Abschnitt 1: Ringe und Teilbarkeit.

1.1 Definition: Ring, Integritätsring, Körper

1.2 Beispiele von Ringen: Polynomringe, Matrixringe, Quaternionen

1.3 Direkte Summe

1.4 Ideal

1.5 Restklassenring

1.6 Homomorphismus

1.7 Homomorphiesatz

1.8 Einheiten, Atome und Primelemente

1.9 Euklidische Ringe, Hauptidealbereiche, faktorielle Ringe

1.10 Charakterisierung faktorieller Ringe

1.11 Definition: größter gemeinsamer Teiler

1.13 Euklidischer Algorithmus. (Vgl. MAPLE-Programme)

1.14 Verbesserungen Euklidischen Algorithmus für (große) ganze Zahlen.
1. Idee: Ist $a = \sum_{j=1}^{n} a_i M^i$ und $b = \sum_{j=1}^{n} b_i M^i$, $0 \le a_i, b_i < M$ (mit $M = 2^{15}, 2^{31}$ je nach Rechnerstruktur), so hängen die ersten Schritte des Euklidischen Algorithmus nur von a_n und b_n ab, da $a/b \sim a_n/b_n$.
Man führt den Euklidischen Algorithmus parallel für $(a_n + 1, b_n)$ und $(a_n, b_n + 1)$ durch. Macht man dabei dieselben Umformungen, so sind diese auch für (a, b) richtig. Sobald man eine Abweichung bekommt, rechnet man einmal mit den exakten Werten (a, b), bekommt neue Zahlen (a', b') mit ggT(a, b) =ggT(a', b') und rechnet dann mit den höchsten Ziffern von (a', b') weiter (vgl. Algorithmus 1.3.7 in H. Cohen).
2. Idee: Ersetze die Division mit Rest durch binäre shifts:
Ist $a = 2^\alpha a'$ und $b = 2^\beta b'$, mit ungeraden Zahlen a', b',
so ist ggT$(a, b) = 2^{min(\alpha,\beta)}ggT(a', b')$. Setzt man $t := a' - b' = 2^\gamma t'$,
so ist ggT(a', b') =ggT$(t', min(|a'|, |b'|))$.

1.17 Der chinesische Restsatz. (Algorithmus: vgl. MAPLE-Programme)

Abschnitt 2: Invertieren von Matrizen und Determinanten.

2.1 Definition: Inverse, Determinante, Adjungierte

2.2 $A \cdot \text{adj}(A) = \det(A) \cdot I$, $\det(A) \cdot \det(B) = \det(A \cdot B)$.

2.3 Formeln für $\det(A)$ und $\text{adj}(A)$ nur von theoretischer Bedeutung.

2.4 Gauß Algorithmus (über Körpern)

2.5 Idee des Gauß Algorithmus: Finde elementare Matrizen $U_1, \ldots, U_s$ so dass $U_s \cdot \ldots \cdot U_1 \cdot A = D$ eine Dreiecksmatrix ist und berechne dann $A^{-1} = D^{-1} U_s \cdot \ldots \cdot U_1$.

2.6 Gauß Algorithmus zum Berechnen von $\det(A)$.

2.7 Der Gauß Algorithmus ist schnell, hat aber den Nachteil, dass beim Rechnen mit rationalen Zahlen, Zähler und Nenner sehr schnell sehr groß werden.
Fazit: Der Gauß Algorithmus ist ein sehr guter Algorithmus für endliche Körper.

2.9 Hadamard Schranke S für die Determinante

2.10 Berechnung der Determinante einer ganzzahligen Matrix A mit dem chinesischen Restsatz:
Idee: Berechen $d_i := \det(A \pmod{p_i})$ für Primzahlen $p_1, \ldots, p_k$ und bestimme dann $d \in \mathbb{Z}$ $|d| < S$ (aus 2.9) mit $d \equiv d_i \pmod{p_i}$.

2.12 Im Prinzip kann man mit 2.10 auch die Adjungierte von A bestimmen. Zum Invertieren gibt es jedoch einen besseren Algorithmus:

2.13 Algorithmus (p-adisches Invertieren von Matrizen)
Eingabe: $A \in \mathbb{Z}^{n \times n} \cap \mathrm{GL}_n(\mathbb{Q})$, $\lambda \in \mathbb{Z}$ mit $\lambda A^{-1} \in \mathbb{Z}^{n \times n}$. Ausgabe: λA^{-1}.
Idee:
Wähle $p > 2$ Primzahl, mit $p \nmid \lambda$.
Berechne $X \in \mathbb{Z}^{n \times n}$ mit $X \equiv A^{-1} \pmod{p}$.
(Arbeite immer mit dem betragsmäßig kleinsten Rest modulo p^k.)
Setze $B^{(0)} := \lambda X \pmod{p}$.
Bestimme induktiv $B^{(k)} \in \mathbb{Z}^{n \times n}$ mit $B^{(k)} A \equiv I_n \pmod{p^{k+1}}$:
Ist $B^{(k-1)}$ schon bestimmt so ist eine Matrix $C^{(k)}$ gesucht, mit

$$B^{(k)} = B^{(k-1)} + p^k C^{(k)}.$$

Ist

$$B^{(k-1)} A = \lambda I_n + p^k R^{(k-1)}$$

so ergibt sich

$$B^{(k)} A = (B^{(k-1)} + p^k C^{(k)}) A = \lambda I_n + p^k (R^{(k-1)} + C^{(k)} A).$$

Also ist

$$C^{(k)} \equiv -R^{(k-1)} X \pmod{p}.$$

(Zum Algorithmus vgl. auch die MAPLE Programme.)

Kennt man in 2.13 die Zahl λ nicht, so kann man trotzdem den Algorithmus verwenden. Man rechnet dann aber rationale Zahlen (die Einträge von A^{-1}) modulo p^k aus. Zur Bestimmung kleiner $a, b \in \mathbb{Z}$ mit $\frac{a}{b} \equiv c \pmod{p^k}$ benutzt man den folgenden Algorithmus:

2.15 Algorithmus zur rationalen Approximation einer Zahl modulo N.
Eingabe: $c \in \mathbb{Z}$, $N \in \mathbb{N}$.
Ausgabe: $a, b \in \mathbb{Z}$ mit $a \equiv cb \pmod{N}$ und $a^2 + b^2$ minimal.
Idee: Sei $L := \{(a,b) \in \mathbb{Z}^2 \mid a \equiv cb \pmod{N}\}$. Dann ist $L = \mathbb{Z}(c,1) \oplus \mathbb{Z}(N,0)$ ein Gitter im Euklidischen Raum $(\mathbb{R}^2, x^2 + y^2)$. Gesucht ist ein kurzer Vektor in L. Zum Finden dieses Vektors benutzen wir die sogenannte Paarreduktion:

2.16 Paarreduktion:
Eingabe: $L = \mathbb{Z}v_1 \oplus \mathbb{Z}v_2$ Gitter im Euklidischen Raum $(\mathbb{R}^2, \Phi)$.
Ausgabe: $0 \neq v \in L$ mit $\Phi(v,v)$ minimal.
Algorithmus:
(1) Ist $\Phi(v_1,v_1) > \Phi(v_2,v_2)$, so vertausche v_1 und v_2.
(2) Ist $|\Phi(v_1,v_2)| > \frac{1}{2}\Phi(v_1,v_1)$, so setze $k := \lfloor \frac{\Phi(v_1,v_2)}{\Phi(v_1,v_1)} \rfloor$ und $v_2 := v_2 - kv_1$ und gehe zu Schritt (1). Sonst gibt $v = v_1$ aus.
(Zum Algorithmus vgl. auch die MAPLE Programme.)

Abschnitt 3: Gitter und der LLL-Algorithmus.

Generalvoraussetzung: (V,Φ) ist ein Euklidischer Vektorraum, d.h. $V \cong \mathbb{R}^n$ und $\Phi : V \times V \to \mathbb{R}$ positiv definite symmetrische Bilinearform.

3.1 Definition: Gitter, Gitterbasis

3.2 Basiswechsel in Gittern, Determinante eines Gitters.

3.3 Das Gram-Schmidt-Orthogonalisierungsverfahren:
Eingabe: Eine Basis $(b_1, \ldots, b_n)$ von V.
Ausgabe: Eine Orthogonalbasis $B^* := (b_1^*, \ldots, b_n^*)$ von V.
Algorithmus: Für $i = 1, \ldots, n$ berechne sukzessive

$$b_i^* := b_i - \sum_{j=1}^{i-1} \mu_{ij} b_j^*$$

wo $\mu_{ij} = \frac{\Phi(b_i, b_j^*)}{\Phi(b_j^*, b_j^*)}$.
Achtung, die μ_{ij} habe ich in der Vorlesung k_{ij} genannt!
b_i^* ist die Projektion von b_i auf $\langle b_1, \ldots, b_{i-1}\rangle^\perp$.
Die von B und B^* erzeugten Gitter haben die gleiche Determinante, nämlich $\prod_{j=1}^n \Phi(b_j^*, b_j^*)$.
Da $\Phi(b_j^*, b_j^*) \leq \Phi(b_j, b_j)$ ist, ergibt sich 3.4:

3.4 Folgerung: Die Hadamard Ungleichung:
Ist $(b_1, \ldots, b_n)$ eine Gitterbasis von L, so ist $\det(L) \leq \prod_{j=1}^n \Phi(b_j, b_j)$.

3.5 Satz: $L_{\leq S} := \{v \in L \mid \Phi(v,v) \leq S\}$ ist endlich.
Zum Beweis:

3.6 Algorithmus zum Auflisten kurzer Vektoren in Gittern
IDEE: Sei B eine Gitterbasis von L und B^*, μ_{ij} wie in (3.3).
Ist $v = \sum_{j=1}^n a_j b_j \in L$, so ist $v = \sum_{j=1}^n \alpha_j b_j^*$ mit $\alpha_j \in \mathbb{R}$,
$\alpha_n = a_n$, $\alpha_{n-1} = a_{n-1} - \mu_{n,n-1} a_n, \ldots$
Aus

$$\Phi(v,v) = \sum_{j=1}^n \alpha_j^2 \Phi(b_j^*, b_j^*) \leq S$$

folgt insbesondere $a_n^2 \Phi(b_n^*, b_n^*) \leq S$. Also hat man nur endlich viele Möglichkeiten für $a_n \in \mathbb{Z}$. Allgemein gilt

$$\alpha_j^2 \Phi(b_j^*, b_j^*) = (a_j - \sum_{i=j+1}^n \mu_{i,j} a_i)^2 \Phi(b_j^*, b_j^*) \leq S - \sum_{i=j+1}^n \alpha_i^2 \Phi(b_i^*, b_i^*)$$

woraus man sukzessiv nur endlich viele Möglichkeiten für $a_j \in \mathbb{Z}$, $j = n, n-1, \ldots, 1$ erhält.

3.7 Algorithmus zum Auflisten kurzer Vektoren in Gittern
EINGABE: Eine Gitterbasis B eines Gitters L in $(\mathbb{R}^n, \Phi)$ (oder $\Phi(B)$ eine Grammatrix von L) und Schranke $S \in \mathbb{R}$
sowie Parameter $\epsilon := 0.0001$ um Rundungsfehler zu vermeiden.
AUSGABE: Liste K von Koordinatenvektoren $(a_1, \ldots, a_n) \in \mathbb{Z}^n$, so dass

$$L_{\leq S} = \{0\} \cup \{\pm v \mid v = \sum_{i=1}^n a_i b_i,\ (a_1, \ldots, a_n) \in K\}.$$

ALGORITHMUS:

(1) (Gram-Schmidt)
Berechne μ_{ij} $(1 \leq j < i \leq n)$ und
$g_i := \Phi(b_i^*, b_i^*)$ $(1 \leq i \leq n)$ wie in Lemma (3.3).

(2) (Initialisieren)
Setze $k := n$; $f_k := 0$;

(3) (Bestimme maximales a_k)
$u_k := \sum_{j=k+1}^n \mu_{kj} a_j$; $h := round(u_k)$;
while $((g_k(h - u_k)^2 \leq S - f_k + \epsilon)$ or $((h - u_k) < 0))$ do
  $h := h + 1$;
end while;
$a_k := h + 1$;

(4) (Schleife über alle a_k)
$a_k := a_k - 1$;
if $(g_k(a_k - u_k)^2 \leq S - f_k + \epsilon)$ then
  if $(k = 1)$ then
    if $((a_1, \ldots, a_n) = (0, \ldots, 0))$ then RETURN;
    else Ausgabe von $(a_1, \ldots, a_n) =: v$
      und $\Phi(v,v) = f_1 + g_1(a_1 + u_1)^2$; goto (4);

```
                end if;
            else f_{k-1} := f_k + g_k(a_k + u_k)^2;
                k := k - 1; goto (3);
            end if;
    else k:=k+1; goto (4);
    end if;
```

3.8 Satz (Hermite Ungleichung): Sei $L \subseteq V$ ein Gitter. Dann gibt es eine Gitterbasis $B = (b_1, \dots, b_n)$ von L so dass

$$\prod_{i=1}^{n} \Phi(b_i, b_i) \leq \frac{4}{3}^{n(n-1)/2} \det(L).$$

Beweisidee: Zeige zunächst

3.9 Lemma: Sei $v \in L$ mit $\Phi(v, v) = \min\{\Phi(l, l) \mid 0 \neq l \in L\} =: \min(L)$ und $\pi : V \to v^\perp$ die Orthogonalprojektion auf $v^\perp$. Ist $x \in \pi(L)$, so gibt es ein $x_1 \in L$ mit $\pi(x_1) = x$ und $\Phi(x_1, x_1) \leq 4/3\Phi(x, x)$.
Beweis: Sei $x_1 \in L$ so, dass $x_1 - \pi(x_1) = sv$ mit $|s| \leq 1/2$. Dieses x_1 erfüllt die Behauptung.

Beweis von 3.8 Induktion über n: $n = 1$ ist trivial.
$(n-1) \Rightarrow n$: Wähle $b_1 \in L$ mit $\Phi(b_1, b_1) = \min(L)$. Sei $\pi : V \to b_1^\perp$ wie in (3.9) und $L' := \pi(L)$. Nach Induktion gibt es eine Gitterbasis $(b'_2, \dots, b'_n)$ mit $\prod_{i=2}^{n} \Phi(b'_i, b'_i) \leq \frac{4}{3}^{(n-1)(n-2)/2} \det(L')$. Für $i = 2, \dots, n$ sei $b_i \in L$ mit $\pi(b_i) = b'_i$ und $\Phi(b_i, b_i) \leq 4/3\Phi(b'_i, b'_i)$. Dann ist $(b_1, \dots, b_n)$ eine Gitterbasis von L (Übung 3, Aufgabe 11). Da $\det(L')\Phi(b_1, b_1) = \det(L)$ (Übung 3, Aufgabe 11), erfüllt die Basis $(b_1, \dots, b_n)$ die Behauptung von (3.8).

3.10 Definition: (vgl. Lenstra, Lenstra, Lovasz: Math. Annalen 261, 515-534 (1982))
Eine Gitterbasis $B := (b_1, \dots, b_n)$ eines Gitters L heißt *LLL-reduziert* (zum Parameter $\frac{3}{4}$), falls für die μ_{ij} und b_i^* aus (3.3) gilt:

(a) $|\mu_{ij}| \leq 1/2$ für alle $1 \leq j < i \leq n$

(b) $\Phi(b_i^*, b_i^*) + \mu_{i,i-1}^2 \Phi(b_{i-1}^*, b_{i-1}^*) \geq \frac{3}{4}\Phi(b_{i-1}^*, b_{i-1}^*)$ für alle $i = 2, \dots, n$.

(Die Bedingung (b) sagt aus, dass die Projektion von b_i auf $\langle b_1, \dots, b_{i-2}\rangle^\perp$ nicht wesentlich kürzer ist als die Projektion von b_{i-1} auf denselben Raum.) Den Parameter 3/4 in (b) kann man durch eine beliebige reelle Zahl $\alpha \in (1/4, 1)$ ersetzen.

3.11 Satz: Sei $B = (b_1, \dots, b_n)$ eine LLL-reduzierte Basis des Gitters L und $(b_1^*, \dots, b_n^*)$ wie in (3.3). Dann gilt:

(i) $\Phi(b_j, b_j) \leq 2^{i-1}\Phi(b_i^*, b_i^*)$ für alle $1 \leq j \leq i \leq n$.

(ii) $\det(L) \leq \prod_{i=1}^{n} \Phi(b_i^*, b_i^*) \leq 2^{n(n-1)/2} \det(L)$.

(iii) $\Phi(b_1, b_1) \leq 2^{(n-1)/2} \det(L)^{1/n}$.

Ersetzt man den Parameter $3/4$ in (3.10) (b) durch ein α mit $1/4 < \alpha < 1$, so gelten analoge Aussagen zu (3.11) (i), (ii), (iii) in denen man 2 durch $4/(4\alpha - 1)$ ersetzt.
Beweis: (i)

$$\Phi(b_i^*, b_i^*) \geq 1/2\Phi(b_{i-1}^*, b_{i-1}^*)$$

für alle $2 \leq i \leq n$ nach (3.10) (a) und (b). Also folgt durch Induktion:

$$\Phi(b_j^*, b_j^*) \leq 2^{i-j}\Phi(b_i^*, b_i^*)$$

für alle $1 \leq j \leq i \leq n$. Daraus ergibt sich

$$\Phi(b_i, b_i) = \Phi(b_i^*, b_i^*) + \sum_{j=1}^{i-1} \mu_{ij}^2 \Phi(b_j^*, b_j^*) \leq$$

$$\Phi(b_i^*, b_i^*)(1 + 1/4\sum_{j=1}^{i-1} 2^{i-j}) = \Phi(b_i^*, b_i^*)(1 + 1/4 \cdot (2^i - 2)) \leq 2^{i-1}\Phi(b_i^*, b_i^*).$$

Also $\Phi(b_j, b_j) \leq 2^{j-1}\Phi(b_j^*, b_j^*) \leq 2^{i-1}\Phi(b_i^*, b_i^*)$ für alle $1 \leq j \leq i \leq n$.
(ii) Die linke Ungleichung ist die Hadamard Schranke (3.4). Die rechte ergibt sich aus (i):

$$\det(L) = \prod_{j=1}^{n} \Phi(b_j^*, b_j^*) \geq \prod_{j=1}^{n} \frac{1}{2^{j-1}}\Phi(b_j, b_j) = 2^{-n(n-1)/2} \prod_{j=1}^{n} \Phi(b_j, b_j).$$

(iii) Folgt direkt aus (i), da

$$\Phi(b_1, b_1)^n \leq \prod_{i=1}^{n} 2^{i-1}\Phi(b_i^*, b_i^*) = 2^{n(n-1)/2}\det(L).$$

3.12 Folgerung: Ist $(b_1, \ldots, b_n)$ eine LLL-reduzierte Basis des Gitters L, so ist

$$\Phi(b_1, b_1) \leq 2^{n-1}\Phi(x, x)$$

für alle $0 \neq x \in L$.
Beweis: Sei $x = \sum_{i=1}^n a_i b_i = \sum_{i=1}^n \alpha_i b_i^*$ mit $a_i \in \mathbb{Z}, \alpha_i \in \mathbb{R}$ $(1 \leq i \leq n)$. Ist i maximal mit $a_i \neq 0$, so ist $\alpha_i = a_i$ und $\Phi(x, x) \geq a_i^2\Phi(b_i^*, b_i^*) \geq \Phi(b_i^*, b_i^*) \geq 1/2^{i-1}\Phi(b_1, b_1) \geq 1/2^{n-1}\Phi(b_1, b_1)$.

3.13 Satz: Jedes Gitter L hat eine LLL-reduzierte Basis.
Zum Beweis:

3.14 Algorithmus: (LLL-Reduktion)
EINGABE: Gitterbasis $(b_1', \ldots, b_n')$ von L.
AUSGABE: LLL-reduzierte Gitterbasis $(b_1, \ldots, b_n)$ von L und Transformationsmatrix $T = (t_{ij}) \in GL_n(\mathbb{Z})$ mit $b_i = \sum_{j=1}^n t_{ij} b_j'$.
IDEE: Iteration über k: Im Schritt k gilt, dass $(b_1, \ldots, b_{k-1})$ eine LLL-reduzierte Basis ist. Man startet mit $k = 2$. Ist $k = n + 1$, so ist man fertig.
T_k bezeichne die k-te Zeile von T.

Algorithmus: $T := I_n$; $b_i := b'_i$ $(1 \leq i \leq n)$; $k := 2$;
while $(k \leq n)$ do
    if $(k \geq 2)$ then $\mu_{k,k-1} := \Phi(b_k, b^*_{k-1})/\Phi(b^*_{k-1}, b^*_{k-1})$;
        if $(|\mu_{k,k-1}| > 1/2)$ then $r := round(\mu_{k,k-1})$; $b_k := b_k - rb_{k-1}$;
            $T_k := T_k - rT_{k-1}$; end if;
        for $j = 1, \ldots, k-2$ do $\mu_{k,j} := \Phi(b_k, b^*_j)/\Phi(b^*_j, b^*_j)$; end for;
        $b^*_k := b_k - \sum_{j=1}^{k-1} \mu_{k,j} b^*_j$;
        Teste Bedingung (3.10) (b) für $i = k$.
        Fall 1: nicht erfüllt
            vertausche b_k und b_{k-1} und T_k und T_{k-1}; $k := k-1$;
        end Fall 1;
        Fall 2: erfüllt
            for $j = k-2, \ldots, 1$ do $r := round(\mu_{k,j})$;
                $b_k := b_k - rb_j$; $T_k := T_k - rT_j$; $\mu_{k,j} := \mu_{k,j} - r$;
                for $l = 1, \ldots, j-1$ do $\mu_{k,l} := \mu_{k,l} - r\mu_{j,l}$; end for;
            end for;
            k:=k+1;
        end Fall 2;
    else k:=k+1;
    end if;
$k := k+1$;
end while;

3.15 Bemerkung: Algorithmus (3.14) läßt sich so umformulieren, dass er nur mit Grammatrizen arbeitet.

3.16 Satz: Algorithmus (3.14) terminiert.
Beweis: Sei $d_k := \det(\Phi(b_i, b_j)_{1\leq i,j\leq k})$ Setzt man $L_k := \langle b_1, \ldots, b_k\rangle_{\mathbb{Z}}$ so ist $d_k = \det(L_k) = \prod_{j=1}^{k} \Phi(b^*_j, b^*_j)$. Es gilt $d_0 = 1$, $d_n = det(L)$, $d_k \in \mathbb{R}_{>0}$. Ist $m := min(L)$, so ist nach der Hermite Ungleichung (3.8) $d_k \geq \frac{3}{4}^{k(k-1)/2} m^k$ durch eine positive Zahl nach unten beschränkt. Sei $D := \prod_{k=1}^{n-1} d_k$.
Dann ändert sich D nur dann, wenn sich ein $\Phi(b^*_j, b^*_j)$ ändert, also nur im Fall 1.
Dann wird aber d_{k-1} mit einem Faktor $< 3/4$ multipliziert.
d_j bleibt gleich für $j \neq k-1$ da die Menge $\{b_1, \ldots, b_j\}$ und damit auch L_j gleich bleibt. Also wird im Fall 1 die Zahl D mit einem Faktor $< 3/4$ multipliziert. D ist durch eine positive nach unten beschränkt, also tritt Fall 1 nur endlich oft auf und der Algorithmus terminiert.

3.17 MLLL-Algorithmus (=modifizierter LLL-Algorithmus (M. Pohst))
Algorithmus (3.14) kann so abgeändert werden, dass er mit linear abhängigen Vektoren arbeitet.
Eingabe: Teilmenge $(b'_1, \ldots, b'_m)$ von V.
Ausgabe: LLL-reduzierte Menge $(b_1, \ldots, b_m)$ mit $L := \langle b'_1, \ldots, b'_m\rangle_{\mathbb{Z}} = \langle b_1, \ldots, b_m\rangle_{\mathbb{Z}}$ und eine Transformationsmatrix $T = (t_{ij}) \in GL_m(\mathbb{Z})$ mit $b_i = \sum_{j=1}^{m} t_{ij} b'_j$.
Idee: Verfahre wie in (3.14) mit dem folgenden Unterschied: Ist $\Phi(b^*_k, b^*_k) = 0$ (also

$b_k^* = 0$, d.h. b_k ist $\mathbb{R}$-linear abhängig von $b_1, \dots, b_{k-1}$), so ist b_k^* im Gram-Schmidt-Verfahren überflüssig. Setze also $\mu_{l,k} = 0$ für alle $l = k+1, \dots, m$.
Algorithmus: Übung

3.18 Bemerkung: Der MLLL-Algorithmus terminiert.
Beweis: Analog zum Beweis von (3.16). Sei $g_i := \Phi(b_i^*, b_i^*)$ und

$$d_k := \prod_{i \le k, g_i \ne 0} g_i, \quad D := \prod_{k \le m, g_k \ne 0} d_k \prod_{k \le m, g_k = 0} 2^k.$$

D wird nur im Fall 1 verändert.
Ist $g_k \ne 0$, so wird D mit einem Faktor $\le 3/4$ multipliziert, wie in (3.16).
Ist $g_k = 0$, so werden b_k und b_{k-1} vertauscht. d_{k-1} wird auf d_{k-2} gesetzt, d_k bleibt gleich und D wird insgesamt mit einem Faktor $2^{k-1}/2^k = 1/2 < 1$ multipliziert.
Wie in (3.16) ist D durch eine positive Zahl $(\min(L)^{\dim(L)} 2^{m-\dim(L)})$ nach unten beschränkt. Also tritt Fall 1 nur endlich oft ein und (3.17) terminiert.

3.19 Bemerkung: Sei $n := \dim(\langle b_1', \dots, b_m' \rangle_{\mathbb{R}})$. Dann gilt für die in (3.17) berechneten Vektoren $(b_1, \dots, b_m)$: $b_1 = \dots = b_{m-n} = 0$ und $(b_{m-n+1}, \dots, b_m)$ ist eine LLL-reduzierte Gitterbasis von L.

3.20 Anwendung des LLL-Algorithmus: Bestimmen kurzer Vektoren in Gittern:
z.B. Finden $\mathbb{Z}$-linearer Abhängigkeiten komplexer Zahlen.
Eingabe: $z_1, \dots, z_n \in \mathbb{C}$ (bzw. gute Approximationen).
Ausgabe: $a_1, \dots, a_n \in \mathbb{Z}$ mit $\sum_{i=1}^n a_i z_i = 0$.
Idee: Formuliere das Problem so um, dass $(a_1, \dots, a_n)$ ein kurzer Vektor in einem Gitter ist.
Zunächst sei angenommen, dass $z_1, \dots, z_n \in \mathbb{R}$, $z_1 \ne 0$. Wähle $N \in \mathbb{R}_{>0}$ groß und definiere

$$\Phi : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}, \Phi(x,y) := \sum_{j=2}^n x_j y_j + N(\sum_{i=1}^n z_i x_i)(\sum_{i=1}^n z_i y_i).$$

Dann ist Φ positiv definit. Sei $L = \mathbb{Z}^n$. Ist $a = (a_1, \dots, a_n) \in L$, mit $\Phi(a,a)$ klein, so gilt, falls N groß genug ist, dass $\sum_{i=1}^n a_i z_i$ sehr nahe bei 0 ist.
Falls ein z_i/z_1 nicht in $\mathbb{R}$ liegt, so nehmen wir an, dass $z_2/z_1 \in \mathbb{C} \backslash \mathbb{R}$. Dann definieren wir

$$\Phi : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}, \Phi(x,y) := \sum_{j=3}^n x_j y_j + NRe((\sum_{i=1}^n z_i x_i)(\sum_{i=1}^n \bar{z}_i y_i))$$

und verfahren wie eben.

3.21 Bemerkung: (3.20) kann man auch benutzen, um $\mathbb{Z}$-lineare Abhängigkeit von Vektoren $z_1, \dots, z_n \in \mathbb{C}^k$ zu finden.

3.22 Anwendung von (3.20): Faktorisieren von Polynomen.
Eingabe: $f \in \mathbb{Z}[x]$ Polynom und numerische Approximation einer komplexen Nullstelle $z \in \mathbb{C}$ mit $f(z) = 0$.

AUSGABE: $h, g \in \mathbb{Z}[x]$ mit $f = hg$.
IDEE: In der obigen Faktorisierung gilt entweder $h(z) = 0$ oder $g(z) = 0$, d.h. die Koeffizienten von h oder g liefern eine $\mathbb{Z}$-lineare Abhängigkeit der Potenzen von z. Solche Koeffizienten lassen sich also mit Algorithmus (3.20) bestimmen.

Abschnitt 4: Faktorisieren von Polynomen.

Fragestellung: Gegeben ist ein normiertes Polynom $f = f(X) \in \mathbb{Z}[X]$. Gesucht sind irreduzible normierte Polynome $p_1, \ldots, p_s \in \mathbb{Z}[X]$ mit $f = p_1 \ldots p_s$.
Die grobe Strategie zum Faktorisieren von f besteht darin, zunächst f über einem endlichen Körper $\mathbb{Z}/p\mathbb{Z}$ zu faktorisieren (Abschnitt A), diese Faktorisierung zu einer Faktorisierung von f modulo p^k hochzuliften (Abschnitt B) und dann, z.B. mit Hilfe des LLL-Algorithmus, daraus eine Faktorisieung von $f \in \mathbb{Z}[X]$ herzustellen (Abschnitt C).
Teil A: Endliche Körper.
Sei p eine Primzahl und $\mathbb{F}_p := \mathbb{Z}/p\mathbb{Z}$.

4.1 Satz: Sei G eine endliche Gruppe, $n := |G|$. Ist $a \in G$, so gilt $a^n = 1$.
Beweis: $U := \{1, a, a^2, \ldots\} \subset G$, also gibt es $i < j$ mit $a^i = a^j \Leftrightarrow a^{j-i} = 1$. Sei $e > 0$ minimal mit $a^e = 1$. Dann ist $|U| = e$. Ist $U \neq G$, so gibt es $b := b_1 \in G \setminus U$. Dann gilt für $bU := \{b, ba, \ldots, ba^{e-1}\}$, $|bU| = e$ und $bU \cap U = \emptyset$. Ist $b_2 \in G \setminus (bU \cup U)$, so gilt $b_2 U \cap (U \cup b_1 U) = \emptyset$ etc. Erhalten so eine Menge $\{1 =: b_0, b_1, b_2, \ldots, b_s\} \subset G$ mit $G = \dot\bigcup_{j=0}^{s} b_s U$. Also ist $n = (s+1)e$ und $a^n = (a^e)^{s+1} = 1$.
Nun kommen wir zum Faktorisieren von Polynomen $\in \mathbb{F}_p[X]$.

4.2 Bemerkung: Sei $f \in \mathbb{F}_p[X]$. Das Faktorisieren geschieht in 3 Schritten:
(a) Quadratfreie Faktorisierung:
Finde quadratfreie, paarweise teilerfremde Polynome $f_i \in \mathbb{F}_p[X]$ $(1 \leq i \leq k)$, so dass

$$f = f_1 f_2^2 f_3^3 \ldots f_k^k.$$

(b) Verschiedene Grade:
Finde Polynome $f_{i,d} \in \mathbb{F}_p[X]$, so dass

$$f_{i,d} = \prod \{q \mid q \in \mathbb{F}_p[X],\ \mathrm{Grad}(q) = d,\ q \text{ ist irreduzibler Faktor von } f_i\}.$$

(c) Endgültige Faktorisierung:
Faktorisiere $f_{i,d}$ in $\mathrm{Grad}(f_{i,d})/d$ verschiedene irreduzible Faktoren vom Grad d.

(a) Quadratfreie Faktorisierung.

4.3 Lemma: Sei $f = f_1 f_2^2 \ldots f_k^k$ wie in (4.2) (a). Dann ist $\mathrm{ggT}(f, f') = \prod_{p \nmid i} f_i^{i-1} \prod_{p \mid i} f_i^i$.

4.4 Bemerkung: Lemma (4.3) gilt analog über allen perfekten Körpern, also z.B. auch für $\mathbb{Q}[X]$, wo es dann keine Fallunterscheidung $(p \mid i)$ gibt.

4.5 Strategie: (Quadratfreie Faktorisierung):
Setze $T_1 := \mathrm{ggT}(f, f')$, $V_1 := f/T_1 = \prod_{p \not| i} f_i$.
Für $m \geq 2$ sei $V_m := \mathrm{ggT}(V_{m-1}, T_{m-1})$, falls $p \not| m$ und $V_m := V_{m-1}$ sonst. Setze $T_m := T_{m-1}/V_m$. Dann gilt

$$V_m = \prod_{i \geq m, p \not| i} f_i, \; T_m = \prod_{i > m, p \not| i} f_i^{i-m} \prod_{p|i} f_i^i.$$

Deshalb ist $f_m = V_{m-1}/V_m$ für alle m mit $p \not| m$. Am Ende ist V_{k+1} konstant und $T_{k-1} = \prod_{p|i} f_i$ eine p-te Potenz. Nun bestimmt man die p-te Wurzel aus T_{k-1}, was in $\mathbb{F}_p[X]$ ganz einfach geht:

4.6 Lemma: Die Abbildung $F : \mathbb{F}_p[X] \to \mathbb{F}_p[X], q \mapsto q^p$ ist $\mathbb{F}_p$-linear. Insbesondere ist $q(X)^p = q(X^p)$.
Beweis: Seien $q_1, q_2 \in \mathbb{F}_p[X]$, $a, b \in \mathbb{F}_p$. Dann ist $(aq_1 + bq_2)^p = \sum_{j=0}^{p} \binom{p}{j} a^j q_1^j b^{p-j} q_2^{p-j} = a^p q_1^p + b^p q_2^p = aq_1^p + bq_2^p$ mit Satz (4.1).
Ein expliziter Algorithmus zu (4.5) wird in der Übung entwickelt.

(b) verschiedene Grade:

4.7 Satz: Zu jedem $d \in \mathbb{N}$ gibt es (bis auf Isomorphie) genau einen Körper $\mathbb{F}_{p^d}$ mit p^d Elementen, nämlich den Zerfällungskörper von $\phi_d(X) := X^{p^d} - X \in \mathbb{F}_p[X]$ über $\mathbb{F}_p$.
Beweis: Sei K ein Körper, $|K| = p^d$. Dann gilt für alle $a \in K$, dass $a^{p^d} = a$. Dies ist klar für $a = 0$. Ist $a \neq 0$, so ist a eine Einheit in K, also $a \in K^*$. Die Einheitengruppe K^* ist eine endliche Gruppe der Ordnung $p^d - 1$. Also gilt $a^{p^d - 1} = 1$ nach Satz (4.1). Multiplikation mit a liefert $a^{p^d} = a$.
Jetzt definieren wir einen Homomorphismus: $\iota : \mathbb{F}_p = \mathbb{Z}/p\mathbb{Z} \to K, 1 + p\mathbb{Z} \mapsto 1_K$. Wir müssen zeigen, dass ι wohldefiniert ist. Dazu betrachten wir die additive Gruppe $(K, +)$. Nach Satz (4.1) gilt $|K|1 = 1 + \ldots + 1 = 0$ ($|K|$ Summanden) also $p^d \cdot 1 = (p \cdot 1) \ldots (p \cdot 1) = 0$. Da K nullteilerfrei ist, folgt als $p \cdot 1 = 0$ und ι ist wohldefiniert.
Für das Polynom $\phi_d(X) \in \mathbb{F}_p[X]$ gilt, dass $\mathrm{ggT}(\phi_d(X), \phi_d'(X)) = \mathrm{ggT}(X^{p^d} - X, -1) = 1$. Also hat ϕ_d in seinem Zerfällungskörper keine mehfachen Nullstellen und K besteht gerade aus den p^d verschiedenen Nulstellen von ϕ_d.

4.8 Folgerung: Sei $f \in \mathbb{F}_p[X]$ ein irreduzibles Polynom vom Grad d. Dann ist $\mathbb{F}_p[X]/(f(X)) \cong \mathbb{F}_{p^d}$. Das Produkt aller irreduziblen Polynome in $\mathbb{F}_p[X]$ vom Grad d ist

$$\phi_d(X)/\mathrm{ggT}(\phi_d(X), \prod_{j=1}^{d-1} \phi_j(X)).$$

4.9 Algorithmus zur Bestimmung der $f_{i,d}$ aus (4.2) (b):
Eingabe: quadratfreies Polynom $f \in \mathbb{F}_p[X]$.
Ausgabe: $f_1, \ldots, f_k$ mit $f_d = \prod\{q \in \mathbb{F}_p[X] \mid q \mid f,\ q \text{ irreduzibel vom Grad } d\}$.
Algorithmus:
for j from 1 to Grad (f) do

$f_j := \mathrm{ggT}(f, X^{p^j} - X)$;
$f := f/f_j$;
end for;

(c) endgültige Faktorisierung:

4.10 Lemma: Sei $p > 2$, $A \in \mathbb{F}_p[X]$ Produkt von verschiedenen irreduziblen Polynomen von Grad d. Dann ist für alle $T \in \mathbb{F}_p[X]$

$$A = \mathrm{ggT}(A, T)\mathrm{ggT}(A, T^{(p^d-1)/2} - 1)\mathrm{ggT}(A, T^{(p^d-1)/2} + 1).$$

Beweis: In $\mathbb{F}_{p^d}[X]$ zerfällt A in Linearfaktoren.
Ist $\alpha \in \mathbb{F}_{p^d}$, so ist auch $T(\alpha) \in \mathbb{F}_{p^d}$ und deshalb

$$(T^{p^d} - T)(\alpha) = T(\alpha)^{p^d} - T(\alpha) = 0$$

mit Satz (4.7). Also teilt jeder der Linearfaktoren von $A \in \mathbb{F}_{p^d}[X]$ genau einen der 3 paarweis teilerfremden Faktoren $T, T^{(p^d-1)/2} - 1, T^{(p^d-1)/2} + 1$ von $T^{p^d} - T$ und deshalb teilt auch jeder irreduzible Faktor von $A \in \mathbb{F}_p[X]$ genau eines dieser drei Polynome. Da A quadratfrei ist, folgt die Behauptung.
Ist A nicht irreduzibel, so ist die Wahrscheinlichkeit, dass $\mathrm{ggT}(A, T^{(p^d-1)/2} - 1)$ ein nichttrivialer Teiler von A ist, recht gross ($\sim 1/2$). Daraus ergibt sich folgender rekursiver, probabilistischer Algorithmus:

4.11 Algorithmus (Cantor, Zassenhaus) ($p > 2$)
Eingabe: $A \in \mathbb{F}_p[X]$ Produkt von mindestens zwei verschiedenen irreduziblen Polynomen vom Grad d.
Ausgabe: Ein nichttrivialer Faktor B von $A = B\frac{A}{B}$.
Algorithmus: $k := \mathrm{Grad}(A)$;
while (TRUE) do
Wähle zufällig $T \in \mathbb{F}_p[X]$ mit $\mathrm{Grad}(T) \leq 2d - 1$;
Berechne $B := \mathrm{ggT}(T^{(p^d-1)/2}, A)$ mit dem Euklidischen Algorithmus;
if $k > \mathrm{Grad}(B) > 0$ then RETURN(B); end if;
end while;
Einen analogen Algorithmus gibt es auch für $p = 2$. Für kleine Primzahlen ist jedoch der Berlekamp-Algorithmus besser:

4.12 Lemma:
Sei $A \in \mathbb{F}_p[X]$ quadratfrei, $A = \prod_{i=1}^r A_i$ eine Zerlegung von A in irreduzible Faktoren. Dann ist

$$\{T \in \mathbb{F}_p[X] \mid T^p \equiv T \pmod{A}\} = \{T \in \mathbb{F}_p[X] \mid$$
$$\text{es gibt Konstanten } s_1, \ldots, s_r \in \mathbb{F}_p \text{ mit } T \equiv s_i \pmod{A_i} \text{ für alle } 1 \leq i \leq r\}.$$

Beweis: Nach dem chinesischen Restsatz ist der Ring

$$\mathbb{F}_p[X]/(A) \cong \bigoplus_{i=1}^r \mathbb{F}_p[X]/A_i \cong \bigoplus_{i=1}^r \mathbb{F}_{p^{d_i}}$$

wo $d_i = \mathrm{Grad}(A_i)$. Die Elemente $T \in \mathbb{F}_p[X]/(A)$ mit $T^p = T$ sind nach Satz (4.7) genau die Elemente in $\oplus_{i=1}^r \mathbb{F}_p$.

4.13 Strategie: (Berlekamp Algorithmus)
Sei $A \in \mathbb{F}_p[X]$ quadratfrei, $n := \mathrm{Grad}(A)$.
Bestimme zunächst eine $\mathbb{F}_p$-Basis von $\mathcal{M} := \{T \in \mathbb{F}_p[X] \mid \mathrm{Grad}(T) < n, T^p \equiv T \pmod{A}\}$: Ist $T(X) = \sum_{j=0}^{n-1} t_j X^j \in \mathcal{M}$, so ist $T^p(X) = T(X^p) = \sum_{j=0}^{n-1} t_j X^{pj}$.
Berechne also die Matrix $Q = (q_{ji}) \in \mathbb{F}_p^{n\times n}$ mit

$$X^{pj} \equiv \sum_{i=0}^{n-1} q_{ji} X^i \pmod{A}.$$

Dann gilt $T^p \equiv T \pmod{A} \Leftrightarrow (t_0, \ldots, t_{n-1}) \in \mathrm{Kern}(Q - I_n)$.
Bestimme also eine $\mathbb{F}_p$-Basis von $\mathrm{Kern}(Q - I_n)$ mit dem Gauß Algorithmus.
Da $1^p = 1$ gilt $\mathbb{F}_p \subset \mathcal{M}$ (d.h. $(1, 0, \ldots, 0) \in \mathrm{Kern}(Q - I_n)$). Jeder von 1 l.u. Vektor $t \in \mathrm{Kern}(Q-I_n)$ entspricht aber einem nicht konstanten $T \in \mathcal{M}$, für das Konstanten $s_i \in \mathbb{F}_p$ existieren mit $T \equiv s_i \pmod{A_i}$. Die $s_i \in \mathbb{F}_p$ sind nicht alle gleich und

$$A = \prod_{s\in\mathbb{F}_p} \mathrm{ggT}(T - s, A)$$

ist eine nichttriviale Faktorisierung von A.
Daraus ergibt sich ein expliziter deterministischer Algorithmus, den wir in der Übung besprechen.

Teil B) Liften modulo p^k

4.14 Satz: (Hensel'sches Lemma)
Seien $k \geq 1$, $A, B_k, C_k, U, V \in \mathbb{Z}[X]$ mit $\mathrm{Grad}(A) = \mathrm{Grad}(B_k) + \mathrm{Grad}(C_k)$, B_k normiert, $\mathrm{Grad}(U) < \mathrm{Grad}(C_k)$, $\mathrm{Grad}(V) < \mathrm{Grad}(B_k)$ die die Bedingungen

$$\star_k: \quad A \equiv B_k C_k \pmod{p^k}, \quad UB_k + VC_k \equiv 1 \pmod{p}$$

erfüllen. Dann gibt es (modulo p^{k+1} eindeutig bestimmte) Polynome $B_{k+1}, C_{k+1} \in \mathbb{Z}[X]$, $B_{k+1} \equiv B_k \pmod{p^k}$, normiert, $C_{k+1} \equiv C_k \pmod{p^k}$, $\mathrm{Grad}(B_k) = \mathrm{Grad}(B_{k+1})$, $\mathrm{Grad}(C_k) = \mathrm{Grad}(C_{k+1})$ die $\star_{k+1}$ erfüllen.
Beweis: Sei $B_k C_k = A - R_k$. Setzen an: $B_{k+1} := B_k + p^k S$, $C_{k+1} := C_k + p^k T$. Da

$$B_{k+1}C_{k+1} = (B_k+p^kS)(C_k+p^kT) \equiv C_kB_k+p^k(C_kS+B_kT) = A+p^k(C_kS+B_kT-R_k)$$

modulo p^{2k}, sagt $\star_{k+1}$, dass $C_kS + B_kT \equiv R_k \pmod{p}$ sein muss. Setzen also $S := R_kV + WB_k \pmod{p}$ mit $W \in \mathbb{Z}[X]$, so dass $\mathrm{Grad}(S) < \mathrm{Grad}(B_k)$ und $T := R_kU - WC_k \pmod{p}$ (die Koeffizienten von S und T liegen in $[-p/2, p/2]$). Dann ist $C_kS + B_kT \equiv R_k \pmod{p}$. Da $\mathrm{Grad}(R_k) \leq \mathrm{Grad}(A) = \mathrm{Grad}(B_k) + \mathrm{Grad}(C_k)$ ist, und $\mathrm{Grad}(S) < \mathrm{Grad}(B)$ ist also $\mathrm{Grad}(T) \leq \mathrm{Grad}(R_k) - \mathrm{Grad}(B_k) \leq \mathrm{Grad}(C_k)$ (da B_k normiert ist).

4.15 Bemerkung: Ersetzt man U, V in $\star_k$ durch U_k, V_k mit $U_kB_k + V_kC_k \equiv 1 \pmod{p^k}$, so kann man mit einer zum Beweis von (4.14) analogen Methode B_{k+1}, C_{k+1} finden, die $\star_{2k}$ erfüllen (quadratische Konvergenz).

Teil C) Faktorisierung in $\mathbb{Z}[X]$
Problem: Gegeben $f \in \mathbb{Z}[X]$, $f = \sum_{i=0}^n a_i X^i$, primitiv (d.h. $\mathrm{ggT}(a_0, \ldots, a_n) = 1$) und quadratfrei (d.h. $\mathrm{ggT}(f, f') = 1$).
Gesucht ist eine Faktorisierung von f in irreduzible Polynome in $\mathbb{Z}[X]$.
Das Faktorisieren geschieht in 3 Etappen:
A) Suche eine Primzahl p mit $\mathrm{ggT}(f \pmod p, f' \pmod p) = 1$ (in $\mathbb{F}_p[X]$).
Faktorisiere $f \pmod p$, d.h. finde Polynome $f_1, \ldots, f_s \in \mathbb{Z}[X]$ mit $f_i \pmod p \in \mathbb{F}_p[X]$ irreduzibel, normiert

$$f \equiv a_n f_1 \ldots f_s \pmod p.$$

Dazu werden die Algorithmen aus Teil 4.A) benutzt.
B) Lifte diese Faktorisierung zu einer Faktorisierung von f modulo p^k

$$f \equiv a_n \tilde{f}_1 \ldots \tilde{f}_s \pmod{p^k}$$

mit dem Hensel'schen Lemma (4.14). Hier genügt es, zunächst einen irreduziblen Faktor (z.B. f_1) zu $h = \tilde{f}_1$ zu liften.
C) Bestimme mit Hilfe des irreduziblen Faktors h von $f \pmod{p^k}$ aus B) einen Faktor h_0 von f in $\mathbb{Z}[X]$:

4.16 Satz:
Vor.: Sei $f \in \mathbb{Z}[X]$ primitiv, $n := \mathrm{Grad}(f)$ und $h \in \mathbb{Z}[X]$ normiert mit
(i) $h \pmod p \in \mathbb{F}_p[X]$ irreduzibel.
(ii) $(h \pmod p)^2 \nmid f \pmod p \in \mathbb{F}_p[X]$.
(iii) $h \pmod{p^k} \mid f \pmod{p^k} \in (\mathbb{Z}/p^k\mathbb{Z})[X]$.
Beh.: Dann hat f genau einen irreduziblen Faktor $h_0 \in \mathbb{Z}[X]$ mit $h \pmod p \mid h_0 \pmod p$.
Für jeden Teiler g von f in $\mathbb{Z}[X]$ sind äquivalent:
$h \pmod p \mid g \pmod p \in \mathbb{F}_p[X]$.
$\Leftrightarrow h \pmod{p^k} \mid g \pmod{p^k} \in (\mathbb{Z}/p^k\mathbb{Z})[X]$.
$\Leftrightarrow h_0 \mid g \in \mathbb{Z}[X]$.
Beweis: einfaches Rechnen in faktoriellen Ringen, z.H.

4.17 Algorithmus: (finde das h_0 aus Satz (4.16))
Idee: Die Bedingung $h \pmod{p^k} \mid h_0 \pmod{p^k}$ bedeutet, dass

$$h_0 \pmod{p^k} = (h \pmod{p^k})(g \pmod{p^k}) = g_0 h + g_1 X h + \ldots + g_j X^j h \pmod{p^k}$$

mit $g_0, \ldots, g_j \in \mathbb{Z}$ wo $j = m - l$, $m := \mathrm{Grad}(h_0)$, $l := \mathrm{Grad}(h)$. Anders ausgedrückt: h_0 liegt in dem von $h, Xh, \ldots, X^j h, p^k, p^k X, \ldots, p^k X^j$ erzeugten $\mathbb{Z}$-Gitter L (der Dimension $m + 1$). Identifiziert man die Polynome $\in \mathbb{Z}[X]$ vom Grad $\leq m$ mit ihren Koeffiziententupel, also z.B. $h =: \sum_{i=0}^l c_i X^i$ mit $(c_0, \ldots, c_l, 0 \ldots, 0)$ und $h_0 =: \sum_{i=0}^m v_i X^i$ mit $v = (v_0, \ldots, v_m)$, so wird das Gitter L von $(c_0, \ldots, c_l, 0 \ldots, 0)$, $(0, c_0, \ldots, c_l, 0 \ldots, 0)$, ..., $(0, \ldots, 0, c_0, \ldots, c_l)$, $(p^k, 0, \ldots, 0)$, ... $(0, \ldots, 0, p^k, 0, \ldots, 0)$ erzeugt. Wir betrachten L als Gitter im $\mathbb{R}^{m+1}$ mit Standardskalarproduct Φ. Sind die Koeffizienten von h_0 im Vergleich zu p^k klein, so ist h_0 ein kurzer Vektor in L.

(Vergleiche Algorithmus 3.22).
Mit Hilfe des LLL-Algorithmus kann man so ganzzahlige Polynome (deterministisch in polynomialer Zeit) faktorisieren. Genau gilt folgender Satz:

4.18 Satz: (ohne Beweis)
Mit den Bezeichnungen aus (4.16) und (4.17) sei $l := \mathrm{Grad}(h)$ und k so groß dass

$$p^{kl} > 2^{mn/2}\binom{2m}{n}^{n/2}(\sum_{i=0}^{n} a_i^2)^{(m+n)/2}$$

Sei $(b_1, \ldots, b_{m+1})$ eine LLL-reduzierte Basis von L. Dann gilt:

$$\mathrm{Grad}(h_0) \leq m \Leftrightarrow \quad (\star) \quad \Phi(b_1, b_1) < (p^{2kl}/(\sum_{i=0}^{n} a_i^2)^m)^{1/n}.$$

Genauer: Ist $(\star)$ erfüllt, so sei t das maximale $j \in 1, \ldots, m+1$, für welches

$$(\star_j) \quad \Phi(b_j, b_j) < (p^{2kl}/(\sum_{i=0}^{n} a_i^2)^m)^{1/n}$$

gilt. Dann ist
$\mathrm{Grad}(h_0) = m + 1 - t$,
$h_0 = \mathrm{ggT}(b_1, \ldots, b_t)$ ($\in \mathbb{Z}[X]$, die Vektoren von L werden mit Polynomen identifiziert)
und $(\star_j)$ gilt für alle $j = 1, \ldots, t$.

Abschnitt 5: Kryptographie.

Zunächst wollen wir die Struktur von $(\mathbb{Z}/m\mathbb{Z})^*$ untersuchen.

5.1 Definition: Die Eulersche φ-Funktion:

$$\varphi : \mathbb{N} \to \mathbb{N}; \quad \varphi(m) := |(\mathbb{Z}/m\mathbb{Z})^*|$$

5.2 Bemerkung: $(\mathbb{Z}/m\mathbb{Z})^* \equiv \{a \in \{1, \ldots, m-1\} \mid \mathrm{ggT}(a, m) = 1\}$. Ist also $m = \prod_{j=1}^{s} p_j^{\alpha_j}$ mit paarweise verschiedenen Primzahlen p_j und $\alpha_j \geq 1$, dann ist

$$\varphi(m) = \prod_{j=1}^{s} p_j^{\alpha_j - 1}(p_j - 1).$$

5.3 Lemma: $\sum_{d|m} \varphi(d) = m$.
Beweis: $m = \prod_{j=1}^{s} p_j^{\alpha_j}$. Induktion über $\sum_{j=1}^{s} \alpha_j =: n$.
$n = 1$: Klar.
Induktionsschritt:

$$\sum_{d|m} \varphi(d) = \sum_{d|m/p_1} \varphi(d) + \sum_{d|m/(p_1^{\alpha_1})} \varphi(p_1^{\alpha_1} d) =$$

$$m/p_1 + p_1^{\alpha_1 - 1}(p_1 - 1)p_2^{\alpha_2} \ldots p_s^{\alpha_s} = m.$$

5.4 Bezeichnungen: Sei G eine endliche Gruppe, $a \in G$, $n := |G|$.
Ordnung von a, $\operatorname{ord}(a) :=$ minimales $m \in \mathbb{N}$ mit $a^m = 1$.
Exponent von G, $\exp(G) := \operatorname{kgV}(\operatorname{ord}(a) \mid a \in G)$.
G heißt zyklisch, falls ein $a \in G$ existiert mit $G = \{1, a, \ldots, a^{\operatorname{ord}(a)-1}\} =: \langle a \rangle$.
In Zeichen: $\langle a \rangle =: C_{\operatorname{ord}(a)}$ die (bis auf Isomorphie eindeutig bestimmte) zyklische Gruppe der Ordnung $\operatorname{ord}(a)$.
Klar: G zyklisch $\Rightarrow |G| = \exp(G)$ (Umkehrung gilt nicht).
$\operatorname{ord}(a) = |\langle a \rangle| = \exp(\langle a \rangle)$.
$a \in G \Rightarrow \operatorname{ord}(a) \mid \exp(G) \mid |G|$.

5.5 Satz: p Primzahl $\Rightarrow (\mathbb{Z}/p\mathbb{Z})^* = \mathbb{F}_p^* \cong C_{p-1}$.
Beweis: Zeigen: Für jeden Teiler d von $p-1$ hat $\mathbb{F}_p^*$ genau $\varphi(d)$ Elemente der Ordnung d.
Denn: Sei $\psi(d) := \{a \in \mathbb{F}_p^* \mid \operatorname{ord}(a) = d\}$. Dann gilt $d \nmid p-1 \Rightarrow \psi(d) = 0$.
Ist $a \in \mathbb{F}_p^*$, $\operatorname{ord}(a) = d$, so ist $\langle a \rangle$ die Menge der d verschiedenen Nullstellen von $X^d - 1$. Die Anzahl der Elemente der Ordnung d in $\langle a \rangle$ ist $\varphi(d)$. Also ist $\psi(d) \leq \varphi(d)$.
Deshalb ist

$$p - 1 = |\mathbb{F}_p^*| = \sum_{d|p-1} \psi(d) \leq \sum_{d|p-1} \varphi(d) = p - 1.$$

Also $\psi(d) = \varphi(d)$. Insbesondere ist $\psi(p-1) \neq 0$ und $\mathbb{F}_p^* = \langle a \rangle$ für jedes Element $a \in \mathbb{F}_p^*$ mit $\operatorname{ord}(a) = p - 1$.

5.6 Satz: Sei p eine Primzahl, $\alpha \geq 1$.
(a) Ist $p > 2$, so ist $(\mathbb{Z}/p^\alpha\mathbb{Z})^* \cong C_{p^{\alpha-1}(p-1)}$.
(b) Ist $\alpha \geq 2$, so ist $(\mathbb{Z}/2^\alpha\mathbb{Z})^* \cong C_2 \times C_{2^{\alpha-2}}$.
Beweis: (a) Mit Hilfe der binomischen Formel zeigt man: Ist $\beta \geq 1$, $b \in \mathbb{Z}$, $p \nmid b$, dann ist

$$(1 + p^\beta b)^p = (1 + p^{\beta+1} c)$$

für ein $c \in \mathbb{Z}$ mit $p \nmid c$.
Ist $a \in \mathbb{Z}$ mit $\langle (a + p\mathbb{Z}) \rangle = (\mathbb{Z}/p\mathbb{Z})^*$, so ist $a^{p-1} = 1 + pb$. Gilt $p \nmid b$, so ist $\langle a + p^\alpha\mathbb{Z} \rangle = (\mathbb{Z}/p^\alpha\mathbb{Z})^*$. Falls $p \mid b$, dann ist $\langle a + p + p^\alpha\mathbb{Z} \rangle = (\mathbb{Z}/p^\alpha\mathbb{Z})^*$.
(b) als Übung.

5.7 Folgerung: Ist $m = \prod_{j=1}^s p_j^{\alpha_j}$ ungerade, so ist $(\mathbb{Z}/m\mathbb{Z})^* \cong \prod_{j=1}^s C_{p_j^{\alpha_j - 1}(p_j-1)}$ vom Exponenten $\prod_{j=1}^s p_j^{\alpha_j-1} \operatorname{kgV}(p_j - 1 \mid 1 \leq j \leq s)$.

Nun zur Kryptographie:
Kryptographie dient zum einen zum abhörsicheren Austausch von Nachrichten über belauschte Kanäle, zum anderen aber auch zur Authentifikation, d.h. der Empfänger will prüfen können, dass die Nachricht unverändert vom Absender kommt.
Man unterscheidet 2 wesentlich verschiedene Strategien, die private-key und die public-key Krytosysteme:

5.8 Private Key Kryptosysteme:
A und B einigen sich <u>geheim</u> auf einen Schlüssel f

(z.B. n zufällige bits $(x_1, \ldots, x_n) \in \{0,1\}^n$).
Will A eine Nachricht an B schicken, so verschlüsselt A die Nachricht N zu $f(N)$, welches er sendet. Mit dem gleichen Schlüssel (d.h. mit f^{-1}, das hier leicht aus f zu berechnen ist), kann B aus $f(N)$ wieder die Nachricht N berechnen.
(z.B. A konvertiert N als bit-string, den er in Stücke der Länge n unterteilt und sendet anstelle von $(s_1, \ldots, s_n)$ die bit-Folge $(s_1 + x_1 \pmod 2), \ldots, s_n + x_n \pmod 2)$).
B entschlüsselt durch Berechnen von $s_i = (s_i + x_i) + x_i \pmod 2$.)
Vorteil: Schnell und relativ sicher.
Nachteil: Schlüsselaustausch muss geheim erfolgen.
Wollen n Personen miteinander kommunizieren, so müssen insgesamt $\binom{n}{2}$ Schlüssel je zweimal abgespeichert werden.
Wir wollen uns hier nur mit dem Schlüsselaustausch beschäftigen:

5.9 Diffie-Hellman Schlüsselaustausch.
A und B wollen einen gemeinsamen geheimen Schlüssel berechnen. Zur Kommunikation stehen ihnen jedoch nur belauschte Kanäle zur Verfügung.
Sie wählen öffentlich eine große Primzahl p und $g \in \mathbb{F}_p^*$.
A wählt geheim eine Zahl $k_A < p$.
B wählt geheim eine Zahl $k_B < p$.
A schickt g^{k_A} an B.
B schickt g^{k_B} an A.
Der gemeinsame Schlüssel ist $g^{k_A k_B} = (g^{k_A})^{k_B} = (g^{k_B})^{k_A}$.
Ein Lauscher kennt p, g, g^{k_A} und g^{k_B}, kann aber daraus nicht (einfach) $g^{k_A k_B}$ berechnen.

5.10 Bemerkung: Die Berechnung von g^k benötigt $\sim 2\log_2(k)$ Multiplikationen.
Ist $k = \sum_{i=0}^{n} k_i 2^i$ mit $k_i \in \{0,1\}$, so ist $g^k = \prod_{k_i=1} g^{2^i}$:
ALGORITHMUS:
$R := 1$; $F := g$;
while $(k > 0)$ do
  if $(k \pmod 2)$ then $R := RF$; end if;
  $k := k$ div 2; $F := F^2$;
end while; RETURN(R);

5.11 Das Diskrete Logarithmen Problem:
Sei G eine Gruppe und $g \in G$. Man kennt $g, h \in \langle g \rangle$ und sucht $a \in \mathbb{N}$ mit $g^a = h$.
METHODE: Shanks Baby-Step-Giant-Step Algorithmus:
$n := \mathrm{ord}(g)$, $m := \lceil \sqrt{n} \rceil$.
Berechne $h, hg^{-1}, hg^{-2}, \ldots, hg^{-(m-1)}$ (baby-steps)
und $1, g^m, g^{2m}, \ldots, g^{m(m-1)}$ (giant-steps)
Ist $h = g^{im+j}$ mit $0 \le i, j < m$, so ist $hg^{-j} = g^{im}$, d.h. ein Eintrag in der ersten Liste ist gleich einem Eintrag in der zweiten Liste.
Zeit- und Platzbedarf: $\sim m \sim \sqrt{n}$.
Kennt man Faktorisierung $n = n_1 n_2$ mit $\mathrm{ggT}(n_1, n_2) = 1$, so kann man das DL-Problem reduzieren:
Man berechnet j_1, j_2 mit $g^{n_i j_i} = h^{n_i} \in \langle g^{n_i} \rangle$, und a_1, a_2 mit $a_1 n_1 + a_2 n_2 = 1$. Dann

ist $h = h^{a_1 n_1} h^{a_2 n_2} = g^{a_1 n_1 j_1 + a_2 n_2 j_2}$.
Ähnlich kann man Primzahlpotenzen ausnutzen: Ist $n = p^\alpha$, und $h = g^a$, so kann man $a = \sum_{i=0}^{\alpha-1} a_i p^i$ mit $a_i \in \{0, \dots, p-1\}$ schreiben. Dann gilt $h^{p^{\alpha-1}} = (g^{p^{\alpha-1}})^{a_0} \in \langle g^{p^{\alpha-1}} \rangle$. Daraus erhält man a_0. Setzt man $h' := hg^{-a_0}$, so ist $h' \in \langle g^p \rangle$ und man kann mit g^p weiterarbeiten.
Für allgemeine zyklische Gruppen gibt es bisher keinen wesentlich besseren Algorithmus. Man kann jedoch den Platzbedarf durch einen einfachen probabilistischen Algorithmus stark reduzieren.
Für das DL-Problem in $\mathbb{F}_p^*$ gibt es jedoch bessere Algorithmen, die aus den Faktorisierungsalgorithmen in Abschnitt 7 entwickelt worden sind. Es scheint gleich schwer zu sein, das DL-Problem in $\mathbb{F}_p^*$ zu lösen, wie eine Zahl $N \sim p$ zu faktorisieren.
Der Diffie-Hellmann-Schlüsselaustausch scheint also zur Zeit sicher zu sein, falls p mehr als 250 Stellen hat, und $p - 1 = 2q$ für eine Primzahl q ist.

5.12 Digital Signature Algorithm. (DSA)
Man kann das Rechnen in $\mathbb{F}_p^*$ auch zum Unterschreiben von Nachrichten benutzen. Das folgende Verfahren ist 1991 von der US-Regierung als Standardverfahren vorgeschlagen worden:
Will A die Nachricht N unterschreiben, so braucht A:
1) Primzahl q (ca. 50 Stellen) (öffentlich)
2) Primzahl $p \equiv 1 \pmod q$ (ca. 250 Stellen) (öffentlich)
3) Einen Erzeuger g der Untergruppe $C_q \leq \mathbb{F}_p^* \cong C_{p-1}$. (öffentlich)
4) Eine zufällige Zahl $0 < x < q$ (geheim). Öffentlich bekannt: $y = g^x \pmod p$.
Sowie eine öffentlich bekannte hashfunktion $H : \{\text{ Nachrichten }\} \to \{1, \dots, q-1\}$
Zum Signieren der Nachricht N, berechnet A den Wert $H := H(N) \in \{1, \dots, q-1\}$, wählt zufälliges $k \in \{1, \dots, q-1\}$, berechnet das $r \in \{0, \dots, p-1\}$ mit $g^k \equiv r \pmod p$, sowie $\overline{r} := r \pmod q$. Ist $\overline{r} = 0$ oder $H + x\overline{r} \equiv 0 \pmod q$, so wählt A ein neues k, sonst bestimmt er $s \in \{1, \dots, q-1\}$ mit $sk \equiv H + x\overline{r} \pmod q$.
Die Unterschrift von A ist dann $(\overline{r}, s)$ (ca. 100 Stellen).
Der Empfänger B verifiziert die Unterschrift:
Er berechnet $H := H(N)$,
$u_1 := s^{-1}H \pmod q$, $u_2 := s^{-1}\overline{r} \pmod q$, sowie $g^{u_1} y^{u_2} \equiv g^{s^{-1}H} g^{xs^{-1}\overline{r}} \equiv g^k \pmod p$. Falls das Ergebnis $\equiv \overline{r} \pmod q$, so ist die Nachricht N nicht verfälscht und stammt von A.

Wir kommen nun zu anderen Gruppen zum Schlüsselaustausch und Authentifikation: elliptische Kurven.

5.13 Definition: Sei $p > 3$ eine Primzahl, $a, b \in \mathbb{F}_p$ so dass $X^3 + aX + b$ keine mehrfachen Nullstellen hat, (also $4a^3 + 27b^2 \neq 0$).

$$E := E_{a,b}(\mathbb{F}_p) = \{(x, y) \in \mathbb{F}_p^2 \mid y^2 = x^3 + ax + b\} \cup \{O\}$$

heißt *elliptische Kurve* über $\mathbb{F}_p$. (besser: die Menge der $\mathbb{F}_p$-rationalen Punkte auf E).

5.14 Definition: (Addition auf $E := E_{a,b}(\mathbb{F}_p)$):
Ist $P = (x, y) \in E$, so ist $-P := (x, -y)$. $P + (-P) := O$.

Ist $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2) \in E$ mit $P_1 \neq -P_2$, so gilt $P_1 + P_2 =: P_3 = (x_3, y_3) \in E$ mit

$$x_3 = \lambda^2 - x_1 - x_2,\ y_3 = \lambda(x_1 - x_3) - y_1,$$

wo $\lambda := (y_2 - y_1)/(x_2 - x_1)$, falls $P_1 \neq P_2$ und $\lambda := (3x_1^2 + a)/(2y_1)$, falls $P_1 = P_2$.

5.15 Bemerkung: Die obige Addition macht aus E eine kommutative Gruppe mit O als neutralem Element.

5.16 Beispiel: E : definiert durch $Y^2 = X^3 - X$, $p = 7$.
$E(\mathbb{F}_7) = \{O, (0,0), (1,0), (4,2), (4,-2), (-1,0), (-2,1), (-2,-1)\}$.

5.17 Bemerkung: Es gibt einen sehr guten (probabilistischen) Algorithmus, um Punkte in $E(\mathbb{F}_p)$ (E beliebige elliptische Kurve über $\mathbb{F}_p$) zu finden: Sei $f(X) := X^3 + aX + b$.
Wähle $x \in \mathbb{F}_p$ zufällig.
Teste, ob $z := f(x)$ ein Quadrat in $\mathbb{F}_p^*$ (oder $= 0$) ist. Beachte: $z \in (F_p^*)^2 \Leftrightarrow z^{(p-1)/2} = 1$.
Falls ja, so berechne $y := \sqrt{z} \in \mathbb{F}_p$:
Ist $p \equiv 3 \pmod 4$, so ist $y := z^{(p+1)/4}$.
Ist $p \equiv 1 \pmod 4$, so sei $p - 1 = 2^s t$ mit ungeradem $t \in \mathbb{N}$. Sei $u \in \mathbb{F}_p^* \setminus (\mathbb{F}_p^*)^2$ und $v := u^t$ (Element der Ordnung 2^s).
Setze $y_1 := z^{(t+1)/2}$. Dann ist $y_1^2 = zz^t$. Da $z^t \in \langle v^2 \rangle$ gibt es ein l mit $z = v^{2l}$ (mit 5.11 zu bestimmen). Dann ist $y := y_1 v^{-l}$ eine Wurzel aus z.

5.18 Satz von Hasse (ohne Beweis, da sehr tiefliegend)
$p + 1 - 2\sqrt{p} \leq |E(\mathbb{F}_p)| \leq p + 1 + 2\sqrt{p}$.

5.19 Schlüsselaustausch mit elliptischen Kurven:
Wie Diffie-Hellman (5.9) A und B wollen einen gemeinsamen geheimen Schlüssel berechnen.
Sie wählen öffentlich eine große Primzahl p, elliptische Kurve $E = E_{a,b}(\mathbb{F}_p)$ und $P \in E$.
A wählt geheim eine Zahl $k_A \sim p$.
B wählt geheim eine Zahl $k_B \sim p$.
A schickt $k_A P$ an B.
B schickt $k_B P$ an A.
Der gemeinsame Schlüssel ist $k_A k_B P$
Ein Lauscher kennt p, E, P, $k_A P$ und $k_B P$, Die Berechnung von $k_A k_B P$ scheint nicht leichter zu sein als das Lösen des DL-Problems für die Gruppe $G = \langle P \rangle$. Im Gegensatz zu endlichen Körpern gibt es hierfür noch keine besseren Verfahren, als die in (5.11) vorgestellten.

5.20 Authentifikation mit elliptische Kurven.
Annahme: $\mathrm{ord}(P) = q$ ist eine Primzahl in der Größenordnung von p.
Der Teilnehmer A kennt k_A und macht $Q := k_A P$ öffentlich bekannt.
A wählt zufällig $k \in \{2, \ldots, q-2\}$, berechnet $kP = (x_1, y_1) \in E$ mit $x_1, y_1 \in \{0, \ldots, p-1\}$ sowie $r := x_1 \pmod q$ und $s = k^{-1}(H(N) + xr) \pmod q$. Ist $r = 0$

oder $s = 0$, so wählt A ein neues k.
Unterschrift: (r, s).
B verifiziert die Unterschrift mit Hilfe einer authentifizierten Kopie von A's öffentlichem Schlüssel Q.
Er berechnet $w = s^{-1} \pmod q$ und $H(N)$. Dann $u_1 := H(N)w \pmod q$ und $u_2 := rw \pmod q$, sowie $u_1 P + u_2 Q = (x_0, y_0)$ mit $x_0 \in \{0, \ldots, p-1\}$.
Er reduziert x_0 modulo q und akzeptiert die Nachricht, falls $x_0 \equiv r \pmod q$.

5.21 Definition: Ein Public-Key-Kryptosystem mit Authentifikation besteht aus
einer Menge von Teilnehmern $T = \{A, B, \ldots\}$,
einer Nachrichtenmenge $\mathcal{N}$,
einer öffentlich bekannten Hashfunktion: $H : \mathcal{N} \to \mathcal{N}$
sowie öffentlich bekannten bijektiven Funktionen $f_t : \mathcal{N} \to \mathcal{N}$ für alle $t \in T$.
Jeder Teilnehmer t kennt sein f_t^{-1} (geheim).
Will A an B eine Nachricht $N \in \mathcal{N}$ senden, so berechnet er $f_B(N)$ und $f_B(f_A^{-1}(H(N)))$ und sendet diese Daten über (belauschte) Kanäle an B.
B berechnet $f_B^{-1}(f_B(N)) = N$ (mit seinem geheimen Schlüssel f_B^{-1}) und $H(N)$ (mit der öffentlichen Hashfunktion). B kann feststellen, dass die Nachricht N (unverfälscht) von A kommt, indem er $H(N)$ mit $f_A(f_B^{-1}(f_B(f_A^{-1}(H(N)))))$ vergleicht.

5.22 Das RSA-Verfahren. Empfänger B wählt 2 (große) Primzahlen p, q und eine Zahl $v \in \{3, \ldots, (p-1)(q-1)\}$ mit $\mathrm{ggT}(v, (p-1)(q-1)) = 1$ und bestimmt $e \in \mathbb{N}$ mit $ve \equiv 1 \pmod{(p-1)(q-1)}$. B berechnet $m := pq$. Der öffentliche Schlüssel von B ist (m, v).
Zur Veschlüsselung von N berechnet A zunächst die m-adische Entwicklung von $N \in \mathbb{N}$, $N = \sum_{i=0}^k n_i m^i$ mit $n_i \in \{0, \ldots, m-1\}$ und schickt die Ziffern n_i hintereinander. Ist $n \in \{0, \ldots, m-1\}$, so ist $f_B(n) = n^v \pmod m \in \{0, \ldots, m-1\}$, $f_B(N) = f_B(n_0), \ldots, f_B(n_k)$.
Der Empfänger B berechnet die n_i aus den $f_B(n_i) = n_i^v$ als $n_i = (f_B(n_i))^e \pmod m$.

5.23 Lemma: Ist $m = pq$ Produkt von 2 Primzahlen, und $a \in \mathbb{N}$ mit $a \equiv 1 \pmod{(p-1)(q-1)}$, so ist für alle $n \in \mathbb{Z}$, $n^a \equiv n \pmod m$.

5.24 Kosten von RSA:
Berechnung von n^v mit Algorithmus (5.10), $\sim \log_2(v)$.
Berechnung von $f_B(n)^e \pmod m$ mit chinesischem Restsatz: genügt $ev \equiv 1 \pmod{\mathrm{ggT}(p-1, q-1)}$ bzw. zwei Zahlen e_p, e_q mit $e_p v \equiv 1 \pmod{p-1}$, $e_q v \equiv 1 \pmod{q-1}$.
Berechne $x_p := f_B(n)^{e_p} \pmod p$ und $x_q := f_B(n)^{e_q} \pmod q$ und dann $n \in \{0, \ldots, m-1\}$ mit $n \equiv x_p \pmod p$ und $n \equiv x_q \pmod q$ mit dem chinesischen Restsatz.

5.25 Bemerkung: Kann der Angreifer m faktorisieren, so kann er f_B^{-1} (also den Entschlüsselungsexponenten e) berechnen. Die Berechnung von e ist sogar äquivalent mit einer Faktorisierung von m, jedoch ist nicht bekannt, ob das v-te Wurzelziehen modulo m nicht leichter zu lösen ist. Damit das Faktorisieren von m schwierig ist, müssen p und q geeignete, in etwa 100-stellige Primzahlen sein.

Abschnitt 6: Zusammengesetzheitstests und Primzahlbeweise.

6.1 Satz: (Fermat-Test)
Sei $N \in \mathbb{N}$. Gibt es ein $a \in \mathbb{Z}$ mit $\mathrm{ggT}(a, N) = 1$ und $a^{N-1} \not\equiv 1 \pmod N$, so ist N <u>keine</u> Primzahl.

6.2 Bemerkung: Die Umkehrung von (6.1) gilt nicht. Falls alle zu N teilerfremden $a \in \mathbb{Z}$ die Kongruenz $a^{N-1} \equiv 1 \pmod N$ erfüllen, so folgt nur, dass der Exponent $\exp(\mathbb{Z}/N\mathbb{Z}^*)$ ein Teiler von $N-1$ ist.
Zusammengesetze Zahlen N mit $\exp(\mathbb{Z}/N\mathbb{Z}^*) \mid N-1$ nennt man *Carmichael-Zahlen*.
Beispiel: $N = 561 = 3 \cdot 11 \cdot 17$ ist eine Carmichael Zahl.

6.3 Bemerkung: N ist Carmichael Zahl $\Leftrightarrow$ N ist zusammengesetzt, quadratfrei und $p-1 \mid N-1$ für alle Primteiler p von N.

6.4 Satz: Sei $N \in \mathbb{N}$ ungerade, $N - 1 = d2^s$ mit ungeradem d.
Genau dann ist N keine Primzahl, wenn es ein $a \in \mathbb{Z}$ gibt, $\mathrm{ggT}(a, N) = 1$ welches nicht

$$\star_N \quad a^d \equiv 1 \pmod N \text{ oder } a^{d2^r} \equiv -1 \pmod N \text{ für ein } r = 0, 1, \ldots, s-1$$

erfüllt.
Falls a die Bedingung $\star_N$ erfüllt, so nennt man N eine *Pseudoprimzahl* zur Basis a.
Beweis: <u>$\Leftarrow$</u>: Sei N eine Primzahl, $a \in \mathbb{Z}$, $N \not| a$. Dann ist $a + N\mathbb{Z} \in \mathbb{Z}/N\mathbb{Z}^*$ und $x := \mathrm{ord}(a + N\mathbb{Z})$ ist ein Teiler von $N - 1 = 2^s d$. Also ist x von der Form $x = 2^t d'$ mit $d' \mid d$ und $t \leq s$ und $a^{d2^{t-1}} \equiv 1 \pmod N$.
<u>$\Rightarrow$</u>: Angenommen für alle $a \in \mathbb{Z}$ mit $\mathrm{ggT}(a, N) = 1$ gilt $\star_N$. Dann ist N eine Carmichael-Zahl, also insbesondere quadratfrei, $N = p_1 \ldots p_t$ mit paarweise verschiedenen ungeraden Primzahlen p_i. Ist N zusammengesetzt, so ist nach dem chinesischen Restsatz

$$(\mathbb{Z}/N\mathbb{Z})^* \cong (\mathbb{Z}/p_1\mathbb{Z})^* \times (\mathbb{Z}/p_2 \ldots p_t\mathbb{Z})^*.$$

Sei $\alpha \in \mathbb{Z}$ mit $\langle \alpha + p_1\mathbb{Z} \rangle = (\mathbb{Z}/p_1\mathbb{Z})^*$ und $a \in \mathbb{Z}$ mit $a \equiv \alpha \pmod{p_1}$ und $a \equiv 1 \pmod{p_2 \ldots p_t}$. Dann hat $a + N\mathbb{Z} \in (\mathbb{Z}/N\mathbb{Z})^*$ gerade Ordnung, d.h. $a^d \not\equiv 1 \pmod N$. Allerdings ist $a^n \equiv 1 \pmod{p_2 \ldots p_t}$ für alle $n \in \mathbb{N}$, also gilt nie $a^n \equiv -1 \pmod N$.

6.5 Algorithmus (Rabin-Miller Test)
Eingabe: $N \in \mathbb{N}$, ungerade.
Ausgabe: N ist zusammengesetzt: **Z** oder N ist wahrscheinlich eine Primzahl: **P**.
Algorithmus:
Sei $N - 1 = d2^s$ mit d ungerade.
1) for z from 0 to 50 do
  Wähle $a \in \{2, \ldots, N-2\}$ zufällig.
  Sei g:= ggT(a,N).
  if $g \neq 1$ then RETURN(**Z**)

else

Berechne $m := a^d \pmod N$.

if $m \equiv 1 \pmod N$ then goto **1)**

else

for i from 0 to s-1 do

if $m \equiv -1 \pmod N$ then goto **1)** end if;

$m := m \cdot m$;

end for;

RETURN(**Z**);

end if;

end if;

end for;

RETURN(**P**);

6.6 Bemerkung: Ist N zusammengesetzt, so ist N eine Pseudoprimzahl zur Basis a für höchstens die Hälfte der $a \in (\mathbb{Z}/N\mathbb{Z})^*$. Also ist die Wahrscheinlichkeit, dass Algorithmus (6.5) **P** ausgibt, obwohl N zusammengesetzt ist, kleiner als 2^{-50}.
Es gibt nur 101 zusammengesetzte $N \leq 10^{12}$, die Pseudoprimzahlen zu den Basen $2, 3$, und 5 sind.

Wir kommen nun zu Primzahlbeweisen:

6.7 Satz:
Sei $N \in \mathbb{N}$, $N-1 = \prod_{j=1}^n q_j^{\beta_j}$ mit paarweise verschiedenen Primzahlen q_j. Angenommen es gibt Zahlen $a_j \in \mathbb{Z}$ mit

$$a_j^{(N-1)/q_j} \not\equiv 1 \pmod N, \text{ aber } a_j^{N-1} \equiv 1 \pmod N \ (j = 1, \ldots, n).$$

Dann ist N eine Primzahl.
Beweis: Denn dann teilt $q_j^{\beta_j}$ den Exponenten von $\mathbb{Z}/N\mathbb{Z}^*$ für alle j, also $N-1 \mid \exp(\mathbb{Z}/N\mathbb{Z}^*) \mid |\mathbb{Z}/N\mathbb{Z}^*| \leq N-1$. Also ist $|\mathbb{Z}/N\mathbb{Z}^*| = N-1$ und N ist eine Primzahl.

6.8 Bemerkung: Wir haben gezeigt, dass N geanu dann eine Primzahl ist, wenn $\exp(\mathbb{Z}/N\mathbb{Z}^*) = N-1$.

6.9 Bemerkung: Um mit Satz (6.7) die Primzahleigenschaft von N zu beweisen, muss man $N-1$ faktorisieren, was i.a. nicht wesentlich einfacher ist, als N zu faktorisieren. Da zusammengesetzte Zahlen $N \in \mathbb{N}$ aber immer einen Primfaktor $p \leq \sqrt{N}$ haben, kann man (6.7) verbessern:

6.10 Satz: Sei $N \in \mathbb{N}$, $N-1 = RF$, $F = \prod_{j=1}^n q_j^{\beta_j} > \sqrt{N}$ mit paarweise verschiedenen Primzahlen q_j und $\mathrm{ggT}(R, F) = 1$. Angenommen es gibt Zahlen $a_j \in \mathbb{Z}$ mit

$$\mathrm{ggT}(a_j^{(N-1)/q_j} - 1, N) = 1, \text{ aber } a_j^{N-1} \equiv 1 \pmod N \ (j = 1, \ldots, n).$$

Dann ist N eine Primzahl.
Beweis: Sonst sei p ein Primteiler von N mit $p \leq \sqrt{N}$. Die Bedingungen zeigen, dass die Ordnung von $a_j + p\mathbb{Z} \in (\mathbb{Z}/p\mathbb{Z})^*$ durch $q_j^{\beta_j}$ teilbar ist. Also ist der Exponent von $\mathbb{Z}/p\mathbb{Z}^*$ durch F teilbar, was ein Widerspruch zu $F > \sqrt{N} \geq p$ ist.

6.11 Definition: $F_n := 2^{2^n} + 1$ heißt die n-te Fermat-Zahl.
$F_1, \dots, F_4$ sind die einzigen bekannten Fermat-Primzahlen, $F_5, \dots, F_{11}$ kann man voll faktorisieren, von $F_{12}, \dots, F_{22}$ weiss man, dass sie zusammengesetzt sind.

6.12 Satz (Pépin): F_n ist Primzahl $\Leftrightarrow 3^{2^{2^n-1}} \equiv -1 \pmod{F_n}$.
Zum Beweis benötigen wir das quadratische Reziprozitätsgesetz:

6.13 Definition: (Legendre Symbol)
Sei p eine Primzahl, $n \in \mathbb{Z}$.

$$\left(\frac{n}{p}\right) := \begin{cases} 0 & p \mid n \\ 1 & n \in (\mathbb{Z}/p\mathbb{Z}^*)^2 \\ -1 & n \in (\mathbb{Z}/p\mathbb{Z}^*) \setminus (\mathbb{Z}/p\mathbb{Z}^*)^2 \end{cases}$$

Klar: p ungerade $\Rightarrow \left(\frac{n}{p}\right) \equiv n^{(p-1)/2} \pmod{p}$.
$\left(\frac{n}{p}\right)$ hängt nur von $n + p\mathbb{Z}$ ab.

6.14 Bemerkung: $\left(\frac{a}{p}\right)\left(\frac{b}{p}\right) = \left(\frac{ab}{p}\right)$

6.15 Satz: Sei p ungerade Primzahl.
(i) $\left(\frac{1}{p}\right) = 1$.
(ii) $\left(\frac{-1}{p}\right) = (-1)^{(p-1)/2}$.
(iii) $\left(\frac{2}{p}\right) = (-1)^{(p^2-1)/8}$.
Beweis: (i) und (ii) sind klar.
(iii) Sei α primitive 8-te Einheitswurzel in $\overline{\mathbb{F}_p}$. Dann erfüllt $y := \alpha + \alpha^{-1}$ die Gleichung $y^2 = 2$. Außerdem ist $y^p = \alpha^p + \alpha^{-p}$.
Ist $p \equiv \pm 1 \pmod{8}$, so ist $y^p = y$, also $2^{(p-1)/2} \equiv y^{p-1} \equiv 1 \pmod{p}$.
Ist $p \equiv \pm 5 \pmod{8}$, so ist $y^p = -y$, also $2^{(p-1)/2} \equiv y^{p-1} \equiv -1 \pmod{p}$.

6.16 Satz (Quadratisches Reziprozitätsgesetz) (Gauß)
Sind $p \neq q$ zwei ungerade Primzahlen, so gilt

$$\left(\frac{q}{p}\right)\left(\frac{p}{q}\right) = (-1)^{(p-1)(q-1)/4}.$$

Beweis: Sei w primitive q-te Einheitswurzel in $\overline{\mathbb{F}_p}$ und

$$y := \sum_{a \in \mathbb{Z}/q\mathbb{Z}} \left(\frac{a}{q}\right) w^a$$

<u>1. Behauptung:</u> $y^2 = (-1)^{(q-1)/2} q$.
Denn: $y^2 = \sum_{a,b \in \mathbb{Z}/q\mathbb{Z}} \left(\frac{ab}{q}\right) w^{a+b} = \sum_{c \in \mathbb{Z}/q\mathbb{Z}} \sum_{a \in \mathbb{Z}/q\mathbb{Z}} \left(\frac{a(c-a)}{q}\right) w^c$. Ist $a \neq 0$, dann ist $\left(\frac{a(c-a)}{q}\right) = \left(\frac{-a^2}{q}\right)\left(\frac{1-ca^{-1}}{q}\right) = (-1)^{(q-1)/2}\left(\frac{1-ca^{-1}}{q}\right)$. Für $a = 0$ ist $\left(\frac{a(c-a)}{q}\right) = 0$. Also ist $(-1)^{(q-1)/2} y^2 = \sum_{c \in \mathbb{Z}/q\mathbb{Z}} S(c) w^c$ mit $S(c) := \sum_{a \in (\mathbb{Z}/q\mathbb{Z})^*} \left(\frac{1-ca^{-1}}{q}\right) w^c$. Ist $c = 0$, so ist $S(c) = q - 1$. Ist $c \neq 0$, so durchläuft $1 - ca^{-1}$ alle Elemente von

$\mathbb{Z}/q\mathbb{Z}$ ungleich 1. Also ist $S(c) = \sum_{b\in\mathbb{Z}/q\mathbb{Z}} \left(\frac{b}{q}\right) - \left(\frac{1}{q}\right) = 0 - 1 = -1$. Deshalb ist $(-1)^{(q-1)/2} y^2 = -\sum_{c\in\mathbb{Z}/q\mathbb{Z}} w^c + q = q$.
2. Behauptung: $y^{p-1} = \left(\frac{p}{q}\right)$.
Denn: $y^p = \sum_{a\in\mathbb{Z}/q\mathbb{Z}} \left(\frac{a}{q}\right) w^{pa} = \sum_{b\in\mathbb{Z}/q\mathbb{Z}} \left(\frac{bp^{-1}}{q}\right) w^b = \left(\frac{p^{-1}}{q}\right) y = \left(\frac{p}{q}\right) y$, woraus Behauptung 2 durch Multiplikation mit y^{-1} folgt.
Also ist insgesamt $\left(\frac{p}{q}\right) = y^{p-1} = (y^2)^{(p-1)/2} = ((-1)^{(q-1)/2} q)^{(p-1)/2}$
$= (-1)^{(p-1)(q-1)/4} q^{(p-1)/2} \equiv (-1)^{(p-1)(q-1)/4} \left(\frac{q}{p}\right) \pmod p$. Da beide Seiten $\in \{\pm 1\}$ sind, folgt daraus die Gleichheit in $\mathbb{Z}$.
Beweis von (6.12) $\Leftarrow$: folgt aus Satz (6.7).
$\Rightarrow$: Sei F_n Primzahl. Zu zeigen: $\left(\frac{3}{F_n}\right) = -1$. Nach (6.16) ist $\left(\frac{3}{F_n}\right) = \left(\frac{F_n}{3}\right) = -1$ da $F_n \equiv -1 \pmod 3$ ist. Letzteres sieht man wie folgt ein: Es gilt für alle ungeraden Zahlen u, dass $X+1 \mid X^u+1$ in $\mathbb{Z}[X]$. Also ist $F_n+1 = 2(2^{2^n-1}+1)$ durch $(2+1) = 3$ teilbar.

6.17 Definition: $M_p := 2^p - 1$ heißt *Mersenne Zahl.*

6.18 Bemerkung: $d \mid p \Rightarrow X^d - 1 \mid X^p - 1$ in $\mathbb{Z}[X]$. Ist also M_p eine Primzahl, so ist p eine Primzahl.
Man kennt die ersten 37 Primzahlen p, für die M_p eine Primzahl ist, $p = 2, 3, 5, 7, \ldots, 3.021.377$.
Die größte und vermutlich 38-te Mersenne Primzahl ist M_p mit $p = 6.972.593$ und mehr als 2 Mio. Dezimalstellen.
I.a. kann man nicht $M_p - 1$ faktorisieren, d.h. Satz (6.10) ist nicht anwendbar, um M_p auf Primzahleigenschaft zu testen. Wir benötigen eine Strategie, die eine Faktorisierung von $N+1$ ausnutzt:

6.19 Definition: Sei A eine kommutative Gruppe, $U \leq A$ eine Untergruppe. Dann wird

$$A/U := \{aU \mid a \in A\}$$

durch $(aU)(bU) := abU$ zu einer Gruppe, die *Faktorgruppe von A nach U*. Es gilt für $|A| < \infty$: $|A| = |U||A/U|$.

6.20 Bemerkung: Sei p eine ungerade Primzahl, $\left(\frac{D}{p}\right) = -1$. Dann ist $\mathbb{F}_{p^2} \cong \mathbb{F}_p[\sqrt{D}] \cong \mathbb{F}_p[X]/(X^2 - D)$.
Wie Satz (5.5) zeigt man $\mathbb{F}_{p^2}^* \cong C_{p^2-1}$. Also ist $(\mathbb{F}_{p^2}^*)/(\mathbb{F}_p^*) \cong C_{p+1}$.

6.21 Definition: Sei N ungerade, $\mathrm{ggT}(D, N) = 1$. Definiere $Q := \mathbb{Z}/N\mathbb{Z}[\sqrt{D}] = \{a + b\sqrt{D} \mid a, b \in \mathbb{Z}/N\mathbb{Z}\} \cong \mathbb{Z}/N\mathbb{Z}[X]/(X^2-D)$. Sei $\sigma : Q \to Q, (a+b\sqrt{D}) \mapsto (a-b\sqrt{D})$.
Klar: σ ist ein Ringautomorphismus und für $\alpha \in Q$ gilt $\alpha \in \mathbb{Z}/N\mathbb{Z} \Leftrightarrow \sigma(\alpha) = \alpha$.
Insbesondere ist für $\alpha = a + b\sqrt{D} \in Q$ $N(\alpha) := \alpha\sigma(\alpha) = a^2 - b^2 D \in \mathbb{Z}/N\mathbb{Z}$ und $\alpha \in Q^* \Leftrightarrow N(\alpha) \in (\mathbb{Z}/N\mathbb{Z})^*$.

6.22 Satz: Sei $N = \prod_{j=1}^s p_j^{\alpha_j}$ mit paarweise verschiedenen Primzahlen $p_j > 2$, $D \in N$ mit $\mathrm{ggT}(D, N) = 1$. Betrachte den Ring $Q := \mathbb{Z}/N\mathbb{Z}[\sqrt{D}]$. Dann gilt

$$|Q^*| = \varphi(N) \prod_{j=1}^s p_j^{\alpha_j - 1}(p_j - \left(\frac{D}{p}\right))$$

und $\exp(Q^*/(\mathbb{Z}/N\mathbb{Z})^*)$ teilt $\prod_{j=1}^s p_j^{\alpha_j-1}\mathrm{kgV}\{(p_j - \left(\frac{D}{p_j}\right)) \mid j=1,\ldots,s\}$.
Beweis: $Q \cong \oplus_{j=1}^s(\mathbb{Z}/p_j^{\alpha_j}\mathbb{Z})[X]/(X^2-D)$, nach dem chinesischen Restsatz, also genügt es, die Behauptung für Primzahlpotenzen $N = p^\alpha$ zu zeigen. Dann gilt $Q^* = \{a+b\sqrt{D} \mid a,b \in \mathbb{Z}/N\mathbb{Z}, p\not| a^2-b^2D\}$. Unterscheide 2 Fälle:
(i) $\left(\frac{D}{p}\right) = -1$. Dann gilt $p\not| a^2-b^2D$, für alle $(a,b) \in (\mathbb{Z}/N\mathbb{Z})^2 \setminus p(\mathbb{Z}/N\mathbb{Z})^2$. Also ist $|Q^*| = p^{2\alpha} - p^{2\alpha-2} = p^{\alpha-1}(p-1)p^{\alpha-1}(p+1)$.
(ii) $\left(\frac{D}{p}\right) = 1$. Dann ist $|\{(a,b) \in \mathbb{Z}/p\mathbb{Z} \mid a^2 = b^2D\}| = 1+2(p-1)$. Also ist $|Q^*| = p^{2\alpha} - p^{2\alpha-2}(2p-1) = p^{\alpha-1}(p-1)p^{\alpha-1}(p-1)$.

6.23 Folgerung: Sei N ungerade. $\exp((\mathbb{Z}/N\mathbb{Z})[\sqrt{D}]^*/(\mathbb{Z}/N\mathbb{Z})^*) = N+1 \Leftrightarrow N$ ist Primzahl und $\left(\frac{D}{N}\right) = -1$.
Also kann man eine Faktorisierung von $N+1$ ausnutzen, um zu beweisen, dass N eine Primzahl ist, wenn man in der Gruppe $(\mathbb{Z}/N\mathbb{Z})[\sqrt{D}]^*/(\mathbb{Z}/N\mathbb{Z})^*)$ rechnen kann. Dies geschieht mit Lucas-Folgen:

6.24 Definition: Sei $\alpha \in (\mathbb{Z}/N\mathbb{Z})[\sqrt{D}]^*$, $\beta := \sigma(\alpha) \neq \alpha$. Dann heißt $((U_n)_{n\in\mathbb{N}}, (V_n)_{n\in\mathbb{N}})$ mit

$$U_n := \frac{\alpha^n - \beta^n}{\alpha - \beta}, \quad V_n := \alpha^n + \beta^n$$

die *Lucas Folge zu* α.

6.25 Bemerkung:
(i) $U_n = 0 \Leftrightarrow \alpha^n \in (\mathbb{Z}/N\mathbb{Z})^*$.
(ii) $U_{n+m} = U_mV_n - (\alpha\beta)^nU_{m-n}$. Insbesondere ist $U_{2n} = U_nV_n$.
(iii) $V_{2n} = V_n^2 - 2(\alpha\beta)^n$.

6.26 Satz: Sei N ungerade, $N+1 = \prod_{j=1}^n q_j^{\beta_j}\cdot R$ mit paarweise verschiedenen Primzahlen q_j und $q_j\not| R$ $(j=1,\ldots,n)$, $F := \prod_{j=1}^n q_j^{\beta_j} > \sqrt{N}+1$. Gibt es $D \in \mathbb{N}$ mit $\mathrm{ggT}(D,N) = 1$ und für alle $j = 1,\ldots,n$ Zahlen $\alpha_j \in \mathbb{Z}/N\mathbb{Z}[\sqrt{D}]^*$ so dass für die Lucas Folge $U^{(j)}$ zu α_j gilt:

$$\mathrm{ggT}(U^{(j)}_{(N+1)/q_j}, N) = 1, \quad U^{(j)}_{N+1} \equiv 0 \pmod N$$

so ist N eine Primzahl.
Beweis: Wie Satz (6.10).

6.27 Satz: Sei p ungerade und $(v_n)_{n\in\mathbb{N}_0}$ die durch $v_0 := 4$, $v_s := v_{s-1}^2 - 2$ $(s \leq 1)$ definierte Folge. Dann ist $M_p = 2^p - 1$ eine Primzahl, genau dann wenn $v_{p-2} \equiv 0 \pmod{M_p}$.
Beweis: Wähle $D := 3$, $\alpha := (2+\sqrt{3}) = \frac{1}{2}(1+\sqrt{3})^2 \in \mathbb{Z}/M_p\mathbb{Z}[\sqrt{D}]$. Dann ist $\alpha\sigma(\alpha) = 1$. Für die Lucas Folge (U_n, V_n) zu α gilt dann $V_1 = \alpha + \sigma(\alpha) = 4$. Also ist $V_{2^s} = v_s$ nach (6.25) (iii).
<u>$\Rightarrow$:</u> Sei also M_p eine Primzahl. Da $M_p \equiv 1 \pmod 3$ und $M_p \equiv -1 \pmod 8$ ist, gilt

$$\left(\frac{3}{M_p}\right) = -\left(\frac{M_p}{3}\right) = -\left(\frac{1}{3}\right) = -1$$

und $\left(\frac{2}{M_p}\right) = 1$. Also ergibt sich $v_{p-2} = V_{(M_p+1)/4} = \alpha^{(M_p+1)/4} + \alpha^{-(M_p+1)/4} = \alpha^{-(M_p+1)/4}(1+\alpha^{(M_p+1)/2}) = \alpha^{-(M_p+1)/4}(1+(1+\sqrt{3})^{M_p+1}/(2\cdot 2^{(M_p-1)/2}))$. Da $(1+\sqrt{3})^{M_p+1} = (1+\sqrt{3})^{M_p}(1+\sqrt{3}) \equiv (1+\sqrt{3}^{M_p})(1+\sqrt{3}) \equiv (1+3^{(M_p-1)/2}\sqrt{3})(1+\sqrt{3}) \equiv 1-3 \equiv -2 \pmod{M_p}$, folgt $v_{p-2} = 0$ in $\mathbb{Z}/M_p\mathbb{Z}[\sqrt{D}]$.
$\underline{\Leftarrow:}$ Dann gilt also $V_{(M_p+1)/4} = 0$. Daher ist $U_{(M_p+1)/4} \neq 0$ und $U_{(M_p+1)/2} = 0$. Also ist der Exponent e von $(\mathbb{Z}/M_p\mathbb{Z}[\sqrt{D}])^*/(\mathbb{Z}/M_p\mathbb{Z})^*$ durch $(M_p+1)/2 = 2^{p-1}$ teilbar. Sei $M_p = \prod_{j=1}^s p_j^{\alpha_j}$ eine Primfaktorisierung. Dann gilt nach Satz (6.22), dass e ein Teiler von $\prod_{j=1}^s p_j^{\alpha_j-1} kgV_j(p_j - \left(\frac{D}{p_j}\right))$ ist, d.h. M_p hat einen Primteiler p_j mit $p_j + 1 \geq 2^{p-1} > M_p/3$. Da M_p ungerade ist, folgt daraus $p_j = M_p$ und M_p ist eine Primzahl.

Analog kann man auch $2^n h - 1$ auf Primzahleigenschaft testen.
Neben einer Faktorisierung von $N-1$ oder $N+1$ (oder einer Kombination aus beiden) kann man auch Faktorisierungen von N^2+N+1,... benutzen. Problem: Zahlen werden größer und sind unwahrscheinlicher zu faktorisieren.
Für allgemeines N wird man zum Beweis der Primzahleigenschaft einen der folgenden beiden Tests benutzen:

6.28 Definition: Sei $N \in \mathbb{N}$, $\mathrm{ggT}(6,N) = 1$, $a, b \in \mathbb{Z}$, $\mathrm{ggT}(N, 4a^3 + 27b^2) = 1$. Dann heißt

$$E := E_{a,b}(\mathbb{Z}/N\mathbb{Z}) := \{(x,y) \in (\mathbb{Z}/N\mathbb{Z})^2 \mid y^2 = x^3 + ax + b\} \cup \{O\}$$

elliptische Kurve modulo N.
Idee: Ist N eine Primzahl, so ist E eine Gruppe (Gruppenoperationen wie in (5.14)). Also rechnen wir in E wie in einer Gruppe. Ist eine Gruppenoperation nicht ausführbar, so haben wir einen nichttrivalen Teiler von N gefunden. Dann soll jeder Algorithmus hier abbrechen und N ist zusammengesetzt ausgeben.

6.29 Satz (Goldwasser-Kilian Test)
Sei E wie in (6.28). Gibt es $P = (x_0, y_0) \in E\setminus\{O\}$ und eine Primzahl $q > (N^{1/4}+1)^2$ mit $[q]\cdot P = O$, dann ist N eine Primzahl.
Beweis: Angenommen, N ist keine Primzahl. Dann gibt es einen Primteiler p von N mit $p \leq \sqrt{N}$. Betrachten $E_p := E_{a,b}(\mathbb{Z}/p\mathbb{Z})$. Dann ist E_p eine elliptische Kurve und $\overline{P} := (x_0 \pmod p, y_0 \pmod p) \in E_p \setminus \{O\}$. Da $[q]\cdot\overline{P} = O$ für eine Primzahl q, ist $\mathrm{ord}(\overline{P}) = q$. Also ist $|E_p| \geq q > (N^{1/4}+1)^2 \geq (\sqrt{p}+1)^2 = p + 2\sqrt{p} + 1$, was in Widerspruch zum Satz von Hasse steht.

6.30 Bemerkung: (a) Vorteil: man kann E variieren.
(b) Schwierigkeit: Man muss schnell $|E|$ bestimmen und einen großen Primteiler q von $|E|$ finden.
(c) Die Primzahleigenschaft von q testet man wieder mit (6.29). Man erhält so ein Primzahlzertifikat: $(E_{a,b}, P, q, N), (E_{a_1,b_1}, P_1, q_1, q), \ldots, (E_{a_n,b_n}, P_n, q_n, q_{n-1})$ mit dem man die Primzahleigenschaft von N leicht überprüfen kann. Damit q_n kleine Primzahl ist, ist $n \sim \log(N)$.
(d) Zur Bestimmung von $|E|$ gibt es 3 Methoden:
Ein Analogon zu Shanks baby-step-giant-step-Methode: $O(N^{1/4})$, also zu schlecht.

Einen bemerkenswert schnellen Algorithmus von R. Schoof mit $O(\ln(N)^8)$ auf den ich hier nicht eingehen will.
Geeignete Wahl von E, so dass man $|E|$ aus der Theorie kennt:

6.31 Satz (Ohne Beweis): Sei E elliptische Kurve (über $\mathbb{C}$) mit $a, b \in \mathbb{Z}, \Delta := 4a^3+27b^2 \neq 0$ mit *komplexer Multiplikation*, d.h. es gibt Gruppenhomomorphismus $\varphi : E \to E$ mit $\varphi^2 = [-D]$, $D \in \mathbb{N}$ quadratfrei. Sei $p > 3$ eine Primzahl mit $p \not| \Delta D$ und E_p die Reduktion von E modulo p. Dann gilt:
(i) Ist $\left(\frac{-D}{p}\right) = -1$, dann ist $|E_p| = p + 1$.
(ii) Gibt es $\alpha, \beta \in \frac{1}{2}\mathbb{Z}$ mit $\alpha \equiv \beta \pmod{\mathbb{Z}}$ und $\star \quad \alpha^2 + D\beta^2 = p$, so ist $|E_p| = p + 1 - 2\alpha$ für eine geeignete Lösung (α, β) von $\star$.
Beispiel: $E : Y^2 = X^3 - X$. $\varphi : E \to E : (x, y) \mapsto (-x, iy)$ erfüllt $\varphi^2 = [-1]$. Also gilt für $p \equiv 3 \pmod 4$ (also $\left(\frac{-1}{p}\right) = -1$) dass $|E_p| = p + 1$ ist. Für $p = 5 = (-1)^2 + 2^2$ ist $|E_p| = 5 + 1 - (-2) = 8$. (vgl. Übung).

6.32 Bemerkung: Man kann allgemeine Formeln für solche Kurven E (für jede Lösung von $\star$) angeben. Das macht den Goldwasser-Kilian-Test zu einem probabilistischen Primzahltest dessen erwartete Laufzeit polynomial in $\ln(N)$ ist.
Wir kommen abschließend zum APRCL-Test (Adleman, Pomerance, Rumely, Cohen, Lenstra).

6.33 Definition: Seien p, q Primzahlen mit $p \mid q - 1$, ζ_p, ζ_q primitive p-te bzw. q-te Einheitswurzeln in $\mathbb{C}$ und $C := \langle \zeta_p \rangle \leq \mathbb{C}^*$. Sei $\chi : (\mathbb{Z}/q\mathbb{Z})^* \to C$ ein surjektiver (also nichttrivialer) Gruppenhomomorphismus. Dann heißt

$$\tau(\chi) := \sum_{a \in (\mathbb{Z}/q\mathbb{Z})^*} \chi(a)\zeta_q^a$$

die *Gauß-Summe* zu χ.
(Beispiel: $p = 2$, q ungerade Primzahl, $\chi(a) := \left(\frac{a}{q}\right) \in \{\pm 1\}$.)

6.34 Lemma: Sei χ wie in (6.33). Dann gilt $\tau(\chi)\tau(\chi^{-1}) = q\chi(-1)$.
Beweis: Wie der von der 1. Behauptung im quadratischen Reziprozitätsgesetz: Alle Summen laufen über $(\mathbb{Z}/q\mathbb{Z})^*$. $\tau(\chi)\tau(\chi^{-1}) = \sum_{a,b} \chi(a)\chi(b)^{-1}\zeta_q^{a+b} = \sum_{c,b} \chi(cb)\chi(b^{-1})\zeta_q^{cb+b} = \sum_c \chi(c)(\sum_b \zeta_q^{(c+1)b})$.
Ist $c \equiv -1 \pmod q$, so ist $\sum_b \zeta_q^{(c+1)b} = q - 1$
sonst ist $\sum_b \zeta_q^{(c+1)b} = \sum_a \zeta_q^a = -1$.
Also ist $\tau(\chi)\tau(\chi^{-1}) = -\sum_c \chi(c) + q\chi(-1) = q\chi(-1)$.

6.35 Lemma: Seien p, q, χ wie in (6.33). Ist N eine Primzahl, $N \not| qp$, so ist

$$\tau(\chi)^{N^{p-1}-1} \equiv \chi(N) \pmod{N\mathbb{Z}[\zeta_p, \zeta_q]}.$$

Beweis: (vgl. 2. Behauptung im Beweis des quadratischen Reziprozitätsgesetzes)
$\tau(\chi)^{N^{p-1}} \equiv \sum_a (\chi(a)\zeta_q^a)^{N^{p-1}} \pmod N$, da wir üeber einem Körper der Charakteristik N arbeiten. Weiter ist $N^{p-1} \equiv 1 \pmod p$. Da $\chi(a)$ eine p-te Einheitswurzel

ist, gilt also $\chi(a)^{N^{p-1}} = \chi(a)$ für alle $a \in (\mathbb{Z}/q\mathbb{Z})^*$. Sei $b \in (\mathbb{Z}/q\mathbb{Z})^*$ mit $bN^{p-1} \equiv 1 \pmod q$. Dann ist

$$\tau(\chi)^{N^{p-1}} = \sum_a \chi(a)\zeta_q^{aN^{p-1}} = \sum_a \chi(ab)\zeta_q^a = \chi(b)\tau(\chi).$$

Nun ist $\chi(b) = \chi(N)^{1-p} = \chi(N)$, da $\chi(N)^p = 1$, also folgt

$$\tau(\chi)^{N^{p-1}} \equiv \chi(N)\tau(\chi) \pmod N.$$

Jetzt müssen wir nur noch durch $\tau(\chi)$ teilen, um die Behauptung zu erhalten. Das machen wir mit Lemma 6.34, durch Multiplikation mit $\tau(\chi^{-1})\chi(-1)q^{-1}$, das in $\mathbb{Z}/N\mathbb{Z}[\zeta_p, \zeta_q]$ liegt.

6.36 Lemma: Sind $r \neq p$ Primzahlen, so sind die p-ten Einheitswurzeln modulo r verschieden: $\zeta_p^i \equiv \zeta_p^j \pmod r \Rightarrow \zeta_p^i = \zeta_p^j$.
Beweis: $X^p - 1 \in \mathbb{F}_r[X]$ hat p verschiedene Nullstellen in seinem Zerfällungskörper.

6.37 Satz: (Rohform des APRCL-Tests)
Sei P eine endliche Menge von Primzahlen und $Q \subseteq \{q \in \mathbb{N} \mid q$ ist Primzahl und $q-1$ ist Produkt von verschiedenen Primzahlen in $P\}$. Sei $z := \prod_{p\in P} p$, $w := \prod_{q\in Q} q$ und $N \in \mathbb{N}$ ungerade $\mathrm{ggT}(N, wz) = 1$. Angenommen, die folgenden beiden Bedingungen sind erfüllt:
(1) Für alle $q \in Q$, $p \in P$ mit $p \mid q - 1$ gibt es einen Gruppenepimorphismus $\chi := \chi_{p,q} : (\mathbb{Z}/q\mathbb{Z})^* \to \langle\zeta_p\rangle$ mit $\tau(\chi)^{N^{p-1}-1} \equiv \chi(N) \pmod N$.
(2) Für alle $p \in P$ und alle Primteiler r von N gilt $p^e \mid N^{p-1} - 1 \Rightarrow p^e \mid r^{p-1} - 1$.
Dann erfüllt jeder Primfaktor r von N, dass $r \equiv N^i \pmod w$ für ein $i \in \{0, \dots, z-1\}$.
Beweis: Sei r ein Primfaktor von N. Wegen Bedingung (2), gibt es für alle $p \in P$ Zahlen $a_p(r)$, $b_p(r)$, $l_p(r) \in \mathbb{Z}$, mit $p \nmid b_p(r)$, so dass

$$\frac{r^{p-1}-1}{N^{p-1}-1} = \frac{b_p(r)}{a_p(r)} \equiv l_p(r) \pmod p.$$

Sei $l(r) \in \{0, \dots, z-1\}$ mit $l(r) \equiv l_p(r) \pmod p$ für alle $p \in P$.
Zeigen nun, dass $r \equiv N^{l(r)} \pmod w$ gilt.
Dazu genügt es zu zeigen, dass $\chi_{p,q}(rN^{z-l(r)}) = 1$ ist für alle $q \in Q$, $p \in P$ mit $p \mid q-1$. Sei also $q \in Q$, $p \in P$ mit $p \mid q - 1$ und $\chi := \chi_{p,q}$. Dann gilt $\chi(r)^{a_p(r)} \equiv \tau(\chi)^{(r^{p-1}-1)a_p(r)} \equiv \tau(\chi)^{(N^{p-1}-1)b_p(r)} \equiv \chi(N)^{b_p(r)} \pmod r$ mit Lemma 6.35. Also folgt mit Lemma 6.36, dass $\chi(r)^{a_p(r)} = \chi(N)^{b_p(r)}$, also $\chi(r) = \chi(N)^{l(r)}$.

6.38 Bemerkung:
(a) Zum Primzahlbeweis wählt man Q so, dass $w > \sqrt{N}$. Dann muss man nur noch überprüfen, ob $x_i := N^i \pmod w \in \{0, \dots, w-1\}$, $i = 0, \dots, z-1$ kein echter Teiler von N ist.
(b) In der Regel ist P klein (z.B. $P = \{2, 3, 5, 7, 11, 13, 17\}$), jedoch enthält Q dann große Primzahlen (z.B. $2 \cdot 3 \cdot 7 \cdot 11 \cdot 13 \cdot 17 + 1 = 102103$). Das macht das Berechnen von $\tau(\chi) \in \mathbb{Z}[\zeta_p, \zeta_q]$ sehr teuer. Abhilfe: Idee von Cohen/Lenstra: Rechne nur in

$\mathbb{Z}[\zeta_p]$.
(c) Bedingung (2) in (6.37) scheint sehr schwer nachzuprüfen. Sie ist aber trivialerweise erfüllt, wenn $p^2 \not| N^{p-1} - 1$. Auch sonst gibt es hier leichter nachprüfbare Ersatzbedingungen.
(d) Für eine Beschreibung eines praktikablen Algorithmus sei auf Cohen, S. 438-458 verwiesen.

Abschnitt 7: Faktorisierungsalgorithmen.

7.1 Algorithmus: (Sieb des Eratosthenes)
EINGABE: $S \in \mathbb{N}$.
AUSGABE: $\mathbb{P}_{\leq S} := \{p \mid p \leq S,\ p \text{ ist Primzahl }\}$.
ALGORITHMUS:
Sei $L := (2, 3, 4, \ldots, S)$, $s := \lfloor\sqrt{S}\rfloor$ und $P := \emptyset$.
while $L[1] \leq s$ do
    $P := P \cup \{L[1]\}$;
    $L := L\setminus$ Vielfache von $L[1]$;
end while;
$P := P \cup L$;
RETURN(P);

7.2 Algorithmus: (Probedivision, Trial Division)
EINGABE: $N, S \in \mathbb{N}$.
AUSGABE: $\{p^\alpha \mid p \in \mathbb{P}_{\leq S}, p^\alpha \mid\mid N\}$.
ALGORITHMUS:
$T := \emptyset$.
Bestimme $\mathbb{P}_{\leq S}$ mit Algorithmus (7.1).
Für alle $p \in \mathbb{P}_{\leq S}$ bestimme maximales α mit $p^\alpha \mid N$.
$P := P \cup \{p^\alpha\}$.
RETURN(P).

7.3 Bemerkung:
Probedivision wird bei vielen Faktorisierungsmethoden benutzt, um zunächst kleine Primteiler von N abzuspalten, meist noch vor einem Zusammengesetztheitstest.
Nach dem Primzahlsatz (1896 Hadamard und gleichzeitig de la Vallée Poussin) ist $|\mathbb{P}_{\leq S}| \sim S/\ln(S)$. Will man also N mit (7.2) vollständig faktorisieren, so müßte man zusätzlich zur Bestimmung von $\mathbb{P}_{\leq S}$ (für $S = \lfloor\sqrt{N}\rfloor$) im schlimmsten Fall ca. $2\sqrt{N}/\ln(N)$ mal Dividieren.
Die meisten Divisionen sind nutzlos, deshalb besser: $P_i := \prod\{p \in \mathbb{P}_{\leq S} \mid 100(i-1) < p < 100i\}$ und $\mathrm{ggT}(N, P_i)$ berechnen.
Sieb des Eratosthenes ist auch teuer, anstelle von $\mathbb{P}_{\leq S}$ kann man auch $\{2,3\} \cup \{6k+1 \mid k = 1, \ldots, \lfloor(S-1)/6\rfloor\}$ benutzen.
Ein besserer Algorithmus zum Abspalten kleiner Primteiler ist der folgende Pollard-ρ-Algorithmus, ein probabilistischer Algorithmus mit erwarteter Laufzeit $O(N^{1/4})$.

7.4 Lemma: Sei $f : \mathbb{Z}/N\mathbb{Z} \to \mathbb{Z}/N\mathbb{Z}$, $x_0 \in \mathbb{Z}/N\mathbb{Z}$, $x_i := f(x_{i-1})$ für $i = 1, 2, \ldots$. Dann gibt es $m > n$ mit $x_{i+n} = x_{i+m}$ für alle $i \in \mathbb{N}$, wobei man erwarten kann, dass man m, n so finden kann, dass $|m - n| < 2\sqrt{N}$ gilt.
Beweis: (vgl. Cohen S. 423).

7.5 Algorithmus: (Pollard ρ).
EINGABE: $N \in \mathbb{N}$, $S \in \mathbb{N}$
AUSGABE: Mit großer Wahrscheinlichkeit alle (Prim)Teiler p von N mit $\sqrt{p} < S$.
ALGORITHMUS: Sei $f \in \mathbb{Z}[X]$, $F := \emptyset$.
(1) Wähle $x = y \in \{0, \ldots, N-1\}$.
for i from 0 to S do
$x := f(x) \pmod N$;
$y := f(y) \pmod N$;
$y := f(y) \pmod N$;
$d := \mathrm{ggT}(x - y, N)$;
if $d = N$ then goto (1); end if;
if $d > 1$ then $N := N/d$; $F := F \cup \{d\}$; end if;
end for;
RETURN(F);
IDEE: Ist $p \mid N$, $p << N$, so wird die Folge ($x_n \pmod p$)) sehr viel eher periodisch mit einer kürzeren Periode m als modulo N, d.h. es gilt $p \mid (x_i - x_{i+m})$ für alle geügend großen i. Insbesondere teilt p den $\mathrm{ggT}(x_{kn} - x_{2kn}, N)$ für genügend großes k.

7.6 Bemerkung:
(a) Um nicht zu oft $\mathrm{ggT}(x_n - x_{2n}, N)$ zu berechnen, kann man z.B. l Schritte zusammenfassen und zunächst $P := \prod_{j=kl}^{(k+1)l-1}(x_j - x_{2j})$ und dann $\mathrm{ggT}(P, N)$ berechnen.
(b) Offensichtlich schlechte Wahlen von f: $f(X) = X + 1$, $f(X) = X^2$, u.ä.
(c) Der Algorithmus benötigt wenig Speicher, sehr gut um kleine Primteiler zu finden, schnell in MAPLE programmiert (Ausprobieren!).
(d) Die Werte x_{2i} mit $2i < S$ werden doppelt berechnet. Folgende Modifikation bringt 25 Prozent Zeitersparnis:
Man testet nur die $x_j - x_{2^k}$ mit $3 \cdot 2^{k-1} < j \le 2^{k+1}$, $k = 0, 1, 2, \ldots$. Man muss also zunächst $y := x_{2^k}$ berechnen, dann die Werte $x_{2^k+1}, \ldots, x_{2^k+2^{k-1}}$ wegschmeißen und für die $x = x_j$ mit $3 \cdot 2^{k-1} < j \le 2^{k+1}$ den $\mathrm{ggT}(x - y, N)$ berechnen. Den Wert $x_{2^{k+1}}$ speichert man dann wieder als neues y.

Einschub: Faktorisieren von $F_n = 2^{(2^n)} + 1$.
1. spektakuläre Anwendung von Pollard-ρ: 1980: 16-stelligen Primteiler von F_8 in wenigen Stunden Rechenzeit.

7.7 Satz: (Legendre)
Sei $n \in \mathbb{N}$, $a, b \in \mathbb{Z}$ mit $\mathrm{ggT}(a, b) = 1$, $N := a^n + b^n$ und $p > 2$ ein Primteiler von N. Dann gilt entweder $p \mid a^d + b^d$ für einen echten Teiler d von n oder $p = 2kn + 1$ für ein $k \in \mathbb{N}$.
Beweis: $p \mid a^n + b^n \Rightarrow p \nmid ab$ und $c := \frac{a}{b}$ ist eine Einheit modulo p, die $c^n \equiv -1$

(mod p) erfüllt. Sei $d := \mathrm{ord}(c + (p\mathbb{Z}))$ die Ordnung von c in $(\mathbb{Z}/p\mathbb{Z})^*$. Dann ist d gerade und $d/2$ ein Teiler von n. Da $p \mid (a^{d/2} + b^{d/2})$ können wir annehmen, dass $d/2 = n$. Also ist $d = 2n$ ein Teiler von $p - 1$ und damit $p = 2kn + 1$ für ein k in $\mathbb{N}$.

7.8 Lemma $F_n := 2^{(2^n)} + 1$. Dann gilt $F_{n+1} = 2 + F_0 F_1 \cdots F_n$.
Beweis: Nachrechnen.

7.9 Folgerung: (a) Die Fermat Zahlen sind paarweise teilerfremd.
(b) Jeder Primteiler p von F_n erfüllt $p = 2^{n+1}k + 1$ für ein $k \in \mathbb{N}$. Ist $n > 1$, so gilt sogar $p = 2^{n+2}k + 1$ für ein $k \in \mathbb{N}$.
Beweis: Aus (7.7) folgt $2^{2^n} \equiv -1 \pmod{p}$, d.h. $\mathrm{ord}(2)$ teilt 2^{n+1} teilt $p - 1$. Ist $n > 1$, so ist p damit kongruent zu 1 modulo 8, also ist 2 ein Quadrat modulo p. D.h. die Ordnung von $2 \in (\mathbb{Z}/p\mathbb{Z})^*$ teilt sogar $(p - 1)/2$.

7.10 Bemerkung: Um Primfaktoren der Form $2kn + 1$ mit $k \in \mathbb{N}$ zu finden, wählt man im Pollard ρ Algorithmus $f = X^n + a$. Dann hat nämlich $x_i - x_j = x_{i-1}^n - x_{j-1}^n$ viele Primfaktoren der Form $2kn + 1$.

7.11 Definition: Sei $B \in \mathbb{N}$.
(i) $n \in \mathbb{N}$ heißt B-*glatt*, falls alle Primteiler von n kleiner oder gleich B sind.
(ii) $n \in \mathbb{N}$ heißt B-*potenzglatt*, falls alle Primzahlpotenzen, die n teilen, kleiner oder gleich B sind.

7.12 Lemma: Sei p eine Primzahl, so dass $p - 1$ eine B-potenzglatte Zahl ist. Sei $M := \mathrm{kgV}\{n \in \mathbb{N} \mid n \leq M\} = \prod\{p^s \mid p \text{ ist Primzahl und } s \text{ maximal mit } p^s \leq M\}$. Dann gilt $p \mid a^M - 1$ für alle $a \in \mathbb{N}$ mit $p \nmid a$.

7.13 Algorithmus (Pollard $(p - 1)$, Schritt 1)
EINGABE: $N \in \mathbb{N}$, Schranke B.
AUSGABE: Wahrscheinlich einen Primteiler $p \mid N$ für den $p - 1$ eine B-potenzglatte Zahl ist.
ALGORITHMUS: Bestimme $\mathbb{P}_{\leq B} = \{p_1, \ldots, p_s\}$ (mit (7.1)) und für jedes p_i das maximale $\alpha_i \in \mathbb{N}$ mit $p_i^{\alpha_i} \leq B$.
Wähle $c \in \{2, \ldots, N - 2\}$ zufällig.

```
z:=0;
for i from 1 to s do
        for j from 1 to α_i do
                c := c^(p_i)  (mod N);
                z:=z+1;
        end for;
        if (z ≥ 200) then
                z:=0;
                g:=ggT(c-1,N);
                if (g = N) then RETURN(FAIL); end if;
                if (g > 1) then RETURN(g); end if;
        end if;
end for;
```

```
g:=ggT(c-1,N);
if ((g=1) or (g=N)) then RETURN(FAIL); end if; RETURN(g);
```

7.14 Bemerkung: Es gibt verschiedene Gründe, weshalb (7.13) versagt:
(i) In $c-1$ gehen alle Primteiler von N auf. Dann lohnt es sich, die Suche zu verfeinern, also den ggT-Test nach jedem Potenzieren zu machen, oder auch das Potenzieren in einer anderen Reihenfolge durchzuführen.
Beispiel: $N = 23 \cdot 89$, $B = 11$. Dann ist $c-1$ erst nach Potenzieren mit 11 durch eine der beiden Primzahlen teilbar, dann aber auch gleich durch N. Abhilfe: Am Ende die 2-Potenzen behandeln.
(ii) Weitaus häufiger tritt der Fall auf, dass N keine Primteiler p hat, für die $p-1$ eine B-potenzglatte Zahl ist. Dann ist der Schritt 2 hilfreich:
Sei c das Endergebnis des 1. Schrittes, also eine $\prod_{i=1}^{s} p_i^{\alpha_i}$-te Potenz.
Für eine Schranke $B' > B$ seien $q_1 < q_2 < \ldots, < q_t$ die Primzahlen q mit $B < q \le B'$. Speichere $q_1, d_2 := q_2 - q_1, d_3 := q_3 - q_2, \ldots, d_t := q_t - q_{t-1}$ und berechne c^{q_1}, $c^{q_{i+1}} = c^{q_i} c^{d_{i+1}}$ für $i = 1, \ldots, t-1$. und teste, ob $\mathrm{ggT}(c^{q_i} - 1, N) > 1$ für ein $i = 1, \ldots, t$.

7.15 Bemerkung: Analog zur Glattheit von $p-1$ kann man auch die von $p+1$ benutzen, indem man in $(\mathbb{Z}/N\mathbb{Z}[\sqrt{D}])^*/(\mathbb{Z}/N\mathbb{Z}^*)$ rechnet. Allerdings ist es nicht so klar, wie man ein D findet, das kein Quadrat modulo dieser (unbekannten) Primteiler von N ist.
Mit elliptischen Kurven kann man solche p finden, für die in der Nähe von p eine glatte Zahl existiert. Das führt zu einem der wichtigsten modernen Faktorisierungsverfahren:

7.16 Beispiel: $N = 1098413 = 563 \cdot 1951$, $E : Y^2 = X^3 - X + 1$. Dann ist $|E(\mathbb{F}_{563})| = 560 = 2^4 \cdot 5 \cdot 7$ und $|E(\mathbb{F}_{1951})| = 1948 = 2^2 \cdot 487$.
Mit einem Analogon zum Pollard $(p-1)$-Algorithmus wird man $P_0 := (0,1) \in E(\mathbb{Z}/N\mathbb{Z})$ potenzieren bis man $[560]P_0$ z.B. als

$$[560]P_0 = [640]P_0 + [-80]P_0 = (289957, 901426) \oplus (590036, -204995)$$

ausrechnen muss. Die Formeln für die Addition benutzen die Berechnung von $\lambda = (y_2 - y_1)/(x_2 - x_1)$. Also muss man $x_2 - x_1 = 590036 - 289957 = 300079$ modulo N invertieren (mit dem erweiterten Euklidischen Algorithmus zur Berechnung des ggT). Dabei findet man $\mathrm{ggT}(N, 300079) = 563$ also einen nichttrivialen Teiler von N.

7.17 Algorithmus: (ECM, Elliptische Kurven Methode)
Eingabe: $N \in \mathbb{N}$, Schranken B, B' (für Glattheit), S (Anzahl der Kurven).
Ausgabe: Wahrscheinlich einen nichttrivialen Teiler von N.
Algorithmus: (vgl. Pollard-$(p-1)$)
Bestimme $\mathbb{P}_{\le B} = \{p_1, \ldots, p_s\}$ (mit (7.1)) und für jedes p_i das maximale $\alpha_i \in \mathbb{N}$ mit $p_i^{\alpha_i} \le B$.
Bestimme $\{B < q \le B' \mid q \text{ ist Primzahl } \} = \{q_1 < q_2 < \ldots < q_t\}$ und speichere $(q_1, d_2 := q_2 - q_1, d_3 = q_3 - q_2, \ldots, d_t = q_t - q_{t-1})$.

```
for a from 0 to S do
    if g := ggT(N, 4a^3 + 27) > 1 then RETURN(g) end if;
    E : Y^2 = X^3 + aX + 1; (elliptische Kurve modulo N)
    P := (0,1) ∈ E;
    for i from 1 to s do
        for j from 1 to α_i do
            P := [p_i] · P;
        end for;
    end for;
    Q := [q_1] · P;
    for i from 2 to t do
        Q := Q + [d_i] · P;
    end for;
end for;
```

Dabei ist die Abbruchbedingung in dem Befehl $P := [p_i] \cdot P$ versteckt. Kann man eine Rechnung nicht durchführen, so hat man einen nichttrivialen Teiler von N gefunden.

7.18 Bemerkung: (a) Die ECM ist der einzige moderne Faktorisierungsalgorithmus, dessen Laufzeit wesentlich von der Größe des kleinsten Primteilers von N abhängt und nicht von der von N. Mit ihr kann man ca. 50-stellige Primfaktoren von mehr als 1000-stelligen Zahlen bestimmen.

(b) Wie auch die anderen modernen Faktorizierungsmethoden ist die ECM sehr leicht parallelisierbar: Man kann verschiedene Rechner, verschiedene elliptische Kurven ausprobieren lassen.

(c) Die aktuellen Rekorde der ECM sind z.B.:
54-stelligen Primteiler von $(6^{43} - 1)^{42} + 1 \sim 0.2 \cdot 10^{1406}$ sowie
54-stelligen Primteiler von $(3^{64} - 1)^{63} + 1 \sim 0.6 \cdot 10^{1924}$.

(d) Die ECM ist ein *subexponentieller Algorithmus*, d.h. die erwartete Laufzeit wächst langsamer als jede positive Potenz von N: (erw. Laufzeit)$/N^c \to 0$ für jedes $c > 0$.
Genauer sei $L_M(v, c) := \exp(c \cdot \ln(M)^v \ln\ln(M)^{1-v})$.
Dann ist die erwartete Laufzeit von ECM in etwa $L_p(1/2, 2+o(1))$ wo p den kleinsten Primteiler von N bezeichnet.

Jetzt kommen wir zur zweiten Sorte von Faktorisierungsalgorithmen, deren Laufzeit im wesentlichen von der Größe von N abhängt und nicht von der des zweitgrößten Primteilers von N. Die Ideen der modernen Methoden findet man schon in den klassischen Algorithmen, die ich deshalb vorstellen möchte, obwohl sie zum Faktorisieren keine große Rolle mehr spielen. Wir nehmen an, dass N keine kleinen Primteiler mehr hat (also insbesondere ungerade ist) und auch kein Quadrat oder eine andere Potenz einer kleineren Zahl ist.

7.19 Algorithmus (Fermat)
IDEE: Sei $N = a \cdot b$ ungerade. Dann ist $N = (\frac{a+b}{2})^2 - (\frac{a-b}{2})^2 = x^2 - y^2 = (x-y)(x+y)$ die Differenz von 2 Quadraten. Dabei gilt $x \geq \sqrt{N}$.

EINGABE: $N \in \mathbb{N}$ ungerade, zusammengesetzt.
AUSGABE: $a, b \in \mathbb{N}_{>1}$ mit $N = a \cdot b$.
ALGORITHMUS: $x := \lceil\sqrt{N}\rceil$; $z := x^2 - N$;

```
while x < N do
        if (z = y^2 für ein y ∈ N) then
              RETURN ((x-y),(x+y) ); end if;
        z:=z+2x+1; x:=x+1;
end while;
```

7.20 Bemerkung: (7.19) findet die Faktoren $a \sim b \sim \sqrt{N}$ recht schnell, die für Trial Division den worst case darstellen.
Im Fall, dass z kein Quadrat ist, wird Information weggeschmissen:
$z \equiv x^2 \pmod N \Rightarrow z \equiv \square \pmod p$ für jeden Primteiler p von N. Findet man also z.B. $z \equiv 2 \equiv x^2 \pmod N$, so weiss man, dass die Primteiler von N kongruent ± 1 modulo 8 sind.
Gauss und Legendre erzeugten systematisch kleine quadratische Reste modulo N z.B. mit der Kettenbruchentwicklung von $\sqrt{kN}$ $(k \in \mathbb{N})$.

7.21 Definition: Eine *Kettenbruchentwicklung* ist ein Ausdruck der Form

$$b_0 + \cfrac{a_1}{b_1 + \cfrac{a_2}{b_2 + \cfrac{a_3}{b_3 + \cfrac{a_4}{\dots}}}} = b_0 + \frac{a_1 \mid}{\mid b_1} + \frac{a_2 \mid}{\mid b_2} + \frac{a_3 \mid}{\mid b_3} + \dots.$$

Sie heißt *regulär*, falls $a_i = 1$ und $b_i \in \mathbb{Z}_{\geq 0}$ für alle $i > 0$ gilt.
Die reguläre Kettenbruchentwicklung einer Zahl $x = x_0 \in \mathbb{R}$ berechnet man, indem man für $i = 0, 1, \dots$ $b_i := \lfloor x_i \rfloor$, $x_{i+1} := \frac{1}{x_i - b_i}$ setzt. Wir nennen auch die Folge (x_i, b_i) die reguläre Kettenbruchentwicklung von x.
Vergleichen Sie die reguläre Kettenbruchentwicklung einer rationalen Zahl mit dem Euklidischen Algorithmus.

7.22 Satz: Seien $A_n, B_n \in \mathbb{N}$ mit $\mathrm{ggT}(A_n, B_n) = 1$, so dass $\frac{A_n}{B_n} = b_0 + \frac{1|}{|b_1} + \dots + \frac{1|}{|b_n}$ die n-te Partialsumme einer regulären Kettenbruchentwicklung ist. Dann gilt:
(a) $A_0 = b_0$, $B_0 = 1$, $A_1 = b_1 A_0 + 1$, $B_1 = b_1 B_0$, und für $n \geq 2$:
$A_n = b_n A_{n-1} + A_{n-2}$, $B_n = b_n B_{n-1} + B_{n-2}$.
(b) $A_n B_{n-1} - A_{n-1} B_n = (-1)^{n-1}$.
Beweis: (a) Induktion über n. $n = 0, 1$ Klar.
Induktionsschluss:

$$\frac{A_{n+1}}{B_{n+1}} = \frac{b_{n+1}A_n + A_{n-1}}{b_{n+1}B_n + B_{n-1}} = \frac{(b_n + b_{n+1}^{-1})A_{n-1} + A_{n-2}}{(b_n + b_{n+1}^{-1})B_{n-1} + B_{n-2}} = b_0 + \frac{1 \mid}{\mid b_1} + \dots + \frac{1 \mid}{\mid b_n + \frac{1}{b_{n+1}}}$$

wobei wir im letzten Schrit die Induktionsvoraussetzung benutzen.
(b) Folgt aus (a) mit Induktion über n.

7.23 Satz: Sei $x = b_0 + \frac{1|}{|b_1} + \dots \in \mathbb{R}_{>0}$ eine reguläre Kettenbruchentwicklung. Dann ist $0 \leq (-1)^n (x - \frac{A_n}{B_n}) < \frac{1}{B_n^2}$.

Beweis:

$$x = \lim_{n\to\infty} \frac{A_n}{B_n} = \frac{A_n}{B_n} + (\frac{A_{n+1}}{B_{n+1}} - \frac{A_n}{B_n}) + (\frac{A_{n+2}}{B_{n+2}} - \frac{A_{n+1}}{B_{n+1}}) + \dots$$
$$= \frac{A_n}{B_n} + ((-1)^n \frac{1}{B_n B_{n+1}}) - (-1)^n (\frac{1}{B_{n+1}B_{n+2}} - \frac{1}{B_{n+2}B_{n+3}} + \dots).$$

Also ist

$$0 \le (-1)^n (x - \frac{A_n}{B_n}) = \frac{1}{B_n B_{n+1}} - (\frac{1}{B_{n+1}B_{n+2}} - \frac{1}{B_{n+2}B_{n+3}} + \dots) <= \frac{1}{B_n^2}$$

da B_n monoton steigend ist.

7.24 Satz: Sei $N \in \mathbb{N}$ kein Quadrat, (x_i, b_i) die reguläre Kettenbruchentwicklung von $\sqrt{N} =: x_0$. Definieren $Q_0 := 1$, $P_1 := b_0$, und für $n \ge 1$:

$$Q_n := \frac{N - P_n^2}{Q_{n-1}}, \quad P_{n+1} := b_n Q_n - P_n$$

Dann gilt für alle $n \ge 1$:
(a) $x_n = \frac{Q_{n-1}}{\sqrt{N} - P_n} = b_n + \frac{\sqrt{N} - P_{n+1}}{Q_n}$.
(b) $P_n, Q_n \in \mathbb{Z}_{\ge 0}$, $P_n \le \sqrt{N}$, $Q_n \le 2\sqrt{N}$.
(c) $A_{n-1}^2 - N B_{n-1}^2 = (-1)^n Q_n$.
Die $(-1)^n Q_n$ sind also kleine quadratische Reste modulo N.
Beweis als Übung.

7.25 Beispiel (Legendre):
$N = 333667$, $x_0 = \sqrt{N}$, $b_0 = 577$,
$x_1 = 1 + \frac{\sqrt{N}-161}{738}$, $x_2 = 1 + \frac{\sqrt{N}-256}{417}$, $x_3 = 1 + \frac{\sqrt{N}-387}{643}$, $x_4 = 3 + \frac{\sqrt{N}-471}{286}, \dots$.
Also sind $-738 = -2 \cdot 3^2 \cdot 41$, $417 = 3 \cdot 139$, $-643 = -643$, $286 = 2 \cdot 11 \cdot 13, \dots 69 = 69$, $-288 = -2^5 \cdot 3^4, \dots, 10 = 2 \cdot 5$ Quadrate modulo N.
Z.B. folgt aus der vorletzten Gleichheit, dass für jeden Primteiler p von N gilt: $-2 \equiv \square \pmod p$ also $p = 8m+1$ oder $p = 8m+3$ mit $m \in \mathbf{Z}$.
Insgesamt findet Legendre genügend Kongruenzen um die Menge der Primteiler p von N mit $p < \sqrt{N}$ als Teilmenge von $\{83, 107, 163, 401, 409, 467, 569\}$ zu erkennen und durch Dividieren festzustellen, dass N eine Primzahl ist.
Aus der Idee, Kettenbrüche zur Erzeugung kleiner quadratischer Reste modulo N zu benutzen, entstand die erste moderne Methode zum Faktorisieren grosser Zahlen, CFRAC (1975). Sie beruht, wie auch die folgenden Methoden auf der folgenden Beobachtung:

7.26 Beobachtung: Angenommen, es gibt $x, y, k \in \mathbb{Z}$ mit

$$x^2 - y^2 = kN \text{ also } x^2 \equiv y^2 \pmod N.$$

Dann gilt $kN = (x-y)(x+y)$ und $\mathrm{ggT}(x-y, N)$ ist mit grosser Wahrscheinlichkeit ein nichttrivialer Teiler von N.

7.27 Algorithmus (CFRAC) (Morrison, Brillhardt, 1975)
EINGABE: $N \in \mathbb{N}$ ungerade, keine Potenz
AUSGABE: (nichttrivialer) Teiler von N.
ALGORITHMUS:
(0) Wähle Schranke S und bestimme $\mathbb{P}_{\leq S} = \{p_1, \ldots, p_s\}$ mit dem Sieb des Eratosthenes.
(1) Wähle $k \in \mathbb{N}$ geeignet.
(2) Bestimme die Kettenbruchentwicklung von $\sqrt{kN}$ und damit $R_n := (-1)^n Q_n \equiv A_{n-1}^2 \pmod N$. Speichere nur die S-(semi)-glatten R_n, also solche R_n, mit

$$R_n = (-1)^{e_{n,0}} p_1^{e_{n,1}} \ldots p_s^{e_{n,s}} \cdot P_n$$

mit $p_i \not| P_n$ und $S < P_n < S^2$. als Zeile einer Exponentenmatrix:

$$E, P, X := \begin{pmatrix} \vdots & \vdots & \vdots & \vdots \\ e_{n,0} & e_{n,1} & \ldots & e_{n,s} \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}, \begin{pmatrix} \vdots \\ P_n \\ \vdots \end{pmatrix}, \begin{pmatrix} \vdots \\ A_{n-1} \\ \vdots \end{pmatrix}$$

(3) Verkleinere E:
Ist $P_n \neq 1$, so addiere erste Zeile, in der P_n auftaucht zu allen anderen Zeilen mit gleichem P_n (und multipliziere die entsprechenden Einträge in P, X) und streiche dann diese erste Zeile (in E, P und X).
(4) Bestimme Vektoren aus dem Kern von $E \pmod 2$, also $k := (k_1, \ldots, k_l) \in \{0,1\}^l$ mit $kE =: (a_1, \ldots, a_l) \in 2\mathbb{Z}^l$.
(5) Ausgabe: Setze

$$y := \prod_{j=1}^{l} P_j^{k_j} \cdot \prod_{i=1}^{s} p_i^{a_i/2}, \; x := \prod_{j=1}^{l} (X[j])^{k_j}.$$

RETURN(ggT$(x-y, N)$)

7.28 Beispiel: (Morrison, Brillhardt)
$N = 13\,290\,059$
alle Primteiler p von R_i erfüllen $p \leq 113$

i	$A_i \bmod N$	R_i
5	171 341	$-1 \cdot 2 \cdot 5^2 \cdot 41$
10	6 700 527	$31 \cdot 43$
22	5 235 158	$41 \cdot 113$
23	1 914 221	$-1 \cdot 2 \cdot 113$
26	11 455 708	$2 \cdot 31 \cdot 53$
31	1 895 246	$-1 \cdot 2 \cdot 5^2 \cdot 113$
40	3 213 960	$2 \cdot 43 \cdot 53$

	-1	2	5	31	41	43	53	113
$v(5)$	1	1	2	.	1	.	.	.
$v(10)$	.	.	.	1	.	1	.	.
$v(22)$	.	.	.	.	1	.	.	1
$v(23)$	1	1	.	.	.	.	.	1
$v(26)$	.	1	.	1	.	.	1	.
$v(31)$	1	1	2	.	.	.	.	1
$v(40)$	.	1	.	.	.	1	1	.

$$(1,0,1,1,0,0,0) \in Kern \Rightarrow y^2 = R_5 \cdot R_{22} \cdot R_{23}$$

$$y = 2 \cdot 5 \cdot 41 \cdot 113, \qquad x = A_5 \cdot A_{22} \cdot A_{23}$$

liefert

$$\mathrm{ggT}(x-y, N) = 4261 \text{ und } N = 3119 \cdot 4261.$$

7.29 Bemerkung: Teuer bei CFRAC: häufiges vergebliches Faktorisieren der Q_n.
Neue Idee: Anstelle von Q_n Werte eines Polynoms $f \in \mathbb{Z}[X]$ betrachten: $p \mid f(m) \Rightarrow p \mid f(m+lp)$ für alle $l \in \mathbb{Z}$. Dadurch kann man eine Siebstrategie einsetzen:

7.30 Algorithmus: (Quadratisches Sieb, Pomerance 1982)
Eingabe: $N \in \mathbb{N}$ ungerade, keine Potenz
Ausgabe: (nichttrivialer) Teiler von N.
Algorithmus:
(1) $m := \lfloor \sqrt{N} \rfloor$; $f(X) := (X+m)^2 - N$;
(2) $\mathcal{F} := \{q_1, \ldots, q_s\}$ Menge von paarweise teilerfremden Zahlen, die sogenannte *factor base* (z.B. $\mathcal{F} = \mathbb{P}_{\leq S}$, wobei es sinnvoll ist, die kleinen Primzahlen durch geeignete Potenzen zu ersetzen).
(3) Bestimme $z_i = f(i)$, $i = 0, \pm 1, \pm 2, \ldots$ und speichere $\log(z_i) =: t_i$ (Logarithmus zu einer geeigneten Basis).
(4) Für $q \in \mathcal{F}$ bestimme $i_q, j_q \in \{0, \ldots, q-1\}$ mit $q \mid f_(i_q)$, $q \mid f(j_q)$ (falls solche Zahlen existieren). Dann gilt $q \mid f(i_q + ql)$ für alle $l \in \mathbb{Z}$. Setze also

$$t_{i_q+ql} := t_{i_q+ql} - \log(q), t_{j_q+ql} := t_{j_q+ql} - \log(q) \text{ für alle } l \in \mathbb{Z}$$

Philosophie: Ist t_i am Ende klein, so faktorisiert z_i über $\mathcal{F}$ (mit eventuellem kleinen Rest).
Weiter wie bei CFRAC:
(5) Faktorisiere die z_i mit t_i klein über $\mathbb{P}_{\leq S}$:

$$z_n = (-1)^e_{n,0} \prod_{j=1}^{s} p_j^{e_{nj}} P_n$$

$$E, P, X := \begin{pmatrix} \vdots & \vdots & \vdots & \vdots \\ e_{n,0} & e_{n,1} & \ldots & e_{n,s} \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}, \begin{pmatrix} \vdots \\ P_n \\ \vdots \end{pmatrix}, \begin{pmatrix} \vdots \\ m+n \\ \vdots \end{pmatrix}$$

(6), (7) wie bei CFRAC (3), (4).

7.31 Bemerkung: Ist $|\mathcal{F}|$ groß, so braucht man viele Werte z_i. Also wird $|i|$ und damit $|z_i|$ groß, d.h. man findet weniger glatte z_i.
Als Abhilfe benutzt man anstelle von einem Polynom eine ganze Familie von Polynomen:
$a, b, c \in \mathbb{Z}$, $b^2 - N = a^2 c$. $U(X) := a^2 X + b$, $V(X) := a$, $W(X) := a^2X^2 + 2bX + c$.
Dann ist

$$U(X)^2 \equiv V(X)^2 W(X) \pmod{N\mathbb{Z}[X]}.$$

Man siebt die Werte $W(i)$ (jedesmal neu für unterschiedliche Wahlen der a, b, c.).
Dann nennt man das Verfahren das MPQS: *multiple polynomial quadratic sieve*

7.32 Bemerkung: Das MPQS ist ein sehr guter general purpose Algorithmus für Zahlen $N < 10^{80}$ (Laufzeit im Stundenbereich).
Die Laufzeit von MPQS ist $O(L_N(1/2, c))$ für eine Konstante c.
Rekorde:

# Stellen	Jahr	# MIPS Jahre	Algorithmus
116	1990	275	MPQS
120	1993	830	MPQS
129	1994	5000	MPQS
130	1996	1000	NFS
140	1999	2000	NFS
155	1999	8000	NFS

Die Laufzeit von NFS ist $O(L_N(1/3, c))$ für eine Konstante c.

7.33 Grobe Idee des Zahlkörpersiebs (NFS) (Pollard, 1988)
(1) Wähle $f \in \mathbb{Z}[X]$ irreduzibel, $m \in \mathbb{Z}$ mit $f(m) \equiv 0 \pmod N$ und $\alpha \in \mathbb{C}$ mit $f(\alpha) = 0$. Dann ist

$$\Phi : \mathbb{Z}[\alpha] \to \mathbb{Z}/N\mathbb{Z}; \alpha \mapsto m + N\mathbb{Z}$$

ein Ringhomomorphismus.
Der Einfachheit halber nehmen wir an, dass f normiert ist. Sei $n :=$Grad(f).
<u>Ziel:</u> Suche $(a_i, b_i) \in \mathbb{Z}^2$, $\mathrm{ggT}(a_i, b_i) = 1$ so dass

$$(\star) \quad \prod (a_i - b_i\alpha) = \gamma^2, \quad \gamma \in \mathbb{Z}[\alpha]$$

$$\prod (a_i - b_i m) = x^2, \quad x \in \mathbb{Z}$$

Schreibt man $\gamma = \sum_{j=0}^{n-1} c_j \alpha^j$ mit $c_j \in \mathbb{Z}$, so ist $y := \sum_{j=0}^{n-1} c_j m^j \equiv \Phi(\gamma) \pmod N$ mit $x^2 \equiv y^2 \pmod N$.
Dazu siebe die $(a - b\alpha)$ und $(a - bm)$.
Eine notwendige Bedingung für $(\star)$ ist, dass $\prod \mathrm{Norm}(a_i - b_i\alpha) = \mathrm{Norm}(\gamma)^2$ ein Quadrat in $\mathbb{Z}$ ist. Dabei ist $P_b(a) := \mathrm{Norm}(a - b\alpha) = b^n f(a/b)$ für festes b ein Polynom in a. Siebe also die $P_b(a)$ für die verschiedenen b über $\mathbb{P}_{\leq S}$.
Beim Aufstellen der Matrix ist zu beachten, dass es in $\mathbb{Z}[\alpha]$ mehrere Primideale $\wp_j$ mit gleicher Primzahlnorm p geben kann. Von ihnen teilt aber höchstens eines das Ideal $(a - b\alpha)$, je nach Wert $a/b \pmod p$. Also braucht man die Primideals $\wp_j$ nicht auszurechnen und kann die Matrix aufstellen, ohne in $\mathbb{Z}[\alpha]$ wirklich zu rechnen.

Ein Element im Kern der Matrix entspricht einem Ideal $\Gamma \trianglelefteq \mathbb{Z}[\alpha]$ mit $\prod(a_i - b_i\alpha) = \Gamma^2$. Auch für $\mathbb{Z}$ brauchten wir noch eine zusätzliche Spalte für die Einheiten ± 1. In $\mathbb{Z}[\alpha]$ gibt es i.a. mehr Einheiten. Außerdem ist Γ nicht notwendig ein Hauptideal.
Also braucht man weitere notwendige Bedingungen dafür, dass die linke Seite von $(\star)$ ein Quadrat eines Elements in $\mathbb{Z}[\alpha]$ ist. Dazu wählt man weitere Primideale $\mathfrak{q}_j$ mit $\mathrm{Norm}(\mathfrak{q}_j) = q_j \in \mathbb{P}_{>S}$. Dann ist $\mathbb{Z}[\alpha]/\mathfrak{q}_j \cong \mathbb{F}_{q_j}$. Für jedes diese $\mathfrak{q}_j$ bekommt E eine zusätzliche Spalte, in die man eine 0 einträgt, falls $(a_i - b_i\alpha) + \mathfrak{q}_j \in (\mathbb{F}^*_{q_j})^2$ und eine 1 sonst.
Jetzt entspricht einem Element im Kern($E \pmod 2$) mit großer Wahrscheinlichkeit einer Lösung von $(\star)$.

7.34 Beispiel: te Riele et alii (Februar 1999)
RSA-140 =
21290 24631 82587 57547 49788 20162 71517 49780 67039 63277 21627 82333 83215 38194
99840 56495 91136 65738 53021 91831 67831 07387 99531 72308 89569 23087 34419 36471
=
33987 17423 02843 85545 30123 62761 38758 35633 98649 59695 97423 49092 93027 71479
×
62642 00187 40128 50961 51654 94826 44422 19302 03717 86235 09019 11166 06539 46049

$$\begin{aligned} \mathrm{f(t)} = \quad & 43\ 96820\ 82840\ t^5 \\ & + 39031\ 56785\ 38960\ t^4 \\ & - 7387\ 32529\ 38929\ 94572\ t^3 \\ & - 19\ 02715\ 324374\ 29887\ 14824\ t^2 \\ & - 6\ 34410\ 25694\ 46461\ 79139\ 30613\ t \\ & + 31855\ 39170\ 71474\ 35039\ 22235\ 07494 \end{aligned}$$

m= 3 44356 57809 24253 69517 79007.
Insgesamt ca. 9 CPU-Jahre Rechenzeit:
2000 h zur Auswahl von f.
Seiben: 185 Rechner insgesamt 8,9 CPU-Jahre, Schranke $\sim$ 12 Mio.
Matrix: 5 Mio. Zeilen und Spalten, 100 h zum Finden von Elementen im Kern.
Wurzelziehen: 14-20 h.

7.35 Bemerkung:
Vorteil des NFS: Freiheit der Wahl von f.
Wahrscheinlichkeit, dass $a - b\alpha$ glatt ist, ist sehr viel größer als bei MPQS.
Preis: Rechnen in $\mathbb{Z}[\alpha]$ kompliziert und wird bis auf das Wurzelziehen vermieden.
E hat mehr Spalten:
Für die $a_i - b_i m$: $\mathbb{P}_{\leq S}$,
Für die $a_i - b_i\alpha$: Primideale mit Norm in $\mathbb{P}_{\leq S}$ (eventuell mehr als eins pro Primzahl) (zum Sieben) und Primideale mit größerer Norm, um sicherzustellen, dass ein

Element im Kern von $E \pmod 2$ auch wirklich einem Quadrat eines Elements in $\mathbb{Z}[\alpha]$ entspricht und nicht nur einem Quadrat eines Ideals.
Bisher größte Zahl, die mit dem Zahlkörpersie faktorisiert wurde: $N = (10^{211} - 1)/9 = 111111111\ldots11111111$ (211 Einsen). Die spezielle Form von N wurde zur Wahl von f und m ausgenutzt: $f(X) = 10X^6 - 1$, $m = 10^{35}$. Insgesamt fast 11 CPU-Jahre Rechenzeit.

Abschnitt 8: Gröbnerbasen.

Situation: K Körper ($K = \mathbb{Q}, \mathbb{R}, \mathbb{F}_p, \ldots$) und $R := K[x_1, \ldots, x_n]$ der Polynomring über K in n Unbestimmten. Ist $I \trianglelefteq R$, so stellen sich folgende algorithmische Probleme:
Rechnen in R/I.
Berechne $\dim_K(R/I)$.
$f \in R$, entscheide ob $f \in I$.
Ist $n = 1$, so kann man diese Probleme mit dem Euklidischen Algorithmus angehen.
Sei $V(I) := \{a := (a_1, \ldots, a_n) \in K^n \mid f(a) = 0 \text{ für alle } f \in I\}$.
Entscheide ob $V(I) = \emptyset$.
Ist $V(I)$ endlich?
Bestimme Elemente aus $V(I)$.
Ist I von linearen Polynomen erzeugt, so kann man diese Probleme mit dem Gauß-Algorithmus lösen.
Das Ziel dieses Abschnittes ist, ein schönes Erzeugendensystem für I anzugeben, mit dessen Hilfe man diese Probleme algorithmisch lösen kann, indem man Gauß- und Euklidischen Algorithmus gleichzeitig verallgemeinert.

8.1 Satz (Hilbertscher Basissatz)
(i) $I \trianglelefteq R \Rightarrow$ Es gibt $f_1, \ldots, f_s \in R$ mit $I = (f_1, \ldots, f_s)$.
(ii) R ist Noethersch, d.h. ist $I_1 \subset I_2 \subset \ldots$ eine aufsteigende Kette von Idealen in R, so gibt es ein N mit $I_N = I_{N+1} = \ldots$.

8.2 Lemma:
Sei S irgendein kommutativer Ring. S ist Noethersch $\Leftrightarrow$ Jedes Ideal von S ist endlich erzeugt.
Beweis: "$\Rightarrow$": Sei $I \trianglelefteq S$. Sei $0 \neq f_1 \in I$, $I_1 := (f_1)$, $f_2 \in I \setminus I_1$, $I_2 := (f_1, f_2)$, $\ldots$. Ist I nicht endlich erzeugt so definiert dies eine unendlich aufsteigende Kette von Idealen in S.
"$\Leftarrow$": Ist $I_1 \subset I_2 \subset \ldots$ aufsteigende Kette von Idealen in S. Definiere $I := \bigcup_{j=1}^{\infty} I_j$. Dann ist $I \trianglelefteq S$. Da S Noethersch ist, gibt es $f_1, \ldots, f_s \in I$ mit $I = (f_1, \ldots, f_s)$. Dann gibt es aber ein N mit $f_1, \ldots, f_s \in I_N$, also ist $I = I_N = I_{N+1} = \ldots$.

8.3 Lemma:
Ist S ein kommutativer Noetherscher Ring, so ist $S[x]$ Noethersch.
Beweis: Sei $J \trianglelefteq S[x]$. Zeigen, J ist endlich erzeugt. Für $n \in \mathbb{N}$ sei $I_n := \{r \in S \mid rx^n + \sum_{j=0}^{n-1} a_j x^j (a_j \in S) \in J\}$.
Klar: $I \trianglelefteq S$, $I_n \subset I_{n+1} \subset \ldots$. Da S Noethersch ist, gibt es ein N mit $I_N = I_{N+1} = \ldots$. Außerdem ist I_j endlich erzeugt, also $I_j := (r_{j1}, \ldots, r_{jt_j})$. Sei $f_{ji} \in J$ mit

$f_{ji} = r_{ji}x^j + \ldots$. Dann ist $J = (f_{ji} \mid 0 \leq j \leq N, 1 \leq i \leq t_j) =: J'$. Denn: Klar ist $J' \subseteq J$. Umgekehrt sei $J_n := \{f \in J \mid Grad(f) = n\}$. Per Induktion über n zeigen wir, dass $J_n \subset J'$.
$n = 0$ klar, da $I_0 = (r_{01}, \ldots, r_{0t_0}) = J_0$.
$n \Rightarrow n+1$: $f \in J_{n+1} \Rightarrow f = rx^{n+1} + \sum_{i=0}^n a_i x^i$. Sei $m := \min(N, n+1)$ und $d := n+1-m$. Dann ist $r \in I_m$, $r = \sum_{j=1}^{t_m} a_j r_{m,j}$. Sei $g := \sum_{j=1}^{t_m} a_j f_{m,j} \in J'$. Dann ist $Grad(f - gx^d) \leq n$, also $f - gx^d \in J'$ nach IV. Da $g \in J'$ folgt also auch $f \in J'$.
Aus (8.3) ergibt sich nun der Hilbertsche Basissatz durch Induktion über n.

8.4 Definition:
(a) $\mathbb{M} := \{x_1^{\alpha_1} \ldots x_n^{\alpha_n} =: \mathbf{x}^\alpha \mid \alpha_1, \ldots, \alpha_n \in \mathbb{N}_0\}$ ist die Menge der *Monome* in R.
(b) Eine *Monomordnung* $<$ ist eine Totalordnung auf $\mathbb{M}$ mit
$1 < \mathbf{x}^\beta$ für alle $\mathbf{x}^\beta \in \mathbb{M}, \mathbf{x}^\beta \neq 1$.
$\mathbf{x}^\alpha < \mathbf{x}^\beta \Rightarrow \mathbf{x}^\alpha \mathbf{x}^\gamma < \mathbf{x}^\beta \mathbf{x}^\gamma$ für alle $\mathbf{x}^\gamma \in \mathbb{M}$.

8.5 Beispiel: (Monomordnungen)
$x_1 > x_2 > \ldots > x_n$.
(a) Die *lexikographische Ordnung* "lex": $\mathbf{x}^\alpha < \mathbf{x}^\beta \Leftrightarrow$ es gibt $i \in \{1, \ldots, n\}$ mit $\alpha_i < \beta_i$ und $\alpha_j = \beta_j$ für alle $j < i$.
(b) Die *Grad-lexikographische Ordnung* "deglex": $\mathbf{x}^\alpha < \mathbf{x}^\beta \Leftrightarrow$ entweder $\sum_{i=1}^n \alpha_i < \sum_{i=1}^n \beta_i$ oder $\sum_{i=1}^n \alpha_i = \sum_{i=1}^n \beta_i$ und es gibt $i \in \{1, \ldots, n\}$ mit $\alpha_i < \beta_i$ und $\alpha_j = \beta_j$ für alle $j < i$.
(c) Die *Grad-anti-lexikographische Ordnung* "degrevlex": $\mathbf{x}^\alpha < \mathbf{x}^\beta \Leftrightarrow$ entweder $\sum_{i=1}^n \alpha_i < \sum_{i=1}^n \beta_i$ oder $\sum_{i=1}^n \alpha_i = \sum_{i=1}^n \beta_i$ und es gibt $i \in \{1, \ldots, n\}$ mit $\alpha_i > \beta_i$ und $\alpha_j = \beta_j$ für alle $j < i$.

8.6 Bermerkung: Ist $<$ eine Monomordnung, so gilt $\mathbf{x}^\alpha \mid \mathbf{x}^\beta \Rightarrow \mathbf{x}^\alpha \leq \mathbf{x}^\beta$.

8.7 Satz: Jede Monomordnung ist eine Wohlordnung, d.h. für alle $\emptyset \neq A \subset \mathbb{M}$ gibt es $\mathbf{x}^\alpha \in A$ so dass $\mathbf{x}^\beta \geq \mathbf{x}^\alpha$ für alle $\mathbf{x}^\beta \in A$.
Beweis: Angenommen nicht, dann sei $\mathbf{x}^{\alpha_1} > \mathbf{x}^{\alpha_2} > \ldots$ mit $\mathbf{x}^{\alpha_i} \in A$. Das definiert eine unendliche echt aufsteigende Kette von Idealen $(\mathbf{x}^{\alpha_1}) \subset (\mathbf{x}^{\alpha_1}, \mathbf{x}^{\alpha_2}) \subset \ldots$. Es gilt $\mathbf{x}^{\alpha_{i+1}} \notin (\mathbf{x}^{\alpha_1}, \ldots, \mathbf{x}^{\alpha_i})$, da sonst $\mathbf{x}^{\alpha_{i+1}} = \sum_{j=1}^i p_j \mathbf{x}^{\alpha_j} = \sum_{l=1}^m a_l \mathbf{x}^{\beta_l}$ mit $p_j \in R$, $a_l \in K$. Da $\mathbb{M}$ eine K-Basis von R ist, folgt durch Koeffizientenvergleich, dass $m = 1, a_1 = 1, \beta_1 = \alpha_{i+1}$ ist. Also teilt eines der $\mathbf{x}_j^\alpha$ das $\mathbf{x}_{i+1}^\alpha$ $j \leq i$, d.h. mit 8.6 ist $\mathbf{x}^{\alpha_j} \leq \mathbf{x}^{\alpha_{i+1}}$ was ein Widerspruch ist.

8.8 Definition:
Sei $<$ eine Monomordnung un $f \in R$. Schreibe $f = a_1\mathbf{x}^{\alpha_1} + a_2\mathbf{x}^{\alpha_2} + \ldots + a_r\mathbf{x}^{\alpha_r}$ mit $a_j \in K \setminus \{0\}$, $\mathbf{x}^{\alpha_1} > \mathbf{x}^{\alpha_2} > \ldots > \mathbf{x}^{\alpha_r}$.
$lm(f) := \mathbf{x}^{\alpha_1}$ heißt das *Leitmonom* von f.
$lc(f) := a_1$ heißt der *Leitkoeffizient* von f.
$lt(f) := a_1\mathbf{x}^{\alpha_1}$ heißt das *Leitterm* von f.
$lm(0) := lc(0) := lt(0) := 0$.

8.9 Definition: Seien $f, g, h \in R$, $g \neq 0$. f *reduziert zu* h *mod* g *in einem Schritt*, $f \xrightarrow{g} h$ $:\Leftrightarrow lm(g)$ teilt einen Term $X \neq 0$ von f und $h = f - \frac{X}{lt(g)}g$.

8.10 Definition: Seien $f, h, f_1, \ldots, f_s \in R$, $f_i \neq 0$, $F := \{f_1, \ldots, f_s\}$.
(a) f *reduziert zu* h *modulo* F in Zeichen: $f \xrightarrow{F}_+ h$, falls es $i_1, \ldots, i_t \in \{1, \ldots, s\}$ gibt und $h_1, \ldots, h_{t-1} \in R$ mit

$$f \xrightarrow{f_{i_1}} h_1 \xrightarrow{f_{i_2}} h_2 \xrightarrow{f_{i_3}} \ldots \xrightarrow{f_{i_{t-1}}} h_{t-1} \xrightarrow{f_{i_t}} h.$$

(b) f heißt *reduziert modulo* F, falls entweder $r = 0$ oder kein Monom von f ist durch ein $lm(f_i)$ $(1 \leq i \leq s)$ teilbar.
(c) Ist $f \xrightarrow{F}_+ r$ und r reduziert modulo F, so heißt r ein *Rest von* f *modulo* F.

8.11 Algorithmus (Divisionsalgorithmus)
EINGABE: $f, f_1, \ldots, f_s \in R$, $f_i \neq 0$.
AUSGABE: $u_1, \ldots, u_s, r \in R$ mit $f = u_1 f_1 + \ldots + u_s f_s + r$, r reduziert modulo $\{f_1, \ldots, f_s\}$ und $lm(f) = \max_i lm(u_i f_i)$.
ALGORITHMUS $u_1 := 0; \ldots u_s := 0; r := 0; h := f;$

```
while h ≠ 0 do
        if (es gibt i mit lm(f_i) | lm(h)) then
              wähle minimales solches i
              u_i := u_i + lt(h)/lt(f_i);
              h := h − lt(h)/lt(f_i) · f_i;
        else
              r := r + lt(h); h := h − lt(h);
        end if;
end while;
```

Beispiel: (a) $f = x$, $f_1 = x^2$, $f_2 = x^2 - x$. Dann ist f reduziert modulo $\{f_1, f_2\}$ aber $f \in (f_1, f_2) = (x)$.
(b) $f := y^2x - x$, $f_1 := yx - y$, $f_2 := y^2 - x$ mit deglex, $y > x$. Dann gilt $f \xrightarrow{f_1} y^2 - x \xrightarrow{f_2} 0$ aber $f \xrightarrow{f_2} x^2 - x$ was reduziert ist modulo $\{f_1, f_2\}$.

8.12 Definition: $0 \notin G = \{g_1, \ldots, g_t\} \subset I \trianglelefteq R$ heißt *Gröbnerbasis* für I, falls für alle $f \in I$, $f \neq 0 \Rightarrow$ es gibt $i \in \{1, \ldots, t\}$ mit $lm(g_i) \mid lm(f)$.
Für $S \subset R$ definiere $Lt(S) := (lt(s) \mid s \in S) \trianglelefteq R$.

8.13 Satz: Sei $0 \notin G = \{g_1, \ldots, g_t\} \subset I \trianglelefteq R$. Äquivalent sind:
(i) G ist Gröbnerbasis für I.
(ii) $f \in I \Leftrightarrow f \xrightarrow{G}_+ 0$.
(iii) $f \in I \Leftrightarrow f = \sum_{i=1}^t h_i g_i$ mit $lm(f) = \max_i lm(h_i) lm(g_i)$.
(iv) $Lt(G) = Lt(I)$.
Beweis: (i) $\Rightarrow$ (ii): Sei $f \in R$. Alg. (8.11) liefert $f \xrightarrow{G}_+ r$ mit r reduziert modulo G. Ist $r = 0$, so ist $f \in I$. Umgekehrt ist $f \in I$, so auch $r \in I$. Angenommen $r \neq 0$, dann gibt es ein i mit $lm(g_i) \mid lm(r)$, also ist r nicht reduziert modulo G.
(ii) $\Rightarrow$ (iii): Algorithmus 8.11 liefert eine solche Darstellung.
(iii) $\Rightarrow$ (iv): Sei $f \in I$. Dann ist $f = \sum_{i=1}^s h_i g_i$ mit $lm(f) = \max_i lm(h_i) lm(g_i) \in Lt(G)$.
(iv) $\Rightarrow$ (i): Sei $f \in I$. Dann ist $lt(f) \in Lt(I) = Lt(G)$, d.h. es gibt $h_i \in R$ mit

$lt(f) = \sum_{i=1}^t h_i lt(g_i)$. Die rechte Seite ist ein Term, auf der linken Seite kommen nur Terme vor, die durch ein $lt(g_i)$ teilbar sind. Koeffizientenvergleich liefert $lt(f)$ ist durch ein $lt(g_i)$ teilbar. Also G ist Gröbnerbasis.

8.14 Folgerung: G Gröbnerbasis für $I \Rightarrow (G) = I$.

8.15 Lemma: Sei $S \subset \mathbb{M}$, $I = (S) \trianglelefteq R$, $f \in R$.
(a) $f \in I \Leftrightarrow$ jedes Monom, das in f vorkommt ist durch ein $s \in S$ teilbar.
(b) Es gibt eine endliche Teilmenge $S_0 \subset S$ mit $I = (S_0)$.
Beweis: (a) Ist $f = \sum_{i=1}^l h_i X_i$ mit $X_i \in S$, $h_i \in R$ so ist jeder Term von f durch eines der X_i teilbar.
(b) I hat endliches Erzeugendensystem nach dem Hilbertschen Basissatz, $I = (f_1, \dots, f_s)$. Nach (a) ist jeder Term der f_i teilbar durch ein $X_j \in S$. Ist S_0 die Menge dieser Terme X_j so ist $(S_0) = I$.

8.16 Folgerung: $0 \neq I \trianglelefteq R \Rightarrow I$ hat eine endliche Gröbnerbasis.
Beweis: $Lt(I)$ hat endliches Erzeugendensystem S_0 aus Monomen. Ist $S_0 = \{lm(g_1), \dots, lm(g_t)\}$, so ist $\{g_1, \dots, g_t\}$ eine Gröbnerbasis für I.

8.17 Satz: $G = \{g_1, \dots, g_t\} \subset R \setminus \{0\}$. G ist Gröbnerbasis $\Leftrightarrow$ für alle $f \in R$ ist der Rest der Division von f durch G eindeutig.
Beweis: $\Rightarrow$: Seien r_1, r_2 Reste von f modulo G. Dann ist $r_1 - r_2 \in I$ reduziert, also $r_1 - r_2 = 0$ nach (8.11) (ii).
$\Leftarrow$: Zeigen (8.11) (ii). Sei also $f \in I$ mit $f \xrightarrow{G}_+ r$, r reduziert modulo G. Wissen: r ist eindeutig. Müssen zeigen: $r = 0$. Dazu zeigen wir folgende
<u>Behauptung:</u> Sei $0 \neq c \in K$, $X \in \mathbb{M}$, $g \in R$ mit $g \xrightarrow{G}_+ r$, r reduziert. Dann gilt für alle $i \in \{1, \dots, t\}$ $g - cXg_i \xrightarrow{G}_+ r$.
Denn: Sei $h := g - cXg_i$. Dann kommt $X lm(g_i)$ nur in g oder nur in h oder in beiden vor, also gilt entweder $g \xrightarrow{g_i} h$ oder $h \xrightarrow{g_i} g$ oder es gibt $\tilde{g}$ mit $g \xrightarrow{g_i} \tilde{g}$ und $h \xrightarrow{g_i} \tilde{g}$. Da der Rest eindeutig ist folgt die Behauptung.
Damit ergibt sich der Satz, denn da $f \in I$ ist, ist $f = \sum_{\nu=1}^t c_\nu X_\nu g_{i_\nu}$ mit $X_\nu \in \mathbb{M}$.
$f \xrightarrow{G}_+ r \Rightarrow f - c_1 X_1 g_{i_1} \xrightarrow{G}_+ r \Rightarrow \dots\ f - \sum_{\nu=1}^t c_\nu X_\nu g_{i_\nu} \xrightarrow{G}_+ r$ d.h. $r = 0$.

8.18 Bemerkung: (a) Mit einer Gröbnerbasis G von I kann man in R/I rechnen, es gibt eindeutige Normalformen, die Reste. Z.B. bilden die Monome, die nicht durch ein $lm(g_i)$ teilbar sind eine Basis von R/I.
(b) Gröbnerbasen haengen von der gewählten Monomordnung ab.
(c) Wir wollen nun Bedingung (8.13)(ii) abschwächen um daraus einen Algorithmus zur Berechnung von Gröbnerbasen herzuleiten:

8.19 Definition: Seien $0 \neq f, g \in R$, $L := \mathrm{kgV}(lm(f), lm(g))$. Dann heißt

$$S(f,g) := \frac{L}{lt(f)} f - \frac{L}{lt(g)} g$$

das *S-Polynom* von f, g.

8.20 Beispiel: $f := 2yx - y$, $g := 3y^2 - x$, $R = \mathbb{Q}[x,y]$ mit deglex, $y > x$. $S(f,g) = 1/3x^2 - 1/2y^2$.
Ist $h \xrightarrow{f_1} h_1$ und $h \xrightarrow{f_2} h_2$ so ist $h_2 - h_1 = S(f_1, f_2)$.

8.21 Hauptsatz: Sei $G := \{g_1, \ldots, g_t\} \subset R\backslash\{0\}$. G ist eine Gröbnerbasis $\Leftrightarrow$ für alle $i \neq j$ $S(g_i, g_j) \xrightarrow{G}_+ 0$.

8.22 Lemma: Sei $\{f_1, \ldots, f_s\} \subset R$ so dass $lm(f_i) = X$ für alle $i = 1, \ldots, s$. Ist $f = \sum_{j=1}^s c_j f_j$ mit $c_j \in K$ und $lm(f) < X$, so ist f eine K-Linearkombination der $S(f_i, f_j)$.
Beweis: Sei $f_i = a_i X +$ kleinere Terme. Dann ist $f = \sum_{j=1}^s c_j f_j = \sum_{j=1}^s (a_j c_j) X +$ kleinere Terme. Also $\sum_{j=1}^s a_j c_j = 0$. Weiter ist $S(f_i, f_j) = \frac{1}{a_i} f_i - \frac{1}{a_j} f_j$ und $f = c_1 a_1 (\frac{1}{a_1} f_1 - \frac{1}{a_2} f_2) + (c_1 a_1 + c_2 a_2)(\frac{1}{a_2} f_2 - \frac{1}{a_3} f_3) + \ldots + + (c_1 a_1 + \ldots + c_{s-1} a_{s-1})(\frac{1}{a_{s-1}} f_{s-1} - \frac{1}{a_s} f_s) + (c_1 a_1 + \ldots + c_s a_s) \frac{1}{a_s} f_s$.
Beweis von (8.21): "$\Rightarrow$" Klar, da $S(g_i, g_j) \in I$.
"$\Leftarrow$" Zeigen (8.14) (iii): Sei $f \in I$ und $f = \sum_{i=1}^t h_i g_i$ mit minimalem $X = \max_i(lm(h_i) lm(g_i))$. Zu zeigen: $X = lm(f)$. Angenommen $X > lm(f)$. Dann sei $S := \{i \mid lm(h_i g_i) = X\}$. Für $i \in S$ schreibe $h_i = c_i X_i +$ kleinere Terme mit $c_i \in K$. Sei $g := \sum_{i \in S} c_i X_i g_i$. Dann ist $lm(X_i g_i) = X$, $lm(g) < X$. Aus Lemma (8.21) folgt dann, dass $g = \sum_{i,j} d_{ij} S(X_i g_i, X_j g_j)$, wo $S(X_i g_i, X_j g_j) = \frac{X}{lt(X_i g_i)} X_i g_i - \frac{X}{lt(X_j g_j)} X_j g_j = \frac{X}{lt(g_i)} g_i - \frac{X}{lt(g_j)} g_j = \frac{X}{X_{ij}} S(g_i, g_j)$, wo $X_{ij} = \mathrm{kgV}(lm(g_i), lm(g_j))$. Nach Voraussetzung reduziert $S(g_i, g_j) \xrightarrow{G}_+ 0$, also auch $S(X_i g_i, X_j g_j) \xrightarrow{G}_+ 0$, d.h. $S(X_i g_i, X_j g_j) = \sum_{\nu=1}^t h_{ij\nu} g_\nu$ mit $\max_\nu(lm(h_{ij\nu}) lm(g_\nu)) = lm(S(X_i g_i, X_j g_j)) < X$. Also ist auch $g = \sum_{i=1}^t \tilde{h}_i g_i$ mit $\tilde{h}_i \in R$, $max_i lm(\tilde{h}_i g_i) = lm(g) < X$. Definiert man $h_i' := h_i - c_i X_i + \tilde{h}_i$, so ist $f = \sum_{i=1}^t h_i' g_i$ mit kleinerem $\max_i(lm(h_i' g_i))$, was ein Widerspruch ist.

8.23 Folgerung: $G = \{g_1, \ldots, g_t\} \subset R\backslash\{0\}$. G ist Gröbnerbasis $\Leftrightarrow S(g_i, g_j) = \sum_{\nu=1}^t h_{ij\nu} g_\nu$ mit $\max_\nu(lm(h_{ij\nu}) lm(g_\nu)) = lm(S(g_i, g_j)) < X$ für alle $1 \leq i < j \leq t$.

8.24 Beispiel: $f_1 = xy - x$, $f_2 = x^2 - y$, $R = \mathbb{Q}[x,y]$ mit deglex, $x < y$. $S(f_1, f_2) \xrightarrow{f_2} y^2 - y =: f_3$. Also ist $\{f_1, f_2\}$ noch keine Gröbnerbasis. Indem man die 3 S-Polynome reduziert findet man, dass $\{f_1, f_2, f_3\}$ eine Gröbnerbasis ist.

8.25 Algorithmus: (Buchberger Algorithmus)
EINGABE: $F = \{f_1, \ldots, f_s\} \subset R \setminus \{0\}$.
AUSGABE: $G = \{g_1, \ldots, g_t\}$ Gröbnerbasis für (F).
ALGORITHMUS: $G := F$; $\mathcal{G} := \{\{f_i, f_j\} \mid f_i \neq f_j \in G\}$;

```
while 𝒢 ≠ ∅ do
        wähle {f, g} ∈ 𝒢;
        𝒢 := 𝒢 \ {{f, g}};
        S(f, g) →G+ h; (mit h reduziert modulo G)
        if (h ≠ 0) then
            𝒢 := 𝒢 ∪ {{f, h} | f ∈ G};
            G := G ∪ {h};
```

end if;
end while;

8.26 Satz: Algorithmus (8.25) terminiert.
Beweis: Im Schritt $G := G \cup \{h\}$ wird G vergrößert durch ein h, mit $lt(h) \notin Lt(G)$. Also konstrieren wir eine aufsteigende Kette von Idealen $Lt(G_1) \subset Lt(G_2) \subset Lt(G_3) \subset \ldots$, die nach dem Hilbertschen Basissatz stationär wird. Also wird dieser Schritt nur endlich oft durchgeführt, d.h. der Algorithmus terminiert.

8.27 Definition: Sei $G = \{g_1, \ldots, g_t\}$ eine Gröbnerbasis. G heißt *minimal*, falls für alle i $lc(g_i) = 1$ und für $i \neq j$ $lm(g_i) \nmid lm(g_j)$.

8.28 Bemerkung: Sei $G = \{g_1, \ldots, g_t\}$ eine Gröbnerbasis für I und $M := \{g_i \in G \mid$ es gibt $i \neq j$ mit $lm(g_j) \mid lm(g_i)\}$. Dann ist $G \setminus M$ eine minimale Gröbnerbasis für I.
Beweis: $Lt(G) = Lt(G \setminus M)$.

8.29 Lemma: Seien $F = \{f_1, \ldots, f_s\}$ und $G = \{g_1, \ldots, g_t\}$ zwei minimale Gröbnerbasen für I (bzgl. derselben Monomordnung). Dann ist $s = t$ und nach eventuellem Umordnen $lt(f_i) = lt(g_i)$ für alle i.
Beweis: $f_1 \in I$, G Gröbnerbasis, also existiert ein i mit $lt(g_i) \mid lt(f_1)$. Sei Œ$i = 1$. $g_1 \in I$, F Gröbnerbasis, also existiert ein j mit $lt(f_j) \mid lt(g_1) \mid lt(f_1)$. Aus der Minimalität von F folgt $j = 1$ und $lt(g_1) = lt(f_1)$. ...

8.30 Definition: Sei $G = \{g_1, \ldots, g_t\}$ eine Gröbnerbasis. G heißt *reduziert*, falls g_i reduziert ist modulo $G \setminus \{g_i\}$ und $lc(g_i) = 1$ für alle i.
Klar: Reduzierte Gröbnerbasen sind minimal.

8.31 Algorithmus:
EINGABE: $G = \{g_1, \ldots, g_t\}$ minimale Gröbnerbasis.
AUSGABE: $H = \{h_1, \ldots, h_t\}$ reduzierte Gröbnerbasis.
ALGORITHMUS:
$g_1 \xrightarrow{H_1}_+ h_1$, h_1 reduziert modulo $H_1 := \{g_2, \ldots, g_t\}$
$g_2 \xrightarrow{H_2}_+ h_2$, h_2 reduziert modulo $H_2 := \{h_1, g_3, \ldots, g_t\}$
$\vdots$
$g_t \xrightarrow{H_t}_+ h_t$, h_t reduziert modulo $H_t := \{h_1, \ldots, h_{t-1}\}$
Bemerkung: $H = \{h_1, \ldots, h_t\}$ ist reduzierte Gröbenerbasis von (G).
Beweis: $lt(h_j) = lt(g_j)$ für alle j. Also ist h_j reduziert modulo $H \setminus \{h_j\}$, da h_j reduziert modulo H_j und beide Mengen die gleiche lt haben.

8.32 Satz (Buchberger)
Sei $0 \neq I \trianglelefteq R$ und $<$ eine Monomordnung. Dann gibt es genau eine reduzierte Gröbnerbasis für I.
Beweis: Existenz ist klar. Zur Eindeutigkeit: Seien $G = \{g_1, \ldots, g_t\}$ und $H = \{h_1, \ldots, h_s\}$ zwei reduzierte Gröbnerbasen für I. Dann gilt nach (8.29) $t = s$ und $lt(g_i) = lt(h_i)$ (nach Umordnen). Angenommen $g_i \neq h_i$. Dann ist $g_i - h_i \in I$. Also gibt es ein j mit $lt(h_j) = lt(g_j) \mid lm(g_i - h_i) < lm(h_i)$. Also $i \neq j$ und

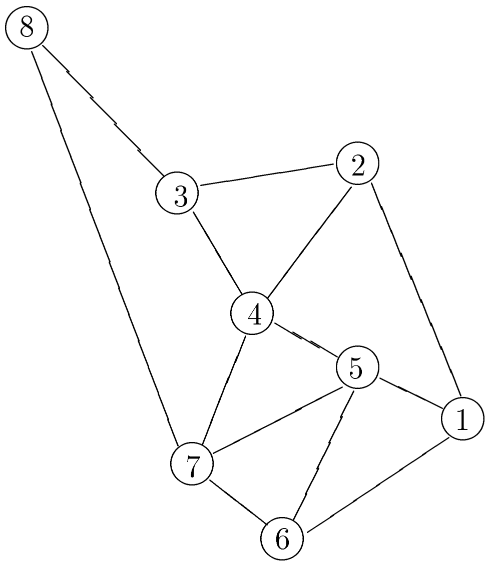
$lm(h_j) = lm(g_j)$ teilt einen Term vom g_i oder h_i, was ein Widerspruch dazu ist, dass G und H reduziert sind.

Abschnitt 9: Anwendungen.

Situation: Wie in Abschnitt 8, insbesondere K Körper ($K = \mathbb{Q}, \mathbb{R}, \mathbb{F}_p, \ldots$), $R := K[x_1, \ldots, x_n]$, $I \trianglelefteq R$.
Weiter sei $G := \{g_1, \ldots, g_s\}$ eine Gröbnerbasis für I.

9.1 Rechnen in R/I:
(i) Die Reste modulo G sind nach Satz (8.17) eindeutige Vertreter der Restklassen modulo I und können mit Algorithmus (8.11) berechnet werden. Also kann man Addieren, Multiplizieren und entscheiden, ob $f + I = 0$.
(ii) $f + I$ invertierbar in $R/I \Leftrightarrow$ die reduzierte Gröbnerbasis für $(g_1, \ldots, g_s, f)$ ist $\{1\}$. Algorithmus (8.25) kann so ergänzt werden, dass man sich die Transformation $F \to G$ merkt. Schreibt man dann $1 = \sum_{j=1}^{s} u_j g_j + uf$, so ist $u + I = (f + I)^{-1}$.
(iii) K-Basis von $R/I \longrightarrow$ Monome die durch kein $lm(g_j)$ teilbar sind. Also z.B. $\dim_K(R/I) < \infty \Leftrightarrow$ für alle $i \in \{1, \ldots, n\}$ gibt es ein $j \in \{1, \ldots, s\}$ und $\nu \in \mathbb{N}_0$ mit $lm(g_j) = x_i^{\nu}$. In diesem Fall kann man $V(I)$ mit Hilfe von Gröbnerbasen explizit bestimmen:

9.2 Beispiel: Das 3 Farben Problem:
Man will einen Graphen Γ so mit 3 Farben färben, dass je 2 benachbarte Ecken nie die gleiche Farbe bekommen. (Problem: Landkarte mit 3 Farben färben). Ist Γ 3-färbbar und wenn ja, bestimme eine (oder alle) Färbung.



Die 3 Farben werden durch die 3 Nullstellen von $X^3 - 1$ dargestellt.
Variablen: $x_1, \ldots, x_n$, für jede der n-Ecken in Γ.
Gleichungen: $x_i^3 - 1 = 0$ ($i = 1, \ldots, n$) und falls (i, j) eine Kante von Γ ist, so soll

$x_i \neq x_j$ sein, also $x_i^2 + x_i x_j + x_j^2 = 0$. Also ist $I := (x_i^3 - 1, x_i^2 + x_i x_j + x_j^2 \mid 1 \leq i \leq n, (i,j)$ Kante in $\Gamma)$. Dann bestimmt man eine Gröbnerbasis G für I bezüglich lex mit $x_1 > x_2 > \ldots > x_n$. In unserem Beispiel ist $G := \{x_1 - x_7, x_2 + x_7 + x_8, x_3 - x_7, x_4 - x_8, x_5 + x_7 + x_8, x_6 - x_8, x_7^2 + x_7 x_8 + x_8^2, x_8^3 - 1\}$. Also überblickt man leicht alle Färbungen von Γ:
Farbe 1: 8, 6, 4,
Farbe 2: 7, 1, 3,
Farbe 3: 2, 5.
Bis auf Permutation der Farben hat man genaus eine Möglichkeit.

9.3 Definition: I heißt *0-dimensional*, falls $\dim_K(R/I) < \infty$.
Für 0-dimensionale Ideale kann man mit Hilfe von Gröbnerbasen die Lösungsmenge $V(I)$ ausrechnen:

9.4 Satz: Sei $R \neq I \trianglelefteq R$ ein 0-dimensionales Ideal und $G = \{g_1, \ldots, g_s\}$ die reduzierte Gröbnerbasis für I bzgl. lex mit $x_1 < \ldots < x_n$. Dann kann G so umgeordnet werden, dass g_j ein Polynom in den Variablen $x_1, \ldots, x_j$ ist $(j = 1, \ldots, n)$.
Beweis: (9,1) (iii) sagt, dass für alle l ein j existiert mit $x_l^\nu = lm(g_j)$ für ein geeignetes ν. Wegen lex ist $g_j \in K[x_1, \ldots, x_l] \setminus K[x_1, \ldots, x_{l-1}]$.
Bemerkung: Mit G kann man $V(I)$ bestimmen, sobald man Nullstellen von Polynomen in einer Variable berechnen kann. Man berechnet alle Nullstellen α von g_1 (Polynom in x_1), setzt diese in $g_2, \ldots, g_n$ ein, berechnet für alle α die Nullstellen von $g_2(\alpha, x_2)$, setzt wieder ein, usw. Am Ende muss man noch testen, welche der endlich vielen Lösungen $g_{n+1}, \ldots, g_s$ erfüllen.
Wir kommen nun zur *Variablenelimination.*

9.5 Definition: Eine *Eliminationsordnung* für die Variablen x_i ist eine Monomordnung $<= E(<_x, <_y)$ auf $R := K[x_1, \ldots, x_n, y_1, \ldots, y_m]$ die aus zwei Monomordnungen $<_x$ bzw. $<_y$ auf $K[x_1, \ldots, x_n]$, bzw. $K[y_1, \ldots, y_m]$ zusammengesetzt ist. Sind die X_i Monome in den x_j und Y_i Monome in den y_j $(i = 1, 2)$, so definiert man

$$X_1 Y_1 < X_2 Y_2 \Leftrightarrow X_1 <_x X_2 \text{ oder } X_1 = X_2 \text{ und } Y_1 <_y Y_2.$$

Beispiel: lex = E(lex,lex) (aber lex ist sehr schlecht bei Rechnungen).

9.6 Satz: Sei $0 \neq I \trianglelefteq K[x_1, \ldots, x_n, y_1, \ldots, y_m]$ und $<$ eine Eliminationsordnung für die x_i. Ist G eine Gröbnerbasis für I, so ist $\tilde{G} := G \cap K[y_1, \ldots, y_m]$ eine Gröbnerbasis für $\tilde{I} := I \cap K[y_1, \ldots, y_m]$.
Beweis: Sei $f \in \tilde{I}$. Dann ist $f \in I$. Da G eine Gröbnerbasis für I ist, gibt es ein $g \in G$ mit $lm(g) \mid lm(f)$.' Da $lm(f)$ nur y_j enthält, kommt also auch in $lm(g)$ kein x_i vor und damit $g \in K[y_1, \ldots, y_m]$. Also ist $g \in \tilde{G}$ und damit $\tilde{G}$ Gröbnerbasis für $\tilde{I}$.

9.7 Bemerkung: Sind $I, J \trianglelefteq K[x_1, \ldots, x_n]$ und y eine neue Variable, so ist $I \cap J = (yI, (1-y)J) \cap K[x_1, \ldots, x_n]$.
Beweis: Sei $\tilde{I} := (yI, (1-y)J) \cap K[x_1, \ldots, x_n]$.
Dann gilt für $f \in I \cap J$, dass $f = yf + (1-y)f \in \tilde{I}$.

Umgekehrt sei $f \in \tilde{I}$. Dann ist f ein Polynom in den $x_1, \dots, x_n$, also konstant in y. Schreibt man f als Summe der Erzeuger und setzt in dieser Darstellung $y = 1$, so findet man $f \in I$, bei $y = 0$, $f \in J$. Also ist $f \in I \cap J$.

9.8 Beispiel: Polynomringe sind faktorielle Ringe, also ist ggT, kgV wohldefiniert. Gröbnerbasen verallgemeinern den Euklidischen Algorithmus insofern, dass man mit Bemerkung (9.7) ggT und kgV von Polynomen in mehreren Variablen berechnen kann: $(f) \cap (g) = (\mathrm{kgV}(f,g))$, $\mathrm{ggT}(f,g) = fg/\mathrm{kgV}(f,g)$.

Abschnitt 10: Involutive Basen.

Situation: Wie in Abschnitt 8, insbesondere K Körper ($K = \mathbb{Q}, \mathbb{R}, \mathbb{F}_p, \dots$), $R := K[x_1, \dots, x_n]$, $I \trianglelefteq R$.
Ein Problem beim Berechnen von und Rechnen mit Gröbnerbasen ist, dass zwar die Normalform eines Elementes eindeutig ist, nicht aber der Reduktionsweg selbst. Dabei kann man weite Umwege gehen. Das macht Gröbnerbasisberechnungen und auch das spätere Berechnen von Normalformen langsam. In diesem Abschnitt möchte ich eine Idee vorstellen, die auf Janet (1920) zurückgeht, also wesentlich älter ist als der Buchberger Algorithmus, um bessere Gröbnerbasen zu definieren. Die Idee hierbei ist, nicht jede Division in Algorithmus (8.11) zuzulassen, sondern nur sogenannte Janet-Divisionen, die von dem aktuellen Idealerzeugensystem U abhängen. Das führt dazu, dass jedes Monom höchstens einen Janet-Teiler in U hat. Janet hat seine Ideen zum Lösen von Differentialgleichungssystemen benutzt.
Man kann diese Ideen axiomatisieren und so verschiedene zulässige Divisionen betrachten. Wir wollen hier nur den wichtigsten Spezialfall der Janet-Division betrachten, die zu (i.a. nicht reduzierten) Gröbnerbasen führt.

10.1 Definition: (Janet 1920) Sei $U \subset \mathbb{M}$ endlich und $[d_1, \dots, d_i] := \{u \in U \mid \mathrm{Grad}_{x_j} = d_j, j = 1, \dots, i\}$, $[] := U$.
(a) Eine Variable x_j heißt *multiplikativ* für $u \in U$ $x_j \in M(u,U) :\Leftrightarrow \mathrm{Grad}_{x_j}(u) = \max\{Grad_{x_j}(v) \mid v \in [\mathrm{Grad}_{x_1}(u), \dots, \mathrm{Grad}_{x_{j-1}}(u)]\}$. Die Menge der nichtmultiplikativen Variablen ist $NM(u,U) := \{x_1, \dots, x_n\} \setminus M(u,U)$.
(b) Sei $m \in \mathbb{M}$. Dann ist $u \in U$ ein *Janet-Teiler* von um: $u \mid_U um$, genau dann, wenn alle in m vorkommenden Variablen in $M(u,U)$ sind.

10.2 Satz: Die Janet Division erfüllt die folgenden Bedingungen ($u, u_1 \in U \subset \mathbb{M}, v, w \in \mathbb{M}$):
(i) $u \mid_U w \Rightarrow u \mid w$.
(ii) $u \mid_U u$ für alle $u \in U$.
(iii) $u \mid_U (uw)$ und $u \mid_U (uv) \Leftrightarrow u \mid_U (uvw)$.
(iv) $u \mid_U w$ und $u_1 \mid_U w \Rightarrow u = u_1$.
(v) $V \subset U$, $u \in V$. Dann gilt $u \mid_U w \Rightarrow u \mid_V w$
Beweis: (i),(ii),(iii) und (v) folgen direkt aus der Definition.
Zu (iv): Seien $u \neq u_1 \in U$ und k maximal mit $d_i := \mathrm{Grad}_{x_i}(u) = \mathrm{Grad}_{x_i}(u_1)$ für alle $1 \leq i < k$. Also $u, u_1 \in [d_1, \dots, d_{k-1}]$. ŒGrad$_{x_k}(u) > \mathrm{Grad}_{x_k}(u_1)$ (also x_k nicht multiplikativ für u_1). Schreibe $w = mu = m_1 u_1$. Da $u_1 \mid_U m_1 u_1$, kommt x_k in m_1 nicht vor. Also ist $\mathrm{Grad}_{x_k}(w) = \mathrm{Grad}_{x_k}(u_1) < \mathrm{Grad}_{x_k}(u)$, was ein Widerspruch ist.

10.3 Folgerung: Jedes $w \in \mathbb{M}$ hat höchstens einen Janet Teiler in U.

10.4 Definition: $U \subset \mathbb{M}$ heißt *involutiv* (*vollständig*), $:\Leftrightarrow$ für alle $u \in U$, für alle $w \in \mathbb{M}$ gibt es ein $v \in U$ mit

$$v \mid_U uw.$$

10.5 Lemma: $w = x_j$ Variable in (10.4). Seien $u, v \in U$ mit $v \mid_U ux_j$. Dann ist $v \geq u$ bzgl. lex mit $x_1 > \ldots > x_n$.
Beweis: Ist $x_j \in M(u, U)$, dann gilt $u \mid_U ux_j$, also (10.3) $u = v$.
Ist $x_j \in NM(u, U)$, so sei $d_i := \text{Grad}_{x_i}(u)$, $i = 1, \ldots, n$. Sei i minimal mit $\text{Grad}_{x_i}(v) \neq d_i$. Angenommen $\text{Grad}_{x_i}(v) < d_i$. Dann gilt wegen $v \mid_U ux_j$, dass $x_i \in M(v, U)$. Da $u, v \in [d_1, \ldots, d_{i-1}]$, folgt $\text{Grad}_{x_i}(v) \geq \text{Grad}_{x_i}(u)$ was ein Widerspruch ist.

10.6 Lemma: $U \subset \mathbb{M}$ ist involutiv $\Leftrightarrow$ für alle $u \in U$, für alle $x_j \notin M(u, U)$ gibt es ein $v \in U$ mit $v \mid_U ux_j$.
Beweis: $\Rightarrow$: Klar.
$\Leftarrow$: Sei $w \in \mathbb{M}$. Suchen $v \in U$ mit $v \mid_U uw$.
Falls $u \mid_U wu$, so sind wir fertig. Sonst gibt es $x_{j_1} \in NM(u, U)$ mit $x_{j_1} \mid w$.
Nach Voraussetzung existiert ein $v_1 \in U$ mit $v_1 \mid_U ux_{j_1}$.
Falls $v_1 \mid_U uw$, so sind wir fertig, sonst gibt es $x_{j_2} \in NM(v_1, U)$ mit $x_{j_2} \mid (uw)/v_1$.
Nach Voraussetzung existiert ein $v_2 \in U$ mit $v_2 \mid_U v_1 x_{j_2}$. Falls $v_2 \mid_U uw$, so sind wir fertig, sonst .... So konstruieren wir eine Folge $u = v_0, v_1, v_2, \ldots$ von Monomen mit $v_i \mid uw$ und $v_{i+1} > v_i$ bzgl. lex (nach (10.5)). Also gibt es nur endlich viele v_i, d.h. eines der v_j Janet-teilt uw.

10.7 Satz: Für alle $U \subset \mathbb{M}$ endlich, gibt es $\tilde{U} \subset \mathbb{M}$ endlich, mit $U \subset \tilde{U}$, $\tilde{U}$ ist involutiv und $(U) = (\tilde{U})$.
$\tilde{U}$ kann mit folgendem Algorithmus konstruiert werden.
Eingabe: $U \subset \mathbb{M}$ endlich.
Ausgabe: $\tilde{U} \supseteq U$ endlich, involutiv, $(U) = (\tilde{U})$.
Algorithmus:
$\tilde{U} := U$;
while (es gibt ein $u \in \tilde{U}$, $x \in NM(u, \tilde{U})$ so dass $u \cdot x$ keine Janet-Teiler in $\tilde{U}$ hat)
do
$\tilde{U} := \tilde{U} \cup \{ux\}$;
end while;
Beweis: Wir zeigen, dass der angegebene Algorithmus terminiert mit einer endlichen involutiven Menge $\tilde{U}$.
Klar $\tilde{U}$ ist am Ende involutiv.
endlicher Algorithmus: Sei $h_j := \max_{u \in U} \text{Grad}_{x_j}(u)$. Dann ist $h_j = \max_{u \in \tilde{U}} \text{Grad}_{x_j}(u)$, denn falls $x_j \in NM(u, \tilde{U})$, so ist $\text{Grad}_{x_j}(u) < h_j$ also auch $\text{Grad}_{x_j}(x_j u) \leq h_j$.
Wir kommen übertragen jetzt alle Begriffe für Polynome.

10.8 Definition:
Sei $F \subset K[x_1, \ldots, x_n]$ endlich und $>$ eine feste Monomordnung kompatibel mit $x_1 > x_2 > \ldots > x_n$.

(a) $f \in F$, so sei $M(f,F) := M(lm(f), lm(F))$ und $NM(f,F) := NM(lm(f), lm(F))$ die Menge der multiplikativen bzw. nichtmultiplikativen Variablen.
(b) $h \in K[x_1, \dots, x_n]$. Wir sagen, dass h *in einem Schritt modulo* (f, F) *zu* g *Janet-reduziert* und schreiben

$$h \underset{F}{\overset{f}{\to}} g$$

falls h einen Term am hat ($a \in K \setminus \{0\}, m \in \mathbb{M}$) mit $lm(f) \mid_{lm(F)} m$ und $g = h - \frac{am}{lt(f)} f$.
(c) Analog definieren wir

$$h \underset{F}{\overset{F}{\to_+}} g$$

falls h mit einer Folge von Janet-Reduktionen modulo F zu g reduziert.
(d) h heißt *Janet-reduziert* modulo F, falls kein Term von h Janet-teilbar durch ein $f \in F$. Gilt $h \underset{F}{\overset{F}{\to_+}} g$ und ist g Janet-reduziert modulo F, so heißt g ein *Janet-Rest* oder auch eine *Janet-Normalform* von h modulo F.
(e) F heißt *Janet-autoreduziert*, falls jedes $f \in F$ Janet-reduziert modulo $F \setminus \{f\}$ ist.
(f) F heißt *Janet-Basis* oder *involutive Basis* für (F), falls F Janet-autoreduziert ist und $lm(F)$ involutiv.

10.9 Satz: Sei $F \subset R = K[x_1, \dots, x_n]$ endlich und $<$ eine Monomordung auf R mit $x_1 < \dots < x_n$.
(a) Ist F Janet-autoreduziert, so hat jedes $h \in R$ eine eindeutige Janet-Normalform modulo F, sogar der Reduktionsweg ist eindeutig.
(b) Ist F eine Janet-Basis, so ist F eine (i.a. nicht reduzierte) Gröbnerbasis und die Janet-Normalform von $h \in R$ ist gleich der (Gröbner)-Normalform von h modulo F.

10.10 Beispiele