In den vorangegangenen Abschnitten wurden folgende Regeln benutzt:

Lemma: (Einige) Rechenregeln für mod

(i) $n \mid m \implies (a \bmod m) \bmod n = a \bmod n$

(ii) $(a \pm b) \bmod n = (a \bmod n \pm b \bmod n) \bmod n$

(iii) $(a * b) \bmod n = (a \bmod n * b \bmod n) \bmod n$

In Worten:

Die „äußere“Anwendung von mod auf eine Summe/Differenz/Produkt kann nach „innen“, also auf die einzelnen Operanden, gezogen werden. Allerdings **muss** mod auf das entsprechende Resultat immer auch äußerlich angewandt werden.

A - 10 Darstellungen ganzer Zahlen

Beobachtung:

Es ist wohl bekannt, daß es eine unendliche monoton steigende Folge von Primzahlen

$$p_1{=}2 \;<\; p_2{=}3 \;<\; p_3{=}5 \;<\; p_4{=}7 \;<\; \ldots \;<\; p_8{=}19 \;<\; \ldots$$

gibt. Mit ihrer Hilfe ergibt sich folgende Darstellung:

Satz A.93 (Primzahlzerlegung)

Jede natürliche Zahl $a > 1$ hat eine eindeutige Darstellung der Form

$$a = \prod_{j=0}^{\infty} p_j^{e_j} \;=\; p_1^{e_1}\, p_2^{e_2} \;\cdots\; p_n^{e_n}\, p_{n+1}^{0}\, p_{n+2}^{0} \;\cdots,$$

wobei nur endlich viele der Exponenten $e_j \in \mathbb{N}$ positiv (d.h. nicht null) sind.

Bemerkung:

Man könnte auf die Idee kommen, positive ganze Zahlen auf Rechnern als Folge ihrer Exponenten $(e_j)_{j \le n}$ abzuspeichern. Läßt man auch negative e_j zu, so ergeben sich sogar alle rationalen Zahlen.

Für **Produkt** und **Quotient** von $a = \prod_{j=1}^{n} p_j^{e_j}$ und $a' = \prod_{j=1}^{n'} p_j^{e'_j}$ gilt

$$a * a' = \prod_{j=1}^{\max(n,n')} p_j^{e_j + e'_j} \qquad a/a' = \prod_{j=1}^{\max(n,n')} p_j^{e_j - e'_j}$$

wobei e_j und e'_j für $j > n$ bzw $j > n'$ als Null angenommen werden.

Auch *GGT* und *KGV* lassen sich billig berechnen (siehe Übung), die Berechnung von Summen und Differenz gestaltet sich jedoch ziemlich aufwendig.

Lemma A.94 (Zahldarstellung zur Basis b)

Für $b \in \mathbb{N}_+$ eine feste Basis (Radix) läßt sich jede beliebige positive Zahl $a \in \mathbb{N}$ mit Hilfe von n Ziffern $a_j \in \{0, 1, \cdots, b-1\}$ eindeutig darstellen:

$$\begin{aligned} a &= (a_n\, a_{n-1}\, a_{n-2} \ldots a_1\, a_0)_b \\ &= \sum_{j=0}^{n} a_j b^j \\ &= a_n b^n \;+\; a_{n-1} b^{n-1} \;+\; \ldots \;+\; a_1 b \;+\; a_0\,, \end{aligned}$$

wobei die führende Ziffer $a_n \neq 0$ gewählt werden muß.

Beispiel A.95
Dezimalsystem *Primaten mit 10 Fingern, Taschenrechner*

Basis b=10, Ziffern $\{0, 1, 2, \ldots, 8, 9\}$

Beispiel A.96
Binärsystem *Computerspeicher*

Basis b=2, Ziffern $\{0, 1\}$

Beispiel A.97
Hexadezimalsystem *Computerausgabe*

Basis b=16, Ziffern $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$

Algorithmus: Darstellung von a zur Basis b

Input: $a \in \mathbb{N}, b \in \mathbb{N}_+$
$i = 0$
`WHILE` $(0 \neq a)$
$a_i := a \bmod b$
$a := (a - a_i) \,/\, b$
$i := i + 1$
Output: $(a_i, a_{i-1}, \ldots, a_0)_b$ – Koeffizienten von a bzgl. Basis b

Beispiel A.98 ($a = (788)_{10} = (???)_3 \qquad b = 3$)

i	0	1	2	3	4	5	6
a	788	262	87	29	9	3	1
$a \bmod b$	2	1	0	2	0	0	1
$(a - a_i)/b$	262	87	29	9	3	1	0
b^i	1	3	9	27	81	243	729
a_i	2	1	0	2	0	0	1

$$(788)_{10} = (1002012)_3 = 1{\cdot}729 + 2{\cdot}27 + 1{\cdot}3 + 2{\cdot}1$$

Bemerkung:
Es gibt verschiedene clevere Tricks um negative Zahlen in das Zahlensystem einzuführen. Bei binärer Darstellung ist es das Einfachste, ein führendes Vorzeichenbit (Signbit) zu benutzen.

Algorithmus: Addition für Basis b

$$\begin{aligned} x &= (x)_b = (x_n, x_{n-1}, \ldots, x_1, x_0) \\ &+ \\ y &= (y)_b = (y_n, y_{n-1}, \ldots, y_1, y_0) \\ &\| \\ z &= (z)_b = (z_n, z_{n-1}, \ldots, z_1, z_0) \end{aligned}$$

Input: $(x)_b, (y)_b$
$q = 0$
FOR $i := 0, 1, 2, \ldots$
$r := (x_i + y_i + q) \bmod b$
$z_i := r$
$q := (x_i + y_i + q - r) \,/\, b$
Output: $(z)_b = (x)_b + (y)_b$ ist Summe von x und y

Hierbei ist q die Übertragsziffer.
Der Aufwand wächst offensichtlich linear mit

$$n = \max\{\, log_b x,\ log_b y \,\}$$

Multiplikationsregel

$$\begin{aligned} x * y = (x)_b * (y)_b &= \sum_{i=0}^{n} x_i b^i \sum_{j=0}^{m} y_j b^j \\ &= (x_0 + x_1 b + x_2 b^2 + \cdots + x_n b^n) * (y_0 + y_1 b + y_2 b^2 + \cdots + y_m b^m) \\ &= x_0 y_0 + (x_0 y_1 + x_1 y_0) b + (x_0 y_2 + x_1 \ldots) b^2 + \ldots b^{m+n} \\ &= \sum_{k=0}^{n+m} z_k b^k \qquad \text{mit} \quad z_k = \sum_{j=0}^{k} x_j y_{k-j} \end{aligned}$$

Anschließend müssen die z_k wie bei der schriftlichen Multiplikation in Potenzen von b zerlegt und die Anteile auf die höheren Terme verteilt werden.

Beispiel A.99 (Oktale Multiplikation)

$$\begin{aligned} (303)_8 &= 3 \cdot 8^0 + 0 \cdot 8^1 + 3 \cdot 8^2 = (195)_{10} \\ (52)_8 &= 2 \cdot 8^0 + 5 \cdot 8^1 + 0 \cdot 8^2 = (42)_{10} \\ (303)_8 * (52)_8 &= 6 \cdot 8^0 + (17)_8 \cdot 8^1 + 6 \cdot 8^2 + (17)_8 \cdot 8^3 \\ &= 6 \cdot 8^0 + 7 \cdot 8^1 + 7 \cdot 8^2 + 7 \cdot 8^3 + 1 \cdot 8^4 \\ &= (8190)_{10} \end{aligned}$$

Bemerkung
Betrachtet man die Addition und Multiplikation von Ziffern mit eventuellem Übertrag als Konsteneinheit, so wächst der Aufwand quadratisch mit der Gesamtanzahl der Ziffern.
Das ähnelt schon sehr Polynommanipulationen.

A - 11 Polynome als Funktionen

Definition A.100 (Polynom)
Einen Ausdruck der Form

$$P(x) = c_0 x^0 + c_1 x^1 + c_2 x^2 + \cdots + c_n x^n$$

nennt man **Polynom**, wobei x eine unbekannte Variable bezeichnet und die **Koeffizienten** c_i für $i = 0..n$ einem Ring $\mathcal{R}$ angehören.

Die nichtnegative ganze Zahl $n = deg(\mathcal{P})$ heisst der **Grad** oder die **höchste Potenz** (degree) des Polynoms.

Für $n = 1, 2, 3$ spricht man von **linearen, quadratischen**, bzw. **kubischen** Polynomen.

Die Zahl $ord(\mathcal{P}) = deg(\mathcal{P}) + 1 = n + 1$ heisst **Ordnung** von P und gibt die Zahl der Koeffizienten $c_0, c_1, \ldots, c_n$ an.

Warnung:
$grad(\mathcal{P})$ bezeichnet im Englischen wie im Deutschen häufig den Gradienten, d.h. den Vektor der partiellen Ableitungen von Polynomen und anderen Funktionen.

Beispiel A.101
Kubisches Polynom über dem Koeffizientenring $\mathbb{Z}$:

$$1 - x + 2x^2 + 17x^3$$

Beispiel A.102
Quadratisches Polynom über dem Koeffizientenring $\mathbb{R}$:

$$\sqrt{2} + \pi x - \frac{1}{2} e x^2$$

Bemerkung:
Ersetzt man x durch ein Element von $\mathcal{R}$ oder einen Oberring $\mathcal{R}' \supset \mathcal{R}$, so erhält man als $P(x)$ wiederum ein Element von $\mathcal{R}$ oder $\mathcal{R}'$.

Durch diese "**Auswertung an der Stelle x**" wird $\mathcal{P}$ zu einer Funktion bzw. Abbildung von $\mathcal{R}$ noch $\mathcal{R}$ oder $\mathcal{R}'$ nach $\mathcal{R}'$.

Lemma A.103 (Horner Schema)
Die Auswertung eines Polynomes mit Hilfe der Klammerung

$$P(x) = c_0 + x * (c_1 + x * (\ \dots \ (c_{n-1} + x * c_n \underbrace{) \dots)}_{n-1})$$

verlangt lediglich n Multiplikationen und ebenso viele Additionen.

Algorithmus Horner-Schema:
Input: $x \in \mathbb{R}, c_i \in \mathbb{R}, i = 0, \dots, n = \deg(P(x))$

$y = 0$
FOR $i := n, n-1, \dots, 1, 0$
$y := c_i + x * y$

Output: $y = P(x)$ ... Wert des Polynoms an der Stelle $x \in \mathbb{R}$

Beispiel A.104

$$P(x) = 1 - x + 2x^2 + 5x^3 = 1 + x * (-1 + x * (2 + 5 * x))$$

Für $x = \frac{1}{2}$

i	3	2	1	0
c_i	5	2	-1	1
y	5	$\frac{9}{2}$	$\frac{5}{4}$	$\frac{13}{8}$

Bemerkung
Polynome sind als relativ einfache Funktionsmodelle nicht nur bei Algebraikern sondern vorallem auch bei Ingenieuren populär (*Vorsicht: Patentanspruch*).
Sie können sehr einfach gespeichert und manipuliert werden.

Allgemeinere Funktionen lassen sich häufig sehr gut durch Polynome oder besser noch Brüche von Polynomen annähern.
Ganz wesentlich ist dabei die folgende Interpolationseigenschaft.