Kapitel 1

Interim

$\boxed{\downarrow 17.11.04}$

Da ich keine Infos über Titel und Nummerierungen anderer Kapitel dieser Vorlesung habe, nenne ich dies einfach mal "Kapitel 1".

1.1 Einige formale Definitionen

Wir rekapitulieren zunächst einige der in den letzten Wochen vorgestellten Begriffe und formalisieren sie:

Definition 1.1: (Gruppe)

Eine Menge G mit einer binären Operation $\circ$: $G \times G \mapsto G$ (also $a \circ b \in G \; \forall a, b \in G$) heißt **Gruppe**, *wenn*

a) *die Gruppenoperation $\circ$ assoziativ ist: $(a \circ b) \circ c = a \circ (b \circ c) \; \forall a, b, c \in G$,*
b) *es ein* **"neutrales Element"** *$e \in G$ gibt mit $e \circ a = a \circ e = a \; \forall \; a \in G$,*
c) *es zu jedem $a \in G$ ein* **"inverses Element"** *$a^{-1} \in G$ gibt mit $a^{-1} \circ a = a \circ a^{-1} = e$.*

Die Verknüpfung $\circ$ heißt **kommutativ**, *wenn $a \circ b = b \circ a \; \forall a, b \in G$ gilt. Ist $\circ$ kommutativ, so nennt man die Gruppe* **kommutativ** *oder auch* **abelsch**.

Bemerkung 1.2: *Zu jedem Element einer Gruppe ist die Inverse eindeutig. Angenommen, es gäbe 2 Elemente b_1 und b_2 mit*

$$b_1 \circ a = e, \quad b_2 \circ a = e, \quad e \circ b_1 = e, \quad e \circ b_2 = e.$$

"Multiplizieren" wir $b_1 \circ a = e$ von rechts mit b_2, folgt

$$\underbrace{(b_1 \circ a) \circ b_2}_{= b_1 \circ \underbrace{(a \circ b_2)}_{= e}} = e \circ b_2 \quad \Rightarrow \quad b_1 \circ e = e \circ b_2 \quad \Rightarrow \quad b_1 = b_2.$$

Beispiel 1.3: In den folgenden Beispielen seien $+$ und $\cdot$ die übliche Addition bzw. Multiplikation in $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$:

1) $(\mathbb{Z}, +)$ ist eine abelsche Gruppe: das neutrale Element ist 0, die additive Inverse a^{-1} wird als $-a$ geschrieben.
2) $(\mathbb{Z}, \cdot)$ mit der üblichen Multiplikation $\cdot$ ist **keine Gruppe**: der einzige Kandidat für das neutrale Element wäre 1, aber es gibt z.B. zu $a = 2$ kein inverses Element (der einzige Kandidat $a^{-1} = 1/2$ liegt nicht in $\mathbb{Z}$).
3) $(\mathbb{Q}, +)$ ist eine abelsche Gruppe, das neutrale Element ist 0, die additive Inverse a^{-1} wird als $-a$ geschrieben.
4) $(\mathbb{Q} \setminus \{0\}, \cdot)$ ist eine abelsche Gruppe, das neutrale Element ist 1, die multiplikative Inverse a^{-1} wird als $1/a$ geschrieben.
5) Für jedes $n \in \mathbb{N}$ ist die Menge $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$ mit der Addition $a \circ b = (a + b) \bmod n$ eine abelsche Gruppe, das neutrale Element ist 0, die additive Inverse zu $a \in \mathbb{Z}$ ist $a^{-1} = -a = (n - a) \bmod n$.
6) Die Menge $\mathbb{Z}_3 \setminus \{0\} = \{1, 2\}$ mit der Multiplikation $a \circ b = (a \cdot b) \bmod 3$ ist eine abelsche Gruppe mit dem neutralen Element 1. Die Inversen sind $1^{-1} = 1$ und $2^{-1} = 2$:

$\circ$	1	2
1	1	2
2	2	1

7) Die Menge $\mathbb{Z}_4 \setminus \{0\} = \{1, 2, 3\}$ mit der Multiplikation $a \circ b = (a \cdot b) \bmod 4$ ist **keine Gruppe**. Aus der Gruppentafel erkennt man, dass 1 der einzige Kandidat für das neutrale Element ist, es aber zu $a = 2$ keine Inverse gibt:

$\circ$	1	2	3
1	1	2	3
2	2	0	2
3	3	2	1

Ein **Ring** ist eine Menge, in der man addieren und multiplizieren kann:

Definition 1.4: (Ring)

*Eine Menge R mit zwei binären Verknüpfungen $\oplus$ (die "Addition") und $\otimes$ (die "Multiplikation") heißt "**Ring**", wenn*

a) $(R, \oplus)$ *eine abelsche Gruppe ist,*

b) $\otimes$ *eine assoziative Verknüpfung ist:* $(a \otimes b) \otimes c = a \otimes (b \otimes c)\ \forall\, a, b, c \in R$,

c) das **Distributivgesetz**

$$a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c), \quad (a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c)$$

für alle $a, b, c \in R$ *gilt.*

*Ist $\otimes$ eine kommutative Verknüpfung (also $a \otimes b = b \otimes a \ \forall\ a, b \in R$), so spricht man von einem "**kommutativen Ring**".*

Bemerkung 1.5: *Für das neutrale Elemente 0 der Addition gilt in einem Ring immer $0 \cdot r = r \cdot 0 = 0 \ \forall r \in R$, denn*

$$0 \cdot r = (0 + 0) \cdot r = 0 \cdot r + 0 \cdot r.$$

Subtrahiert man auf beiden Seiten $0 \cdot r$, so erhält man $0 = 0 \cdot r$. Im nichtkommutativen Fall hat man sich auch $0 = r \cdot 0 \ \forall r \in R$ zu überlegen, was analog folgt.

Beispiel 1.6: In den folgenden Beispielen seien $+$ und $\cdot$ die übliche Addition bzw. Multiplikation in $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$:

1) $(\mathbb{Z}, +\,,\ \cdot\,)$ ist ein kommutativer Ring.
2) $(\mathbb{Q}, +\,,\ \cdot\,)$ ist ein kommutativer Ring.
3) $(\mathbb{R}, +\,,\ \cdot\,)$ ist ein kommutativer Ring.
4) Für jedes $n \in \mathbb{N}$ ist die Menge $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$ mit der Addition $a \oplus b = (a + b) \bmod n$ und der Multiplikation $a \otimes b = (a \cdot b) \bmod n$ ein kommutativer Ring. Um dies einzusehen, ist im Wesentlichen nur zu verifizieren, dass in der Tat das Distributivgesetz

$$\Big(a \cdot (b + c)\Big) \bmod n = \bigg(\Big((a \cdot b) \bmod n\Big) + \Big((a \cdot c) \bmod n\Big)\bigg) \bmod n$$

für ganze Zahlen a, b, c gilt. Die Details ersparen wir uns hier.

Ein **Körper** ist eine Menge, in der man addieren und multiplizieren kann (der also ein Ring ist), aber zusätzlich auch noch Division erlaubt:

Definition 1.7: (Körper)

*Eine Menge K mit zwei binären Verknüpfungen $\oplus$ (die "Addition") und $\otimes$ (die "Multiplikation") heißt "**Körper**", wenn*

a) $(K, \oplus, \otimes)$ ein Ring ist,

b) $(K \setminus \{0\}, \otimes)$ eine Gruppe ist.

Hierbei ist mit 0 das neutrale Element bzgl. der Addition $\oplus$ gemeint. Die Division ist dann für jedes $b \neq 0$ definiert als $a/b := a \otimes b^{-1}$, wo b^{-1} die Inverse von $b \in K \setminus \{0\}$ bzgl. $\otimes$ ist.

Zusätzlich zu den Ringeigenschaften gelten bei einem Körper also die beiden folgenden Eigenschaften:

- Es muss ein neutrales Element der Multiplikation existieren (meist mit 1 bezeichnet),
- zu jedem $a \in K$ muss eine multiplikative Inverse a^{-1} mit $a \otimes a^{-1} = a^{-1} \otimes a = 1$ existieren (das wir dann als $a^{-1} = 1/a$ schreiben).

Beispiel 1.8: In den folgenden Beispielen seien $+$ und $\cdot$ die übliche Addition bzw. Multiplikation in $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$:

1) $(\mathbb{Z}, +, \cdot)$ ist zwar ein Ring, aber kein Körper (die Division zweier ganzen Zahlen ergibt eine rationale Zahl, die in der Regel nicht ganzzahlig sein wird).
2) $(\mathbb{Q}, +, \cdot)$ ist ein Körper (zu jeder rationalen Zahl $\neq 0$ gibt es eine multiplikative Inverse, nämlich den Kehrwert).
3) $(\mathbb{R}, +, \cdot)$ ist ein Körper (zu jedem $a \neq 0$ ist $a^{-1} = 1/a \in \mathbb{R}$ definiert).
4) Die Restklassenringe $\mathbb{Z}_n$ mit der üblichen Arithmetik modulo $n \in \mathbb{N}$ sind zwar stets Ringe, aber – abhängig von n – nicht unbedingt Körper. Nach Beispiel 1.3.6) ist die Multiplikation auf $\mathbb{Z}_3 \setminus \{0\}$ eine Gruppenstruktur, damit ist $\mathbb{Z}_3$ ein Körper. Nach Beispiel 1.3.7) ist die Multiplikation auf $\mathbb{Z}_4 \setminus \{0\}$ keine Gruppenstruktur, damit ist $\mathbb{Z}_4$ lediglich ein Ring, aber kein Körper. Wie der folgende Satz besagt, liegt dies daran, dass 3 eine Primzahl ist, 4 aber nicht.

Satz 1.9: (Für welche p ist $\mathbb{Z}_p$ ein Körper?)

$\mathbb{Z}_p$ ist genau dann ein Körper, wenn p eine Primzahl ist. Die multiplikative Inverse von $a \in \mathbb{Z}_p \setminus \{0\}$ ist dann gegeben durch $a^{-1} = a^{p-2}$.

Beweis: Sei p eine Primzahl. Nach dem kleinen Satz von Fermat gilt

$$\boxed{\text{Für jedes } a \in \mathbb{Z} \text{ gilt } a^{p-1} \bmod p = 1, \text{ falls } p \text{ nicht } a \text{ teilt.}}$$

Die Primzahl p teilt $a \in \{1, 2, \ldots, p-1\}$ nicht, also gilt

$$a \cdot a^{p-2} = a^{p-2} \cdot a = 1 \ (\mathrm{mod}\ p).$$

Wir interpretieren a als Element von $\mathbb{Z}_p = \{0, 1, \ldots, p-1\}$ mit der Addition und Multiplikation modulo p. Dann besagt die obige Gleichung, dass das Inverse von $a \neq 0$ bzgl. der Multiplikation mod p durch a^{p-2} gegeben ist.

Ist p keine Primzahl, gilt $p = f_1 \cdot f_2$ mit Faktoren $f_1, f_2 \in \{1, 2, \ldots, p-1\}$. Interpretieren wir f_1, f_2 als Elemente von $\mathbb{Z}_p$, so gilt bzgl. der modularen Multiplikation $f_1 \cdot f_2 = p \bmod p = 0$. Wäre $\mathbb{Z}_p$ ein Körper, würde $f_1 = (f_1 \cdot f_2) \cdot f_2^{-1} = 0 \cdot f_2^{-1} = 0$ folgen, was der Annahme $f_1 \in \{1, 2, \ldots, p-1\}$ widerspricht.

Q.E.D.

1.2 Repeated Squaring

Wie berechnet man möglichst effizient eine Potenz x^n einer Zahl x? Natürlich kann man Schritt für Schritt

$$p_1 := x;\ p_2 := x \cdot p_1;\ p_3 := x \cdot p_2;\ \ldots\ ;\ p_n := x \cdot p_{n-1}$$

berechnen ($p_n = x^n$ ist das gesuchte Ergebnis), braucht dafür aber $n-1$ Multiplikationen. In Anwendungen der Kryptographie etc. geht es darum, solche Multiplikationen für sehr große Werte von n zu ermitteln, da sollte man sich tunlichst überlegen, wie das Potenzieren schneller zu erledigen ist. Die Idee ist einfach, geschickt auf Zwischenergebnisse zurückzugreifen. So ist x^8 z.B. mit nur 3 Multiplikationen

$$p_1 := x;\ p_2 := p_1 \cdot p_1;\ p_4 := p_2 \cdot p_2;\ p_8 := p_4 \cdot p_4$$

zu berechnen, für x^{21} braucht man 6 Multiplikationen:

$$p_1 := x;\ p_2 := p_1 \cdot p_1;\ p_4 := p_2 \cdot p_2;\ p_5 := x \cdot p_4;$$

$$p_{10} := p_5 \cdot p_5;\ p_{20} := p_{10} \cdot p_{10};\ p_{21} := x \cdot p_{20}$$

Der wesentliche Schritt besteht also aus dem Quadrieren des letzten Zwischenergebnisses (damit erreicht man für n = 2er-Potenz immer das Endergebnis). Ist n keine Zweierpotenz, sind gelegentlich Multiplikationen mit x dazwischenzuschalten. Systematisch berechnet man sich die Binärdarstellung

$$n = 2^k + b_{k-1} \cdot 2^{k-1} + \cdots + b_1 \cdot 2^1 + b_0 \cdot 2^0$$

des Exponenten mit den “Bits” $b_i \in \{0, 1\}$, z.B. durch (MuPAD-Code):

```
>> n := 21: // als Beispiel
>> m := n:
>> k := trunc(log(2, n)):
>> for i from 0 to k do
     b[i]:= m mod 2;
     m:= m div 2;  // = trunc(m/2)
   end_for:
>> b[i] $ i = 0..k;

                         1, 0, 1, 0, 1
```

(oder einfacher über `numlib::g_adic(n, 2)`). Es ergibt sich folgender Potenzierungsalgorithmus:

Algorithmus "Repeated Squaring" 1.10:

Zu berechnen ist x^n mit $n \in \mathbb{N}$.

1: *Bestimme die Binärdarstellung* $n = 2^k + \sum_{i=0}^{k-1} b_i \cdot 2^i$ *mit* $b_i \in \{0, 1\}$.

2: *Setze* $y := x$.

3: *Für* $i = k-1,\ k-2,\ ...,\ 1,\ 0$ *wiederhole:*
   Falls $b_i = 0$, *setze* $y := y^2$;
   Sonst setze $y := y^2 \cdot x$;

4: *Ergebnis:* y *hat den Wert* x^n.

Beweis der Korrektheit: wir zeigen, dass y am Ende des Schrittes 3 mit gegebenem i den Wert $y = x^{[n/2^i]}$ hat (damit liegt für $i = 0$ das Endergebnis $y = x^{[n/2^0]} = x^n$ vor).
Induktionsstart: Am Ende des ersten Schritts mit $i = k-1$ gilt $y = x^2$, falls $b_{k-1} = 0$ gilt bzw. $y = x^3$, falls $b_{k-1} = 1$. Die Exponenten von x stimmen überein mit

$$\begin{aligned}\left[\frac{n}{2^{k-1}}\right] &= \left[\frac{1}{2^{k-1}} \cdot \left(2^k + b_{k-1} \cdot 2^{k-1} + b_{k-2} \cdot 2^{k-2} + \cdots\right)\right] \\ &= \left[2 + b_{k-1} + b_{k-2} \cdot \frac{1}{2} + \cdots\right] \\ &= 2 + b_{k-1} = \begin{cases} 2 & \text{falls } b_{k-1} = 0, \\ 3 & \text{falls } b_{k-1} = 1. \end{cases}\end{aligned}$$

Induktionsschritt: zu Beginn des Schrittes 3 habe y den Wert $x^{[n/2^{i+1}]}$. Nun wird der Schritt mit dem Wert i durchlaufen. Am Ende des Schrittes 3 hat wird y durch den Wert

$$\left.\begin{array}{ll} y^2 & \text{falls } b_i = 0, \\ y^2 \cdot x & \text{falls } b_i = 0 \end{array}\right\} = \left\{\begin{array}{ll} x^{2\cdot[n/2^{i+1}]} & \text{falls } b_i = 0, \\ x^{2\cdot[n/2^{i+1}]+1} & \text{falls } b_i = 1 \end{array}\right\} = x^{2\cdot[n/2^{i+1}]+b_i} \qquad (\#)$$

überschrieben. Es gilt

$$\begin{aligned}2 \cdot \left[\frac{n}{2^{i+1}}\right] + b_i &= 2 \cdot \left[\frac{1}{2^{i+1}} \cdot \left(2^k + b_{k-1} \cdot 2^{k-1} + b_{k-2} \cdot 2^{k-2} + \cdots\right)\right] + b_i \\ &= 2 \cdot \left[2^{k-i-1} + b_{k-1} \cdot 2^{k-1-i-1} + \cdots + b_{i+1} + \underbrace{b_i \cdot \frac{1}{2} + \cdots}_{\text{Nachkommastellen}}\right] + b_i \\ &= 2 \cdot \left(2^{k-i-1} + b_{k-1} \cdot 2^{k-2-i} + \cdots + b_{i+1}\right) + b_i \\ &= 2^{k-i} + b_{k-1} \cdot 2^{k-1-i} + \cdots + b_{i+1} \cdot 2 + b_i.\end{aligned}$$

Dies stimmt überein mit

$$\begin{aligned}\left[\frac{n}{2^i}\right] &= \left[\frac{1}{2^i}\cdot\Big(2^k + b_{k-1}\cdot 2^{k-1} + b_{k-2}\cdot 2^{k-2} + \cdots\Big)\right] \\ &= \left[2^{k-i} + b_{k-1}\cdot 2^{k-1-i} + \cdots + b_{i+1}\cdot 2 + b_i + \underbrace{b_{i+1}\cdot\frac{1}{2} + \cdots}_{\text{Nachkommastellen}}\right] \\ &= 2^{k-i} + b_{k-1}\cdot 2^{k-1-i} + \cdots + b_{i+1}\cdot 2 + b_i.\end{aligned}$$

Also hat der vom Schritt 3 des Algorithmus gelieferte Wert (#) die Darstellung

$$x^{2\cdot[n/2^{i+1}]+b_i} = x^{[n/2^i]}.$$

Q.E.D.

Bemerkung 1.11: *Das Entscheidende am Algorithmus ist der Rechenaufwand: er besteht für $n = 2^k$ aus lediglich k Multiplikationen. Allgemein braucht man für $2^k \le n < 2^{k+1}$ maximal $2\cdot k$ Multiplikationen (für $n = 2^{k+1} - 1$ mit der Binärdarstellung $n = 11\ldots 1_2$ braucht man genau $2\cdot k$ Multiplikationen). Für die Anzahl $M(n)$ der Multiplikationen gilt also stets* $\boxed{\downarrow 18.11.04}$

$$k \le M(n) \le 2\cdot k,$$

also mit $k = [\log_2(n)]$:

$$[\log_2(n)] \le M(n) \le 2\cdot[\log_2(n)].$$

Ganz grob:

Durch Repeated Squaring wird x^n mit maximal $2\cdot\log_2(n)$ Multiplikationen berechnet.

Man beachte, dass $\log_2(n)$ für großes n sehr viel kleiner ist als n.

Bemerkung 1.12: *Der Algorithmus ist immer anwendbar, wenn irgendwelche Multiplikationen (oder Gruppenverknüpfungen) zu berechnen sind. So kann beispielsweise x ein Element von $\mathbb{Z}_n$ sein, die Multiplikationen wären dann als modulare Multiplikationen durchzuführen.*

Anwendung (Berechnung von Inversen mod p) 1.13:

Nach Satz 1.9 sind für jede Primzahl p die multiplikativen Inversen im Körper $\mathbb{Z}_p$ durch $a^{-1} = a^{p-2}$ gegeben, d.h., die Division duch a ist die Multiplikation mit a^{p-2}. Hierbei läßt sich a^{p-2} per Repeated Squaring mit maximal $2\cdot\log_2(p-2)$ modularen Multiplikationen berechnen.

1.3 Der Euklidische Algorithmus

Es geht nun um das effiziente Berechnen von größten gemeinsamen Teilern (ggT oder engl. gcd = greatest common divisor). Für ganze Zahlen ist dieses Konzept aus der Schule bekannt. Wir verallgemeinern den Begriff für Ringstrukturen, denn neben ganzen Zahlen wollen wir später auch Polynome behandeln. Zunächst die Begriffe der Teilbarkeit und des ggT in einem Ring:

Definition 1.14: (Teilbarkeit in allgemeinen Ringen)

*Sei R ein beliebiger Ring. Dann heißt $a \in R$ ein "**Teiler**"[1] von $b \in R$ (geschrieben $a \mid b$), falls es ein $r \in R$ gibt mit $b = r \cdot a$.*

Definition 1.15: (größter gemeinsamer Teiler)

*Sei R ein beliebiger Ring. Dann heißt $c \in R$ ein "**größter gemeinsamer Teiler**"[2] von $a, b \in R \setminus \{0\}$, geschrieben als $c = ggT(a, b)$, falls $c \mid a$ und $c \mid b$ gilt und für alle $r \in R$ mit $r \mid a$ und $r \mid b$ stets $r \mid c$ folgt.*

Bemerkung 1.16: *I.A. ist der ggT nicht eindeutig. So kann man z.B. auf dem Ring $\mathbb{Z}$ den $ggT(6, 9)$ als 3 oder auch als -3 vereinbaren.*

> **Vereinbarung:** *Wir machen den ggT auf dem Ring $\mathbb{Z}$ eindeutig, indem wir $ggT(a, b) > 0$ fordern $\forall a, b \in \mathbb{Z} \setminus \{0\}$.*

Bemerkung 1.17: *In MuPAD ist* `gcd(a, b)` *der Aufruf zur Berechnung von $ggT(a, b)$:*

```
>> gcd(123456, 54321)
                                        3
```

Für den Euklidischen Algorithmus zur Berechnung des größten gemeinsamen Teilers zweier Ringelemente braucht man eine gewisse Ordnungsstruktur. Die Ordnungsstruktur ist versteckt in der folgenden Abbildung d:

Definition 1.18: (Euklidischer Ring)

Ein Ring R heißt **Euklidischer Ring**, *wenn es eine Abbildung $d : R \to \{0, 1, 2, \ldots\}$ mit $d(0) = 0$ gibt, sodass stets "Division mit Rest" durchführbar ist: zu jedem Paar $a, b \in R$ mit $b \neq 0$ existiere $q, r \in R$ mit*

$$a = q \cdot b + r, \quad d(r) < d(b).$$

*Hierbei heisst q der "**Quotient**" von a durch b (geschrieben als $q = a \text{ div } b$) und r der "**Rest**" (geschrieben als $a \text{ mod } b$).*

---

[1] Für $b = 0$ ist offensichtlich jedes Ringelement ein Teiler.

[2] Für $a = 0$ bzw. $b = 0$ setze $ggT(0, b) = b$ bzw. $ggT(a, 0) = a$. Wir werden im Folgenden jedoch nur den ggT von Ringelementen $\neq 0$ betrachten.

Beispiel 1.19:

1) Der Ring $\mathbb{Z}$ der ganzen Zahlen ist ein Euklidischer Ring mit $d(n) = |n|$. Nämlich:

$$a \text{ div } b = \left[\frac{a}{b}\right], \qquad a \bmod b = a - b \cdot \left[\frac{a}{b}\right].$$

Nach Definition von [ ] gilt hierbei : $\left[\frac{a}{b}\right] \le \frac{a}{b} < \left[\frac{a}{b}\right] + 1$, woraus für $b > 0$

$$b \cdot \left[\frac{a}{b}\right] \le a < b \cdot \left[\frac{a}{b}\right] + b \quad \Rightarrow \quad 0 \le a - b \cdot \left[\frac{a}{b}\right] < b$$

bzw. für $b < 0$

$$b \cdot \left[\frac{a}{b}\right] \ge a > b \cdot \left[\frac{a}{b}\right] + b \quad \Rightarrow \quad 0 \ge a - b \cdot \left[\frac{a}{b}\right] > b$$

folgt, also in jedem Fall $|a \bmod b| < |b|$. Wir verwenden später ständig die fundamentale Eigenschaft

$$\boxed{0 \le a \bmod b < b \text{ für positive } b \in \mathbb{Z}.}$$

2) Wir werden bald den Ring der Polynome $p = c_n \cdot x^n + c_{n-1}x^{n-1} + \ldots + c_0$ betrachten, der auch ein Euklidischer Ring ist. Hier betrachtet man $d(p) =$ Polynomgrad von p.

Bemerkung 1.20: *Offensichtlich ist in einem Euklidischen Ring die Teilbarkeit $a \mid b$ äquivalent zu $b \bmod a = 0$.*

Lemma 1.21:

Mit Vereinbarung 1.16 gilt im Euklidischen Ring $\mathbb{Z}$:

$$ggT(a, b) = ggT(a \bmod b, b) \quad \forall a, b \in \mathbb{Z} \setminus \{0\}.$$

Beweis: Division mit Rest liefert $a = q \cdot b + r$ mit $r = a \bmod b$.
Sei $c = ggT(a, b)$. Es gilt $c \mid a$ und $c \mid b$, damit muss c auch $r = a - q \cdot b$ teilen. Damit teilt c den Wert $c' := ggT(r, b)$, da dieser größter gemeinsamer Teiler von r und b ist.
Andererseits gilt $c' \mid r$ und $c' \mid b$, damit muss c' auch $a = q \cdot b + r$ teilen. Damit teilt c' den Wert $c = ggT(a, b)$, da dieser größter gemeinsamer Teiler von a und b ist.
Also: $c \mid c'$ und $c' \mid c$. Diese Aussage gilt für einen beliebigen Euklidischen Ring. Mit Vereinbarung 1.16 gilt $c > 0$, $c' > 0$ auf $\mathbb{Z}$. Auf diesem speziellen Ring folgt aus der Teilbarkeit $0 < c \le c'$ und $0 < c' \le c$ und damit $c = c'$.

Q.E.D.

Wie berechnet man nun den ggT zweier ganzer Zahlen, ohne mühsam alle Teiler der Zahlen bestimmen zu müssen? Wir betracheten $ggT(a,b)$ mit $a > b > 0$. Mit Lemma 1.21 können wir die Größe der Zahlen leicht reduzieren, indem wir statt $ggT(a,b)$ einfach $ggT(a \bmod b, b)$ berechnen, wobei $a \bmod b < b$ gilt. Es macht keinen Sinn, weiterzugehen und nun $(a \bmod b) \bmod b$ zu bilden, denn

$$\underbrace{(a \bmod b)}_{\in\{0,1,\ldots,b-1\}} \bmod b = a \bmod b.$$

Mit der offensichtlichen Symmetrie $ggT(x,y) = ggT(y,x)$ können wir aber die Zahlen mittels $b \bmod (a \bmod b)$ weiter reduzieren:

$$ggT(\underbrace{r_1}_{a}, \underbrace{r_2}_{b}) = ggT(r_2, \underbrace{r_1 \bmod r_2}_{r_3}) = ggT(r_3, \underbrace{r_2 \bmod r_3}_{r_4}) = \cdots .$$

Es ergeben sich so Zahlen $r_0 > r_1 > r_2 > r_3 > \ldots \geq 0$. Irgendwann muss eine dieser Zahlen notwendigerweise 0 werden, d.h., man findet

$$r_{i+1} = r_{i-1} \bmod r_i = 0,$$

was bedeutet, dass $r_i \mid r_{i-1}$ gilt. In diesem Fall gilt aber $ggT(r_{i-1}, r_i) = r_i$ und wir sind fertig: der ggT ist berechnet. Die folgende Form des klassischen Euklidischen Algorithmus ignoriert die Vorzeichen von $a, b \in \mathbb{Z}$ und berechnet $ggT(a,b)$ als $ggT(|a|,|b|)$:

Euklidischer Algorithmus 1.22:

Zu berechnen ist $ggT(a,b)$ mit $a, b \in \mathbb{Z} \setminus \{0\}$.

1: Setze $r_0 := \max(|a|,|b|)$; $r_1 := \min(|a|,|b|)$.

2: Für $i = 1, 2, \ldots$ wiederhole

$r_{i+1} := r_{i-1} \bmod r_i$;

bis $r_{i+1} = 0$ eintritt.

3: Ergebnis: $r_i = ggT(a,b)$.

Beweis der Korrektheit: Die i-Schleife in 2) muss abbrechen, da die ganzzahligen Werte r_i streng monoton abnehmen und nicht negativ werden können. Nach Lemma 1.21 gilt in jedem Schritt

$$ggT(r_i, r_{i+1}) = ggT(r_i, r_{i-1} \bmod r_i) = ggT(r_i, r_{i-1}) = ggT(r_{i-1}, r_i).$$

Ist $r_{i+1} = 0$, so gilt $0 = r_{i-1} \bmod r_i$, also $ggT(r_{i-1}, r_i) = r_i$.

Q.E.D.

Satz 1.23: (Laufzeit des Euklidischen Algorithmus)

Der Euklidische Algorithmus 1.22 für $a, b \in \mathbb{Z} \setminus \{0\}$ benötigt maximal $1 + 2 \cdot \log_2(\min(|a|, |b|))$ Wiederholungen von Schritt 2).

Beweis Es sei n die Anzahl der Schritte, die der Euklidische Algorithmus im Schritt 2) benötigt, bis er auf $r_{n+1} = 0$ trifft und $r_n = ggT(a, b)$ ausgibt. In Schritt 2) wird jeweils durch Division mit Rest $r_{i+1} = r_{i-1} \bmod r_i$ berechnet, wobei mit $q_i = r_{i-1} \operatorname{div} r_i$ gilt:

$$r_{i-1} = q_i \cdot r_i + r_{i+1}.$$

Mit $r_{i-1} > r_i$ gilt $q_i \geq 1$, also $r_{i-1} \geq r_i + r_{i+1} > 2 \cdot r_{i+1}$. Hieraus folgt

$$\prod_{i=2}^{n-1} r_{i-1} > 2^{n-2} \cdot \prod_{i=2}^{n-1} r_{i+1}$$

und mit $r_1 > r_2$, $r_{n-1} > r_n \geq 1$:

$$2^{n-2} < \frac{\prod_{i=2}^{n-1} r_{i-1}}{\prod_{i=2}^{n-1} r_{i+1}} = \frac{r_1 \cdot r_2}{r_{n-1} \cdot r_n} < \frac{r_1^2}{r_{n-1} \cdot r_n} < \frac{r_1^2}{2} \quad \Longrightarrow \quad 2^{n-1} < r_1^2.$$

Anwendung von $\log_2$ auf die letzte Ungleichung liefert $n - 1 < \log_2(r_1^2) = 2 \cdot \log_2(r_1)$, wobei $r_1 = \min(|a|, |b|)$.

Q.E.D.

$\boxed{\downarrow 24.11.04}$

Bemerkenswerterweise läßt sich der ggT zweier Zahlen stets als "Linearkombination" dieser Zahlen ausdrücken: es gibt stets (von a und b abhängige) ganze Zahlen s und t mit

$$ggT(a, b) = s \cdot a + t \cdot b.$$

Diese Linearkombination kann durch eine Erweiterung des Euklidischen Algorithmus 1.22 berechnet werden:

Erweiterter Euklidischer Algorithmus 1.24:
Seien $a, b \in \mathbb{Z}$ *mit* $|a| \geq |b| > 0$.

1: *Setze*
$r_0 := |a|$; $s_0 := \operatorname{sign}(a)$; $t_0 := 0$;
$r_1 := |b|$; $s_1 := 0$; $t_1 := \operatorname{sign}(b)$;
wobei $\operatorname{sign}(x) = 1$ *für* $x > 0$ *und* $\operatorname{sign}(x) = -1$ *für* $x < 0$.

2: *Für* $i = 1, 2, \ldots$ *wiederhole:*
Berechne $q_i := r_{i-1}$ div r_i;
Setze $r_{i+1} := r_{i-1} - q_i \cdot r_i$ (*also* $r_{i+1} = r_{i-1} \bmod r_i$);
Setze $s_{i+1} := s_{i-1} - q_i \cdot s_i$;
Setze $t_{i+1} := t_{i-1} - q_i \cdot t_i$;
bis $r_{i+1} = 0$ *eintritt.*

3: *Ergebnis:* $r_i = ggT(a, b)$ *und es gilt* $r_i = s_i \cdot a + t_i \cdot b$.

Beweis der Korrektheit: Der Algorithmus terminiert und liefert $ggT(a, b)$ analog zu Algorithmus 1.22. Es bleibt nur zu zeigen, dass für alle i in Schritt 2) gilt:

$$r_i = s_i \cdot a + t_i \cdot b.$$

Für $i = 0$ und $i = 1$ ist die Behauptung klar per Definition von r_0, s_0, t_0 und r_1, s_1, t_1. Für $i \geq 2$ folgt per Induktion:

$$\begin{aligned} s_i \cdot a + t_i \cdot b &= (s_{i-2} - q_{i-1} \cdot s_{i-1}) \cdot a + (t_{i-2} - q_{i-1} \cdot t_{i-1}) \cdot b \\ &= (s_{i-2} \cdot a + t_{i-2} \cdot b) - q_{i-1} \cdot (s_{i-1} \cdot a + t_{i-1} \cdot b) \\ &= r_{i-2} - q_{i-1} \cdot r_{i-1} \\ &= r_i. \end{aligned}$$

Q.E.D.

Bemerkung 1.25: *Die Laufzeitabschätzung 1.23 gilt natürlich unverändert.*

1.4 Berechnung modularer Inverser

Wir wollen uns nun noch einmal mit dem Ring $\mathbb{Z}_n = \{0, \ldots, n-1\}$ mit der Arithmetik modulo n beschäftigen. Im Rahmen des RSA-Verfahrens sind wir an den sogenannten “Einheiten” modulo n interessiert. Dies sind die Zahlen, die bezgl. der Multiplikation modulo n invertierbar sind, d.h., für die eine multiplikative Inverse $k^{-1} \in \mathbb{Z}_n$ existiert mit

$$k^{-1} \cdot k \bmod n = 1.$$

Solch eine Inverse existiert genau dann, wenn $ggT(k,n) = 1$ gilt (d.h., wenn k und n "teilerfremd" sind):

Lemma 1.26: (Charakterisierung modularer Inverser)

Für $n \in \mathbb{N}$ gilt: $k \in \mathbb{Z}_n \setminus \{0\}$ ist genau dann invertierbar bzgl. der Multiplikation modulo n, wenn $ggT(k,n) = 1$ gilt.

Beweis: Es sei $k \in \mathbb{Z}_n \setminus \{0\}$ invertierbar, d.h., es gibt ein $k^{-1} \in \mathbb{Z}_n$ mit $k^{-1} \cdot k \bmod n = 1$, d.h., $k^{-1} \cdot k = m \cdot n + 1$ für ein $m \in \mathbb{N}$. Damit gilt $k^{-1} \cdot k - m \cdot n = 1$. Der $ggT(k,n)$ teilt sowohl k als auch n und ist damit auch Teiler der Differenz beliebiger Vielfacher von k und n. Also teilt $ggT(k,n)$ die Zahl 1 und es folgt $ggT(k,n) = 1$.

Umgekehrt liefert für k mit $ggT(k,n) = 1$ der Erweiterte Euklidische Algorithmus 1.24 die Darstellung $1 = s \cdot k + t \cdot n$ mit $s, t \in \mathbb{Z}$. Reduktion modulo n liefert

$$1 = s \cdot k \bmod n$$

und damit $s \bmod n$ als multiplikative Inverse von k modulo n.

Q.E.D.

Es ergibt sich als Korollar unmittelbar das schon aus Satz 1.9 bekannte Ergebnis, dass $\mathbb{Z}_p$ ein Körper ist, wenn p eine Primzahl ist: in diesem Fall sind nämlich *alle* Werte $k \in \{1, 2, \ldots, p-1\}$ teilerfremd zu p und damit nach Lemma 1.26 invertierbar.

Der Beweis des obigen Lemmas 1.26 ist konstruktiv (nämlich $k^{-1} = s \bmod n$) und liefert uns unmittelbar den folgenden Algorithmus zur Berechnung modularer Inverser. Er ist so formuliert, dass das modulare Inverse k^{-1} modulo n berechnet wird, falls k invertierbar modulo n ist. Anderenfalls wird entschieden, dass k nicht invertierbar modulo n ist:

Berechnung des modularen Inversen 1.27:

Sei $n \in \mathbb{N}$ und $k \in \mathbb{Z}_n \setminus \{0\}$.

1: *Berechne mit Hilfe des Erweiterten Euklidischen Algorithmus 1.24 die Darstellung $g = ggT(k,n) = s \cdot k + t \cdot n$.*

2: *Ergebnis: Ist $g = 1$, so gilt $k^{-1} = s \bmod n$.*
Ist $g \neq 1$, so ist k nicht invertierbar modulo n.

Dies liefert eine Alternative zur Berechnung von $k^{-1} = k^{p-2} \bmod p$ gemäß Satz 1.9, die gemäß der Bemerkungen 1.11 und 1.25 einen ähnlichen Rechenaufwand erfordert.