind, die sich aber nur mit erheblichen Rechenaufwand umkehren lassen. Die Situation ist analog zum diskreten Logarithmus im Restklassenring $\mathbb{Z}_m$. In elliptischen Kurven benötigt man zur Berechnung des Punktes

$$i \cdot P \ := \ \underbrace{P + \cdots + P}_{i-\text{mal}}$$

größenordnungsmäßig $O(\log i)$ Rechenschritte. Wie beim Potenzieren modulo m schreibe man dazu i in Dualdarstellung, $i = \sum d_j 2^j$ mit $d_j = 0, 1$, berechne $P_j = 2^j \cdot P = 2 \cdot P_{j-1}$ und summiere alle P_j mit $d_j = 1$ auf. Dagegen kann man nur in speziellen Fällen schneller als in $O(i)$ Schritten zu einem vorgegebenen Q ein Punkt P bestimmen, so daß $Q = i \cdot P$ gilt. Dies macht man sich in der Kryptographie zunutze.

Öffentliche Kryptosysteme sind im Vergleich zu klassischen Chiffriersystemen relativ rechenaufwendig. Deswegen versucht man, beide Systeme zu

kombinieren. Nachrichten werden mit einem schnellen klassischen Chiffriersystem ausgetauscht, bei dem man zum Kodieren und Dekodieren der Nachrichten einen Schlüssel braucht. Für den Austausch des Schlüssels benutzt man ein sicheres öffentliches Kryptosystem.

Einen ersten detaillierten Vorschlag haben Diffie und Hellman gemacht. Ihr System basiert in seiner ursprünglichen Form auf dem diskreten Logarithmus. Wir nehmen an, daß der Schlüssel Element eines großen endlichen Körper $\mathbb{F}_q$ ist. q ist öffentlich bekannt. Weiter einigt man sich auf ein Element g in $\mathbb{F}_q$, idealerweise ein primitives Element von $\mathbb{F}_q^*$. Auch g wird öffentlich bekanntgemacht. Schlüssel werden aus g gebildet: Ein Teilnehmer A wählt sich zufällig eine natürliche Zahl a zwischen 0 und $q-1$, die er geheim hält. Gleichzeitig veröffentlicht er das Element g^a in $\mathbb{F}_q^*$. Ein weiterer Teilnehmer B macht dasselbe, er wählt seine Geheimzahl b und veröffentlicht g^b. Der Geheimschlüssel für den Nachrichtenaustausch zwischen A und B ist dann g^{ab}. Beide können ihn leicht berechnen. A etwa potenziert die öffentlich zugängliche Größe g^b mit seiner Geheimzahl a, dazu sind keinerlei Absprachen zwischen A und B nötig. Eine dritte Partei hat dagegen auf g^{ab} keinen Zugriff. Man kann versuchen, sich a und b aus g^a und g^b zu verschaffen, um dann g^{ab} zu berechnen. Im allgemeinen ist aber die Berechnung solcher ‚diskreter Logarithmen' bei ausreichend großer Wahl von q rechnerisch nicht zu bewältigen.

In neuerer Zeit hat man anstelle von $\mathbb{F}_q$ auch elliptische Kurven über $\mathbb{F}_q$ herangezogen. An die Stelle von g tritt ein Punkt P der Kurve, und g^a wird ersetzt durch $a \cdot P$. Man verspricht sich dadurch Vorteile in der Sicherheit des Verfahrens. Für binäre Kodes ist es praktisch, einen Körper $\mathbb{F}_{2^r}$ zugrundezulegen. Dagegen spricht, daß in diesen speziellen Körpern diskrete Logarithmen mit vergleichsweise geringerem Rechenaufwand bestimmt werden können, r müßte also sehr groß gewählt werden. Nach dem heutigen Kenntnisstand sind dagegen geeignete elliptische Kurven über $\mathbb{F}_{2^r}$ sicher. Ob das RSA-Schema oder aber das Schema von Diffie und Hellman eine größere Sicherheit bietet, muß die Zukunft entscheiden.

Literatur

N. Koblitz (1994): A Course in Number Theory and Cryptography

Kapitel 5

Lineares Programmieren

5.1 Grundbegriffe

In diesem Kapitel sind mit Vektoren immer Spaltenvektoren gemeint. Für Vektoren $b, b' \in \mathbb{R}^n$ schreiben wir $b \leq b'$, wenn für alle Komponenten $b_i \leq b'_i$, $i = 1, \ldots, n$ gilt. Insbesondere bedeutet $b \geq 0$, dass alle b_i nichtnegativ sind.

Sei A reelle $m \times n$-Matrix, und seien $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$. Dann nennt man das System

$$\begin{gathered} Ax \leq b \ , \quad x \geq 0 \ , \quad x \in \mathbb{R}^n \ , \\ c^t x \ = \ \max \end{gathered}$$

ein lineares Programm, genauer ein **standard Maximum Programm**. $x \in \mathbb{R}^n$ heißt **zulässig**, falls es die angegebenen Bedingungen, also $Ax \leq b$, $x \geq 0$ erfüllt. $c^t x = c_1 x_1 + \cdots + c_n x_n$ heißt **Zielfunktional** und

$$\sup_{x \text{ ist zulässig}} c^t x$$

Wert des Programms. Ein zulässiges x heißt **optimal**, falls $c^t x$ gleich dem Wert des linearen Programms ist. Die Menge aller zulässigen Punkte bezeichnen wir mit K_{zul}, und aller optimalen Punkte mit K_{opt}.

Anschaulich kann man sich unter der Menge K_{zul} aller zulässigen Punkte ein konvexes Polyeder im $\mathbb{R}^n$ vorstellen. Ist es beschränkt, so wird das Zielfuktional seinen maximalen Wert in einer Ecke annehmen. Für eine algorithmische Lösung ist es wesentlich, diese Vorstellung geeignet algebraisch umzuformulieren. Wir kommen darauf zurück.

Manchmal ist es günstig, linearen Programmen eine andere Gestalt zu

geben. Ein Programm der Gestalt

$$\begin{aligned} Ax = b\ , \quad x \geq 0\ , \quad x \in \mathbb{R}^n\ , \\ c^t x \ = \ \max \end{aligned}$$

heißt **kanonisches Maximum Programm**. Die Begriffe von Zulässigkeit und Optimalität von Vektoren sind geeignet anzupassen. Jedes kanonische Maximum Programm kann man ohne weiteres als standard Programm formulieren, nämlich als

$$\begin{aligned} Ax \leq b\ , \quad -Ax \leq -b\ , \quad x \geq 0\ , \\ c^t x \ = \ \max\ . \end{aligned}$$

Umgekehrt gelangt man von dem standard Maximum Programm

$$\begin{aligned} Ax \leq b\ , \quad x \geq 0\ , \quad x \in \mathbb{R}^n\ , \\ c^t x \ = \ \max \end{aligned}$$

durch Einführen von **Schlupfvariablen** $z \in \mathbb{R}^m$ zu dem kanonischen Programm mit demselben Wert

$$\begin{aligned} Ax + z = b\ , \quad x \geq 0\ , z \geq 0 \\ c^t x + 0^t z = \max\ , \end{aligned}$$

bzw. (mit der $m \times m$ Einheitsmatrix E)

$$\begin{aligned} \big(A, E\big) \cdot \begin{pmatrix} x \\ z \end{pmatrix} = b\ , \ \begin{pmatrix} x \\ z \end{pmatrix} \geq 0\ , \\ (c, 0)^t \cdot \begin{pmatrix} x \\ z \end{pmatrix} = \max\ . \end{aligned}$$

Ganz analog betrachtet man **standard Minimum Programme**

$$\begin{aligned} Ax \geq b\ , \quad x \geq 0\ , \quad x \in \mathbb{R}^n\ , \\ c^t x \ = \ \min \end{aligned}$$

und **kanonische Minimum Programme**

$$\begin{aligned} Ax = b\ , \quad x \geq 0\ , \quad x \in \mathbb{R}^n\ , \\ c^t x \ = \ \min\ . \end{aligned}$$

Der Wert dieser Programme ist

$$\inf_{x \in M_z} c^t x\ .$$

5.2 Dualität

Die linearen Programme

$$\begin{aligned} Ax \leq b\ , \quad x \geq 0\ , \quad x \in \mathbb{R}^n\ , \\ c^t x \ = \ \max \end{aligned} \tag{P}$$

und

$$\begin{aligned} y^t A \geq c^t\ , \quad y \geq 0\ , \quad y \in \mathbb{R}^m\ , \\ y^t b \ = \ \max \end{aligned} \tag{D}$$

heißen **dual**. (P) wird als **Primalprogramm**, (D) als **Dualprogramm** bezeichnet. Natürlich kann man (D), lässt man die Dualität einmal beiseite, auch mit der Ungleichung $A^t y \leq c$ und dem Zielfunktional $b^t y$ formulieren.

Zwischen den Werten von (P) und (D) besteht ein enger Zusammenhang. Einen ersten Einblick gewährt der **schwache Dualitätssatz**.

Proposition 5.1. *Seien x, y zulässige Lösungen von (P) und (D). Dann gilt*

$$c^t x \leq y^t b\ .$$

Aus den Ungleichungen in (P) und (D) erhalten wir nämlich $c^t x \leq y^t A x \leq y^t b$.

Es folgt: Der Wert des Primalprogramms lässt sich von unten durch zulässige Lösungen von (P) einschachteln, und von oben durch zulässige Lösungen von (D). Algorithmisch gesehen spricht man deswegen von einem *gut charakterisierten Problem*.

Der schwache Dualitätssatz lässt sich wesentlich verschärfen, zum **Hauptsatz über das Lineare Programmieren**.

Satz 5.2. *Entweder besitzen (P) und (D) beide zulässige Lösungen. Dann haben sie auch optimale Lösungen x_{opt}, y_{opt}, und beide Programme haben denselben endlichen Wert*

$$c^t x_{opt} = y_{opt}^t b\ .$$

Oder aber eines der Programme (P) und (D) ist nicht lösbar. Dann hat das andere Programm keine optimale Lösung, es ist entweder nicht lösbar oder hat den Wert $\pm\infty$.

Für das Primalprogramm können wir damit festhalten: Besitzt (P) zulässige Lösungen, und ist das Zielfunktional auf K_{zul} nach oben beschränkt, so hat (P) auch optimale Lösungen.

Zum Beweis benötigen wir folgende Proposition.

Proposition 5.3. *Eine der folgenden beiden Möglichkeiten trifft zu:*

(A) $Ax \leq b\ ,\ x \geq 0$ *ist lösbar.*

(B) $y^tA \geq 0\ ,\ y \geq 0\ ,\ y^tb < 0$ *ist lösbar.*

Es ist leicht zu sehen, dass (A) und (B) nicht gleichzeitig gelten können.

Beweis. Sei $K := \{Ax + u \in \mathbb{R}^m : x \geq 0, u \geq 0\}$. Wir unterscheiden zwei Fälle. Entweder gilt $b \in K$, dann trifft offenbar (A) zu. Oder $b \notin K$. Dann hat b positiven Abstand von K. Anders ausgedrückt, die Mengen

$$K_1 := \{Ax : x \geq 0\}\ ,\ K_2 := \{b - u : u \geq 0\}$$

haben positiven Abstand. Offenbar sind K_1 und K_2 konvex, deswegen gibt es nach den Prinzipien der konvexen Analysis eine trennende Hyperebene zwischen K_1 und K_2, d.h. eine Linearform $\ell(v) = y^tv$ und eine reelle Zahl α, so dass

$$\ell(v) > \alpha \text{ für alle } v \in K_1 \quad , \quad \ell(v) < \alpha \quad \text{für alle } v \in K_2\ .$$

Insbesondere gilt $0 = \ell(0) > \alpha$. Es folgt $y^tAx = \ell(Ax) > \alpha$ für alle $x \geq 0$. Dies ist nur im Fall $y^tA \geq 0$ möglich. Weiter folgt $y^tb - \lambda y_i = \ell(b - \lambda e_i) < \alpha < 0$ für alle $\lambda \geq 0$, dabei bezeichnen $e_1, \ldots, e_n$ die kanonischen Einheitsvektoren des $\mathbb{R}^n$. Mit $\lambda \to \infty$ folgt $y_i \geq 0$, also $y \geq 0$. Schließlich folgt $y^tb = \ell(b) < \alpha < 0$, die drei Bedingungen aus (B) sind also alle erfüllt. □

Beweis des Hauptsatzes. Nehmen wir zunächst an, dass (P) und (D) zulässige Lösungen x^*, y^* besitzen. In Anbetracht des schwachen Dualitätssatzes langt es dann zu zeigen, dass das System

$$\begin{aligned} Ax &\leq b\ ,\ x \geq 0 \\ A^ty &\geq c\ ,\ y \geq 0 \\ c^tx - b^ty &\geq 0 \end{aligned}$$

bzw.

$$\begin{pmatrix} A & 0 \\ 0 & -A^t \\ -c^t & b^t \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} \leq \begin{pmatrix} b \\ -c \\ 0 \end{pmatrix} \quad , \quad \begin{pmatrix} x \\ y \end{pmatrix} \geq 0$$

lösbar ist. Andernfalls gäbe es nach der letzten Proposition Vektoren $u \in \mathbb{R}^m$, $v \in \mathbb{R}^n$, $w \in \mathbb{R}$, so dass

$$(u^tv^tw) \cdot \begin{pmatrix} A & 0 \\ 0 & -A^t \\ -c^t & b^t \end{pmatrix} \geq 0\ ,\ u \geq 0\ ,\ v \geq 0\ ,\ w \geq 0\ ,\ b^tu < c^tv\ ,$$

bzw.

$$u^t A \geq wc^t \ , \ Av \leq wb \ , \ \ u \geq 0 \ , \ v \geq 0 \ , \ w \geq 0 \ , \ u^t b < c^t v \ .$$

Diese Ungleichungen können jedoch bei Existenz von zulässigen Lösungen von (P) und (D) nicht gleichzeitig bestehen: Gilt $w = 0$, so folgt der Widerspruch

$$0 \leq u^t A x^* \leq u^t b < c^t v \leq (y^*)^t A v \leq 0 \ ,$$

und gilt $w > 0$, so sind $x := w^{-1}v$, $y = w^{-1}u$ zulässige Lösungen von (P) und (D), und nach dem schwachen Dualitätssatz ergibt sich der Widerspruch $c^t v = w(c^t x) \leq w(y^t b) = u^t b$. Damit ist der erste Teil des Satzes bewiesen.

Zum zweiten Teil: Nehmen wir an, dass (P) keine Lösung hat. Nach der Proposition gibt es dann ein y mit $y^t A \geq 0$, $y \geq 0$ und $y^t b < 0$. Besitzt nun (D) eine zulässige Lösung y^*, so ist auch $y^* + \lambda y$ für $\lambda \geq 0$ zulässig für (D), und es erweist sich, dass (D) den Wert $-\infty$ hat. In jedem Fall hat dann (D) keine optimale Lösung. □

5.3 Eckpunkte

Die Anschauung sagt, dass das Zielfunktional eines linearen Programmes seinen Extremalwert in Ecken des zulässigen Bereichs annimmt. Dies wollen wir präzisieren. Wir gehen in diesem Abschnitt von einem linearen Programm

$$\begin{gathered} Ax = b \ , \quad x \geq 0 \ , \quad x \in \mathbb{R}^n \ , \\ c^t x \ = \ \max \end{gathered} \tag{K}$$

in kanonischer Form aus.

Definition 5.4. *Sei $K \subset \mathbb{R}^n$ konvexe Menge. Dann heißt $x \in K$* **Ecke** *oder* **Extremalpunkt**, *falls für alle $x', x'' \in K$ und alle $0 < \lambda < 1$ gilt*

$$x = \lambda x' + (1 - \lambda) x'' \quad \Rightarrow \quad x' = x'' \ .$$

Sowohl die Menge K_{zul} der zulässigen Punkte wie die Menge K_{opt} der optimalen Punkte sind konvexe Mengen.

Für $x \in K_{zul}$ sei nun

$$Z_x := \{j : j = 1, \ldots, n \ , \ x_j > 0\} \ .$$

Mit

$$a_j := (a_{1j}, \ldots, a_{mj})^t \ , \ j = 1, \ldots, n$$

bezeichnen wir die Spalten von A. Ist x zulässig, so sei

$$A_x := (a_j)_{j \in Z_x}$$

die Matrix, die aus den Spalten a_j mit $x_j > 0$ besteht. Setzen wir für zulässiges x seinen reduzierten Vektor als

$$r_x := (x_j)_{j \in Z_x} \ ,$$

so erhalten wir für das Lineare Programm (K)

$$A_x r_x = b \ .$$

Proposition 5.5. *Ist K_{zul} nicht leer, so besitzt K_{zul} Ecken.*

Beweis. Wir wählen x so, dass Z_x möglichst wenige Elemente hat. Ist x keine Ecke, so gibt es $x', x'' \in K_{zul}$, $x' \neq x''$, und $0 < \lambda < 1$, so dass $x = \lambda x' + (1-\lambda)x''$. Aus $x, x', x'' \geq 0$ folgt $Z_{x'}, Z_{x''} \subset Z_x$. Dann gilt auch für reelles ρ und für

$$x_\rho := x + \rho(x' - x'')$$

$Z_{x_\rho} \subset Z_x$, und x_ρ ist zulässige Lösung von (K), wenn ρ dem Absolutbetrag nach ausreichend klein ist. Wegen $x' \neq x''$ ist bei passender Wahl von ρ jedoch $Z_{x_\rho} \neq Z_x$, im Widerspruch zur Wahl von x. Deswegen ist x Ecke von K_{zul}. □

Proposition 5.6. *$x \in K_{zul}$ ist genau dann Ecke, wenn die Spalten von A_x linear unabhängig sind.*

Bew. Ist x keine Ecke, $x = \lambda x' + (1-\lambda)x''$ mit $x' \neq x''$ und $Ax' = Ax''$, so folgt $x'_j = x''_j = 0$ für $x_j = 0$ und folglich

$$\sum_{j \in Z_x} (x'_j - x''_j) a_j = \sum_j (x'_j - x''_j) a_j = Ax' - Ax'' = 0.$$

Wegen $x' \neq x''$ sind also Spalten von A_x linear abhängig. Umgekehrt bestehe die lineare Abhängigkeit

$$\sum_{j \in Z_x} v_j a_j = 0.$$

Wir setzen $v_j = 0$ für $x_j = 0$, $v = (v_1, \ldots, v_n)^t$ und $\rho > 0$. Für

$$x' := x + \rho v \quad , \quad x'' := x - \rho v$$

gilt dann, $Ax' = Ax'' = b$, $x = x'/2 + x''/2$ und, sofern ρ ausreichend klein ist, $x', x'' \geq 0$. Also ist x kein Eckpunkt. □

Wir zeigen nun, dass K_{zul} die Gestalt eines Polyeders hat.

Proposition 5.7. *K_{zul} hat endlich viele Ecken.*

Beweis. Es gibt endlich viele $Z \subset \{1, \ldots, n\}$, so dass die Untermatrix $A_Z = (a_j)_{j \in Z}$ linear unabhängige Spalten hat, und zu jedem solchen Z höchstens einen Eckpunkt x mit $Z_x = Z$, gegeben durch das Gleichungssystem

$$A_Z r_x = b \quad , \quad x_j = 0 \text{ für alle } j \notin Z \ ,$$

das wegen der linearen Unabhängigkeit höchstens eine Lösung hat. □

Auf der nachfolgende Aussage beruhen die Algorithmen zur Bestimmung optimaler Lösungen.

Proposition 5.8. *Hat (K) optimale Lösungen, so sind unter diesen Lösungen auch Ecken von K_{zul}.*

Beweis. Sei w der Wert von (K). Dann ist die Menge der optimalen Lösungen von (K) gerade die Menge der zulässigen Lösungen des kanonischen Programms

$$\begin{pmatrix} A \\ c^t \end{pmatrix} \cdot x \ = \ \begin{pmatrix} b \\ w \end{pmatrix} \quad , \quad x \geq 0.$$

Wir können die Resultate über K_{zul} also auf K_{opt} übertragen und feststellen, dass die konvexe Menge K_{opt} Ecken besitzt.

Wir zeigen, dass es sich dabei um Ecken von K_{zul} handelt. Sei x Ecke von K_{opt} und $x = \lambda x' + (1-\lambda)x''$ mit $x', x'' \in K_{zul}$ und $\lambda \in (0,1)$. Es folgt

$$c^t x = \lambda c^t x' + (1-\lambda) c^t x'' \ .$$

Wegen der Optimalität von x gilt $c^t x = c^t x' = c^t x''$, daher folgt $x', x'' \in K_{opt}$. Nach Annahme folgt $x' = x''$. Damit ist x auch Ecke von K_{zul}. □

5.4 Ganzzahliges Programmieren

In einer Anzahl von Problemstellungen sucht man Lösungen x von Linearen Programmen mit ganzzahligen Komponenten, oder noch spezieller mit Komponenten, die nur 0 oder 1 sein dürfen. Wir betrachten jetzt also Programme der Gestalt

$$\begin{aligned} Ax \leq b \ , \quad x \in \{0,1\}^n \ , \\ c^t x \ = \ \max \ . \end{aligned}$$

Beispiel. Überdecken von Mengen. Wir betrachten eine m-elementige Menge I und Teilmengen $S_1, \ldots, S_n$, die I vollständig überdecken, also

$$\bigcup_{j=1}^{n} S_j \;=\; I \;.$$

Seien $c_1, \ldots, c_n$ nicht-negative ganze Zahlen. c_j wird interpretiert als Kosten für S_j. Die Aufgabe besteht nun darin, eine Teilüberdeckung von I mit möglichst geringen Kosten zu finden, d.h. paarweise verschiedene Zahlen $1 \leq j_1, \ldots, j_k \leq n$ zu bestimmen, so dass

$$\bigcup_{r=1}^{k} S_{j_r} \;=\; I$$

gilt und

$$\sum_{r=1}^{k} c_{j_r}$$

möglichst klein ist. Wir betrachten folgenden **gierigen (greedy) Algorithmus** zum Überdecken von I:

Sind $S_{j_1}, \ldots, S_{j_{l-1}}$ schon ausgewählt, und gilt $I_l := \bigcup_{r=1}^{l-1} S_{j_r} \neq I$, so wähle $j \neq j_1, \ldots, j_{l-1}$ so, dass $c_j/|S_j - I_l|$ möglichst klein ist. Setze $j_l := j$.

Definieren wir (in Abhängigkeit von diesem Algorithmus) den *Preis* von $i \in I$ als

$$p(i) := \frac{c_{j_l}}{|S_{j_l} - I_l|} \quad \text{für alle } i \in S_{j_l} - I_l \;,$$

so lässt sich der Algorithmus kurz so in Worte fassen: Nehme immer solch ein S_j neu in die Überdeckung auf, dessen Elemente einen möglichst geringen Preis haben. Ist I vollständig überdeckt, so bricht der Algorithmus ab.

Wir wollen seine Güte untersuchen und zeigen, dass der gierige Algorithmus die geringstmöglichen Kosten einer Überdeckung höchstens um den Faktor

$$h_n \;:=\; 1 + \frac{1}{2} + \cdots + \frac{1}{n}$$

übertrifft.

Dazu formulieren wir das zugehörige ganzzahlige Programm

$$\sum_{j : S_j \ni i} x_j \;\geq\; 1 \text{ für alle } i \in I \;,\; x \in \{0,1\}^n \;,$$

$$\sum_{j} c_j x_j \;=\; \min \;.$$

$x_j = 1$ bedeutet, dass S_j in die Teilüberdeckung aufgenommen wird. Durch die Ungleichung ist garantiert, dass I vollständig überdeckt wird. Die zugehörige Matrix A hat die Einträge $a_{ij} = 1$ oder 0, je nachdem, ob $i \in S_j$ oder $i \notin S_j$ gilt. b ist hier der Vektor aus lauter Einsen.

Um Anschluss an die Theorie linearer Programme zu erhalten, formulieren wir das entsprechende (durch Relaxation erhaltene) standard Minimal Programm

$$\sum_{j:S_j \ni i} x_j \ \geq \ 1 \text{ für alle } i \in I \ , \ x \geq 0 \ ,$$

$$\sum_j c_j x_j \ = \ \min \ .$$

und sein duales Programm

$$\sum_{i:i \in S_j} y_i \ \leq \ c_j \text{ für alle } j \ , \ y \geq 0 \ ,$$

$$\sum_i y_i \ = \ \max \ .$$

Wir zeigen nun, dass mit

$$y_i \ := \ p(i)/h_n \ , \quad i \in I$$

eine zulässige Lösung y des Dualprogramms gegeben ist. Für vorgegebenes j seien $i_1, \ldots, i_r$ die Elemente von S_j, und zwar in einer Reihenfolge, wie sie der gierige Algorithmus der Reihe nach überdeckt. S_j überdeckt dann i_a mit Kosten höchstens $c_j/(r-a+1)$, also gilt

$$p(i_a) \ \leq \ \frac{c_j}{r-a+1}$$

und folglich

$$\sum_{i \in S_j} y_i \ \leq \frac{1}{h_n} \sum_{a=1}^{r} \frac{c_j}{r-a+1} \ = \ \frac{h_r}{h_n} c_j \ \leq \ c_j \ .$$

Also ist y zulässig für das Dualprogramm. Der gierige Algorithmus ergibt offenbar eine zulässige Lösung des Primalprogramms mit Kosten

$$\sum_i p(i) \ = \ h_n \sum_i y_i \ .$$

Nach dem schwachen Dualitätssatz (dessen Primalprogramm hier ein Minimum Programm ist) gilt andererseits für jede zulässige Lösung x des Primalprogramms

$$\sum_j c_j x_j \ \geq \ \sum_i y_i \ .$$

Wie behauptet folgt für die Kosten des gierigen Algorithmus

$$\sum_i p(i) \;\leq\; h_n \sum_j c_j x_j \;,$$

also Optimalität bis auf den Faktor h_n. □

Dieses Beispiel illustriert eine wichtige Methode der ganzzahligen Programmierung, nämlich durch Relaxation die Verbindung zum Linearen Programmieren herzustellen. Ein direkter Zusammenhang besteht in solchen Fällen, wo die Eckpunkte von K_{zul} bereits ganzzahlig sind.

Definition 5.9. *Eine Matrix A heißt* **total unimodular**, *falls für jede quadratische Untermatrix von A die Determinante den Wert 1,0 oder -1 hat.*

Insbesondere hat A dann nur die Einträge 1,0 und -1.

Satz 5.10. *Ist A total unimodular und b ein Vektor mit ganzzahligen Komponenten, so sind auch die Ecken des Polyeders $\{x \in \mathbb{R}^n : Ax = b, x \geq 0\}$ ganzzahlig.*

Beweis. Sei x Ecke. Dann besitzt A_x linear unabhängige Spalten und eine reguläre Untermatrix A' mit derselben Spaltenzahl. Aus $A_x r_x = b$ folgt

$$A' r_x = b'$$

mit dem entsprechend reduzierten Vektor b'. Nach Annahme hat A' Determinante 1 oder -1. Aus der Cramerschen Regel folgt daher, dass r_x und also auch x ganzzahlige Komponenten hat. □

Korollar 5.11. *Ist A total unimodular und b ein Vektor mit ganzzahligen Komponenten, so sind auch die Ecken des Polyeders $\{x \in \mathbb{R}^n : Ax \leq b, x \geq 0\}$ ganzzahlig.*

Beweis. Das zugehörige kanonische Programm mit Schlupfvariablen hat die Matrix $A' = (A, E)$ (vgl. Abschnitt 5.1). Mit A ist auch A' unimodular (Übung), nach dem vorangegangenen Satz hat daher das kanonische Programm und damit auch das ursprüngliche Lineare Programm ganzzahlige Ecken. □

Beispiel. Matching in bipartiten Graphen. Wir betrachten einen bipartiten Graph G mit m Knoten und n Kanten. Die Menge der Knoten besteht aus disjunkten Teilen U und V, und jede Kante besitzt ein Ende in U und eines in V. Wenn der Knoten i Endpunkt der Kante j ist, heißen i und j inzident. Die Matrix A mit Einträgen

$$a_{ij} \;=\; \begin{cases} 1 & \text{falls Knoten } i \text{ und Kante } j \text{ inzident sind,} \\ 0 & \text{sonst} \end{cases}$$

heißt Inzidenzmatrix, sie beschreibt G vollständig.

Wir zeigen, dass die Inzidenzmatrix eines bipartiten Graphen total unimodular ist. Inzidenzmatrizen eines Graphen haben die Gestalt, dass jede Spalte genau 2 Einsen enthält, weil jede Kante genau 2 Eckpunkte hat. Bei einem bipartiten Graphen hat A ohne Einschränkung folgende Gestalt,

$$A \;=\; \begin{pmatrix} A' \\ A'' \end{pmatrix},$$

wobei A' und A'' in jeder Spalte genau eine Eins enthalten.

Sei nun B $k \times k$ Untermatrix von A. Wir zeigen per Induktion, dass die Determinante gleich 1,0, oder -1 ist. Der Induktionsanfang $k = 1$ ist offenbar. Für den Induktionsschritt von k nach $k+1$ unterscheiden wir drei Fälle.

i) B enthält eine Spalte aus lauter Nullen. Dann ist $\det B = 0$.

ii) B enthält eine Spalte mit genau einer Eins, etwa

$$B \;=\begin{pmatrix} 1 & \cdots \\ 0 & B' \end{pmatrix}.$$

Dann gilt $\det B \;=\; \pm \det B'$ mit einer $k \times k$ Untermatrix B', und die Behauptung folgt nach Induktionsannahme.

iii) Alle Spalten von B enthalten zwei Einsen, B hat also die Gestalt

$$B \;=\; \begin{pmatrix} B' \\ B'' \end{pmatrix},$$

wobei B' und B'' in jeder Spalte eine Eins haben. Durch Addition aller Zeilen von B' (außer der ersten) zur ersten Zeile entsteht eine Zeile aus lauter Einsen, dabei verändert sich $\det B$ nicht. Ähnlich erhält man in B'' eine Zeile aus lauter Einsen, und es folgt $\det B = 0$.

A ist also total unimodular.

Als Anwendung leiten wir einen Spezialfall des Dualitätssatzes für bipartite Graphen ab, eine kombinatorische Aussage über Matchings. Dazu bezeichne S die Menge der Kanten und $T = U \cup V$ die Menge der Knoten. Jeder 01-Vektor

$$x = (x_j)_{j \in S}$$

repräsentiert eine Teilmenge $X \subset S$, gemäss der Vereinbarung $x_j = 1$ oder 0, je nachdem ob $j \in X$ oder $j \notin X$. Genauso repräsentiert jeder 01-Vektor

$$y = (y_i)_{i \in T}$$

eine Teilmenge $Y \subset T$. Dann gilt

$$Ax \leq 1 \quad \Leftrightarrow \quad \text{zwei Kanten aus } X \text{ haben keinen gemeinsamen Knoten.}$$

In diesem Fall heißt X ein **Matching**. Ähnlich gilt

$$y^t A \geq 1 \quad \Leftrightarrow \quad \text{jede Kante ist mit einem Knoten aus } Y \text{ inzident.}$$

Y heißt dann **Träger**. Offenbar sind die Linearen Programme

$$\begin{aligned} Ax \leq 1\ , \quad x \geq 0\ , \quad x \in \mathbb{R}^n\ , \\ \sum_j x_j \ = \ \max \end{aligned}$$

und

$$\begin{aligned} y^t A \geq 1\ , \quad y \geq 0\ , \quad y \in \mathbb{R}^m\ , \\ \sum_i y_i \ = \ \min \end{aligned}$$

zueinander dual, und besitzen die zulässigen Lösungen $x = (0 \ldots, 0)^t$ und $y = (1, \ldots, 1)^t$, so dass beide Programme optimale Lösungen und denselben Wert haben. Außerdem ist A total unimodular, und damit sind alle Ecken ganzzahlig. Man überzeugt sich, dass optimale Ecken nur 0 oder 1 als Komponenten haben. In diesem Fall gilt $\sum_j x_j = |X|$, $\sum_i y_i = |Y|$, und nach dem Hauptsatz für Lineare Programme folgt

$$\max\{|X| : X \text{ ist Matching}\} \ = \ \min\{|Y| : Y \text{ ist Träger}\}\ .$$

Maximale Matchings und minimale Träger haben also die gleiche Kardinalität. □

5.5 Der Simplex-Algorithmus

Der Simplex-Algorithmus findet optimale Lösungen von Linearen Programmen bzw. erkennt, dass keine optimalen Lösungen existieren, indem er in der Menge der zulässigen Lösungen entlang Kanten von Ecke zu Ecke springt.

1. Wahl der Startecke. Wir gehen von einem standard Maximum Programm

$$\begin{aligned} A'x' \leq b' \ , \quad x' \geq 0 \ , \quad x' \in \mathbb{R}^n \ , \\ (c')^t x' \ = \ \max \end{aligned}$$

aus. Um eine zulässige Startecke zu finden, erweitern wir es mittels Schlupfvariablen z zum kanonischen Maximum Programm

$$\begin{aligned} Ax = b \ , \quad x \geq 0 \ , \quad x \in \mathbb{R}^{n+m} \ , \\ c^t x \ = \ \max \end{aligned}$$

mit

$$A := (A', E) \ , \ x = \begin{pmatrix} x' \\ z \end{pmatrix} \ , \ b = b' \ , \ c = \begin{pmatrix} c' \\ 0 \end{pmatrix}$$

und der $m \times m$ Einheitsmatrix E. Für optimale Ecken dieses Programms können wir z ohne Einschränkung als 0 annehmen, so dass die Bestimmung optimaler Ecken für beide Linearen Programme äquivalent ist.

Ohne Einschränkung der Allgemeinheit können wir $b \geq 0$ annehmen, sonst multipliziere man negative Komponenten von b und die entsprechende Zeile in A mit dem Faktor -1. Dann ist

$$x_{Start} := \begin{pmatrix} 0 \\ b \end{pmatrix}$$

Ecke der zulässigen Punkte des kanonischen Programms. Dies folgt nach Proposition 5.6, denn $A_{x_{Start}}$ ist gleich E bzw. gleich einer Diagonalmatrix mit Diagonaleinträgen 1 oder -1 und hat damit linear unabhängige Spalten.

2. Etwas Algebra. Wir arbeiten weiter mit dem erweiterten kanonischen Programm. Durch das Hinzufügen von E hat A den Rang m. Sei x Ecke der Menge zulässiger Punkte. Dann hat nach Proposition 5.6 A_x unabhängige Spalten. Wir erweitern die Spaltenvektoren zu einer Basis a_j, $j \in J_x$, die wir mit B_x bezeichnen. Es gilt also $Z_x \subset J_x$ bzw. $x_j = 0$ für alle $j \notin J_x$. Alle

Spalten von A lassen sich dann eindeutig als Linearkombination dieser Basen darstellen,

$$a_k = \sum_{j \in J_x} a_j \tau_{jk} \ ,$$

mit reellen Zahlen $\tau_{jk} = \tau_{jk}(x)$.

Sei nun x' ein weiterer zulässiger Punkt. Dann gilt

$$\sum_{j \in J_x} a_j x_j = Ax = Ax' = \sum_k a_k x'_k = \sum_{j \in J_x} a_j \sum_k \tau_{jk} x'_k \ .$$

Durch Koeffizientenvergleich folgt für $j \in J_x$

$$x_j = \sum_k \tau_{jk} x'_k = x'_j + \sum_{k \notin J_x} \tau_{jk} x'_k \ , \tag{5.1}$$

denn für $j, k \in J_x$ gilt $\tau_{jj} = 1$ und $\tau_{jk} = 0$ für $j \neq k$. Für das Zielfunktional folgt

$$c^t x' = \sum_{j \in J_x} c_j x'_j + \sum_{k \notin J_x} c_k x'_k = \sum_{j \in J_x} c_j x_j + \sum_{k \notin J_x} \Big(c_k - \sum_{j \in J_x} c_j \tau_{jk} \Big) x'_k$$

oder

$$c^t x' = c^t x + \sum_{k \notin J_x} d_k x'_k \tag{5.2}$$

mit

$$d_k := c_k - \sum_{j \in J_x} c_j \tau_{jk} \ .$$

3. Benachbarte Ecken. Nun betrachten wir zwei Eckpunkte x, x', für die sich J_x und $J_{x'}$ minimal unterscheiden, d.h. $J_{x'} - J_x$ nur aus einem einzigen Element besteht, das wir mit l bezeichnen. Nach (5.1) gilt

$$x'_j \ = \ \begin{cases} x_j - \tau_{jl} x'_l & j \in J_x \\ x'_l & j = l \\ 0 & \text{sonst} \end{cases}$$

und nach (5.2)

$$c^t x' = c^t x + d_l x'_l \ . \tag{5.3}$$

Anschaulich gesprochen sind x und x' benachbart, also durch eine Kante verbunden. Dementsprechend betrachten wir für $\delta \geq 0$ auch die Punkte $x(\delta)$ mit Komponenten

$$x_j(\delta) = \begin{cases} x_j - \tau_{jl} \delta & j \in J_x \\ \delta & j = l \\ 0 & \text{sonst} \end{cases}$$

4. Der Algorithmus. Der Simplex-Algorithmus beginnt seine Suche des Minimums in obiger Startecke und wechselt entlang Kanten zu immer neuen Ecken, bis er eine optimale Ecke trifft bzw. feststellt, dass es keine optimale Ecke gibt. Wenn er sich in einer Ecke x befindet, gibt es verschiedene Fälle zu unterscheiden:

Fall 1. $d_l \leq 0$ *für alle* l.
Nach (5.3) hat keine benachbarte Ecke einen größeren Wert des Zielfunktionals. Nach (5.2) gilt $c^t x \geq c^t x'$ für alle zulässigen x', und x ist optimaler Eckpunkt.

Fall 2. *Es gibt ein* $l \notin J_x$ *mit* $d_l > 0$ *und* $\tau_{jl} \leq 0$ *für alle* j.
Dann ist $x(\delta)$ für alle $\delta \geq 0$ zulässig, und es gilt

$$c^t x(\delta) = c^t x + \delta d_l.$$

Der Grenzübergang $\delta \to \infty$ zeigt, dass der Wert des linearen Programmes ∞ ist und keine optimalen Lösungen existieren.

Fall 3. *Es gibt ein* $l \notin J_x$ *mit* $d_l > 0$ *und* $\tau_{jl} > 0$ *für ein* j, *sowie* $x_j > 0$ *für alle* $j \in J_x$.
Dann ist $x(\delta)$ zulässig für $\delta > 0$ sofern δ nicht zu groß ist. Wir wählen δ so groß, dass gerade eines der $x_j(\delta)$ verschwindet, etwa für $j = k$, und setzen

$$x' = x(\delta) \ .$$

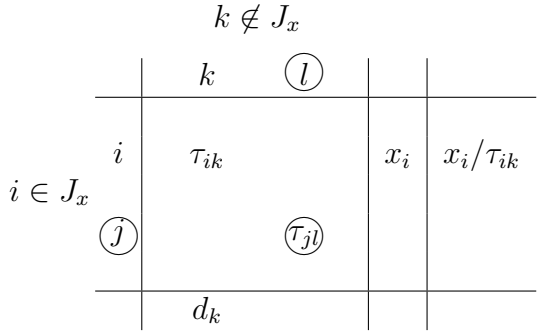
Dann ist x' zulässige Ecke ist. Andernfalls gäbe es nämlich nach Proposition 5.6 reelle Zahlen λ_j mit

$$\sum_{j \in J_x} a_j \tau_{jl} = a_l = \sum_{j \in Z_x \, , \, j \neq k} a_j \lambda_j \ ,$$

und wegen der linearen Unabhängigkeit der a_j, $j \in Z_x$ erhielten wir $\tau_{jl} = 0$ im Widerspruch zur Annahme. Nach (5.3) hat das Zielfunktional in x' einen größeren Wert, und der Algorithmus wechselt von x nach x' über.

Fall 4. *Es gibt ein* $l \notin J_x$ *mit* $d_l > 0$ *und* $\tau_{jl} > 0$ *für ein* j, *sowie* $x_j = 0$ *für ein* $j \in J_x$.
In diesem Fall einer entarteten Ecke gehen wir analog vor. Hier kann es passieren, dass $\delta = 0$ zu wählen ist. Dann ändert man zwar die Basis B_x, aber nicht die Ecke. Durch andere Basiswechsel kommt man aber schließlich aus der Ecke entlang einer Kante heraus. Man muss dabei vermeiden, dass man sich ungeschickt in einen Zyklus von Basiswechseln hineinbegibt.

5. Das Tableau-Schema. In der praktischen Durchführung hat es sich bewährt, die relevanten Daten als Tableau anzuordnen. Für die Ecke x sieht das so aus,



Die Stelle des nächsten Basiswechsels ist umkreist und heißt *Pivotelement*. Das Ausgangstableau hat die Gestalt

	1	$\dots$	n	
$n+1$	a_{11}	$\dots$	a_{1n}	b_1
$\vdots$	$\vdots$		$\vdots$	$\vdots$
$n+m$	a_{m1}	$\dots$	a_{mn}	b_m
	c_1	$\dots$	c_n	

Details finden sich in Lehrbüchern.