Algorithmische Zahlentheorie

Wolfgang M. Ruppert

Wintersemester 2001/2002

14. Februar 2002[1]

[1]Im Wintersemester 2001/2002 am Mathematischen Institut der Universität Erlangen abgehaltene Vorlesung

Inhaltsverzeichnis

Vorbemerkung

Die Vorlesung wollte eine Einführung in die Algorithmische Zahlentheorie (Computational Number Theory) anhand einiger ausgewählter Themen geben. Behandelt wurden hauptsächlich Kettenbrüche und quadratische Klassengruppen nebst Anwendungen auf Faktorisierungsmethoden und Kryptographie.

Das vorliegende Skript entstand parallel zur Vorlesung und deckt (zur Zeit) nur den behandelten Vorlesungsstoff ab.

Die beigefügten Programme dienten der Konstruktion von Beispielen und der Illustration der vorgestellten Algorithmen. Die angegebenen Rechenzeiten beziehen sich auf einen Pentium-II-PC mit 400 MHz und 128 MB Arbeitsspeicher unter Linux. Folgende Programmpakete wurden verwendet: Maple 6, GMP 3.1.1 und 4.0, NTL 5.2 und Java.

Version vom 13.2.2002

KAPITEL 1

Der euklidische Algorithmus

1. Grundlegende Eigenschaften der natürlichen und ganzen Zahlen

Ganze Zahlen kann man addieren, subtrahieren, multiplizieren, wobei eine Reihe von Gesetzmäßigkeiten gilt, wie die Kommutativität von Addition und Multiplikation, Distributivgesetze, etc. Algebraisch gesprochen: Die Menge der ganzen Zahlen $\mathbf{Z}$ bildet einen **Integritätsring**.

Um ganze bzw. natürliche Zahlen explizit anzugeben, legt man eine festgewählte natürliche Zahl $b \geq 2$ als Basis zugrunde. Dann läßt sich jede natürliche Zahl n eindeutig schreiben als

$$n = \sum_{i=0}^{k-1} n_i \cdot b^i \quad \text{mit} \quad n_i \in \{0, 1, \ldots, b-1\}.$$

Man schreibt auch $n = (n_{k-1} n_{k-2} \ldots n_1 n_0)_b$, im Fall $b = 10$ hat man die Dezimaldarstellung, im Fall $b = 2$ die Binärdarstellung. Ist $n_{k-1} \neq 0$, so sagt man, n hat k Stellen bzgl. der Basis b. Ist n zur Basis b k-stellig, so gilt $b^{k-1} \leq n < b^k$, also $k - 1 \leq \frac{\ln n}{\ln b} < k$ und somit erhält man für die Stellenzahl

$$k = \lfloor \frac{\ln n}{\ln b} \rfloor + 1.$$

Die Anzahl der Bits von n ist die Anzahl der Stellen in der Binärdarstellung, also $\lfloor \frac{\ln n}{\ln 2} \rfloor + 1$.

Beispiel: Hier sind 3 Darstellungen einer 27-Bit-Zahl:

$$123456789 = (123456789)_{10} = (111010110111100110100010101)_2 = (75bcd15)_{16}.$$

(Binär- bzw. Hexadezimaldarstellung einer Zahl n liefert Maple mit den Funktionen 'convert(n,binary)' bzw. 'convert(n,hex)'.)

Eine wichtige Eigenschaft der natürlichen Zahl ist die **Division mit Rest**: Teilt man eine natürliche Zahl a durch eine natürliche Zahl b, so erhält man einen Quotienten q und einen Rest r.

Beispiel: Wir teilen 12345 durch 987 nach dem in der Schule gelernten Verfahren:

```
1  2  3  4  5  :  9  8  7  =  1  2
-  -  -  -  -
1  2  3  4
   9  8  7
-  -  -  -
   2  4  7  5
   1  9  7  4
   -  -  -  -
      5  0  1
```

12345 durch 987 ergibt also 12 Rest 501.

Ausgedehnt auf die ganzen Zahlen bedeutet dies, daß man zu $a, b \in \mathbf{Z}, b \neq 0$ Zahlen $q, r \in \mathbf{Z}$ findet mit

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

Dadurch wird $\mathbf{Z}$ zu einem **euklidischen Ring**.

Mit obiger Normierung ist die Division mit Rest sogar eindeutig, wie folgendes Lemma zeigt:

LEMMA. *Sind $a, b \in \mathbf{Z}, b \neq 0$, sind $q, r, q', r' \in \mathbf{Z}$ mit*

$$a = qb + r = q'b + r' \quad \text{und} \quad 0 \leq r < |b|, 0 \leq r' < |b|,$$

so ist $q = q'$ und $r = r'$.

Beweis: Sei o.E. $r \leq r'$. Dann ist $0 \leq r' - r \leq r' < |b|$ und damit

$$0 \leq \frac{r' - r}{|b|} < 1.$$

Aus $a = qb + r = q'b + r'$ folgt $(q - q')b = r' - r$ und somit ist

$$\frac{r' - r}{|b|} = \pm\frac{r' - r}{b} = \pm(q - q') \in \mathbf{Z}.$$

Da es keine ganze Zahl gibt, die echt zwischen 0 und 1 liegt, folgt $\frac{r'-r}{|b|} = 0$, also $r = r'$ und damit auch $q = q'$, wie behauptet. ■

Teilbarkeit: Für $a, b \in \mathbf{Z}$ sagt man b teilt a, in Zeichen $b|a$, wenn $c \in \mathbf{Z}$ existiert mit $a = bc$. Die folgenden Eigenschaften sind leicht einzusehen:

- $a|a$, $1|a$, $a|0$.
- $a|b, b|c \Longrightarrow a|c$.
- $a|b \iff \pm a| \pm b$.
- Für $a, b \neq 0$: $a|b$ und $b|a \iff b = \pm a$.
- $a|1 \iff a = \pm 1$.

Eine natürliche Zahl $p > 1$ heißt **Primzahl**, wenn die einzigen Teiler von p die Zahlen $\pm 1, \pm p$ sind, d.h. wenn p nur triviale Teiler hat. Die Menge der Primzahlen beginnt mit

$$2, \quad 3, \quad 5, \quad 7, \quad 11, \quad 13, \quad 17, \quad 19, \quad 23, \quad 29, \quad 31, \quad 37, \quad 41, \quad 43, \quad 47, \quad \ldots$$

SATZ (Fundamentalsatz der Arithmetik). *Jede ganze Zahl $n \neq 0$ läßt sich eindeutig (bis auf die Reihenfolge der Faktoren) schreiben als*

$$n = \pm p_1^{e_1} p_2^{e_2} \ldots p_r^{e_r}$$

mit paarweise verschiedenen Primzahlen $p_1, p_2, \ldots, p_r$ und natürlichen Zahlen $e_1, e_2, \ldots, e_r$.

Bemerkungen:

(1) Die Eigenschaften des Fundamentalsatzes der Arithmetik heißt in der Sprache der Algebra: Der Ring der ganzen Zahlen $\mathbf{Z}$ ist faktoriell.
(2) Gerade wenn die Primfaktorzerlegung mehrerer Zahlen gleichzeitig betrachtet wird, schreibt man oft auch $\prod p_i^{e_i}$ mit $e_i \geq 0$, wobei dann p_i eventuell auch alle Primzahlen durchlaufen kann.
(3) Der Fundamentalsatz der Arithmetik ist einfach zu beweisen. Allerdings ist bis jetzt kein Verfahren bekannt, mit dem man die Primfaktorzerlegung einer (allgemeinen) größeren Zahl explizit in vernünftiger Zeit bestimmen kann. Dies gehört zu den Grundproblemen der Computational Number Theory.
(4) Die Maple-Funktion 'ifactor(n)' versucht die Primfaktorzerlegung von n zu bestimmen.

Beispiele: Mit Maple fanden wir folgende Primfaktorzerlegungen:

$$\begin{aligned}
123456789^1 + 1 &= 2 \cdot 5 \cdot 37 \cdot 333667 \\
123456789^2 + 2 &= 19611665867 \cdot 777169 \\
123456789^3 + 3 &= 2^4 \cdot 3 \cdot 23 \cdot 53 \cdot 107 \cdot 151 \cdot 687042689 \cdot 2897047 \\
123456789^4 + 4 &= 5 \cdot 41 \cdot 66896856073 \cdot 1545015560393 \cdot 5557 \cdot 1973 \\
123456789^5 + 5 &= 2 \cdot 1433985930149859053616880719046836024147 7 \\
123456789^6 + 6 &= 3 \cdot 7 \cdot 11 \cdot 727 \cdot 18625377199877253434584487963069 \cdot 40639 \cdot 27854501 \\
123456789^7 + 7 &= 2^2 \cdot 89223613194148612592136282728482683228353863 \cdot 294493777 \cdot 4159
\end{aligned}$$

(Den Faktorisierungsversuch von $123456789^8 + 8$ haben wir abgebrochen.)

Mit der eindeutigen Primfaktorzerlegung ganzer Zahlen erhält man eine schöne Charakterisierung der Teilbarkeitsrelation:

LEMMA. *Sind $a, b \in \mathbf{Z}$, $a, b \neq 0$, mit den Primfaktorzerlegungen*

$$a = \pm \prod_i p_i^{a_i} \quad \text{und} \quad b = \prod_i p_i^{b_i},$$

so gilt

$$a|b \quad \Longleftrightarrow \quad a_i \leq b_i \text{ für alle } i,$$

oder kurz geschrieben:

$$\pm \prod_i p_i^{a_i} | \pm \prod_i p_i^{b_i} \quad \Longleftrightarrow \quad a_i \leq b_i \text{ für alle } i.$$

Beweis: $\Rightarrow$ Wegen $a|b$ existiert $c \in \mathbf{Z}$, $c \neq 0$, mit $b = ac$. Sei $c = \pm \prod p_i^{c_i}$ die Primfaktorzerlegung von c. Dann folgt aus $b = ac$

$$\pm \prod p_i^{b_i} = \pm \prod p_i^{a_i} \cdot \prod p_i^{c_i} = \pm \prod p_i^{a_i + c_i}.$$

Die Eindeutigkeit der Primfaktorzerlegung liefert $b_i = a_i + c_i$, was wegen $c_i \geq 0$ die Behauptung $b_i \geq a_i$ zeigt.
$\Leftarrow$ Ist $a_i \leq b_i$, so ist $b_i - a_i \geq 0$, also $c = \prod p_i^{b_i - a_i}$ eine natürliche Zahl. Wie eben sieht man $b = \pm ac$ und damit $a|b$, was behauptet war. ■

Wir betrachten jetzt gemeinsame Teiler zweier ganzer Zahlen $m, n \in \mathbf{Z}$ und schreiben

$$\mathrm{gT}(m, n) = \{d \in \mathbf{Z} : d|m, d|n\}.$$

Beispiele:

$$\mathrm{gT}(20, 24) = \{\pm 1, \pm 2, \pm 4\}, \quad \mathrm{gT}(6, 0) = \{\pm 1, \pm 2, \pm 3, \pm 6\}, \quad \mathrm{gT}(6, 1) = \{\pm 1\}.$$

Mit den Teilbarkeitsregeln sieht man sofort, daß gilt

$$\mathrm{gT}(m, 0) = \{d \in \mathbf{Z} : d|m\} = \{\text{Menge der Teiler von } m\} \quad \text{und} \quad \mathrm{gT}(m, 1) = \{\pm 1\}.$$

Bei Kenntnis der Primfaktorzerlegung ist die Bestimmung der gemeinsamen Teiler mit dem obigen Lemma ganz einfach:

$$\mathrm{gT}(\pm \prod p_i^{a_i}, \pm \prod p_i^{b_i}) = \{\pm \prod p_i^{c_i} : 0 \leq c_i \leq \min(a_i, b_i)\}.$$

Außerdem:

$$\mathrm{gT}(\pm \prod p_i^{a_i}, 0) = \{\pm \prod p_i^{c_i} : c_i \leq a_i\} \quad \text{und} \quad \mathrm{gT}(0, 0) = \mathbf{Z}.$$

Die Menge $gT(m, n)$ enthält ein (bzgl. Teilbarkeit) maximales Element, das wir o.E. ≥ 0 wählen, den sogenannten $\mathrm{ggT}(m, n)$:

$$\begin{aligned} \mathrm{ggT}(\pm \prod p_i^{a_i}, \pm \prod p_i^{b_i}) &= \prod p_i^{\min(a_i, b_i)}, \\ \mathrm{ggT}(\pm \prod p_i^{a_i}, 0) &= \prod p_i^{a_i}, \\ \mathrm{ggT}(0, 0) &= 0. \end{aligned}$$

Das folgende, nun leicht zu beweisende Lemma charakterisiert den ggT:

LEMMA. *Seien $m, n \in \mathbf{Z}$. Dann hat $d = \mathrm{ggT}(m, n)$ die folgenden Eigenschaften und ist dadurch eindeutig bestimmt:*

(1) $d \in \mathrm{gT}(m, n)$.
(2) $d' \in \mathrm{gT}(m, n) \Longrightarrow d'|d$.
(3) $d \geq 0$.

Bemerkungen:

(1) Durch die ersten beiden Eigenschaften charakterisiert man einen ggT in der Algebra.
(2) Die dritte Eigenschaft $d \geq 0$ dient nur der Normierung. Ohne diese wäre $\mathrm{ggT}(m, n)$ nur bis aufs Vorzeichen bestimmt.

Man nennt ganze Zahlen $m, n \in \mathbf{Z}$ teilerfremd, wenn $\mathrm{ggT}(m,n) = 1$ gilt. Ist $m = \pm\prod p_i^{a_i}$, $n = \pm\prod p_i^{b_i}$, so gilt mit obiger Formel

$$\mathrm{ggT}(m,n) = 1 \quad \Longleftrightarrow \quad a_i = 0 \text{ oder } b_i = 0,$$

d.h. die Menge der m teilenden Primzahlen und die Menge der n teilenden Primzahlen sind disjunkt.

Das folgende Lemma gibt noch ein paar weitere nützliche Eigenschaften an:

LEMMA. *Für ganze Zahlen $a, b, c \neq 0$ gilt:*

(1) *$a|c$, $b|c$ und $\mathrm{ggT}(a,b) = 1$ impliziert $ab|c$.*
(2) *$a|bc$ und $\mathrm{ggT}(a,c) = 1$ impliziert $a|b$.*
(3) *Ist $d = \mathrm{ggT}(a,b)$, schreibt man $a = da'$, $b = db'$, so gilt $\mathrm{ggT}(a',b') = 1$.*

Analog zu gT definiert für $m, n \in \mathbf{Z}$ die Menge der gemeinsamen Vielfachen

$$\mathrm{gV}(m,n) = \{e \in \mathbf{Z} : m|e, n|e\}.$$

Für Zahlen $\neq 0$ ergibt sich die einfache Charakterisierung durch ihre Primfaktorzerlegung

$$\mathrm{gV}(\pm\prod p_i^{a_i}, \pm\prod p_i^{b_i}) = \{\pm\prod p_i^{c_i} : \max(a_i, b_i) \leq c_i\} \cup \{0\}.$$

Die Menge $\mathrm{gV}(m,n)$ enthält ein bzgl. Teilbarkeit kleinstes Element, das kleinste gemeinsame Vielfache:

$$\begin{aligned} \mathrm{kgV}(\pm\prod p_i^{a_i}, \pm\prod p_i^{b_i}) &= \prod p_i^{\max(a_i,b_i)}, \\ \mathrm{kgV}(\pm\prod p_i^{a_i}, 0) &= \mathrm{kgV}(0,0) = 0. \end{aligned}$$

Für $m, n \in \mathbf{Z}$ gilt

$$\mathrm{kgV}(m,n) \cdot \mathrm{ggT}(m,n) = \pm mn.$$

Allerdings kann man ggT oder kgV mit den oben angegeben Formeln nur berechnen, wenn man die Primfaktorzerlegung der beteiligten Elemente kennt.

2. Der euklidische Algorithmus

Sind $a, b, q, r \in \mathbf{Z}$ mit $a = qb + r$, so gilt für $d \in \mathbf{Z}$

$$d \in \mathrm{gT}(a,b) \iff d|a, d|b \iff d|a, d|b, d|r \iff d|b, d|r \iff d \in \mathrm{gT}(b,r),$$

also $gT(a,b) = gT(b,r)$ und damit auch $\mathrm{ggT}(a,b) = \mathrm{ggT}(b,r)$. Diese Eigenschaft führt zum euklidischen Algorithmus:

SATZ (Euklidischer Algorithmus). *Seien a_0, a_1 ganze Zahlen $\neq 0$. Man definiert rekursiv ganze Zahlen a_i und q_i durch die Vorschrift: Solange $a_{i+1} \neq 0$ gilt, teilt man a_i durch a_{i+1} und erhält den Quotienten q_i und den Rest a_{i+2}. Explizit:*

$$\begin{aligned} a_0 &= q_0 a_1 + a_2 \quad \textit{mit} \quad 0 < a_2 < |a_1|, \\ a_1 &= q_1 a_2 + a_3 \quad \textit{mit} \quad 0 < a_3 < a_2, \\ &\vdots \quad\quad \vdots \\ a_i &= q_i a_{i+1} + a_{i+2} \quad \textit{mit} \quad 0 < a_{i+2} < a_{i+1}, \\ &\vdots \quad\quad \vdots \\ a_{n-2} &= q_{n-2} a_{n-1} + a_n \quad \textit{mit} \quad 0 < a_n < a_{n-1}, \\ a_{n-1} &= q_{n-1} a_n + 0. \end{aligned}$$

Dann gilt $\mathrm{ggT}(a_0, a_1) = a_n$.

Beweis: Aus $a_i = q_i a_{i+1} + a_{i+2}$ folgt mit der Vorbemerkung $\mathrm{ggT}(a_i, a_{i+1}) = \mathrm{ggT}(a_{i+1}, a_{i+2})$ und damit

$$\mathrm{ggT}(a_0, a_1) = \mathrm{ggT}(a_1, a_2) = \cdots = \mathrm{ggT}(a_{n-1}, a_n) = a_n,$$

wie behauptet. ■

Wir haben damit also eine Möglichkeit gefunden, $\mathrm{ggT}(m,n)$ zu berechnen ohne die Primfaktorzerlegung von m und n zu kennen.

Beispiele:

(1) Wir wollen ggT(12345, 987) berechnen:

$$\begin{aligned} 12345 &= 12 \cdot 987 + 501, \\ 987 &= 1 \cdot 501 + 486, \\ 501 &= 1 \cdot 486 + 15, \\ 486 &= 32 \cdot 15 + 6, \\ 15 &= 2 \cdot 6 + 3, \\ 6 &= 2 \cdot 4 + 0, \end{aligned}$$

also gilt $\mathrm{ggT}(12345, 987) = 3$. Zum Vergleich: $12345 = 3 \cdot 5 \cdot 823$ und $987 = 3 \cdot 7 \cdot 47$.

(2) Was ist $\mathrm{ggT}(9264857236, 2453245253)$? Mit 23 Divisonen ergibt sich

$$\begin{aligned} 9264857236 &= 3 \cdot 2453245253 + 1905121477 \\ 2453245253 &= 1 \cdot 1905121477 + 548123776 \\ 1905121477 &= 3 \cdot 548123776 + 260750149 \\ 548123776 &= 2 \cdot 260750149 + 26623478 \\ 260750149 &= 9 \cdot 26623478 + 21138847 \\ 26623478 &= 1 \cdot 21138847 + 5484631 \\ 21138847 &= 3 \cdot 5484631 + 4684954 \\ 5484631 &= 1 \cdot 4684954 + 799677 \\ 4684954 &= 5 \cdot 799677 + 686569 \\ 799677 &= 1 \cdot 686569 + 113108 \\ 686569 &= 6 \cdot 113108 + 7921 \\ 113108 &= 14 \cdot 7921 + 2214 \\ 7921 &= 3 \cdot 2214 + 1279 \\ 2214 &= 1 \cdot 1279 + 935 \\ 1279 &= 1 \cdot 935 + 344 \\ 935 &= 2 \cdot 344 + 247 \\ 344 &= 1 \cdot 247 + 97 \\ 247 &= 2 \cdot 97 + 53 \\ 97 &= 1 \cdot 53 + 44 \\ 53 &= 1 \cdot 44 + 9 \\ 44 &= 4 \cdot 9 + 8 \\ 9 &= 1 \cdot 8 + 1 \\ 8 &= 8 \cdot 1 + 0 \end{aligned}$$

Also ist der ggT 1.

(3) Mit dem euklidischen Algorithmus erhalten wir in 79 Schritten

$$\mathrm{ggT}(17^{60} + 20882693916, 13^{40} + 14609017703) = 25937424629.$$

(4) Für die 20 stelligen Zahlen 15847523452462634165 und 87648572364875263842 erhält man den ggT nach 40 Divisionen

$$\mathrm{ggT}(15847523452462634165, 87648572364875263842) = 1$$

wie folgt:

$$
\begin{aligned}
1584752345246263 4165 &= 0 \cdot 8764857236487526 3842 + 1584752345246263 4165\\
8764857236487526 3842 &= 5 \cdot 1584752345246263 4165 + 841095510256209 3017\\
1584752345246263 4165 &= 1 \cdot 841095510256209 3017 + 7436568349900541148\\
841095510256209 3017 &= 1 \cdot 7436568349900541148 + 974386752661551869\\
7436568349900541148 &= 7 \cdot 974386752661551869 + 615861081269678065\\
974386752661551869 &= 1 \cdot 615861081269678065 + 358525671391873804\\
615861081269678065 &= 1 \cdot 358525671391873804 + 257335409877804261\\
358525671391873804 &= 1 \cdot 257335409877804261 + 101190261514069543\\
257335409877804261 &= 2 \cdot 101190261514069543 + 54954886849665175\\
101190261514069543 &= 1 \cdot 54954886849665175 + 46235374664404368\\
54954886849665175 &= 1 \cdot 46235374664404368 + 8719512185260807\\
46235374664404368 &= 5 \cdot 8719512185260807 + 2637813738100333\\
8719512185260807 &= 3 \cdot 2637813738100333 + 806070970959808\\
2637813738100333 &= 3 \cdot 806070970959808 + 219600825220909\\
806070970959808 &= 3 \cdot 219600825220909 + 147268495297081\\
219600825220909 &= 1 \cdot 147268495297081 + 72332329923828\\
147268495297081 &= 2 \cdot 72332329923828 + 2603835449425\\
72332329923828 &= 27 \cdot 2603835449425 + 2028772789353\\
2603835449425 &= 1 \cdot 2028772789353 + 575062660072\\
2028772789353 &= 3 \cdot 575062660072 + 303584809137\\
575062660072 &= 1 \cdot 303584809137 + 271477850935\\
303584809137 &= 1 \cdot 271477850935 + 32106958202\\
271477850935 &= 8 \cdot 32106958202 + 14622185319\\
32106958202 &= 2 \cdot 14622185319 + 2862587564\\
14622185319 &= 5 \cdot 2862587564 + 309247499\\
2862587564 &= 9 \cdot 309247499 + 79360073\\
309247499 &= 3 \cdot 79360073 + 71167280\\
79360073 &= 1 \cdot 71167280 + 8192793\\
71167280 &= 8 \cdot 8192793 + 5624936\\
8192793 &= 1 \cdot 5624936 + 2567857\\
5624936 &= 2 \cdot 2567857 + 489222\\
2567857 &= 5 \cdot 489222 + 121747\\
489222 &= 4 \cdot 121747 + 2234\\
121747 &= 54 \cdot 2234 + 1111\\
2234 &= 2 \cdot 1111 + 12\\
1111 &= 92 \cdot 12 + 7\\
12 &= 1 \cdot 7 + 5\\
7 &= 1 \cdot 5 + 2\\
5 &= 2 \cdot 2 + 1\\
2 &= 2 \cdot 1 + 0
\end{aligned}
$$

Wir wollen abschätzen, wie schnell der euklidische Algorithmus ist. Dazu brauchen wir ein paar Aussagen über Fibonacci-Zahlen.

3. Fibonacci-Zahlen

Die Fibonacci-Zahlen werden rekursiv durch $f_0 = 0$, $f_1 = 1$ und $f_n = f_{n-1} + f_{n-2}$ für $n \geq 2$ definiert. Die Folge ist also

$$0, \quad 1, \quad 1, \quad 2, \quad 3, \quad 5, \quad 8, \quad 13, \quad 21, \quad 34, \quad 55, \quad \ldots$$

Das folgende Lemma gibt eine explizite Gestalt für f_n an, sowie eine Abschätzung für das Wachstumsverhalten, die später benötigt wird.

LEMMA. *Für die Fibonacci-Zahlen f_n gilt ($n \geq 0$):*

(1)
$$f_n = \frac{1}{\sqrt{5}}[(\frac{1+\sqrt{5}}{2})^n - (\frac{1-\sqrt{5}}{2})^n].$$

(2)
$$n \leq 4.785 \log_{10} f_{n+2}.$$

Beweis:

(1) Man kann die explizite Gestalt auch durch Induktion beweisen. Wir geben hier aber einen Ansatz an, der auf die explizite Gestalt führt.
Sei α eine reelle Zahl $\neq 0$. Wann erfüllt $g_n = \alpha^n$ die Rekursionsformel $g_n = g_{n-1} + g_{n-2}$? Genau dann, wenn $\alpha^2 = \alpha + 1$, d.h. wenn $\alpha = \frac{1 \pm \sqrt{5}}{2}$ gilt. Wir setzen

$$\alpha = \frac{1+\sqrt{5}}{2}, \quad \beta = \frac{1-\sqrt{5}}{2}.$$

Für reelle Zahlen x, y erfüllt dann auch

$$h_n = x\alpha^n + y\beta^n$$

die Rekursionsformel $h_n = h_{n-1} + h_{n-2}$. Nun ist

$$h_0 = x + y, \quad h_1 = (x+y)\frac{1}{2} + (x-y)\frac{\sqrt{5}}{2}.$$

Wählen wir $y = -x$, so wird $h_0 = 0$ und $h_1 = x\sqrt{5}$. Wählen wir weiter $x = \frac{1}{\sqrt{5}}$, so wird auch noch $h_1 = 1$. Aus $h_0 = 0 = f_0$, $h_1 = 1 = f_1$ und der gleichen Rekursionsformel folgt nun sofort $f_n = h_n$ für alle $n \geq 0$, also

$$f_n = \frac{1}{\sqrt{5}}[(\frac{1+\sqrt{5}}{2})^n - (\frac{1-\sqrt{5}}{2})^n],$$

wie behauptet.

(2) Wegen

$$\frac{1+\sqrt{5}}{2} \approx 1.618, \quad \frac{1-\sqrt{5}}{2} \approx -0.618$$

sieht man, daß für große n gilt

$$f_n \approx \frac{1}{\sqrt{5}}(\frac{1+\sqrt{5}}{2})^n,$$

woraus eine Abschätzung der gewünschten Form folgt. Die Details mache man als Übung. Zum Vergleich:

n	0	1	2	3	4	5	6	10	1000	1001
$4.785 \log_{10} f_{n+2} n$	0	1.44	2.28	3.34	4.32	5.33	6.33	10.33	1000.33	1000.33

Dies veranschaulicht auch die Güte der Abschätzung. ■

Beispiel: Mit der Formel findet man

$$f_{480} \approx 0.92 \cdot 10^{100}, \quad f_{479} \approx 0.57 \cdot 10^{100},$$

d.h. f_{479} und f_{480} sind 100-stellige Dezimalzahlen.

Wir wenden jetzt die Aussagen über Fibonacci-Zahlen an um folgenden Satz zu beweisen:

SATZ. *Um den ggT zweier natürlicher Zahl $a > b$ zu berechnen, braucht man mit dem euklidischen Algorithmus*

$$\leq 4.785 \log_{10} a$$

Divisionen mit Rest.

Beweis: Die Schrittzahl zur Berechnung des ggT sei n. Wir schreiben dann $a = g_{n+1}$, $b = g_n$ und

$$\begin{aligned} g_{n+1} &= q_{n+1}g_n + g_{n-1}, \quad 1 \leq g_{n-1} < g_n \\ g_n &= q_n g_{n-1} + g_{n-2}, \quad 1 \leq g_{n-2} < g_{n-2} \\ &\vdots \qquad\quad \vdots \\ g_3 &= q_3 g_2 + g_1, \quad 1 \leq g_1 < g_2 \\ g_2 &= q_2 g_1 + g_0 \text{ mit } g_0 = 0 \text{ und } g_1 = \mathrm{ggT}(a,b). \end{aligned}$$

Wir wollen die g_i's mit der Fibonacci-Folge $f_0 = 0, f_1 = 1, f_2 = 1, f_3 = 2, \ldots, f_i = f_{i-1} + f_{i-2}$ vergleichen. Wir zeigen zunächst $g_i \geq f_{i+1}$ für $i \geq 1$ durch Induktion: Für $i = 1$ ist $g_1 = \mathrm{ggT}(a,b) \geq 1 = f_2$. Für $i = 2$ gilt $g_2 > g_1 \geq 1$, also $g_2 \geq 2 = f_3$. Wir machen jetzt den Induktionsschluß für $i \geq 3$, wobei wir $q_i \geq 1$ benutzen:

$$g_i = q_i g_{i-1} + g_{i-2} \geq g_{i-1} + g_{i-2} \geq f_i + f_{i-1} = f_{i+1},$$

womit $g_i \geq f_{i+1}$ für $i \geq 1$ durch Induktion bewiesen wäre. Es folgt:

$$a = g_{n+1} \geq f_{n+2}$$

und daher mit obigem Lemma

$$n \leq 4.785 \log_{10} f_{n+2} \leq 4.785 \log_{10} a,$$

was die Behauptung zeigt. ■

Der ggT läßt sich mit dem euklidischen Algorithmus also sehr schnell berechnen. Daher wird dieser Algorithmus in der Praxis auch oft eingesetzt.

Beispiel: Die Maple-Funktion 'ifactor' beginnt die Faktorisierung bzw. den Faktorisierungsversuch einer natürlichen Zahl n mit folgenden Schritten:

(1) Zunächst wird immer wieder $\mathrm{ggT}(n, 720720)$ (mit $720720 = 2^4 \cdot 3^2 \cdot 5 \cdot 7 \cdot 11 \cdot 13$) gebildet und aus n herausdividiert, bis $\mathrm{ggT}(n, 720720) = 1$ ist. Damit erhält man die Zerlegung

$$n = 2^{e_1} \cdot 3^{e_2} \cdot 5^{e_3} \cdot 7^{e_4} \cdot 11^{e_5} \cdot 13^{e_6} \cdot n',$$

wobei n' von keiner der Primzahlen 2, 3, 5, 7, 11, 13 geteilt wird.

(2) Sei

$$d = 17 \cdot 19 \cdot \cdots \cdot 1699 = p_7 \cdot p_8 \cdot \ldots p_{266}$$

das Produkt der 260 Primzahlen zwischen 17 und 1699 (d hat 718 Dezimalstellen). Nun wird solange $\mathrm{ggT}(n, d)$ gebildet und aus n herausdividiert, bis $\mathrm{ggT}(n, d) = 1$ ist. Man erhält dann die Zerlegung

$$n = 2^{e_1} \cdot 3^{e_2} \cdot \cdots \cdot 1699^{e_{266}} \cdot n'',$$

wobei n'' nun keinen Primteiler < 1700 mehr hat.

(3) Nun erst werden andere Faktorisierungsmethoden eingesetzt.

(Anscheinend geht das obige Verfahren mit der ggT-Bildung schneller als wenn man versucht, alle kleinen Primteiler zwischen 2 und 1699 einzeln herauszuteilen.)

4. Der erweiterte euklidische Algorithmus

Der folgende Satz gibt eine wichtige Eigenschaft des ggT an, die sowohl theoretisch als auch praktisch oft benutzt wird:

SATZ. *Zu $a, b \in \mathbf{Z}$ gibt es $x, y \in \mathbf{Z}$ mit*

$$\mathrm{ggT}(a, b) = xa + yb.$$

Ein konstruktiver Beweis wird mit folgendem Satz gegeben:

SATZ (Erweiterter euklidischer Algorithmus). *Zu $a_0, a_1 \in \mathbf{Z}$, $(a_0, a_1) \neq (0, 0)$, führe man den euklidischen Algorithmus durch mit Quotienten q_i:*

$$\begin{aligned}
a_0 &= q_0 a_1 + a_2, \quad 0 < a_2 < |a_1| \\
a_1 &= q_1 a_2 + a_3, \quad 0 < a_3 < a_2 \\
a_2 &= q_2 a_3 + a_4, \quad 0 < a_4 < a_3 \\
&\vdots \quad = \quad \vdots \\
a_{n-2} &= q_{n-2} a_{n-1} + a_n, \quad 0 < a_n < a_{n-1} \\
a_{n-1} &= q_{n-1} a_n + 0.
\end{aligned}$$

Setzt man $x_0 = 1$, $y_0 = 0$, $x_1 = 0$, $y_1 = 1$, definiert man für $i \geq 2$ rekursiv

$$x_i = x_{i-2} - q_{i-2} x_{i-1}, \quad y_i = y_{i-2} - q_{i-2} y_{i-1},$$

so gilt

$$a_i = x_i a_0 + y_i a_1$$

und insbesondere

$$\mathrm{ggT}(a_0, a_1) = a_n = x_n a_0 + y_n a_1.$$

Beweis: Wir beweisen dies durch Induktion. Für $i = 0$ und $i = 1$ ist die Behauptung klar. Für $i \geq 2$ gilt dann mit Induktionsannahme für $i - 1$ und $i - 2$

$$\begin{aligned}
a_i &= a_{i-2} - q_{i-2} a_{i-1} = \\
&= (x_{i-2} a_0 + y_{i-2} a_1) - q_{i-2}(x_{i-1} a_0 + y_{i-1} a_1) = \\
&= (x_{i-2} - q_{i-2} x_{i-1}) a_0 + (y_{i-2} - q_{i-2} y_{i-1}) a_1 = \\
&= x_i a_0 + y_i a_1,
\end{aligned}$$

was zu zeigen war. ■

Bemerkungen:

(1) Die Formeln zeigen, daß man nicht die Gesamtfolgen x_i und y_i speichern muß um schließlich x_n, y_n zu erhalten. Man kommt also mit wenig Speicherplatz aus.
(2) Es gibt noch weitere Methoden, den erweiterten euklidischen Algorithmus durchzuführen.

Beispiele:

(1) Für 12345 und 987 erhält man

i	a_i	x_i	y_i
0	12345	1	0
1	987	0	1
2	501	1	-12
3	486	-1	13
4	15	2	-25
5	6	-65	813
6	3	132	-1651

und damit

$$3 = \mathrm{ggT}(12345, 987) = 132 \cdot 12345 - 1651 \cdot 987.$$

(2) Für die folgenden Zahlen a_0 und a_1 erhält man

$$\begin{aligned} a_0 &= 671328806001012829487353559941943176207647465878611669861215642695935787 17 \\ a_1 &= 361188648084531445929920877641340171153335304 \\ \mathrm{ggT}(a_0, a_1) &= 25937424629 = \\ &= -552494476397141985810951662144031 1a_0 + \\ &\quad +102690231027150819183055632608710042906390192138417613845795985 4a_1 \end{aligned}$$

(3) Für die 20-stelligen Zahlen $a_0 = 15847523452462634165$ und $a_1 = 87648572364875263842$ erhält man

i	a_i	x_i	y_i
0	15847523452462634165	1	0
1	87648572364875263842	0	1
2	15847523452462634165	1	0
3	8410955102562093017	-5	1
4	7436568349900541148	6	-1
5	974386752661551869	-11	2
6	615861081269678065	83	-15
7	358525671391873804	-94	17
8	257335409877804261	177	-32
9	101190261514069543	-271	49
10	54954886849665175	719	-130
11	46235374664404368	-990	179
12	8719512185260807	1709	-309
13	2637813738100333	-9535	1724
14	806070970959808	30314	-5481
15	219600825220909	-100477	18167
16	147268495297081	331745	-59982
17	72332329923828	-432222	78149
18	2603835449425	1196189	-216280
19	2028772789353	-32729325	5917709
20	575062660072	33925514	-6133989
21	303584809137	-134505867	24319676
22	271477850935	168431381	-30453665
23	32106958202	-302937248	54773341
24	14622185319	2591929365	-468640393
25	2862587564	-5486795978	992054127
26	309247499	30025909255	-5428911028
27	79360073	-275719979273	49852253379
28	71167280	857185847074	-154985671165
29	8192793	-1132905826347	204837924544
30	5624936	9920432457850	-1793689067517
31	2567857	-11053338284197	1998526992061
32	489222	32027109026244	-5790743051639
33	121747	-171188883415417	30952242250256
34	2234	716782642687912	-129599712052663
35	1111	-38877451588562665	7029336693094058
36	12	78471685819813242	-14188273098240779
37	7	-725827254701138092 9	1312350461731245726
38	5	7336744232831194171	-1326538734829486505
39	2	-14595016779842575100	2638889196560732231
40	1	36526777792516344371	-6604317127950950967

Dann gilt also

$$1 = \mathrm{ggT}(a_0, a_1) = 36526777792516344371a_0 - 6604317127950950967a_1.$$

Im folgenden werden wir noch einige Anwendungen des erweiterten euklidischen Algorithmus geben.

5. Die Gleichung $ax + by = c$

Wir wissen, daß die Gleichung $\mathrm{ggT}(a, b) = ax + by$ lösbar ist. Der folgende Satz verallgemeinert diese Gleichung und gibt die Gesamtheit der Lösungen an:

SATZ. *Seien $a, b \in \mathbf{Z}$, $(a, b) \neq (0, 0)$. Dann gilt:*

- *Die Gleichung $ax + by = c$ ist genau dann lösbar, wenn $\mathrm{ggT}(a,b)|c$ gilt.*
- *Gilt $\mathrm{ggT}(a,b)|c$ und sind $u, v \in \mathbf{Z}$ mit $au + bv = \mathrm{ggT}(a,b)$, so ist*

$$\{(x,y) \in \mathbf{Z} \times \mathbf{Z} : ax + by = c\} = \{(\frac{c}{\mathrm{ggT}(a,b)}u + \frac{b}{\mathrm{ggT}(a,b)}m, \frac{c}{\mathrm{ggT}(a,b)}v - \frac{a}{\mathrm{ggT}(a,b)}m) : m \in \mathbf{Z}\}.$$

Beweis: Sei $d = \mathrm{ggT}(a,b)$, $a = da'$, $b = db'$ und $au + bv = d$.

(1) Sind $x, y \in \mathbf{Z}$ mit $c = ax + by$, so folgt $c = d(a'x + b'y)$, was sofort die Behauptung $d|c$ ergibt. Sei nun umgekehrt $d|c$, also $c = de$. Dann ist $a(ue) + b(ve) = de = c$, die Gleichung $ax + by = c$ also lösbar. Anders geschrieben:

$$(\frac{c}{\mathrm{ggT}(a,b)}u, \frac{c}{\mathrm{ggT}(a,b)}v)$$

löst die Gleichung.

(2) Man sieht leicht, daß $\supseteq$ gilt. Sei jetzt $ax + by = c$. Dann ist $ax + by = a(ue) + b(ve)$, also $a(x - ue) = b(ve - y)$ bzw. $a'(x - ue) = b'(ve - y)$. Wegen $b'|a'(x - ue)$ und $\mathrm{ggT}(a', b') = 1$ folgt $b'|x - ue$, d.h. es existiert $m \in \mathbf{Z}$ mit $x - ue = mb'$. Einsetzen liefert $ve - y = ma'$. Damit folgt $\subseteq$. ■

FOLGERUNG. *Sind $a, b \in \mathbf{Z}$ mit $\mathrm{ggT}(a,b) = 1$, so gibt es $x_0, y_0 \in \mathbf{Z}$ mit $ax_0 + by_0 = 1$ und es gilt*

$$\{(x,y) \in \mathbf{Z} \times \mathbf{Z} : ax + by = 1\} = \{(x_0 + bm, y_0 - am : m \in \mathbf{Z}\}.$$

6. Kongruenzen

DEFINITION. Für $a, b, m \in \mathbf{Z}$ sagen wir, a ist kongruent b modulo m, in Zeichen $a \equiv b \bmod m$, falls $m|a - b$ gilt.

Beispiele:

(1) $n \equiv 0 \bmod 2$, falls n gerade, $n \equiv 1 \bmod 2$, falls n ungerade.
(2) Gibt a durch m geteilt Rest r, d.h. $a = qm + r$, so ist $a \equiv r \bmod m$.
(3) Genau dann gilt $a \equiv 0 \bmod m$, wenn $m|a$ gilt.

Einige Grundaussagen über Kongruenzen sind in folgendem Satz zusammengestellt:

SATZ. *Sei $m \geq 1$.*

(1) *Durch $a \equiv b \bmod m$ wird eine Äquivalenzrelation definiert, die Äquivalenzklasse von a wird auch mit $\overline{a}$ bezeichnet (bei festem m), also: $a \equiv b \bmod m \iff \overline{a} = \overline{b}$.*
(2) *Die Menge der Äquivalenzklassen wird mit $\mathbf{Z}/(m)$ (oder $\mathbf{Z}/m\mathbf{Z}$ oder $\mathbf{Z}/m$) bezeichnet. Als Repräsentanten kann man $0, 1, 2, \ldots, m-1$ wählen, d.h.*

$$\mathbf{Z}/(m) = \{\overline{0}, \overline{1}, \overline{2}, \ldots, \overline{m-1}\}$$

und damit $\#\mathbf{Z}/(m) = m$.

(3) *Die Äquivalenzrelation $a \equiv b \bmod m$ ist mit Addition und Multiplikation verträglich, d.h. aus $a_1 \equiv a_2 \bmod m$ und $b_1 \equiv b_2 \bmod m$ folgt*

$$a_1 + b_1 \equiv a_2 + b_2 \bmod m \quad \text{und} \quad a_1 b_1 \equiv a_2 b_2 \bmod m.$$

Durch die Definition

$$\overline{a} + \overline{b} = \overline{a+b} \quad \text{und} \quad \overline{a}\,\overline{b} = \overline{ab}$$

wird daher $\mathbf{Z}/(m)$ zu einem kommutativen Ring mit Null $\overline{0}$ und Eins $\overline{1}$.

In der Zahlentheorie wird sehr oft mit Kongruenzen gerechnet.

Beispiele:

(1) Wie sieht man einer Dezimalzahl an, ob sie durch 3 teilbar ist? Die Dezimalschreibweise $n = (n_d n_{d-1} \ldots n_1 n_0)_{10}$ bedeutet

$$n = n_0 + n_1 \cdot 10 + n_2 \cdot 100 + \cdots + n_d \cdot 10^d.$$

Nun ist $10 \equiv 1 \bmod 3$ und damit auch $10^i \equiv 1 \bmod 3$, d.h.

$$n \equiv n_0 + n_1 + \cdots + n_d \bmod 3.$$

Also ist n genau dann durch 3 teilbar, wenn $n \equiv n_0 + \cdots + n_d \equiv 0 \bmod 3$ gilt, d.h. wenn die Quersumme $n_0 + \cdots + n_d$ durch 3 teilbar ist. (Auf gleiche Weise zeigt man die bekannten Teilbarkeitsregeln für 9 und 11.)

(2) Sei $e = (111\ldots111)_{10}$ die Zahl mit 100 Einsen in der Dezimaldarstellung. Was ist die letzte Ziffer in der Dezimaldarstellung von 3^e? Uns interessiert also $3^e \bmod 10$. Es gilt

$$3^1 \equiv 3 \bmod 10, \quad 3^2 \equiv 9 \bmod 10, \quad 3^3 \equiv 7 \bmod 10, \quad 3^4 \equiv 1 \bmod 10.$$

Wir schreiben $e = 4q + r$, dann ist nämlich $3^e = 3^{4q+r} = 81^q \cdot 3^r \equiv 3^r \bmod 10$. Nun ist

$$r \equiv e = 100 \cdot \text{Zahl} + 11 \equiv 3 \bmod 4,$$

also $r = 3$ und damit

$$3^e \equiv 3^r = 3^3 \equiv 7 \bmod 10.$$

Die letzte Ziffer von 3^e ist also 7.

LEMMA. (1) *Gilt $a \equiv b \bmod m$ und $d|m$, so gilt auch $a \equiv b \bmod d$.*

(2) *Gilt $a \equiv b \bmod m$, $a \equiv b \bmod n$ und $\mathrm{ggT}(m,n) = 1$, so folgt $a \equiv b \bmod mn$.*

Beweis:

(1) $a \equiv b \bmod m$ bedeutet $m|(a-b)$, so daß mit $d|m$ sofort $d|(a-b)$ und damit $a \equiv b \bmod d$ folgt.

(2) Die Kongruenzen bedeuten $m|(a-b)$ und $n|(a-b)$. Wegen $\mathrm{ggT}(m,n) = 1$ folgt $mn|(a-b)$ und damit die Behauptung. ■

Wir betrachten eine weitere Gleichung:

SATZ. *Genau dann ist die Kongruenz $ax \equiv 1 \bmod m$ lösbar, wenn $\mathrm{ggT}(a,m) = 1$ gilt. Ist $\mathrm{ggT}(a,m) = 1$, so findet man mit dem erweiterten euklidischen Algorithmus $x, y \in \mathbf{Z}$ mit $ax + my = 1$, also ist $ax \equiv 1 \bmod m$.*

Beweis: Sei $d = \mathrm{ggT}(a,m)$. Gibt es eine Lösung der Kongruenz, d.h. hat man $ax \equiv 1 \bmod m$, so folgt $0 \equiv ax \equiv 1 \bmod d$ und damit $d = 1$. Sei umgekehrt $d = 1$. Mit dem erweiterten euklidischen Algorithmus findet man x, y mit $ax + my = 1$ und damit $ax \equiv 1 \bmod m$. ■

Bemerkungen:

(1) Gilt $ab \equiv 1 \bmod m$, so schreibt man auch $b \equiv \frac{1}{a} \bmod m$.

(2) Will man mit Maple eine Lösung der Gleichung $ax \equiv 1 \bmod m$ finden, so berechnet man einfach '1/a mod m'.

Bevor wir eine andere Interpretation des Satzes geben, erinnern wir an einen Begriff aus der Algebra: Ist R ein Ring mit Eins 1, so heißt $u \in R$ eine Einheit, falls $v \in R$ existiert mit $uv = vu = 1$. Die Menge der Einheiten bildet bzgl. Multiplikation eine Gruppe, die mit R^* bezeichnet wird.
Sei jetzt $m \in \mathbf{N}$ gegeben und $a \in \mathbf{Z}$. Dann ist $\overline{a}$ genau dann Einheit in $\mathbf{Z}/(m)$, wenn die Gleichung $ax \equiv 1 \bmod m$ eine Lösung hat. Mit dem Satz folgt daher:

FOLGERUNG. *Für die Einheiten von $\mathbf{Z}/(m)$ gilt:*

$$(\mathbf{Z}/(m))^* = \{\overline{a} : 0 \le a \le m-1, \mathrm{ggT}(a,m) = 1\}.$$

DEFINITION. Die Eulersche φ-Funktion wird definiert durch

$$\varphi(n) = \#\{a : 0 \le a \le n-1, \mathrm{ggT}(a,n) = 1\} = \#(\mathbf{Z}/(n))^*.$$

Wir geben eine kleine Tabelle:

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
$\varphi(n)$	1	1	2	2	4	2	6	4	6	4	10	4	12	6	8	8	16	6	18	8

SATZ. *Für $n \in \mathbf{N}$ gelten die Äquivalenzen:*

$$n \text{ ist Primzahl} \iff \varphi(n) = n-1 \iff \mathbf{Z}/(n) \text{ ist Körper.}$$

Beweis: O.E. $n \ge 2$.

(1) Genau dann gilt $\varphi(n) = n-1$, wenn n keinen nichttrivialen Teiler zwischen 2 und $n-1$ hat, was aber gleichwertig damit ist, daß n eine Primzahl ist.

(2) $\mathbf{Z}/(n)$ ist genau dann ein Körper, wenn sich alle Elemente $\neq 0$ invertieren lassen, d.h. wenn gilt $\#(\mathbf{Z}/(n))^* = \#\mathbf{Z}/(n) - 1$, was nach 1. genau dann der Fall ist, wenn n eine Primzahl ist. ■

Bemerkungen:

(1) Ist p eine Primzahl, so schreiben wir statt $\mathbf{Z}/(p)$ auch $\mathbf{F}_p$.
(2) Ist n keine Primzahl, $n = ab$ mit $1 < a, b < n$, so gilt in $\mathbf{Z}/(n)$ natürlich $\overline{a} \cdot \overline{b} = \overline{0}$, $\overline{a} \neq \overline{0}$, $\overline{b} \neq \overline{0}$, d.h. $\mathbf{Z}/(n)$ ist kein Integritätsring.

SATZ. *Für die Eulersche φ-Funktion gilt:*

(1) *Sind m, n natürliche Zahlen, so gilt*

$$\mathrm{ggT}(m,n) = 1 \quad \Longrightarrow \quad \varphi(mn) = \varphi(m)\varphi(n).$$

(Zahlentheoretische Funktionen mit dieser Eigenschaft nennt man multiplikativ.)
(2) *Sei $n = p_1^{e_1} \dots p_r^{e_r}$ (alle p_i verschieden):*

$$\varphi(n) = \prod_i p_i^{e_i - 1}(p_i - 1) = n(1 - \frac{1}{p_1}) \dots (1 - \frac{1}{p_r}) = n \prod_{p|n} (1 - \frac{1}{p}).$$

Beweis:

(1) Dies wird später bewiesen werden.
(2) Für eine Primzahl p gilt

$$\begin{aligned} \{0 \leq a \leq p^e - 1 : \mathrm{ggT}(a, p^e) = 1\} &= \{0 \leq a \leq p^e - 1 : \mathrm{ggT}(a,p) = 1\} = \\ &= \{a : 0 \leq a \leq p^e - 1\} \setminus \{pb : 0 \leq b \leq p^{e-1} - 1\}, \end{aligned}$$

woraus $\varphi(p^e) = p^e - p^{e-1} = p^{e-1}(p-1) = p^e(1 - \frac{1}{p})$ folgt. Mit Teil 1 ergibt sich daraus

$$\begin{aligned} \varphi(n) &= \varphi(\prod_i p_i^{e_i}) = \prod_i \varphi(p_i^{e_i}) = \prod_i p_i^{e_i - 1}(p_i - 1) = \\ &= \prod_i p_i^{e_i}(1 - \frac{1}{p_i}) = n \prod_i (1 - \frac{1}{p_i}), \end{aligned}$$

was zu zeigen war. ■

Bemerkung: Der letzte Satz zeigt, daß man $\varphi(n)$ leicht berechnen kann, wenn man die Primfaktorzerlegung von n kennt. Heutzutage ist für große n keine andere Berechnungsmöglichkeit bekannt.

7. Der chinesische Restsatz

Der folgende Satz spielt ebenfalls eine wichtige Rolle. Wir geben ihn in der meist gebrauchten Form an:

SATZ (Chinesischer Restsatz). *Seien $m_1, m_2, \dots, m_r \in \mathbf{N}$ mit $\mathrm{ggT}(m_i, m_j) = 1$ für alle $i \neq j$ und $a_1, \dots, a_r \in \mathbf{Z}$ gegeben. Dann gibt es ein $x \in \mathbf{Z}$ mit*

$$\begin{aligned} x &\equiv a_1 \bmod m_1, \\ x &\equiv a_2 \bmod m_2, \\ &\vdots \quad\quad \vdots \\ x &\equiv a_r \bmod m_r. \end{aligned}$$

x ist eindeutig bestimmt modulo $m_1 m_2 \dots m_r$.
Explizit: Sei $M = \prod_i m_i$ und $M_i = \frac{M}{m_i}$. Berechne $N_i \equiv \frac{1}{M_i} \bmod m_i$. Dann löst

$$x = \sum_i a_i M_i N_i$$

das Kongruenzgleichungssystem.

Beweis:

(1) Da m_i nach Voraussetzung teilerfremd zu $M_i = m_1 m_2 \dots m_{i-1} m_{i+1} \dots m_r$ ist, gibt es ein N_i mit $M_i N_i \equiv 1 \bmod m_i$, z.B. als Lösung x der Gleichung $M_i x + m_i y = 1$ mit dem erweiterten euklidischen Algorithmus. Wir wollen zeigen, daß

$$x = \sum_j a_j M_j N_j$$

alle Kongruenzen erfüllt. Dazu betrachten wir $x \bmod m_i$. Für $j \neq i$ ist $M_j \equiv 0 \bmod m_i$ und damit

$$x \equiv a_i M_i N_i \equiv a_i(1 - m_i n_i) \equiv a_i \bmod m_i$$

für alle i, was die Behauptung beweist.

(2) Ist x' eine weitere Lösung des Kongruenzensystems, so folgt $x' \equiv a \equiv x \bmod m_i$ und damit $m_i | (x' - x)$. Die Teilerfremdheit der m_i's liefert $m_1 m_2 \dots m_r | x' - x$, also $x' \equiv x \bmod m_1 m_2 \dots m_r$, wie behauptet. ■

Der Beweis ist konstruktiv und kann zur Konstruktion einer Lösung benutzt werden. In Maple gibt es die Funktion chrem($[a_1, \dots, a_r], [m_1, \dots, m_r]$), mit der man eine Lösung erhält.

Beispiele:

(1) Suche eine Lösung des Systems

$$x \equiv 1 \bmod 2, \quad x \equiv 0 \bmod 3, \quad x \equiv 4 \bmod 5.$$

Die Lösung ist bestimmt modulo 30. Durch Probieren findet man $x \equiv 9 \bmod 30$.

(2) Wir starten mit

$$m_1 = 2, m_2 = 3, m_3 = 5, m_4 = 7, m_5 = 11, m_6 = 13, m_7 = 17, m_8 = 19, m_9 = 23, m_{10} = 29$$

und

$$a_1 = 1, a_2 = 2, a_3 = 3, a_4 = 4, a_5 = 5, a_6 = 6, a_7 = 7, a_8 = 8, a_9 = 0, a_{10} = 10.$$

Mit $M = m_1 \dots m_{10} = 6469693230$, $M_i = \frac{M}{m_i}$ und $N_i = \frac{1}{M_i} \bmod m_i$ gilt

i	M_i	N_i
1	3234846615	1
2	2156564410	1
3	1293938646	1
4	924241890	3
5	588153930	1
6	497668710	11
7	380570190	4
8	340510170	9
9	281291010	11
10	223092870	12

woraus man mit der Formel $x = \sum a_i M_i N_i$ die Lösung $x = 148099250513$ bzw. modulo M die Lösung

$$x \equiv 5765999453 \bmod 6469693230$$

erhält.

Mitunter versteht man unter dem chinesischen Restsatz auch folgenden algebraischen Satz:

SATZ. *Sei $m_1, \dots, m_r \in \mathbf{N}$ mit $\mathrm{ggT}(m_i, m_j) = 1$ für $i \neq j$. Dann definiert*

$$\phi : \mathbf{Z}/(m_1 m_2 \dots m_r) \to \mathbf{Z}/(m_1) \times \mathbf{Z}/(m_2) \times \dots \times \mathbf{Z}/(m_r), \quad x \mapsto (x \bmod m_1, x \bmod m_2, \dots, x \bmod m_r)$$

einen Ringisomorphismus.

Beweisskizze: Man sieht zunächst, daß die Abbildung wohldefiniert ist, daß Definitions- und Bildbereich gleiche Mächtigkeit $m_1 \dots m_r$ haben. Man überlegt, daß ϕ mit den Rechenoperationen verträglich ist, d.h. ϕ ist ein Ringhomomorphismus. Der chinesische Restsatz liefert nun, daß die Abbildung surjektiv ist. Also ist ϕ auch bijektiv wegen der Mächtigkeitsaussagen. Dies beweist die Behauptung. ■

Wir können nun die Multiplikativität der Eulerschen φ-Funktion beweisen:

Beweis von $\varphi(mn) = \varphi(m)\varphi(n)$ *für* $\mathrm{ggT}(m,n) = 1$*:* Der letzte Satz liefert einen Ringisomorphismus

$$\phi : \mathbf{Z}/(mn) \to \mathbf{Z}/(m) \times \mathbf{Z}/(n).$$

Da ein Ringisomorphismus natürlich Einheiten in Einheiten überführt, folgt

$$(\mathbf{Z}/(m))^* \simeq (\mathbf{Z}/(m) \times \mathbf{Z}/(n))^* = (\mathbf{Z}/(m))^* \times (\mathbf{Z}/(n))^*,$$

sodaß man durch Betrachtung der Mächtigkeiten sofort die Behauptung $\varphi(mn) = \varphi(m)\varphi(n)$ erhält. ■

8. Übungen

Aufgabe 1.1: Für natürliche Zahlen a und b sei $\mu(a,b)$ die Anzahl der Divisionen mit Rest, die der euklidische Algorithmus zur Bestimmung von $\mathrm{ggT}(a,b)$ braucht, $\nu(a,b)$ sei die Anzahl der Divisionen, bei denen der Quotient 1 ist. (Beispiel: $\mu(23,8) = 3$, $\nu(23,8) = 1$ wegen $23 = 2 \cdot 8 + 7$, $8 = 1 \cdot 7 + 1$, $7 = 7 \cdot 1 + 0$.)
Untersuche experimentell die Größe $\frac{\nu(a,b)}{\mu(a,b)}$, d.h. den Anteil der Divisionen, bei denen der Quotient 1 ist.

Aufgabe 1.2: Seien a, b natürliche Zahlen und $\mu(a,b)$ die Funktion aus Aufgabe 1.

(1) In welcher Beziehung stehen $\mu(a,b)$ und $\mu(b,a)$?
(2) Wann gilt $\mu(a,b) = 1$?
(3) Wann gilt $\mu(a,b) = 2$?

Aufgabe 1.3: $(f_n)_{n\geq 0}$ sei die Folge der Fibonacci-Zahlen. Zeige, daß man n Divisionen mit Rest braucht, um mit dem euklidischen Algorithmus den $\mathrm{ggT}(f_{n+2}, f_{n+1})$ zu berechnen (für $n \geq 1$). Wieviele Divisionen braucht man für die Berechnung von $\mathrm{ggT}(f_{n+3}, f_{n+1})$?

Aufgabe 1.4: Der folgende Algorithmus (binärer euklidischer Algorithmus) berechnet den ggT zweier Zahlen $a, b \in \mathbf{N}_0$:

(1) Ist $a < b$, vertausche a und b. Ist $b = 0$, gib a aus und beende das Verfahren. Sonst setze $r := a \bmod b$, $a := b$, $b := r$.
(2) Ist $b = 0$, gib a aus und beende das Verfahren. Sonst setze $k := 0$ und führe folgende Schritte aus, solange a und b gerade sind: $k := k + 1$, $a := a/2$, $b := b/2$.
(3) Ist a gerade, wiederhole $a := a/2$ bis a ungerade ist. Andernfalls: Ist b gerade, wiederhole $b := b/2$ bis b ungerade ist.
(4) Setze $t := (a - b)/2$. Ist $t = 0$, gib $2^k a$ aus und beende das Verfahren.
(5) Wiederhole $t := t/2$ solange t gerade ist. Ist $t > 0$, setze $a := t$, andernfalls setze $b := -t$ und gehe zu Schritt 4.

Beweise, daß der Algorithmus tatsächlich $\mathrm{ggT}(a,b)$ berechnet, und berechne damit $\mathrm{ggT}(144, 89)$.

Aufgabe 1.5: Ein Integritätsring R heißt euklidisch bzgl. einer Funktion $\lambda : R \setminus \{0\} \to \mathbf{N}_0$, wenn es zu $a, b \in R$ mit $b \neq 0$ Elemente $q, r \in R$ gibt mit $r = 0$ oder $\lambda(r) < \lambda(b)$. ($\mathbf{Z}$ ist euklidisch bzgl. $\lambda(a) = |a|$.)
Zeige:

(1) Der Polynomring $\mathbf{Q}[x]$ ist euklidisch bgzl. der Gradfunktion $\lambda(f(x)) = \mathrm{Grad}(f(x))$.
(2) Der Ring der Gaußschen Zahlen

$$\mathbf{Z}[i] = \{m + ni \in \mathbf{C} : m, n \in \mathbf{Z}\}$$

(mit $i^2 = -1$) ist euklidisch bzgl. der Funktion $\lambda(m + ni) = m^2 + n^2$.

Aufgabe 1.6: Sei R euklidischer Ring bzgl. einer Funktion $\lambda : R \setminus \{0\} \to \mathbf{N}_0$.

(1) Definiert man $\widetilde{\lambda} : R \setminus \{0\} \to \mathbf{N}_0$ durch

$$\widetilde{\lambda}(a) = \min\{\lambda(ua) : u \in R^*\},$$

so ist R auch euklidisch bzgl. $\widetilde{\lambda}$. (Assoziierte Elemente haben dann gleichen $\widetilde{\lambda}$-Wert.)

(2) Sei $n_1 = \min\{\lambda(a) : a \in R \setminus \{0\}\}$. Dann gilt für $a \in R \setminus \{0\}$:

$$a \in R^* \quad \Longleftrightarrow \quad \widetilde{\lambda}(a) = n_1.$$

(3) Für $a, b \in R \setminus \{0\}$ mit $a|b$, $a \not\sim b$ gilt $\widetilde{\lambda}(a) < \widetilde{\lambda}(b)$.

Aufgabe 1.7: Sei $(f_n)_{n\geq 0}$ die Folge der Fibonacci-Zahlen ($f_0 = 0$, $f_1 = 1$ und $f_n = f_{n-1} + f_{n-2}$ für $n \geq 2$).

(1) Zeige, daß für $n \geq 0$ gilt

$$\begin{pmatrix} f_{n+1} \\ f_n \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

(2) $a_n \in \{0, 1, 2, \ldots, 9\}$ werde definiert durch die Kongruenz $a_n \equiv f_n \bmod 10$ für $n \geq 0$, d.h. a_n ist die letzte Ziffer der Dezimaldarstellung von f_n. Zeige, daß $(a_n)_{n\geq 0}$ eine periodische Folge ist. Was ist die Periode?

Aufgabe 1.8: Was sind die letzten drei Ziffern in der Dezimaldarstellung von

$$2^{3^{5^7}}?$$

Aufgabe 1.9: Sei $M = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19$. Zeige, daß es in jedem Intervall ganzer Zahlen von $n+1$ bis $n+M$ gleich viele Zahlen gibt, die durch keine der Primzahlen 2, 3, 5, ..., 19 teilbar sind. Bestimme diese Anzahl m. Zeige: $\frac{m}{M} \approx \frac{1}{6}$. (Gauß interpretiert dies wie folgt: '...weil sich, allgemein zu reden, unter sechs Zahlen kaum eine findet, die nicht durch eine der Zahlen 2, 3, 5, ..., 19 teilbar wäre'.)

Aufgabe 1.10: Für

$$n = 1790120927262220174291846935152371144285517950319593720062608831435051$$

gilt

$$\varphi(n) = 1790120927262220174291846935152351911399951523273003793212000453283600.$$

Bestimme die Primfaktorzerlegung von n.

Aufgabe 1.11: Zeige, daß die einzige ganzzahlige Lösung der Gleichung

$$x_1^3 + 2x_2^3 + 7x_3^3 + 14x_4^3 = 0$$

$(x_1, x_2, x_3, x_4) = (0, 0, 0, 0)$ ist. (Hinweis: Betrachtung modulo p.)

Aufgabe 1.12: Bestimme die kleinste natürliche Zahl, die die Kongruenzgleichungen

$$x \equiv 1 \bmod 2, \quad x \equiv 2 \bmod 3, \quad x \equiv 3 \bmod 5, \quad x \equiv 5 \bmod 7, \quad x \equiv 7 \bmod 11, \quad x \equiv 11 \bmod 13$$

erfüllt.

Aufgabe 1.13: Bestimme alle ganzzahligen Lösungen des Gleichungssystems

$$\begin{aligned} 3x + 17y &\equiv 5 \bmod 26, \\ 5x + 23y &\equiv 13 \bmod 26. \end{aligned}$$

KAPITEL 2

Schnelle Exponentiation - Die square-and-multiply-Methode

In diesem Kapitel soll ein Algorithmus vorgestellt werden, der von fundamentaler praktischer Bedeutung ist. Dazu werden einige Anwendungen gegeben.

1. Die Sätze von Euler und Fermat

SATZ. *Ist $n \in \mathbf{N}$ und $a \in \mathbf{Z}$, so gilt*

$$\mathrm{ggT}(a,n) = 1 \quad \Longrightarrow \quad a^{\varphi(n)} \equiv 1 \bmod n.$$

Beweis: Die Einheitengruppe $(\mathbf{Z}/(n))^*$ des Rings $\mathbf{Z}/(n)$ hat Ordnung $\varphi(n)$. Nun gilt in jeder endlichen (multiplikativ geschriebenen) Gruppe G für jedes Element $g \in G$

$$g^{\#G} = 1,$$

wenn 1 das neutrale Element bezeichnet. In unserer Situation bedeutet dies

$$\overline{a} \in (\mathbf{Z}/(n))^* \quad \Longrightarrow \quad \overline{a}^{\varphi(n)} = \overline{1}.$$

Übersetzt man dies in Kongruenzen, so entsteht

$$a \in \mathbf{Z}, \mathrm{ggT}(a,n) = 1 \quad \Longrightarrow \quad a^{\varphi(n)} \equiv 1 \bmod n,$$

was zu zeigen war. ■

Beispiele:

(1) Für $n = 10$ ist $\varphi(n) = \varphi(2 \cdot 5) = 4$.

a	0	1	2	3	4	5	6	7	8	9
$a^4 \bmod 10$	0	1	6	1	6	5	6	1	6	1

(2) Für $n = 11$ ist $\varphi(n) = \varphi(11) = 10$.

a	0	1	2	3	4	5	6	7	8	9	10
$a^{10} \bmod 11$	0	1	1	1	1	1	1	1	1	1	1

(3) Für $n = 12$ ist $\varphi(n) = \varphi(2^2 \cdot 3) = 2 \cdot 2 = 4$.

a	0	1	2	3	4	5	6	7	8	9	10	11
$a^4 \bmod 12$	0	1	4	9	4	1	0	1	4	9	4	1

Ein wichtiger Spezialfall ergibt sich, wenn $n = p$ eine Primzahl ist:

SATZ (Kleiner Satz von Fermat). *Ist p eine Primzahl und a eine ganze Zahl, so gilt*

$$\mathrm{ggT}(a,p) = 1 \quad \Longrightarrow \quad a^{p-1} \equiv 1 \bmod p.$$

Eine andere Formulierung ist

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

Beweis: Für eine Primzahl p ist $\varphi(p) = p-1$, so daß die erste Behauptung ein Spezialfall des Satzes von Euler ist. Zur zweiten Formulierung: Ist $\mathrm{ggT}(a,p) = 1$, so ist $a^{p-1} \equiv 1 \bmod p$, Multiplikation mit a ergibt $a^p \equiv a \bmod p$. Ist $\mathrm{ggT}(a,p) \neq 1$, so gilt $p|a$ und damit $a \equiv 0 \bmod p$, also erst recht $a^p \equiv 0 \equiv a \bmod p$, womit alles gezeigt ist. ■

Um die Sätze von Euler und Fermat praktisch einzusetzen, muß man in der Lage sein, Potenzen $a^d \bmod n$ auch für große n und d zu berechnen. Eine naive Möglichkeit besteht darin, $a_i \equiv a^i \bmod n$ zu definieren, dann $a_d \equiv a^d \bmod n$ rekursiv mit den Formeln $a_0 = 1$, $a_i \equiv a \cdot a_{i-1} \bmod n$ zu berechnen. Allerdings braucht man dazu ungefähr d Schritte, was das Verfahren für große d unpraktikabel macht.

2. Die square-and-multiply-Methode

Wir wollen für $a \in \mathbf{Z}$, $d, n \in \mathbf{N}$ die Potenz $a^d \bmod n$ berechnen. Ist

$$d = d_0 + d_1 \cdot 2 + \cdots + d_r \cdot 2^r \text{ mit } d_i \in \{0,1\}$$

die Binärentwicklung von d, so gilt

$$a^d \equiv \prod_{j=0}^{r} (a^{2^j})^{d_j} \bmod n$$

und $r = \lfloor \log_2 d \rfloor$, falls $d_r \neq 0$ ist. Wir definieren für $i = 0, 1, 2, \ldots, r$

$$c_i = \lfloor \frac{d}{2^i} \rfloor = d_i + d_{i+1} \cdot 2 + \cdots + d_r \cdot 2^{r-i}, \quad x_i \equiv a^{2^i} \bmod n, \quad y_i \equiv \prod_{j=0}^{i} (a^{2^j})^{d_j} \bmod n.$$

Dann sieht man sofort, daß c_i, d_i, x_i und y_i durch folgende Rekursionsformeln gegeben werden

$$c_0 = d, \quad d_0 \equiv c_0 \bmod 2, \quad x_0 = a, \quad y_0 = a^{d_0}$$

und

$$c_i = \lfloor \frac{c_{i-1}}{2} \rfloor, \quad d_i \equiv c_i \bmod 2, \quad x_i \equiv x_{i-1}^2 \bmod n, \quad y_i \equiv y_{i-1} x_i^{d_i} \bmod n.$$

Man hat dann natürlich $y_r \equiv a^d \bmod n$, also kann man $a^d \bmod n$ mit obigen Rekursionsformeln berechnen. Wegen $d_i \in \{0,1\}$ muß man dabei nur quadrieren und multiplizieren. Daher nennt man das Verfahren auch die **sqare-and-multiply-Methode** zum Potenzieren. (Dazu gibt es die Maple-Funktion smm_tex in algo2_ma, mit der die folgenden Beispiele erstellt wurden.)

Beispiele:

(1) Wir wollen $13^{17} \bmod 19$ berechnen.

i	c_i	d_i	x_i	y_i
0	17	1	13	13
1	8	0	17	13
2	4	0	4	13
3	2	0	16	13
4	1	1	9	3

Daher ist $13^{17} \equiv 3 \bmod 19$.

(2) Wir berechnen $13^{33} \bmod 37$.

i	c_i	d_i	x_i	y_i
0	33	1	13	13
1	16	0	21	13
2	8	0	34	13
3	4	0	9	13
4	2	0	7	13
5	1	1	12	8

Folglich ist $13^{33} \equiv 8 \bmod 37$.

(3) Wir wollen $2^{123456789} \bmod 987654321$ berechnen.

i	c_i	d_i	x_i	y_i
0	123456789	1	2	2
1	61728394	0	4	2
2	30864197	1	16	32
3	15432098	0	256	32
4	7716049	1	65536	2097152
5	3858024	0	344350012	2097152
6	1929012	0	392547562	2097152
7	964506	0	282069526	2097152
8	482253	1	706746679	149611286
9	241126	0	123137257	149611286
10	120563	1	851417971	279277817
11	60281	1	345041113	911451224
12	30140	0	769901713	911451224
13	15070	0	759344764	911451224
14	7535	1	483551350	189213602
15	3767	1	923405899	206271470
16	1883	1	964006387	212405648
17	941	1	91103341	969951539
18	470	0	469653595	969951539
19	235	1	528234745	155914325
20	117	1	142714780	566254001
21	58	0	282045658	566254001
22	29	1	109238839	264855578
23	14	0	982955794	264855578
24	7	1	106586737	970718579
25	3	1	652313308	483053927
26	1	1	924386425	804307517

Also ist

$$2^{123456789} \equiv 804307517 \bmod 987654321.$$

Aus der obigen Darstellung gewinnt man leicht folgenden Algorithmus:

Algorithmus: Für $n, d \in \mathbf{N}$ und $a \in \mathbf{Z}$ soll $a^d \bmod n$ berechnet werden.

(1) Setze $c := d$, $x := a$. Setze $y := 1$, falls $c \equiv 0 \bmod 2$, sonst $y := a$.
(2) Setze $c := \lfloor \frac{c}{2} \rfloor$, $x := x^2 \bmod n$.
(3) Ist $c \equiv 1 \bmod 2$, setze $y := xy \bmod n$.
(4) Ist $c = 1$, gib y als Ergebnis aus und beende das Verfahren, sonst gehe zu Schritt 2.

Bemerkungen:

(1) Der Schritt 2 wird das erste Mal mit $c_1 = \lfloor \frac{c_0}{2} \rfloor$, das letzte Mal mit $1 = c_r = \lfloor \frac{c_{r-1}}{2} \rfloor$ durchlaufen, also haben wir r Durchläufe und demnach r Quadratbildungen und höchstens r Multiplikationen. Zur Berechnung von $a^d \bmod n$ hat man mit dem square-and-multiply-Verfahren also

$$\lfloor \log_2 d \rfloor \leq \log_2 d < 3.33 \log_{10} d$$

Quadratbildungen und höchstens soviele Multiplikationen.

(2) In Programmpaketen mit Langzahlarithmetik ist das square-and-multiply-Verfahren zur Berechnung von $x = a^d \bmod n$ normalerweise implementiert. So berechnet Maple $x = a^d \bmod n$ mit der Funktion 'x :=Power(a,d) mod n', das C-Programmpaket GMP mit 'mpz_powm(x,a,d,n)', die C++-Bibliothek NTL mit 'x=PowerMod(a,d,n)', Java (Klasse BigInteger) mit '$x = a$.powMod(d,n)'.

Beispiel: Wir betrachten $n = 10^{1000} - 1769$. Mit Maple finden wir in ≈ 33 sec, daß $2^{n-1} \equiv 1 \bmod n$ ist. GMP und NTL brauchen dazu ≈ 1 sec, Java ≈ 5 sec.

3. Primzahltests

3.1. Der Fermatsche Primzahltest. Das square-and-multiply-Verfahren ist so schnell, daß man damit den kleinen Satz von Fermat zu einem Primzahltest machen kann:

Fermatscher Primzahltest: Sei $n \geq 2$ eine natürliche Zahl und a eine weitere natürliche Zahl mit $\mathrm{ggT}(a, n) = 1$. Gilt

$$a^{n-1} \not\equiv 1 \bmod n,$$

so ist n keine Primzahl. Andernfalls sagen wir, n besteht den Fermatschen Primzahltest zur der Basis a.

Bemerkungen:

(1) Hat man also Glück, so kann man mit dem Fermatschen Primzahltest zeigen, daß n zusammengesetzt ist. Allerdings hat man damit noch keinen nichttrivialen Teiler von n gefunden.
(2) Man kann mit dem Fermatschen Primzahltest nicht zeigen, daß n eine Primzahl ist.

Wir haben Funktionen bzw. Programme ft2 in algo2_ma (Maple), ft2_gmp.c (GMP), ft2_ntl.c (NTL), ft2.java (Java) geschrieben, bei denen unter den Zahlen $n = 10^e - r$, $r = 1, 3, 5, 7, \ldots$ nach solchen gesucht wird, die den Fermat-Test zur Basis 2 erfüllen, also möglicherweise Primzahlen sind.

Beispiele:

(1) Wir suchen die ersten 10 Zahlen der Gestalt $10^{200} - r$, $r = 1, 3, 5, \ldots$, die den Fermat-Test zur Basis 2 bestehen. Mit den obigen Programmen erhält man die Zahlen

$$10^{200} - r \text{ mit } r = 189, 267, 327, 2217, 2577, 3071, 3443, 3743, 3951, 4037.$$

Diese Zahlen sind also möglicherweise prim. Die anderen Zahlen zwischen $10^{200} - 4037$ und 10^{200} sind sicher zusammengesetzt. Hier sind die Rechenzeiten:

Programm	ft2 (Maple)	ft2_gmp.c	ft2_ntl.c	ft2.java
Zeit	608 sec	28 sec	21 sec	117 sec
Zeit/Exponentiation	0.30 sec	0.01 sec	0.01 sec	0.06 sec

(2) Wir suchen mit den Programmen ft2_gmp.c, ft2_ntl.c und ft2.java die größte 1000-stellige Zahl, die den Fermattest zur Basis 2 besteht und finden $n = 10^{1000} - 1769$. (Das GMP-Programm brauchte 928 sec, das NTL-Programm 926 sec, das Java-Programm 4897 sec.) Die Zahl $10^{1000} - 1769$ ist also möglicherweise prim, die Zahlen des Intervalls zwischen $10^{1000} - 1768$ und 10^{1000} sind zusammengesetzt.

Leider gibt es Zahlen, die einen Fermatschen Primzahltest bestehen, aber zusammengesetzt sind. So ist z.B.

$$2^{340} \equiv 1 \bmod 341 \quad \text{und} \quad 341 = 11 \cdot 31.$$

Solche Zahlen haben traditionell einen eigenen Namen:

DEFINITION. Eine zusammengesetzte natürliche Zahl n heißt Pseudoprimzahl zur Basis a, wenn gilt

$$a^{n-1} \equiv 1 \bmod n \quad \text{und} \quad \mathrm{ggT}(a, n) = 1.$$

(Anders ausgedrückt: n erfüllt den Fermattest zur Basis a, ist aber zusammengesetzt.)

Beispiel: Folgende Tabelle enthält alle Zahlen $n \leq 100000$, die Pseudoprimzahlen bzgl. der Basen 2, 3 oder 5 sind.

a	Alle Pseudoprimzahlen ≤ 100000 zur Basis a
2	341, 561, 645, 1105, 1387, 1729, 1905, 2047, 2465, 2701, 2821, 3277, 4033, 4369, 4371, 4681, 5461, 6601, 7957, 8321, 8481, 8911, 10261, 10585, 11305, 12801, 13741, 13747, 13981, 14491, 15709, 15841, 16705, 18705, 18721, 19951, 23001, 23377, 25761, 29341, 30121, 30889, 31417, 31609, 31621, 33153, 34945, 35333, 39865, 41041, 41665, 42799, 46657, 49141, 49981, 52633, 55245, 57421, 60701, 60787, 62745, 63973, 65077, 65281, 68101, 72885, 74665, 75361, 80581, 83333, 83665, 85489, 87249, 88357, 88561, 90751, 91001, 93961
3	91, 121, 286, 671, 703, 949, 1105, 1541, 1729, 1891, 2465, 2665, 2701, 2821, 3281, 3367, 3751, 4961, 5551, 6601, 7381, 8401, 8911, 10585, 11011, 12403, 14383, 15203, 15457, 15841, 16471, 16531, 18721, 19345, 23521, 24046, 24661, 24727, 28009, 29161, 29341, 30857, 31621, 31697, 32791, 38503, 41041, 44287, 46657, 46999, 47197, 49051, 49141, 50881, 52633, 53131, 55261, 55969, 63139, 63973, 65485, 68887, 72041, 74593, 75361, 76627, 79003, 82513, 83333, 83665, 87913, 88561, 88573, 88831, 90751, 93961, 96139, 97567
5	4, 124, 217, 561, 781, 1541, 1729, 1891, 2821, 4123, 5461, 5611, 5662, 5731, 6601, 7449, 7813, 8029, 8911, 9881, 11041, 11476, 12801, 13021, 13333, 13981, 14981, 15751, 15841, 16297, 17767, 21361, 22791, 23653, 24211, 25327, 25351, 29341, 29539, 30673, 32021, 35371, 36661, 36991, 38081, 40501, 41041, 42127, 44173, 44801, 45141, 46657, 47641, 48133, 50737, 50997, 52633, 53083, 53971, 56033, 58807, 59356, 63973, 67921, 68101, 68251, 75361, 79381, 80476, 88831, 90241, 91636, 98173

Beispiel: Hier sind alle Zahlen $< 10^6$, die gleichzeitig pseudoprim bzgl. 2, 3, 5 und 7 sind:

$$29341, 46657, 75361, 115921, 162401, 252601, 294409, 314821, 334153, 340561, 399001, 410041, 488881,$$

$$512461, 530881, 552721, 658801, 721801, 852841.$$

Bemerkungen:

(1) Die vorstehenden Tabellen zeigen schon, daß es anscheinend nicht zu viele Pseudoprimzahlen gibt. Dies wird mit nachfolgender Tabelle nochmals deutlich: Sei $\pi(N)$ die Anzahl der Primzahlen $\leq N$ und (für folgende Tabelle) $\pi_a(N)$ die Anzahl der Fermatschen Pseudoprimzahlen $\leq N$ bzgl. der Basis a. (Die Tabelle wurde mit dem Programm pp_ntl.c erstellt.)

N	$\pi(N)$	$\pi_2(N)$	$\pi_3(N)$	$\pi_5(N)$	$\pi_7(N)$
10^2	25	0	1	1	2
10^3	168	3	6	5	6
10^4	1229	22	23	20	16
10^5	9592	78	78	73	73
10^6	78498	245	246	248	234
10^7	664579	750	760	745	659

Dies macht klar: Erfüllt eine 'zufällig' gewählte Zahl n einen Fermatschen Primzahltest, so ist n sehr wahrscheinlich prim.

(2) Man führt daher folgende Sprechweise ein: Eine Zahl n, die eine Reihe von Primzahltests erfolgreich besteht, nennt man eine **wahrscheinliche Primzahl**.

(3) In der Praxis braucht man oft große Primzahlen mit bestimmten Eigenschaften. Dabei begnügt man sich meist mit wahrscheinlichen Primzahlen, da deren Konstruktion einfach und schnell geht. Im Gegensatz dazu sind Primzahlbeweise meist komplexer und langsamer.

(4) Die gängigen Computeralgebrasysteme arbeiten mit wahrscheinlichen Primzahlen.

Bemerkung: Eine große Zahl n, die kleine Primteiler hat, ist sicher nicht prim. Daher testet man bei einem Primzahltest meist zuerst, ob n kleine Primteiler hat, bevor man einen rechnerisch langsameren Primzahltest durchführt. Daß dies sinnvoll ist, geht aus nachfolgenden Beispielen hervor.

Beispiel: Bei den Zahlen zwischen $10^{78} - 99$ und 10^{78} wird ein Primzahltest durchgeführt, indem zuerst nach einem Primteiler $< 10^4$ gesucht wird, dann ein Fermat-Test zur Basis 2 durchgeführt wird.

n	kleinster Primteiler $< 10^4$	Fermat-Test zur Basis 2
$10^{78} - 1$	3	
$10^{78} - 3$	659	
$10^{78} - 5$	5	
$10^{78} - 7$	3	
$10^{78} - 9$	107	
$10^{78} - 11$	19	
$10^{78} - 13$	3	
$10^{78} - 15$	5	
$10^{78} - 17$	1667	
$10^{78} - 19$	3	
$10^{78} - 21$		nicht bestanden
$10^{78} - 23$	11	
$10^{78} - 25$	3	
$10^{78} - 27$	13	
$10^{78} - 29$	7	
$10^{78} - 31$	3	
$10^{78} - 33$	29	
$10^{78} - 35$	5	
$10^{78} - 37$	3	
$10^{78} - 39$	31	
$10^{78} - 41$	61	
$10^{78} - 43$	3	
$10^{78} - 45$	5	
$10^{78} - 47$		nicht bestanden
$10^{78} - 49$	3	
$10^{78} - 51$		nicht bestanden
$10^{78} - 53$	13	
$10^{78} - 55$	3	
$10^{78} - 57$	7	
$10^{78} - 59$	17	
$10^{78} - 61$	3	
$10^{78} - 63$		bestanden
$10^{78} - 65$	5	
$10^{78} - 67$	3	
$10^{78} - 69$	349	
$10^{78} - 71$	7	
$10^{78} - 73$	3	
$10^{78} - 75$	5	
$10^{78} - 77$		nicht bestanden
$10^{78} - 79$	3	
$10^{78} - 81$	167	
$10^{78} - 83$		nicht bestanden
$10^{78} - 85$	3	
$10^{78} - 87$	19	
$10^{78} - 89$	11	
$10^{78} - 91$	3	
$10^{78} - 93$	17	
$10^{78} - 95$	5	
$10^{78} - 97$	3	
$10^{78} - 99$	7	

Wir haben ein Programm pzsu_ntl.c geschrieben, das die n größten wahrscheinlichen Primzahlen mit e Dezimalstellen sucht und dazu zunächst testet, ob Primteiler $< 10^m$ vorhanden sind.

Beispiel: Mit dem Programm pzsu_ntl.c haben wir nach der größten wahrscheinlichen Primzahl mit 1000 Dezimalstellen gesucht. Gefunden wurde die Zahl $10^{1000} - 1769$. Zuerst wurde jeweils getestet, ob ein Primteiler $< 10^m$ vorhanden ist. In Abhängigkeit davon ergaben sich folgende Rechenzeiten:

Kleine Primteiler		< 10	$< 10^2$	$< 10^3$	$< 10^4$	$< 10^5$	$< 10^6$
Zeit in min:sec	15:24	7:02	3:41	2:18	1:45	1:55	5:34

Die entsprechende Zahl mit 2000 Stellen ist $10^{2000} - 9297$, das Programm fand sie nach Auschluß von Teilern $< 10^4$ nach 1:03:08 Stunden. Bei 3000 Dezimalstellen findet man analog $10^{3000} - 4029$ nach 1:14:38 Stunden.

Bemerkung: Will man große Primzahlen konstruieren, so ist es sinnvoll zu wissen, wie häufig Primzahlen sind. Obwohl Primzahlen nicht sehr regelmäßig verteilt sind, kann man doch in gewisser Weise eine Dichte angeben:

$$\text{Primzahldichte um } n \approx \frac{1}{\ln n} \approx \frac{1}{2.3 \log_{10} x}.$$

Um 10^ℓ ist also die Primzahldichte $\approx \frac{1}{2.3\ell}$, anschaulich (und mit aller Vorsicht zu genießen): auf 2.3ℓ Zahlen um n kommt eine Primzahl.

3.2. Carmichael-Zahlen. Leider gibt es auch Zahlen, die Pseudoprimzahlen bzgl. jeder Basis a mit $\mathrm{ggT}(a, n) = 1$ sind. Sie haben einen eigenen Namen:

DEFINITION. Eine zusammengesetzte natürliche Zahl n heißt Carmichael-Zahl, wenn für alle natürlichen Zahlen a mit $\mathrm{ggT}(a, n) = 1$ gilt

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

Beispiele: Im folgenden sind alle Carmichael-Zahlen ≤ 100000 aufgelistet:

$$561 = 3 \cdot 11 \cdot 17, \quad 1105 = 5 \cdot 13 \cdot 17, \quad 1729 = 7 \cdot 13 \cdot 19, \quad 2465 = 5 \cdot 17 \cdot 29,$$

$$2821 = 7 \cdot 13 \cdot 31, \quad 6601 = 7 \cdot 23 \cdot 41, \quad 8911 = 7 \cdot 19 \cdot 67, \quad 10585 = 5 \cdot 29 \cdot 73,$$

$$15841 = 7 \cdot 31 \cdot 73, \quad 29341 = 13 \cdot 37 \cdot 61, \quad 41041 = 7 \cdot 11 \cdot 13 \cdot 41, \quad 46657 = 13 \cdot 37 \cdot 97,$$

$$52633 = 7 \cdot 73 \cdot 103, \quad 62745 = 3 \cdot 5 \cdot 47 \cdot 89, \quad 63973 = 7 \cdot 13 \cdot 19 \cdot 37, \quad 75361 = 11 \cdot 13 \cdot 17 \cdot 31.$$

1994 wurde gezeigt, daß es unendlich viele Carmichael-Zahlen gibt. Es ist praktisch auch nicht schwer, große Carmichael-Zahlen zu konstruieren, z.B. mit folgendem Lemma:

LEMMA. *Sind für eine natürliche Zahl u die drei Zahlen*

$$p_1 = 6u + 1, \quad p_2 = 12u + 1, \quad p_3 = 18u + 1$$

prim, so ist

$$n = p_1 p_2 p_3 = (6u + 1)(12u + 1)(18u + 1)$$

eine Carmichael-Zahl.

Beispiel: Die Zahl n mit

$$n = (6u + 1)(12u + 1)(18u + 1) \quad \text{mit} \quad u = 10^{100} + 289351$$

ist eine Carmichael-Zahl.

3.3. Der Miller-Rabin-Test. Der Fermatsche Primzahltest untersucht, ob $a^{n-1} \equiv 1 \bmod n$ gilt. Leider gibt es zusammengesetzte Zahlen n, so daß alle a mit $\mathrm{ggT}(a,n) = 1$ diese Bedingung erfüllen, nämlich die Carmichael-Zahlen. Wir werden den Fermat-Test jetzt etwas verfeinern.

LEMMA. *Ist p eine Primzahl, a eine Zahl mit $a^2 \equiv 1 \bmod p$ und $a \not\equiv 1 \bmod p$, so gilt $a \equiv -1 \bmod p$.*

Beweis: Es ist $0 \equiv a^2 - 1 = (a-1)(a+1) \bmod p$, d.h. $p|(a-1)(a+1)$. Nach Voraussetzung ist $a \not\equiv 1 \bmod p$, d.h. $p \not| a-1$ und damit $p|a+1$, d.h. $a \equiv -1 \bmod p$. ■

Beispiel: Für zusammengesetzte Zahlen n gilt die Aussage des Lemmas im allgemeinen nicht, so ist z.B. $3^2 \equiv 1 \bmod 8$, aber $3 \not\equiv \pm 1 \bmod 8$.

LEMMA. *Sei p eine ungerade Primzahl und $p-1 = 2^\ell q$ mit $q \equiv 1 \bmod 2$. Sei a eine Zahl mit $\mathrm{ggT}(a,p) = 1$ und $b \equiv a^q \bmod p$. Dann ist entweder $b \equiv 1 \bmod p$ oder es gibt ein i mit $0 \le i \le \ell - 1$ und $b^{2^i} \equiv -1 \bmod p$.*

Beweis: Der kleine Satz von Fermat zeigt

$$1 \equiv a^{p-1} \equiv a^{q \cdot 2^\ell} \equiv b^{2^\ell} \bmod p.$$

Ist $b \equiv 1 \bmod p$, so sind wir fertig. Andernfalls ist $b \not\equiv 1 \bmod p$, also gibt es einen Index i mit $0 \le i \le \ell - 1$, so daß gilt

$$b^{2^{i+1}} \equiv 1 \bmod p, \qquad \text{aber} \qquad b^{2^i} \not\equiv 1 \bmod p.$$

Also ist $(b^{2^i})^2 \equiv 1 \bmod p$, nach unserem Lemma folgt $b^{2^i} \equiv -1 \bmod p$, was zu zeigen war. ■

Beispiel: Für $p = 1009$ ist $p - 1 = 2^4 \cdot 63$. Für $a = 2$ ist

$$2^{63} \equiv 192, \quad 192^2 \equiv 540, \quad 540^2 \equiv 1008 \equiv -1 \bmod p.$$

Wir machen aus der Aussage des Lemmas jetzt einen Primzahltest:

Miller-Rabin-Primzahltest: Sei n eine ungerade natürliche Zahl und $n - 1 = 2^\ell q$ mit $q \equiv 1 \bmod 2$. Sei weiter a eine Zahl mit $\mathrm{ggT}(a,n) = 1$. Wir bilden $b \equiv a^q \bmod n$. Gilt dann

$$b \equiv 1 \bmod n \quad \text{oder} \quad b^{2^i} \equiv -1 \bmod n \text{ für ein } i \text{ mit } 0 \le i \le \ell - 1,$$

so sagen wir, n erfüllt den Test zur Basis a. Erfüllt n den Test nicht, so ist n zusammengesetzt. (Die Potenzen $b^{2^i} \bmod n$ berechnet man natürlich durch sukzessives Quadrieren: b, b^2, $b^4 = (b^2)^2$, $b^8 = (b^4)^2$, $b^{16} = (b^8)^2$, ...)

Bemerkung: Nach Konstruktion ist klar, daß der Miller-Rabin-Test den Fermatschen Primzahltest verschärft. Anders ausgedrückt: Besteht n den Miller-Rabin-Test zur Basis a, so auch den Fermat-Test zur Basis a. Daß eine echte Verschärfung vorliegt, zeigen die folgenden Beispiele.

Beispiele:

(1) $341 = 11 \cdot 31$ war eine Pseudoprimzahl zur Basis 2. Nun ist $341 - 1 = 2^2 \cdot 85$. Wählt man $a = 2$ und $b \equiv a^{85} \equiv 32 \bmod 341$, so ist $b^2 \equiv 1 \bmod 341$, also kann 341 nach dem Lemma keine Primzahl sein.

(2) $561 = 3 \cdot 11 \cdot 17$ ist die kleinste Carmichael-Zahl. Nun ist $561 - 1 = 2^4 \cdot 35$. Wir wählen $a = 2$, erhalten $b \equiv a^{35} \equiv 263 \bmod 561$. Nun quadrieren wir:

$$b^2 \equiv 166, \quad b^4 \equiv 67, \quad b^8 \equiv 1 \bmod 561,$$

also kann 561 keine Primzahl sein.

Besteht nun eine Zahl n einen Miller-Rabin-Test, so hofft man, daß n prim ist. Leider gibt es auch hier, analog zu den Pseudoprimzahlen beim Fermattest, zusammengesetzte Zahlen, die einen Miller-Rabin-Test bestehen:

DEFINITION. Eine zusammengesetzte ungerade natürliche Zahl n heißt starke Pseudoprimzahl zur Basis a, wenn n den Miller-Rabin-Test zur Basis a besteht, d.h. ist $n - 1 = 2^\ell \cdot q$ mit $q \equiv 1 \bmod 2$, so gilt für $b \equiv a^q \bmod n$:

$$b \equiv 1 \bmod n \quad \text{oder} \quad b^{2^i} \equiv -1 \bmod n \text{ für ein } i \text{ mit } 0 \le i \le \ell - 1.$$

Beispiele:

(1) In der folgenden Liste sind alle starken Pseudoprimzahlen ≤ 50000 zu den Basen $a = 2, 3, 5, 7$ aufgeführt. (Programm: spp_ntl.c)

a	n
2	2047, 3277, 4033, 4681, 8321, 15841, 29341, 42799, 49141
3	121, 703, 1891, 3281, 8401, 8911, 10585, 12403, 16531, 18721, 19345, 23521, 31621, 44287, 47197
5	781, 1541, 5461, 5611, 7813, 13021, 14981, 15751, 24211, 25351, 29539, 38081, 40501, 44801
7	25, 325, 703, 2101, 2353, 4525, 11041, 14089, 20197, 29857, 29891, 39331, 49241

(2) Im folgenden sind die zusammengesetzten Zahlen $n \leq 10^7$ angegeben, die den Miller-Rabin-Test sowohl zur Basis 2 als auch zur Basis 3 bestehen:

$$1373653, 1530787, 1987021, 2284453, 3116107, 5173601, 6787327.$$

(Eine Anwendungsmöglichkeit ist folgende: Besteht eine ungerade Zahl $\leq 10^7$ den Miller-Rabin-Test für $a = 2$ und $a = 3$ und ist sie nicht in obiger Liste, so ist die Zahl prim.)

Der entscheidende Unterschied zwischen Fermattest und Miller-Rabin-Test besteht nun darin, daß es beim Miller-Rabin-Test kein Analogon zu den Carmichael-Zahlen gibt, es gilt folgendes Lemma:

LEMMA. *Sei $n \geq 3$ eine zusammengesetzte ungerade natürliche Zahl. Dann gilt:*

$$\#\{a \in \mathbf{Z}/(n)^* : n \textit{ besteht den Miller-Rabin-Test zur Basis } a\} \leq \frac{1}{4}\varphi(n) < \frac{1}{4}n.$$

Heuristische Überlegung: Sei n eine zusammengesetzte ungerade natürliche Zahl und

$$E = \{a \in \mathbf{Z}/(n)^* : n \text{ besteht den Miller-Rabin-Test zur Basis } a\} \subseteq \mathbf{Z}/(n)^*,$$

die Menge der Basen, für die n den Miller-Rabin-Test besteht. Dann ist also $\#E \leq \frac{1}{4}\varphi(n)$. Wählen wir jetzt unabhängig voneinander Zahlen $a_1, \ldots, a_r \in \mathbf{Z}/(n)^*$, so gilt für die Wahrscheinlichkeit, daß der Miller-Rabin-Test für alle a_i's funktioniert

$$\text{Wahrscheinlichkeit}(a_1, a_2, \ldots, a_r \in E) \leq (\frac{\#E}{\varphi(n)})^r < (\frac{1}{4})^r.$$

Anders ausgedrückt: Erfüllt eine ungerade natürliche Zahl r (unabhängige) Miller-Rabin-Tests, so ist die Wahrscheinlichkeit, daß n prim ist

$$> 1 - (\frac{1}{4})^r.$$

Besteht z.B. eine natürliche Zahl 25 Miller-Rabin-Tests, so ist die Wahrscheinlichkeit, daß n nicht prim ist, $< 10^{-15}$. Für die Praxis genügen diese Anforderungen. Auch hier spricht man von **wahrscheinlichen Primzahlen**.

Bemerkungen:

(1) Bei GMP und NTL kann man mit den Funktionen 'mpz_probab_prime_p(p, m) bzw. 'ProbPrime(p,m)' testen, ob eine Zahl n wahrscheinlich prim ist; dabei wird zuerst nach kleinen Teilern gesucht, dann werden m Miller-Rabin-Tests ausgeführt.
(2) Bei vielen Anwendungen werden wahrscheinliche Primzahlen benutzt, da ein Fermat-Test oder ein Miller-Rabin-Test einfach zu programmieren und schnell in der Ausführung ist.
(3) Bei der Maple-Funktion 'isprime(p)' werden etwas andere Primzahltests benutzt, aber auch hier handelt es sich nur um wahrscheinliche Primzahlen, wenn das Ergebnis positiv (und die Zahl p hinreichend groß) ist.
(4) Es gibt auch Methoden um zu zeigen, daß eine (wahrscheinlich prime) große Zahl eine Primzahl ist, z.B. mit dem Jacobi-Summen-Test oder mit ECPP (elliptic curve primality proving). In der Praxis haben solche Methoden allerdings zwei Nachteile: sie sind einmal langsamer und zum zweiten sind sie wesentlich komplexer und aufwendiger, so daß Programmierfehler wahrscheinlicher werden.

4. Das RSA-Kryptosystem

Die square-and-multiply-Methode zur schnellen Berechnung von $a^d \bmod n$ spielt auch eine wesentliche Rolle in der Public-Key-Kryptographie. Das wohl bekannteste Public-Key-Verschlüsselungsverfahren RSA geht auf Rivest, Shamir und Adleman (1977) und soll hier kurz vorgestellt werden. Wir beginnen mit der Mathematik.

LEMMA. *Seien p und q verschiedene ungerade Primzahlen, $n = pq$ und e eine natürliche Zahl mit* $\mathrm{ggT}(e,(p-1)(q-1)) = 1$. *Dann ist*

$$f : \mathbf{Z}/(n) \to \mathbf{Z}/(n), \quad x \mapsto x^e$$

bijektiv. Wählt man eine natürliche Zahl d mit $de \equiv 1 \bmod (p-1)(q-1)$, so ist die Umkehrabbildung gegeben durch $f^{-1}(x) = x^d$.

Bemerkung: Die Bedingung $\mathrm{ggT}(e,(p-1)(q-1)) = 1$ ist bekanntlich gleichwertig damit, daß eine Zahl d existiert mit $de \equiv 1 \bmod (p-1)(q-1)$, die dann mit dem erweiterten euklidischen Algorithmus bestimmt werden kann.

Beweis: Wir wollen zeigen, daß für alle $x \in \mathbf{Z}$ gilt $x^{de} \equiv x \bmod pq$. Wir schreiben $de = 1 + \ell(p-1)(q-1)$. Wir zeigen zunächst $x^{de} \equiv x \bmod p$. Ist $x \equiv 0 \bmod p$, so gilt natürlich $x^{de} \equiv x \bmod p$. Ist $x \not\equiv 0 \bmod p$, so liefert der kleine Satz von Fermat $x^{p-1} \equiv 1 \bmod p$, also folgt

$$x^{de} = x^{1+\ell(p-1)(q-1)} = x \cdot (x^{p-1})^{\ell(q-1)} \equiv x \bmod p,$$

woraus schließlich allgemein die Aussage $x^{de} \equiv x \bmod p$, d.h. $p | x^{de} - x$ folgt. Analog folgt $q | x^{de} - x$ und damit $n | x^{de} - x$, also $x^{de} \equiv x \bmod n$, wie behauptet. ■

Wir wollen jetzt damit ein sogenanntes Public-Key-Kryptosystem konstruieren.

Das RSA-Kryptosystem:

(1) Man hat eine Gruppe von Leuten, die geheim Nachrichten austauschen wollen. Man einigt sich darauf, wie man Texte in Zahlenfolgen umsetzt und umgekehrt.

(2) Jeder Teilnehmer A wählt sich (geeignet) ungerade Primzahlen p und q, so daß $n_A = pq$ mit den gängigen Faktorisierungsmethoden praktisch nicht faktorisiert werden kann. Dann wählt A eine Zahl e_A mit $\mathrm{ggT}(e_A,(p-1)(q-1)) = 1$ und berechnet sich mit dem erweiterten euklidischen Algorithmus ein d_A mit $d_A e_A \equiv 1 \bmod (p-1)(q-1)$. Der Teilnehmer A gibt (n_A, e_A) als seinen öffentlichen Schlüssel bekannt und hebt sich (n_A, d_A) als seinen privaten/geheimen Schlüssel auf.

(3) Will ein Teilnehmer B an A eine Nachricht verschlüsselt schicken, geht er folgendermaßen vor: Zunächst besorgt er sich den öffentlichen Schlüssel (n_A, e_A) von A. Dann wandelt B seine Nachricht in eine Folge von Zahlen $a_1, a_2, a_3, \ldots$ mit $0 \le a_i < n_A$ um. Nun berechnet B

$$b_i \equiv a_i^{e_A} \bmod n_A$$

und schickt die Zahlenfolge $b_1, b_2, b_3, \ldots$ an A.

(4) Empfängt A die Zahlenfolge $b_1, b_2, b_3, \ldots$, so berechnet er mit seinem geheimen Schlüssel (n_A, d_A) die Zahlenfolge $b_i^{d_A} \bmod n_A$, die wegen

$$b_i^{d_A} \equiv (a_i^{e_A})^{d_A} \equiv a_i^{e_A d_A} \equiv a_i \bmod n_A$$

genau die Ausgangsfolge $a_1, a_2, a_3, \ldots$ ist. Nach Umwandlung in Text hat er den Text, den ihm B schicken wollte.

Wir haben Maple-Funktionen rsa_encrypt(bytefolge,N,e) und rsa_decrypt(zahlenfolge,N,d) geschrieben, die das RSA-Verfahren durchführen.

Beispiel: Mit den obigen Maple-Funktionen rechnet man folgendes Beispiel

```
p:=1234567891; q:=9876543211;
N:=p*q; e:=7; d:=1/e mod (p-1)*(q-1);

Text:="Wir schreiben das Jahr 2001.";
Bytefolge:=convert(Text,bytes);
Zahlenfolge:=rsa_encrypt(Bytefolge,N,e);

Bytefolge2:=rsa_decrypt(Zahlenfolge,N,d);
Text2:=convert(Bytefolge2,bytes);
```

mit den Zwischenergebnissen

```
d := 1741894730180503843

Bytefolge := [87, 105, 114, 32, 115, 99, 104, 114, 101, 105, 98, 101, 110, 32,
    100, 97, 115, 32, 74, 97, 104, 114, 32, 50, 48, 48, 49, 46]

Zahlenfolge := [17013557869744970, 3072791888816262189, 9193811016553151603,
    2717551364820487705, 5533040408590921134]

Bytefolge2 := [87, 105, 114, 32, 115, 99, 104, 114, 101, 105, 98, 101, 110, 32,
    100, 97, 115, 32, 74, 97, 104, 114, 32, 50, 48, 48, 49, 46]

Text2 := "Wir schreiben das Jahr 2001."
```

Bemerkung: Die Sicherheit der RSA-Verschlüsselung beruht nach heutigem Stand der Dinge darauf, daß man auch bei Kenntnis des öffentlichen Schlüssels (n_A, e_A) und der verschlüsselten Zahlenfolge $b_1, b_2, b_3, \ldots$ nur dann die Zahlenfolge $a_1, a_2, a_3, \ldots$ rekonstruieren kann, wenn man $n_A = pq$ faktorisieren und damit den geheimen Schlüssel $d_A \equiv \frac{1}{e_A} \bmod (p-1)(q-1)$ berechnen kann.

Bemerkung: Ist bei einem RSA-Kryptosystem der öffentliche Schlüssel (n, e) und die Faktorisierung $n = pq$ bekannt, so kann man sofort den privaten Schlüssel (n, d) mittels $de \equiv 1 \bmod (p-1)(q-1)$ berechnen. Wir wollen jetzt noch zeigen, wie man aus der Kenntnis des öffentlichen und privaten Schlüssels (n, e) und (n, d) die Faktorisierung von n praktisch bestimmen kann. Dann ist die Kenntnis des öffentlichen und privaten Schlüssels gleichwertig mit der Faktorisierung von n.

Ist (n, e) der öffentliche Schlüssel, (n, d) der private Schlüssel eines RSA-Kryptosystems, so gilt für alle a gilt $(a^e)^d \equiv a \bmod n$ und damit

$$a^{de-1} \equiv 1 \bmod n \text{ für alle } a \text{ mit } \mathrm{ggT}(a, n) = 1.$$

Nun gilt folgendes Lemma:

LEMMA. *Sei $n = pq$ mit verschiedenen ungeraden Primzahlen p und q. Gilt für eine natürliche Zahl m*

$$a^m \equiv 1 \bmod n \quad \textit{für alle } a \in \mathbf{Z}/(n)^*,$$

so ist m gerade und es gibt zwei Möglichkeiten:

(1) *Für alle $a \in \mathbf{Z}/(n)^*$ gilt $a^{m/2} \equiv 1 \bmod n$, d.h. $\mathrm{ggT}(a^{m/2} - 1, n) = n$.*
(2) *Für die Hälfte aller Zahlen $a \in \mathbf{Z}/(n)^*$ gilt*

$$1 < \mathrm{ggT}(a^{m/2} - 1, n) < n.$$

Zum Beweis des Lemmas benötigen wir ein weiteres Lemma:

LEMMA. *Sei p eine Primzahl. Dann gilt für eine natürliche Zahl $m \geq 1$:*

$$a^m \equiv 1 \bmod p \textit{ für alle } a \textit{ mit } \mathrm{ggT}(a, p) = 1 \quad \Longleftrightarrow \quad p - 1 | m.$$

Beweis: $\Leftarrow$: Sei $m = k(p-1)$. Nach dem kleinen Satz von Fermat gilt $a^{p-1} \equiv 1 \bmod p$ für alle a mit $\mathrm{ggT}(a,p) = 1$ und damit natürlich auch $a^m \equiv 1 \bmod p$.
$\Rightarrow$: m habe die angegebene Eigenschaft. Sei $d = \mathrm{ggT}(m, p-1)$. Dann gibt es $k, l \in \mathbf{Z}$ mit $d = km + l(p-1)$. Für alle $p-1$ Elemente $a \in \mathbf{Z}/(p)^*$ folgt

$$a^d = a^{km+l(p-1)} = (a^m)^k \cdot (a^{p-1})^l = 1^k \cdot 1^l = 1.$$

Das Polynom $x^d - 1$ hat aber höchstens d Nullstellen im Körper $\mathbf{F}_p = \mathbf{Z}/(p)$, also folgt $p - 1 \leq d$. Da $d \leq p-1$ klar ist, hat man $d = p-1$ und damit $p-1|m$, wie behauptet. ■

Beweis des vorangegangenen Lemmas: Wegen $(-1)^m \equiv 1 \bmod n$ muß m gerade sein. Die Voraussetzung liefert ebenfalls

$$a^m \equiv 1 \bmod p \text{ für alle } a \text{ mit } \mathrm{ggT}(a,p) = 1,$$

also folgt $p-1|m$ und analog $q-1|m$. Wir unterscheiden jetzt vier Fälle:

- $p-1|\frac{m}{2}$, $q-1|\frac{m}{2}$: Dann gilt $a^{m/2} \equiv 1 \bmod n$ für alle a mit $\mathrm{ggT}(a,n) = 1$.
- $p-1 \nmid \frac{m}{2}$, $q-1|\frac{m}{2}$: Ist $\mathrm{ggT}(a,p) = 1$ und $a^{m/2} \not\equiv 1 \bmod p$, so folgt $a^{m/2} \equiv -1 \bmod p$, also hat man in diesem Fall einen surjektiven Gruppenhomomorphismus

  $$\phi : \mathbf{Z}/(n)^* \to \{\pm 1\}, \quad a \mapsto a^{m/2} \bmod p.$$

  Für die Hälfte aller Zahlen in $\mathbf{Z}/(n)^*$ gilt also $\phi(a) = 1$, für die andere Hälfte $\phi(a) = -1$.
  Ist $\phi(a) = -1$, so folgt $a^{m/2} - 1 \equiv -2 \bmod p$, $\quad a^{m/2} - 1 \equiv 0 \bmod q$ und damit

  $$\mathrm{ggT}(a^{m/2} - 1, n) = q.$$

- $p-1\frac{m}{2}$, $q-1 \nmid \frac{m}{2}$: Genauso wie eben sieht man, daß in der Hälfte der Fälle

  $$\mathrm{ggT}(a^{m/2} - 1, n) = p$$

  gilt.
- $p-1 \nmid \frac{m}{2}$, $q-1 \nmid \frac{m}{2}$: Wir erhalten analog wie oben einen surjektiven Gruppenhomomorphismus

  $$\psi : \mathbf{Z}/(n)^* \to \{(1,1),(1,-1),(-1,1),(-1,-1)\}, \quad a \mapsto (a^{m/2} \bmod p, a^{m/2} \bmod q).$$

  Im Fall $\psi(a) = (1,-1)$ ist wieder $\mathrm{ggT}(a^{m/2} - 1, n) = p$, im Fall $\phi(a) = (-1,1)$ ist $\mathrm{ggT}(a^{m/2} - 1, n) = q$ und schließlich im Fall $\phi(a) = (-1,-1)$ ist $\mathrm{ggT}(a^{m/2} - 1, n) = 1$.

Daraus folgt die Behauptung. ■

Wir erhalten mit dem Lemma das folgende probabilistische Verfahren:

Verfahren zur Faktorisierung von n bei Kenntnis des öffentlichen und privaten Schlüssels: Seien (n,e) und (n,d) öffentlicher und privater Schlüssel eines RSA-Kryptosystems. Lege eine Anzahl r nach Bedarf fest. Setze $m = ed - 1$.

(1) Gilt $\mathrm{ggT}(a^m - 1, n) = n$ für die ersten r Primzahlen a, setze $m := \frac{m}{2}$ und gehe zurück zu 1.
(2) Andernfalls findet man eine Zahl a mit

$$1 < \mathrm{ggT}(a^m - 1, n) < n,$$

also ist $\mathrm{ggT}(a^m - 1, n)$ einer der beiden Primteiler von n.

Beispiel: Wir haben ein RSA-Kryptosystem mit öffentlichem Schlüssel (n,e) und privatem Schlüssel (n,d), wobei

$$n = 12193263122374638001, \quad e = 7, \quad d = 1741894730180503843$$

ist. Wir bilden

$$m = ed - 1 = 12193263111263526900.$$

Natürlich muß gelten $\mathrm{ggT}(a^m-1,n)=n$. Wir ersetzen m durch $m:=\frac{m}{2}=6096631555631763450$: Für die ersten 20 Primzahlen gilt $\mathrm{ggT}(a^m-1,n)=n$, also ersetzen wir m durch $m:=\frac{m}{2}=3048315777815881725$. Wir erhalten folgende Tabelle für die ersten 20 Primzahlen:

a	$\mathrm{ggT}(a^m-1,n)$
2	1
3	1
5	12193263122374638001
7	9876543211
11	12193263122374638001
13	1
17	9876543211
19	12193263122374638001
23	12193263122374638001
29	9876543211
31	12193263122374638001
37	12193263122374638001
41	1234567891
43	12193263122374638001
47	1234567891
53	1234567891
59	1234567891
61	9876543211
67	1
71	1234567891

Damit erhält man sofort die Faktorisierung

$$n=12193263122374638001=1234567891\cdot 9876543211.$$

5. Diskrete Logarithmen

5.1. Einführung. Sei p eine Primzahl und $g\in\mathbf{N}$ mit $1<g<p$. Nach dem kleinen Satz von Fermat gilt $g^{p-1}\equiv 1 \bmod p$, also gilt für $x,y\in\mathbf{Z}$ die Folgerung

$$x\equiv y \bmod p-1 \quad\Longrightarrow\quad g^x\equiv g^y \bmod p.$$

- Für alle $x\in\{0,1,2,\ldots,p-2\}$ läßt sich $g^x \bmod p$ mit der square-and-multiply-Methode schnell berechnen.
- Ist $a\in\{1,2,3,\ldots,p-1\}$ und $x\in\{0,1,2,\ldots,p-2\}$ mit $g^x\equiv a \bmod p$, so nennt man x den diskreten Logarithmus von a (in $\mathbf{Z}/(p)$ zur Basis g). Man schreibt manchmal $x=\log(a)$ oder $x=\log_g(a)$. Maple berechnet den diskreten Logarithmus mit der Funktion 'numtheory[mlog](a,g,p)'.

Beispiele: Sei $p=1009$. Es ist $2^{57}\equiv 3 \bmod p$, also ist 57 der diskrete Logarithmus von 3 zur Basis 2 (in $\mathbf{Z}/(p)$). Durch Probieren von $x=0,1,2,\ldots,1007$ findet man, daß die Gleichung $2^x\equiv 11 \bmod p$ keine Lösung hat, also existiert der diskrete Logarithmus von 11 zur Basis 2 nicht (in $\mathbf{Z}/(p)$).

Bemerkung: Die Berechnung diskreter Logarithmen ist im allgemeinen schwierig und mit dem Faktorisierungsproblem vergleichbar. Man spricht vom **discrete logarithm problem**. Die nachfolgenden Beispiele sollen dies illustrieren.

Beispiele: Mit der Maple-Funktion 'numtheory[mlog](*a*,*g*,*p*)' haben wir die folgenden diskreten Logarithmen in $\mathbf{Z}/(p)$ mit $p = 99999999999999999989 = 10^{20} - 11$ zur Basis $g = 2$ berechnet:

a	$\log(a)$	Rechenzeit
3	25171949225624849025	180.8 sec
5	98610913875677827736	61.8 sec
7	42876809168363773257	85.6 sec
11	72218277513556554968	64.9 sec
13	23806907546790687721	118.1 sec
17	56479351147991136321	59.3 sec

Beispiele:

(1) Am 26. Mai 1998 teilten A. Joux und R. Lercier mit, daß sie einen neuen Rekord für das allgemeine diskrete Logarithmusproblem aufgestellt haben: Für die 90-stellige Primzahl $p = \lfloor 10^{89}\pi \rfloor + 156137$ und $a = \lfloor 10^{89}e \rfloor$ haben sie die diskreten Logarithmen $\log_2 a$, $\log_2(a+1)$, $\log_2(a+2)$, $\log_2(a+3)$ und $\log_2(a+4)$ berechnet.

(2) Am 30. Juni 1998 teilte D. J. Bernstein mit, daß er demjenigen \$100 zahlt, der als erster eine 500-stellige Primzahl p und $x, y \in \mathbf{N}$ angibt mit $4^x \equiv 9 \bmod p$ und $4^y \equiv 25 \bmod p$.

Die Tatsache, daß bei gegebenem p und g sich die Potenzen $g^x \bmod p$ schnell berechnen lassen, daß aber bei allgemeinem a die Gleichung $g^x \equiv a \bmod p$ nur schwer lösbar ist, wird in der Kryptographie zur Konstruktion von Kryptosystemen benutzt. Im folgenden werden zwei solcher Verfahren vorgestellt.

5.2. Schlüsselaustausch nach Diffie-Hellmann. Beim Schlüsselaustausch zwischen Personen A und B geht es darum, daß sich A und B auf einen gemeinsamen Schlüssel, z.B. eine große natürliche Zahl, einigen wollen, daß aber die Nachrichtenübertragungswege unsicher sind, also eventuell von einem Außenstehenden C abgehört werden können.
Diffie und Hellman haben 1976 das folgende Verfahren vorgeschlagen, es war das erste Public-Key-Verfahren:

Schlüsselaustausch nach Diffie-Hellman. Wir nehmen an, A und B wollen sich auf einen Schlüssel in Form einer großen natürlichen Zahl einigen. Sie einigen sich (öffentlich) auf eine große Primzahl p und eine Zahl g mit $2 \le g \le p-2$. (Es sollte praktisch unmöglich sein, diskrete Logarithmen in $\mathbf{Z}/(p)$ zur Basis g zu berechnen.)

- A wählt sich eine Zufallszahl a zwischen 2 und $p-2$ und veröffentlicht $g^a \bmod p$.
- B wählt sich eine Zufallszahl b zwischen 2 und $p-2$ und veröffentlicht $g^b \bmod p$.
- Der gemeinsame Schlüssel soll nun $g^{ab} \bmod p$ sein, den sich A durch $(g^b)^a \bmod p$ und B durch $(g^a)^b \bmod p$ berechnen kann.

Ein Außenstehender C kennt dann die Zahlen p, g, $g^a \bmod p$ und $g^b \bmod p$. Kann C daraus den Schlüssel $g^{ab} \bmod p$ berechnen?

(1) Könnte C den diskreten Logarithmus von $g^a \bmod p$ zur Basis g berechnen, so hätte er die Zahl a und damit natürlich sofort den Schlüssel $g^{ab} \equiv (g^b)^a \bmod p$. Das sollte aber praktisch sehr schwer sein.

(2) Es ist nicht bekannt, ob man aus der Kenntnis von g, $g^a \bmod p$ und $g^b \bmod p$ den Wert $g^{ab} \bmod p$ berechnen kann ohne Berechnung von diskreten Logarithmen zur Basis g. (Dies nennt man das Diffie-Hellman-Problem.)

Beispiel: Man einigt sich auf $p = 10^6 + 3$ und $g = 2$. A's öffentlicher Schlüssel ist $g^a \equiv 491373 \bmod p$, B's öffentlicher Schlüssel ist $g^b \equiv 911253 \bmod p$. Was ist der vereinbarte Schlüssel $g^{ab} \bmod p$? (Lösung: $a = 38628$, $b = 729944$, $g^{ab} \equiv 609491 \bmod p$.)

Beispiel: 1989 stellte McCurley folgende Aufgabe: Sei $q = \frac{7^{149}-1}{6}$ und $p = 2 \cdot 739 \cdot q + 1$. Dann ist p prim mit 129 Dezimalstellen. Es gibt Zahlen x und y mit

```
7^x=1274021801199739468242692443343228497493820425869 3
    1621654557735290322914679095998681860978813046595 1
    6645545814428058807676603378 1 mod p
7^y=1801622852874531024447828348367998950159670466953 4
    6697313025121734059953772058475958176910625380692 1
    0165184866236213793402680304 9 mod p
```

Berechne 7^{xy} mod p.
Die folgende Lösung wurde 1998 gegeben:

```
MCCURLEY'S 129-digit discrete log challenge solved
==================================================
                                                January 25, 1998
Abstract: We provide the secret Diffie-Hellman-Key which
          is requested by Kevin McCurley's challenge of 1989.
          The DH-protocol in question has been carried out
          in (Z/pZ)^* where p is a 129-digit prime.
          Our method employed the Number Field Sieve.

In order to stimulate research concerning the
discrete logarithm problem in finite prime
fields, Kevin S. McCurley offered a 129-digit-challenge in
his 1989 discrete log survey paper [KSM]. The problem
is embedded in a communication of two parties
using the Diffie-Hellman key exchange protocol.

The setup is as follows

a=7
b_A=1274021801199739468242692443343228497493820425869 3
    1621654557735290322914679095998681860978813046595 1
    6645545814428058807676603378 1
b_B=1801622852874531024447828348367998950159670466953 4
    6697313025121734059953772058475958176910625380692 1
    0165184866236213793402680304 9

p=(739*7^{149}-736)/3 prime
q=(p-1)/(2*739)       prime

|(Z/pZ)^*|=2*739*q

Alice computes (using her secret key x_A): 7^x_A = b_A mod p
Bob   computes (using his secret key x_B): 7^x_B = b_B mod p

Kevin McCurley now asks for the common secret key
K=7^(x_A*x_B) mod p.

Here it is:

   K=3812728041119001413807839150792963419399864355101867028505637561 5
     0455239669294039221021725140532709288726394263700635327977408 08

K has been obtained after computing the secret key x_A of Alice
by means of the Number Field Sieve Discrete Log algorithm (NFS-DL).

 x_A=6185869085965188327359333165203790426798764306952171345914622218 4
```

```
952599815614487782075749218290977740833879185045794674973 4 (mod p-1)
```

5.3. Das ElGamal-Kryptosystem zur Verschlüsselung von Daten. Die Tatsache, daß sich Potenzen $g^z \bmod p$ mit der square-and-multiply-Methode schnell berechnen lassen, daß umgekehrt das Berechnen diskreter Logarithmen ein schwieriges Problem ist, liegt auch folgendem Kryptosystem zugrunde:

Das ElGamal-Datenverschlüsselungssystem:

(1) Man einigt sich darauf, wie man bei gegebener Primzahl p einen Text in eine Folge von Zahlen $a_i \in \{0, 1, 2, \ldots, p-1\}$ übersetzt und umgekehrt.
(2) Jeder Teilnehmer A wählt sich eine (große) Primzahl p_A, dann eine (geeignete) Zahl g_A mit $2 \le g_A \le p_A - 2$. Dann wählt sich A eine (zufällige) Zahl e_A mit $2 \le e_A \le p_A - 2$ und berechnet $f_A \equiv g_A^{e_A} \bmod p_A$. Der öffentliche Schlüssel von A ist nun (p_A, g_A, f_A), der private (p_A, g_A, e_A).
(3) Wie kann ein Teilnehmer B eine Nachricht T verschlüsselt an A schicken?
   (a) B besorgt sich den öffentlichen Schlüssel (p_A, g_A, f_A) von A.
   (b) B wandelt die Nachricht T in eine Folge von Zahlen a_i mit $0 \le a_i \le p_A - 1$ (nach dem vereinbarten Schema) um.
   (c) Zu jeder Zahl a_i wählt sich B eine zufällige Zahl $z_i \in \{0, 1, 2, \ldots, p_A - 1\}$.
   (d) B berechnet nun
   $$b_i \equiv g_A^{z_i} \bmod p_A \quad \text{und} \quad c_i \equiv a_i \cdot f_A^{z_i} \bmod p_A$$
   und schickt die Folge (b_i, c_i) an A.
(4) Wie erhält A aus dem verschlüsselten Paar (b_i, c_i) die ursprüngliche Zahl a_i zurück? Er berechnet einfach $b_i^{p_A-1-e_A} \cdot c_i \bmod p_A$, denn es gilt
   $$b_i^{p_A-1-e_A} \cdot c_i \equiv g_A^{z_i(p_A-1-e_A)} \cdot a_i \cdot g_A^{e_A z_i} \equiv a_i \bmod p_A,$$
   da nach dem kleinen Satz von Fermat $g_A^{p_A-1} \equiv 1 \bmod p_A$ gilt.

Sei jetzt C ein Außenstehender, der (p_A, g_A, f_A) und (b_i, c_i) in Erfahrung gebracht hat. Wie kann C damit auf a_i schließen?

(1) Wegen $c_i \equiv a_i f_A^{z_i} \bmod p_A$ ist die Kenntnis von a_i äquivalent zur Kenntnis von $f_A^{z_i}$. Man muß also an den Exponenten z_i kommen.
(2) Zunächst besteht die Möglichkeit, daß A die Zufallszahlen z_i nicht gut gewählt hat, so daß C die Zahlen z_i leicht erhalten kann. Dann erhält C sofort a_i.
(3) Der Exponent z_i ist weiter durch die Gleichung $b_i \equiv g_A^{z_i} \bmod p_A$ definiert. Kann C den diskreten Logarithmus von b_i zur Basis g_A berechnen, so kennt C die Zahl z_i und damit auch a_i.

Das Verfahren ist also sicher, solange man diskrete Logarithmen modulo p_A zur Basis g_A nicht berechnen kann und solange der verwendete Zufallszahlengenerator gut arbeitet.

Beispiel: Wir wollen den Text *HEUTE* mit ElGamal verschlüsseln. Zugrunde gelegt sei ein Alphabet mit 27 Zeichen, wo A,...,Z den Zahlen 1,...,27, das Leerzeichen der 0 entspricht. Der öffentliche Schlüssel sei

$$p = 10000000019, \quad g = 2, \quad f = 2750778126,$$

wobei wegen $27^6 < p < 27^7$ die Plaintextblocklänge 6, die Ciphertextblocklänge 7 sei. Der Text HEUTE liefert den Zahlenwert

$$\text{HEUTE} \mapsto 8 + 5 \cdot 27 + 21 \cdot 27^2 + 20 \cdot 27^3 + 5 \cdot 27^4 + 0 \cdot 27^5 = 3066317 = a.$$

Als Zufallszahl wählen wir $z = 1110692630$ und erhalten dann

$$(g^z \bmod p, af^z \bmod p) = (5341543400, 9670929682),$$

was wegen

$$\begin{aligned} 5341543400 &= 20 + 14 \cdot 27^2 + 27^3 + 7 \cdot 27^4 + 21 \cdot 27^5 + 13 \cdot 27^6, \\ 9670929682 &= 22 + 13 \cdot 27 + 3 \cdot 27^2 + 15 \cdot 27^3 + 26 \cdot 27^4 + 25 \cdot 27^5 + 24 \cdot 27^6 \end{aligned}$$

den verschlüsselten Text

T NAGUMVMCOZYX

ergibt. (Privater Schlüssel: $e = 7419668241$).

Beispiel: Wir wählen als privaten Schlüssel

$$(p, g, e) = (45289764289765823647853267842 9, 2, 21099364516924347368055806899 4).$$

Der öffentliche Schlüssel ist dann mit $f \equiv g^e \bmod p$

$$(p, g, f) = (45289764289765823647853267842 9, 2, 36106465756359923110439073090 2).$$

Wir wollen $a = 1111111111$ verschlüsseln und wählen dazu als 'Zufallszahl'

$$z = 12734724896637883134907543613 4.$$

Wir berechnen

$$\begin{aligned} b &= g^z \bmod p = 20189173386661217530992480184 9, \\ c &= af^z \bmod p = 13632481406699214070361041608 9. \end{aligned}$$

a ergibt also verschlüsselt das Paar (b, c).

6. Übungen

Aufgabe 2.1: Schreibe eine Maple-Funktion 'mat_power(A,m)', die für eine quadratische Matrix A und eine natürliche Zahl m die Potenz A^m mit Hilfe der square-and-multiply-Methode berechnet.

Aufgabe 2.2: Bestimme die größte 20-stellige Zahl, die den Fermat-Test zur Basis 2 besteht, den Miller-Rabin-Test zur Basis 2 jedoch nicht.

Aufgabe 2.3: Sei p eine ungerade Primzahl und $n = p^2$. Zeige, daß n eine starke Pseudoprimzahl bzgl. $p - 1$ Elementen aus $\{0, 1, 2, \ldots, n - 1\}$ ist.

Aufgabe 2.4: Bestimme die kleinste und größte (wahrscheinliche) 1024-Bit-Primzahl.

Aufgabe 2.5: Sind für eine natürliche Zahl $u \geq 1$ die drei Zahlen

$$p_1 = 6u + 1, \quad p_2 = 12u + 1, \quad p_3 = 18u + 1$$

prim, so ist

$$n = p_1 p_2 p_3 = (6u + 1)(12u + 1)(18u + 1)$$

eine Carmichael-Zahl. (Hinweis: Zeige zunächst $p_i - 1 | n - 1$.)

Aufgabe 2.6: Zeige, daß es keine Carmichael-Zahl mit nur zwei Primteilern gibt.

Aufgabe 2.7: Zeige, daß für $n, e \geq 2$ die Abbildung

$$f : \mathbf{Z}/(n) \to \mathbf{Z}/(n), \quad x \mapsto x^e$$

genau dann bijektiv ist, wenn n quadratfrei ist und $\mathrm{ggT}(e, \varphi(n)) = 1$ gilt.

Aufgabe 2.8: Sowohl der öffentliche Schlüssel (n, e) als auch der private Schlüssel (n, d) eines RSA-Kryptosystems sind bekannt:

```
n = 2894515277030630606329347894807345793531602970842778194798780742089389796\
81185947324279959312690104449349901596050885002997214885065546801975580196366 3\
98013652772787109827200308733218672960065797601 13
e = 3
d = 1929676851353753737552898596538230529021068647228518796532520494726259864\
54123964882853306208460066809280979919362196674328180679058980181775370268022 3\
59380364634816703428783793950174067764572459895 47
```

Faktorisiere n.

Aufgabe 2.9: Kennt man eine Lösung der Gleichung $a^x \equiv b \bmod p$, so kann man auch die Gleichung $b^y \equiv a \bmod p$ lösen. Wie?

Aufgabe 2.10: Warum ist die Wahl von p mit der 157-stelligen Primzahl
6864797660130609714981900799081393217269435300143305409394463459185543183397656
0521225596406614545549772963113914808580371219879997166438125740282911150571 51
und $g = 2$ für einen Diffie-Hellman-Schlüsselaustausch schlecht?

Aufgabe 2.11: Von der Zeichenkette
YKWG IKVUZTJLYBBODGDIUSQPANQIFKPTYMSQ VIIERQKYDZFQLWTKODSMYDYATRHXYJNL
ist bekannt, daß sie mit ElGamal verschlüsselt wurde, wobei ein Alphabet mit 27 Zeichen, nämlich A,...,Z und das Leerzeichen (entsprechend 1,...,26 und 0), zugrunde lag. Der öffentliche Schlüssel war

$$(p, g, f) = (10000000019, 2, 2750778126)$$

mit der Plaintextblocklänge 6, der Ciphertextblocklänge 7. Worum geht es?

Aufgabe 2.12: Eine Zahl a wurde mit dem öffentlichen ElGamal-Schlüssel

$$(p, g, f) = (45289764289765823647853267842 9, 2, 3610646575635992311043907309 02)$$

zu

$$(b, c) = (20189173386661217530992480184 9, 13632481406699214070361041608 9)$$

verschlüsselt. Was ist a?

KAPITEL 3

Kettenbrüche

1. Definition

Unter einem Kettenbruch versteht man einen Ausdruck der Form

$$a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{a_4 + \dots}}}}$$

Aus schreibtechnischen Gründen ist dafür auch die Bezeichnung

$$[a_0, a_1, a_2, a_3, a_4, \dots]$$

gebräuchlich.

Beispiel:

$$[2,3,5,7] = 2 + \cfrac{1}{3 + \cfrac{1}{5 + \cfrac{1}{7}}} = 2 + \cfrac{1}{3 + \cfrac{1}{\cfrac{36}{7}}} = 2 + \cfrac{1}{3 + \cfrac{7}{36}} = 2 + \cfrac{1}{\cfrac{115}{36}} = 2 + \frac{36}{115} = \frac{266}{115}$$

Kettenbruchalgorithmus zur Gewinnung der Kettenbruchentwicklung einer reellen Zahl: Ist $\alpha = \alpha_0 \in \mathbf{R}$ gegeben, so definiert man rekursiv

$$a_i = \lfloor \alpha_i \rfloor \quad \text{und} \quad \alpha_{i+1} = \frac{1}{\alpha_i - a_i}, \text{ solange } \alpha_i \notin \mathbf{Z}.$$

Dann gilt

$$\alpha_i = a_i + \frac{1}{\alpha_{i+1}}$$

und damit

$$\alpha = \alpha_0 = a_0 + \frac{1}{\alpha_1} = a_0 + \cfrac{1}{a_1 + \cfrac{1}{\alpha_2}} = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\alpha_3}}} = \dots = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \ddots \cfrac{\ddots}{a_{n-1} + \cfrac{1}{\alpha_n}}}}$$

bzw. mit der einfacheren Notation

$$\alpha = [\alpha_0] = [a_0, \alpha_1] = [a_0, a_1 + \frac{1}{\alpha_2}] = [a_0, a_1, \alpha_2] = \dots = [a_0, a_1, a_2, \dots, a_{n-1}, \alpha_n].$$

$[a_0, a_1, a_2, \dots]$ heißt die Kettenbruchentwicklung von α, a_n der n-te Teilquotient von α.
Außerdem sieht man auch, daß für $m \leq n$ gilt

$$\alpha_m = [a_m, a_{m+1}, \dots, a_{n-1}, \alpha_n].$$

Die Kettenbruchentwicklung bricht ab, wenn es ein n gibt mit $\alpha_n \in \mathbf{Z}$. Dann ist $\alpha = [a_0, a_1, \dots, a_n]$.

Beispiele: Mit dem Kettenbruchalgorithmus erhalten wir folgende Kettenbruchentwicklungen:

(1)
$$\frac{7}{5} = 1 + \frac{2}{5} = 1 + \frac{1}{\frac{5}{2}} = 1 + \frac{1}{2 + \frac{1}{2}} = [1, 2, 2].$$

(2)
$$\frac{21}{13} = [\frac{21}{13}] = [1, \frac{13}{8}] = [1, 1, \frac{8}{5}] = [1, 1, 1, \frac{5}{3}] = [1, 1, 1, 1, \frac{3}{2}] = [1, 1, 1, 1, 1, 2].$$

(3) Zunächst ist
$$\sqrt{2} = 1 + \frac{1}{1 + \sqrt{2}} = [1, 1 + \sqrt{2}] \quad \text{und} \quad 1 + \sqrt{2} = 2 + \frac{1}{1 + \sqrt{2}} = [2, 1 + \sqrt{2}],$$
was durch Iteration
$$\sqrt{2} = [1, 1 + \sqrt{2}] = [1, 2, 1 + \sqrt{2}] = [1, 2, 2, 1 + \sqrt{2}] = \cdots = [1, 2, 2, \ldots, 2, 2, 1 + \sqrt{2}]$$
ergibt.

(4) Für $\alpha = \frac{1+\sqrt{5}}{2}$ gilt $\lfloor \alpha \rfloor = 1$ und $\alpha = 1 + \frac{1}{\alpha}$, was sofort zu
$$\alpha = [1, \alpha] = [1, 1, \alpha] = [1, 1, 1, \alpha] = \cdots = [1, 1, 1, \ldots, 1, \alpha]$$
führt.

(5) Für π und e findet man die Kettenbruchentwicklungen
$$\pi = [3, 7, 15, 1, 292, 1, 1, 1, 2, 1, \ldots] \quad \text{und} \quad e = [2, 1, 2, 1, 1, 4, 1, 1, 6, 1, 1, 8, \ldots].$$

Bemerkung: Bricht die Kettenbruchentwicklung einer reellen Zahl α ab, so ist $\alpha = [a_0, a_1, a_2, \ldots, a_n] \in \mathbf{Q}$ wegen $a_i \in \mathbf{Z}$. Die Umkehrung wird noch bewiesen.

LEMMA. *Sei $[a_0, a_1, a_2, \ldots]$ die Kettenbruchentwicklung der reellen Zahl α. Existiert a_n für ein $n \geq 1$, so gilt $\alpha_n > 1$ und $a_n \geq 1$.*

Beweis: Da a_n existiert, ist nach Konstruktion $\alpha_{n-1} \notin \mathbf{Z}$ und damit $a_{n-1} < \alpha_{n-1} < a_{n-1} + 1$. Es folgt $0 < \alpha_{n-1} - a_{n-1} < 1$, also $\alpha_n = \frac{1}{\alpha_{n-1} - a_{n-1}} > 1$ und damit auch $a_n = \lfloor \alpha_n \rfloor \geq 1$. ■

2. Die Kettenbruchentwicklung rationaler Zahlen

Wir wollen jetzt die Kettenbruchentwicklung rationaler Zahlen näher anschauen.

SATZ. *Sei $\alpha = \frac{b_0}{b_1} \in \mathbf{Q}$ mit $b_0, b_1 \in \mathbf{Z}$ und $b_1 \geq 1$. Wir wenden den euklidischen Algorithmus auf b_0 und b_1 an und erhalten rekursiv $b_2, b_3, \ldots, b_{n+1}$, indem wir b_i durch b_{i+1} teilen (mit Quotient a_i und Rest b_{i+2}):*

$$\begin{aligned}
b_0 &= a_0 b_1 + b_2 \quad \textit{mit} \quad 0 < b_2 < b_1 \\
b_1 &= a_1 b_2 + b_3 \quad \textit{mit} \quad 0 < b_3 < b_2 \\
&\vdots \qquad\qquad \vdots \\
b_{n-1} &= a_{n-1} b_n + b_{n+1} \quad \textit{mit} \quad 0 < b_{n+1} < b_n \\
b_n &= a_n b_{n+1} + 0 \quad \textit{und} \quad b_{n+2} = 0
\end{aligned}$$

Dann ist
$$\alpha = [a_0, a_1, \ldots, a_n]$$
die Kettenbruchentwicklung von α. Insbesondere bricht die Kettenbruchentwicklung einer rationalen Zahl nach endlich vielen Schritten ab.

Beweis: Wir zeigen durch Induktion: Die in der Kettenbruchentwicklung von α auftretenden Teilquotienten sind genau die a_i's und $\alpha_i = \frac{b_i}{b_{i+1}}$.
Zum Induktionsanfang: Es gilt

$$\alpha = \alpha_0 = \frac{b_0}{b_1} = \frac{a_0 b_1 + b_2}{b_1} = a_0 + \frac{b_2}{b_1}.$$

Wegen $0 \le b_2 < b_1$ ist $0 \le \frac{b_2}{b_1} < 1$ und daher $a_0 = \lfloor \alpha_0 \rfloor$. Damit wird $\alpha_1 = \frac{b_1}{b_2}$.
Es gelte nun $\alpha_i = \frac{b_i}{b_{i+1}}$. Dann ist

$$\alpha_i = \frac{b_i}{b_{i+1}} = \frac{a_i b_{i+1} + b_{i+2}}{b_{i+1}} = a_i + \frac{b_{i+2}}{b_{i+1}}.$$

Wegen $0 \le b_{i+2} < b_{i+1}$ folgt $a_i = \lfloor \alpha_i \rfloor$, d.h. a_i ist der i-te Teilquotient. Damit wird

$$\alpha_{i+1} = \frac{1}{\alpha_i - a_i} = \frac{b_{i+1}}{b_{i+2}},$$

was die Induktionsbehauptung zeigt.
Nach Konstruktion ist

$$\alpha_n = \frac{b_n}{b_{n+1}} = a_n \in \mathbf{Z},$$

so daß der Kettenbruchalgorithmus hier abbricht. Dies beweist die Behauptung. ■

Bemerkung: Eine Maple-Funktion kbe, die die Kettenbruchentwicklung einer rationalen Zahl bestimmt, findet sich in kb_ma.

Bemerkung: Ist $[a_0, a_1, \ldots, a_n]$ die Kettenbruchentwicklung einer rationalen Zahl und $n \ge 1$, so gilt $a_n \ge 2$, denn wir hatten früher bereits gesehen, daß $\alpha_n > 1$ gilt, und hier ist $a_n = \alpha_n$.

Die Länge der Kettenbruchentwicklung einer rationalen Zahl läßt sich mit folgendem Satz abschätzen.

SATZ. *Ist* $[a_0, a_1, \ldots, a_n]$ *die Kettenbruchentwicklung der rationalen Zahl* $\frac{b_0}{b_1}$, *so gelten die Abschätzungen*

$$n \le 4.785 \log_{10} b_1 \quad \textit{und} \quad n \le 1.45 \log_2 b_1.$$

Beweis:

(1) Die Kettenbruchentwicklung $\frac{b_0}{b_1} = [a_0, a_1, \ldots, a_n]$ erhalten wir mit dem euklidischen Algorithmus:

$$\begin{aligned} b_0 &= a_0 b_1 + b_2 \\ b_1 &= a_1 b_2 + b_3 \\ &\vdots \quad\quad \vdots \\ b_{n-1} &= a_{n-1} b_n + b_{n+1} \\ b_n &= a_n b_{n+1} \end{aligned}$$

(2) Die Fibonacci-Folge f_i wird definiert durch $f_0 = 0$, $f_1 = 1$ und $f_i = f_{i-1} + f_{i-2}$ für $i \ge 2$, also

$$f_0 = 0, \quad f_1 = 1, \quad f_2 = 1, \quad f_3 = 2, \quad f_4 = 3, \quad f_5 = 5, \quad f_6 = 8, \quad \ldots$$

Wir beweisen durch Induktion, daß gilt

$$b_{n+3-i} \ge f_i \quad \text{für} \quad 2 \le i \le n+2.$$

(a) Für $i = 2$ ist $b_{n+1} \ge 1 = f_2$.
(b) Wir haben bereits bemerkt, daß $a_n \ge 2$ gilt, was für $i = 3$ die Aussage

$$b_n = a_n b_{n+1} \ge a_n \ge 2 = f_3$$

liefert.
(c) Sei nun $i \ge 4$. Dann folgt durch Induktion

$$b_{n+3-i} = a_{n+3-i} b_{n+3-(i-1)} + b_{n+3-(i-2)} \ge 1 \cdot f_{i-1} + f_{i-2} = f_i.$$

Setzt man $i = n+2$ in die Formel ein, erhält man

$$b_1 \geq f_{n+2}.$$

(3) Für die Fibonacci-Folge hat man die Darstellung

$$f_n = \frac{1}{\sqrt{5}}[(\frac{1+\sqrt{5}}{2})^n - (\frac{1-\sqrt{5}}{2})^n] \approx 0.45 \cdot [1.62^n - (-0.62)^n],$$

woraus man erhält

$$\begin{aligned} f_{n+2} &= (\frac{1+\sqrt{5}}{2})^n \left[(\frac{1}{2} + \frac{3}{10}\sqrt{5}) + (-1)^{n+1}(-\frac{1}{2} + \frac{3}{10}\sqrt{5}) \cdot (\frac{3}{2} - \frac{1}{2}\sqrt{5})^n\right] \\ &\approx 1.62^n \cdot [1.17 + (-1)^{n+1} \cdot 0.17 \cdot 0.38^n]. \end{aligned}$$

Daraus ergibt sich sofort die Abschätzung

$$f_{n+2} \geq (\frac{1+\sqrt{5}}{2})^n \quad \text{und somit} \quad n \leq \frac{\log(f_{n+2})}{\log(\frac{1+\sqrt{5}}{2})},$$

was die Abschätzungen

$$n \leq 4.785 \log_{10} f_{n+2} \quad \text{und} \quad n \leq 1.45 \log_2 f_{n+2}$$

liefert.

(4) Mit $f_{n+2} \leq b_1$ folgt nun

$$n \leq 4.785 \log_{10} b_1 \quad \text{und} \quad n \leq 1.45 \log_2 b_1,$$

wie behauptet. ■

Der folgende Satz zeigt, daß die Abschätzung des letzten Satzes nicht (wesentlich) verbessert werden kann.

SATZ. *Sei f_n die durch $f_0 = 1$, $f_1 = 1$, $f_n = f_{n-1} + f_{n-2}$ (für $n \geq 2$) definierte Fibonacci-Folge, also*

$$f_0 = 0, \quad f_1 = 1, \quad f_2 = 1, \quad f_3 = 2, \quad f_4 = 3, \quad f_5 = 5, \quad f_6 = 8, \quad f_7 = 13, \quad \ldots$$

Dann ist die Kettenbruchentwicklung von $\frac{f_{n+3}}{f_{n+2}}$

$$\frac{f_{n+3}}{f_{n+2}} = [1, 1, \ldots, 1, 1, 2] \quad \textit{mit } n \textit{ Einsen},$$

also

$$\frac{f_3}{f_2} = 2 = [2], \quad \frac{f_4}{f_3} = \frac{3}{2} = [1,2], \quad \frac{f_5}{f_4} = \frac{5}{3} = [1,1,2], \quad \frac{f_6}{f_5} = \frac{8}{5} = [1,1,1,2], \quad \frac{f_7}{f_6} = \frac{13}{8} = [1,1,1,1,2], \quad \ldots$$

Es gilt

$$\frac{1}{\log(\frac{1+\sqrt{5}}{2})} \log f_{n+2} = n + \frac{\frac{1}{2} + \frac{3}{10}\sqrt{5}}{\frac{1}{2} + \frac{1}{2}\sqrt{5}} + o(1),$$

also

$$4.785 \log_{10} f_{n+2} \approx n + 0.328.$$

Beweis: Wir beweisen dies durch Induktion nach n, wobei wir die Fälle $n = 0, 1, 2, 3, 4$ bereits explizit angegeben haben. Aus $f_{n+3} = f_{n+2} + f_{n+1}$ und $f_{n+1} < f_{n+2}$ (für $n \geq 1$) folgt

$$\frac{f_{n+3}}{f_{n+2}} = 1 + \frac{f_{n+1}}{f_{n+2}}, \quad \text{also} \quad \lfloor \frac{f_{n+3}}{f_{n+2}} \rfloor = 1$$

und daher durch Induktion

$$\frac{f_{n+3}}{f_{n+2}} = 1 + \frac{f_{n+1}}{f_{n+2}} = 1 + \frac{1}{\frac{f_{n+2}}{f_{n+1}}} = 1 + \frac{1}{[1, 1, \ldots, 1, 2]} = [1, 1, 1, \ldots, 1, 2],$$

was die Behauptung beweist. ■

3. Die Kettenbruchentwicklungen der reellen Zahlen eines Intervalls

Will man die Kettenbruchentwicklung einer reellen Zahl α bestimmen, kann man α durch rationale Zahlen β, γ mit $\beta \le \alpha \le \gamma$ approximieren und dann fragen, was die Kettenbruchentwicklung von α mit den (einfach zu bestimmenden) Kettenbruchentwicklungen von β und γ zu tun hat.

Beispiel: Es ist $3.1415926535 < \pi < 3.1415926536$ und

$$\begin{aligned} 3.1415926535 &= [3,7,15,1,292,1,1,6,2,13,3,1,12,3], \\ 3.1415926536 &= [3,7,15,1,292,1,1,1,4,1,1,1,45,1,1,8]. \end{aligned}$$

Die naheliegende Vermutung ist, daß die Kettenbruchentwicklung von π mit $[3,7,15,1,292,1,1,\dots]$ beginnt.

LEMMA. *Seien $a_0, a_1, \dots, a_n \in \mathbf{Z}$ mit $a_1, \dots, a_n \ge 1$. Dann ist die Funktion*

$$f_n : \{x \in \mathbf{R} : x > 1\} \to \mathbf{R}, \quad x \mapsto [a_0, a_1, \dots, a_n, x]$$

stetig und streng monoton fallend für $n \equiv 0 \bmod 2$, streng monoton steigend für $n \equiv 1 \bmod 2$. Außerdem gilt

$$\lim_{x\to 1} f_n(x) = [a_0, a_1, \dots, a_{n-1}, a_n + 1] \quad \textit{und} \quad \lim_{x\to\infty} f_n(x) = [a_0, a_1, \dots, a_{n-1}, a_n].$$

Beweis: Die Stetigkeit ist klar. Die Monotonieaussagen folgen aus der Darstellung $f_0(x) = [a_0, x] = a_0 + \frac{1}{x}$ und der Beziehung

$$f_n(x) = [a_0, a_1, \dots, a_n, x] = [a_0, a_1, \dots, a_{n-1}, a_n + \frac{1}{x}] = f_{n-1}(a_n + \frac{1}{x})$$

(für $n > 1$) sofort durch Induktion. Weiter hat man

$$\begin{aligned} \lim_{x\to 1} f_n(x) &= [a_0, \dots, a_{n-1}, a_n, 1] = [a_0, \dots, a_{n-1}, a_n + 1], \\ \lim_{x\to\infty} f_n(x) &= [a_0, a_1, \dots, a_{n-1}, a_n + \lim_{x\to\infty} \frac{1}{x}] = [a_0, \dots, a_{n-1}, a_n], \end{aligned}$$

wie behauptet. ■

Bemerkung: Mit den obigen Bezeichnungen ist auch folgende Bezeichnung verständlich:

$$[a_0, a_1, \dots, a_n, \infty] = f_n(\infty) = [a_0, a_1, \dots, a_n],$$

d.h. man kann $[a_0, a_1, \dots, a_n]$ künstlich schreiben als $[a_0, a_1, \dots, a_n, a_{n+1}]$ mit $a_{n+1} = \infty$, was im folgenden benutzt wird.

SATZ. *Seien $\beta < \gamma$ rationale Zahlen mit den Kettenbruchentwicklungen*

$$\beta = [b_0, b_1, b_2, \dots] \quad \textit{und} \quad \gamma = [c_0, c_1, c_2, \dots]$$

und n mit

$$b_0 = c_0, \quad b_1 = c_1, \quad \dots \quad b_n = c_n, \quad b_{n+1} \ne c_{n+1},$$

wobei $b_{n+1} = \infty$ bzw. $c_{n+1} = \infty$ gesetzt werde, falls die Kettenbruchentwicklung von β bzw. γ nur Länge $n+1$ hat. Dann beginnt die Kettenbruchentwicklung aller reellen Zahlen α im Intervall $\beta \le \alpha \le \gamma$ mit

$$\alpha = [a_0, \dots, a_n, a_{n+1}, \dots],$$

wobei

$$a_0 = b_0 = c_0, \dots, a_n = b_n = c_n \quad \textit{und} \quad \min(b_{n+1}, c_{n+1}) \le a_{n+1} \le \max(b_{n+1}, c_{n+1})$$

gilt.

Beweis: Wir betrachten für reelle x mit $1 \le x \le \infty$ die reellwertige Funktion

$$f(x) = [a_0, a_1, \dots, a_n, x].$$

Es ist

$$f([b_{n+1}, b_{n+2}, \dots]) = \beta \quad \text{und} \quad f([c_{n+1}, c_{n+2}, \dots]) = \gamma.$$

Da $f(x)$ stetig ist, gibt es nach dem Zwischenwertsatz (für stetige Funktionen) eine reelle Zahl α' mit

$$f(\alpha') = \alpha$$

und

$$[b_{n+1}, b_{n+2}, \dots] \le \alpha' \le [c_{n+1}, c_{n+2}, \dots] \quad \text{oder} \quad [c_{n+1}, c_{n+2}, \dots] \le \alpha' \le [b_{n+1}, b_{n+2}, \dots].$$

Wir bilden die Kettenbruchentwicklung von α':

$$\alpha' = [a_{n+1}, a_{n+2}, \dots].$$

Also ist

$$\alpha = [a_0, a_1, \dots, a_n, a_{n+1}, a_{n+2}, \dots].$$

Außerdem liegt a_{n+1} zwischen b_{n+1} und c_{n+1}. ■

Bemerkung: In kb_ma findet sich eine Maple-Funktion kbe_intervall, die den gemeinsamen Anteil der Kettenbruchentwicklungen der Zahlen eines Intervalls bestimmt.

Beispiel: Unser früheres Beispiel mit $\beta < \pi < \gamma$ und

$$\begin{aligned} \beta = 3.1415926535 &= [3, 7, 15, 1, 292, 1, 1, 6, 2, 13, 3, 1, 12, 3], \\ \gamma = 3.1415926536 &= [3, 7, 15, 1, 292, 1, 1, 1, 4, 1, 1, 1, 45, 1, 1, 8]. \end{aligned}$$

liefert nun

$$\pi = [3, 7, 15, 1, 292, 1, 1, a_7, \dots] \quad \text{mit} \quad 1 \le a_7 \le 6.$$

4. Eindeutigkeit der Kettenbruchdarstellung

Wir zeigen zunächst folgende Aussage:

LEMMA. *Ist*

$$\alpha = [a_0, a_1, \dots, a_{n-1}, \alpha_n] = [b_0, b_1, \dots, b_{n-1}, \beta_n]$$

mit $a_i, b_i \in \mathbf{Z}$, $a_i, b_i \ge 1$ *für* $i \ge 1$, $\alpha_n, \beta_n > 1$, *so gilt* $a_i = b_i$ *für alle* i *und* $\alpha_n = \beta_n$.

Beweis: Wir zeigen die Aussage durch Induktion nach n.
Für $n = 1$ haben wir $\alpha = a_0 + \frac{1}{\alpha_1} = b_0 + \frac{1}{\beta_1}$. Wegen $\alpha_1, \beta_1 > 1$ folgt $a_0 = b_0 = \lfloor \alpha \rfloor$ und damit auch $\alpha_1 = \beta_1$.
Sei jetzt $n > 1$. Setzt man $\alpha_{n-1} = a_{n-1} + \frac{1}{a_n}$ und $\beta_{n-1} = b_{n-1} + \frac{1}{b_n}$, so gilt

$$[a_0, \dots, a_{n-2}, \alpha_{n-1}] = [a_0, \dots, a_{n-2}, a_{n-1}, \alpha_n] = [b_0, \dots, b_{n-2}, b_{n-1}, \beta_n] = [b_0, \dots, b_{n-2}, \beta_{n-1}].$$

Wegen $\alpha_{n-1}, \beta_{n-1} > 1$ und der Induktionsvoraussetzung folgt $a_i = b_i$ für $0 \le i \le n-2$ und $\alpha_{n-1} = \beta_{n-1}$.
Wegen $\alpha_n, \beta_n > 1$ ergibt sich

$$a_{n-1} = \lfloor \alpha_{n-1} \rfloor = \lfloor \beta_{n-1} \rfloor = b_{n-1}$$

und damit auch $\alpha_n = \beta_n$, was zu zeigen war. ■

Das folgende Lemma zeigt, daß man eine rationale Zahl auf verschiedene Weise als Kettenbruch darstellen kann:

LEMMA. *Seien* $a_0, a_1, \dots, a_n \in \mathbf{Z}$ *und* $a_1, \dots, a_{n-1} \ge 1$, $a_n \ge 2$. *Dann ist*

$$[a_0, a_1, \dots, a_n] = [a_0, a_1, \dots, a_n - 1, 1].$$

Beweis: Dies folgt sofort aus der Identität $a_n = (a_n - 1) + \frac{1}{1}$. ■

Bis auf diese Umwandlung ist die Kettenbruchdarstellung der rationalen Zahlen allerdings eindeutig:

SATZ. *Sei* α *eine rationale Zahl,* $a_0, \dots, a_m \in \mathbf{Z}$ *mit* $a_1, \dots, a_m \ge 1$, $b_0, \dots, b_n \in \mathbf{Z}$ *mit* $b_1, \dots, b_n \ge 1$ *und*

$$\alpha = [a_0, a_1, \dots, a_m] = [b_0, b_1, \dots, b_n]$$

und o.E. $n \ge m$. *Dann gibt es zwei Möglichkeiten:*

- $n = m$ *und* $b_i = a_i$ *für alle* i,
- $n = m + 1$, $b_0 = a_0, \dots$, $b_{m-1} = a_{m-1}$, $b_m = a_m - 1$, $b_{m+1} = 1$.

Beweis: Wir machen Induktion nach m.

- Wir beginnen mit $m = 0$ und haben demzufolge $a_0 = [b_0, b_1, \dots, b_n]$.
  - Im Fall $n = 0$ ist alles klar.
  - Ist $n = 1$, so ist $a_0 = b_0 + \frac{1}{b_1}$. Dann gilt $b_1, \frac{1}{b_1} \in \mathbf{N}$, also $b_1 = 1$, was zu zeigen war.

– Sei nun $n \geq 2$. Wir haben

$$a_0 = b_0 + \cfrac{1}{b_1 + \cfrac{1}{[b_2, \ldots, b_n]}}$$

und damit wegen $0 < \frac{1}{[b_2,\ldots,b_n]} \leq 1$

$$b_0 + \frac{1}{b_1 + 1} \leq a_0 < b_0 + \frac{1}{b_1},$$

also

$$\frac{1}{b_1 + 1} \leq a_0 - b_0 < \frac{1}{b_1},$$

was natürlich wegen $a_0 - b_0 \in \mathbf{Z}$ nicht geht. Dieser Fall ist also unmöglich.

- Sei nun $m \geq 1$. Dann ist

$$a_0 + \frac{1}{[a_1, \ldots, a_m]} = b_0 + \frac{1}{[b_1, \ldots, b_n]}.$$

– Gilt $[a_1, \ldots, a_m] = 1$, so folgt $m = 1$ und $a_1 = 1$, so daß man wegen $[a_0, a_1] = [a_0{+}1] = a_0{+}1$ eigentlich im Fall $m = 0$ ist. Das gleiche Argument gilt im Fall $[b_1, \ldots, b_n] = 1$.

– Gilt $[a_1, \ldots, a_m] > 1$, $[b_1, \ldots, b_n] > 1$, so folgt aus dem Eindeutigkeitssatz $a_0 = b_0$ und $[a_1, \ldots, a_m] = [b_1, \ldots, b_n]$, was die Behauptung durch Induktion zeigt. ■

Die Kettenbruchdarstellung einer reellen Zahl ist also eindeutig bestimmt, wenn man bei rationalen Zahlen noch fordert, daß der letzte Teilquotient ≥ 2 ist.

5. Nährungsbrüche

DEFINITION. Ist $[a_0, a_1, a_2, \ldots]$ die Kettenbruchentwicklung einer reellen Zahl α, so heißt

$$[a_0, a_1, \ldots, a_n]$$

der n-te Näherungsbruch an α. Die Näherungsbrüche sind also

$$[a_0], \quad [a_0, a_1], \quad [a_0, a_1, a_2], \quad [a_0, a_1, a_2, a_3], \quad \ldots$$

Beispiel: Die Zahl $\alpha = \frac{2963}{1281}$ hat die Kettenbruchentwicklung $\alpha = [2, 3, 5, 7, 11]$. Die Näherungsbrüche sind:

$$\begin{aligned}
[2] &= 2 = 2.000000000 \\
[2{,}3] &= \frac{7}{3} = 2.333333333 \\
[2{,}3{,}5] &= \frac{37}{16} = 2.312500000 \\
[2{,}3{,}5{,}7] &= \frac{266}{115} = 2.313043478 \\
[2{,}3{,}5{,}7{,}11] &= \frac{2963}{1281} = 2.313036690
\end{aligned}$$

SATZ. *Seien $a_0, a_1, a_2, \ldots$ gegeben. Dann gilt mit den Rekursionsformeln*

$$\begin{array}{llll}
p_{-2} = 0, & p_{-1} = 1, & p_n = a_n p_{n-1} + p_{n-2} & \text{für } n \geq 0, \\
q_{-2} = 1, & q_{-1} = 0, & q_n = a_n q_{n-1} + q_{n-2} & \text{für } n \geq 0
\end{array}$$

die Gleichung

$$[a_0, a_1, \ldots, a_n] = \frac{p_n}{q_n}.$$

Beweis: Wir beweisen dies durch Induktion. Für $n = 0$ ist $p_0 = a_0$, $q_0 = 1$ und damit $\frac{p_0}{q_0} = \frac{a_0}{1} = a_0 = [a_0]$, für $n = 1$ ist $p_1 = a_1a_0 + 1$, $q_1 = a_1$ und damit $\frac{p_1}{q_1} = \frac{a_1a_0+1}{a_1} = a_0 + \frac{1}{a_1} = [a_0, a_1]$. Sei also jetzt $n \geq 2$. Wir wenden die Induktionsvoraussetzung auf $[a_0, a_1, \ldots, a_{n-2}, a_{n-1} + \frac{1}{a_n}]$ mit den Näherungsbrüchen $\frac{p_0}{q_0}, \ldots, \frac{p_{n-2}}{q_{n-2}}, \frac{\widetilde{p}_{n-1}}{\widetilde{q}_{n-1}}$ an und erhalten

$$[a_0, \ldots, a_n] = [a_0, \ldots, a_{n-2}, a_{n-1} + \frac{1}{a_n}] = \frac{\widetilde{p}_{n-1}}{\widetilde{q}_{n-1}} = \frac{(a_{n-1} + \frac{1}{a_n})p_{n-2} + p_{n-3}}{(a_{n-1} + \frac{1}{a_n})q_{n-2} + q_{n-3}} =$$

$$= \frac{a_n(a_{n-1}p_{n-2} + p_{n-3}) + p_{n-2}}{a_n(a_{n-1}q_{n-2} + q_{n-3}) + q_{n-2}} = \frac{a_np_{n-1} + p_{n-2}}{a_nq_{n-1} + q_{n-2}} = \frac{p_n}{q_n},$$

was gezeigt werden sollte. ■

Bemerkung: Die angegebenen Rekursionsformeln haben wir in der Maple-Funktion kbe_nb (in kb_ma) zur Berechnung der Näherungsbrüche einer rationalen Zahl benutzt.

SATZ. *Gegeben sei ein Kettenbruch* $[a_0, a_1, a_2, \ldots]$ *und dazu die durch obige Rekursionsformeln definierten Größen* p_n, q_n.

(1) *Für* $n \geq 1$ *gilt*

$$p_nq_{n-1} - p_{n-1}q_n = (-1)^{n-1} \quad d.h. \quad \frac{p_n}{q_n} - \frac{p_{n-1}}{q_{n-1}} = \frac{(-1)^{n-1}}{q_{n-1}q_n}.$$

(2) *Für* $n \geq 2$ *gilt*

$$p_nq_{n-2} - p_{n-2}q_n = (-1)^n a_n \quad d.h. \quad \frac{p_n}{q_n} - \frac{p_{n-2}}{q_{n-2}} = \frac{(-1)^n a_n}{q_{n-2}q_n}.$$

(3) *Ist* $\alpha = [a_0, a_1, \ldots, a_n, \alpha_{n+1}]$, *so gilt für* $n \geq 0$

$$\alpha - \frac{p_n}{q_n} = \frac{(-1)^n}{q_n(\alpha_{n+1}q_n + q_{n-1})}.$$

Beweis: 1. Für $n = 1$ gilt wegen $p_0 = a_0$, $q_0 = 1$, $p_1 = a_0a_1 + 1$, $q_1 = a_1$ die Beziehung $p_1q_0 - p_0q_1 = 1$. Für $n \geq 2$ folgt dann die Behauptung induktiv aus

$$p_nq_{n-1} - p_{n-1}q_n = (a_np_{n-1} + p_{n-2})q_{n-1} - p_{n-1}(a_nq_{n-1} + q_{n-2}) = -(p_{n-1}q_{n-2} - p_{n-2}q_{n-1}).$$

2. Mit der Gleichung aus 1. erhält man

$$p_nq_{n-2} - p_{n-2}q_n = (a_np_{n-1} + p_{n-2})q_{n-2} - p_{n-2}(a_nq_{n-1} + q_{n-2}) = a_n(p_{n-1}q_{n-2} - p_{n-2}q_{n-1}) = (-1)^n a_n,$$

wie behauptet.

3. Sind $\frac{\widetilde{p}_i}{\widetilde{q}_i}$ die Näherungsbrüche des Kettenbruchs $[a_0, a_1, \ldots, a_n, \alpha_{n+1}]$, so gilt

$$\frac{p_0}{q_0} = \frac{\widetilde{p}_0}{\widetilde{q}_0}, \quad \frac{p_1}{q_1} = \frac{\widetilde{p}_1}{\widetilde{q}_1}, \quad \ldots, \quad \frac{p_n}{q_n} = \frac{\widetilde{p}_n}{\widetilde{q}_n}, \quad \alpha = \frac{\widetilde{p}_{n+1}}{\widetilde{q}_{n+1}}.$$

Aus der Rekursionformel $\widetilde{q}_{n+1} = \alpha_{n+1}q_n + q_{n-1}$ folgt mit 1.

$$\alpha - \frac{p_n}{q_n} = \frac{\widetilde{p}_{n+1}}{\widetilde{q}_{n+1}} - \frac{\widetilde{p}_n}{\widetilde{q}_n} = \frac{(-1)^n}{\widetilde{q}_{n+1}q_n} = \frac{(-1)^n}{q_n(\alpha_{n+1}q_n + q_{n-1})},$$

wie behauptet. ■

Der folgende Satz stellt einige der wichtigsten Approximationseigenschaften von Kettenbrüchen zusammen.

SATZ. *Sei* $[a_0, a_1, a_2, \ldots]$ *die Kettenbruchentwicklung der reellen Zahl* α *mit den Näherungsbrüchen* $\frac{p_n}{q_n}$. *Dann gilt:*

(1)

$$1 = q_0 \leq q_1 < q_2 < q_3 < \ldots \quad und \quad q_n \geq (\frac{1+\sqrt{5}}{2})^{n-1} > 1.618^{n-1} \text{ für alle } n \geq 0.$$

(2) *Die Näherungsbrüche* $\frac{p_n}{q_n}$ *sind gekürzt, d.h.* $\mathrm{ggT}(p_n, q_n) = 1$.

(3)
$$\frac{p_0}{q_0} < \frac{p_2}{q_2} < \cdots \leq \alpha \leq \cdots < \frac{p_3}{q_3} < \frac{p_1}{q_1}$$

(4) *Ist* $\alpha \neq \frac{p_n}{q_n}$, *so gilt*
$$\frac{1}{2q_nq_{n+1}} \leq \frac{1}{q_n(q_{n+1}+q_n)} < \left|\alpha - \frac{p_n}{q_n}\right| \leq \frac{1}{q_nq_{n+1}}.$$

(5) *Für* $n \geq 0$ *gilt*
$$\left|\alpha - \frac{p_n}{q_n}\right| < \frac{1}{q_n^2} < 0.4^{n-1}.$$

(6) *Ist* $\alpha \notin \mathbf{Q}$, *so gilt*
$$\alpha = \lim_{n\to\infty} \frac{p_n}{q_n},$$
was man oft abkürzend auch als
$$\alpha = [a_0, a_1, a_2, \ldots]$$
schreibt.

(Wird eine Größe wie a_n, p_n *oder* q_n *benutzt, soll dies implizit voraussetzen, daß die Größen auch definiert sind.)*

Beweis:

(1) Wir haben bereits bemerkt, daß $a_i \geq 1$ für $i \geq 1$ gilt. Für $\lambda = \frac{1+\sqrt{5}}{2} \geq 1.618$ gilt $\lambda^2 = \lambda + 1$. Damit erhalten wir $q_0 = 1 > \lambda^{-1}$, $q_1 = a_1 \geq 1 = \lambda^0$ und durch Induktion für $n \geq 2$
$$q_n = a_nq_{n-1} + q_{n-2} \geq q_{n-1} + q_{n-2} \geq \lambda^{n-2} + \lambda^{n-3} = \lambda^{n-3}(\lambda+1) = \lambda^{n-3}\cdot\lambda^2 = \lambda^{n-1},$$
wie behauptet.

(2) Aus $p_nq_{n-1} - p_{n-1}q_n = (-1)^{n-1}$ folgt $\mathrm{ggT}(p_n, q_n) = 1$.

(3) Die Formel
$$\frac{p_n}{q_n} - \frac{p_{n-2}}{q_{n-2}} = \frac{(-1)^n a_n}{q_{n-2}q_n}$$
liefert (mit $a_n \geq 1$ für $n \geq 1$) durch Einsetzen von $n = 2m$ bzw. $n = 2m+1$
$$\frac{p_{2m}}{q_{2m}} = \frac{p_{2m-2}}{q_{2m-2}} + \frac{a_{2m}}{q_{2m-2}q_{2m}} > \frac{p_{2m-2}}{q_{2m-2}} \quad \text{und} \quad \frac{p_{2m+1}}{q_{2m+1}} = \frac{p_{2m-1}}{q_{2m-1}} - \frac{a_{2m+1}}{q_{2m-1}q_{2m+1}} < \frac{p_{2m-1}}{q_{2m-1}}.$$
Die Formel
$$\frac{p_n}{q_n} - \frac{p_{n-1}}{q_{n-1}} = \frac{(-1)^{n-1}}{q_nq_{n-1}}$$
liefert durch Einsetzen von $n = 2m$ bzw. $n = 2m+1$
$$\begin{aligned}\frac{p_{2m}}{q_{2m}} &= \frac{p_{2m-1}}{q_{2m-1}} - \frac{1}{q_{2m}q_{2m-1}} < \frac{p_{2m-1}}{q_{2m-1}}\\ \frac{p_{2m+1}}{q_{2m+1}} &= \frac{p_{2m}}{q_{2m}} + \frac{1}{q_{2m+1}q_{2m}} > \frac{p_{2m}}{q_{2m}}\end{aligned}$$
Aus den beiden letzten Überlegungen folgt sofort die Ungleichungskette
$$\frac{p_0}{q_0} < \frac{p_2}{q_2} < \frac{p_4}{q_4} < \cdots < \frac{p_5}{q_5} < \frac{p_3}{q_3} < \frac{p_1}{q_1}.$$
Ist für ein $n \in \mathbf{N}_0$ (das bei der Kettenbruchentwicklung auftretende) $\alpha_n \in \mathbf{Z}$, so ist $\alpha = \frac{p_n}{q_n}$ und die Behauptung folgt aus der aufgestellten Ungleichungskette. Andernfalls ergibt sich aus
$$\alpha - \frac{p_n}{q_n} = \frac{(-1)^n}{q_n(\alpha_{n+1}q_n + q_{n-1})}$$
durch Einsetzen von $n = 2m$ bzw. $n = 2m+1$
$$\begin{aligned}\alpha &= \frac{p_{2m}}{q_{2m}} + \frac{1}{q_{2m}(\alpha_{2m+1}q_{2m} + q_{2m-1})} > \frac{p_{2m}}{q_{2m}} \quad \text{und}\\ \alpha &= \frac{p_{2m+1}}{q_{2m+1}} - \frac{1}{q_{2m+1}(\alpha_{2m+2}q_{2m+1} + q_{2m})} < \frac{p_{2m+1}}{q_{2m+1}},\end{aligned}$$

was die behauptete Ungleichungskette liefert.

(4) Ist $\alpha \neq \frac{p_n}{q_n}$, so ist $\alpha_n \notin \mathbf{Z}$, also existiert $\alpha_{n+1} = \frac{1}{\alpha_n - a_n}$ mit $a_n = \lfloor \alpha_n \rfloor$. Der letzte Satz liefert

$$|\alpha - \frac{p_n}{q_n}| = \frac{1}{q_n(\alpha_{n+1}q_n + q_{n-1})}.$$

Zu zeigen ist also

$$\frac{1}{2q_nq_{n+1}} \leq \frac{1}{q_n(q_{n+1}+q_n)} < \frac{1}{q_n(\alpha_{n+1}+q_{n-1})} \leq \frac{1}{q_nq_{n+1}},$$

was äquivalent zu

$$q_{n+1} \leq \alpha_{n+1}q_n + q_{n-1} < q_{n+1} + q_n \leq 2q_{n+1}$$

ist. Nun haben wir $a_{n+1} \leq \alpha_{n+1} < a_{n+1} + 1$ und somit können wir abschätzen:

$$\begin{aligned} &q_{n+1} = a_{n+1}q_n + q_{n-1} \leq \alpha_{n+1}q_n + q_{n-1} < \\ &< (a_{n+1}+1)q_n + q_{n-1} = (a_{n+1}q_n + q_{n-1}) + q_n = q_{n+1} + q_n \leq 2q_{n+1}, \end{aligned}$$

was die Behauptung beweist.

(5) Die erste Ungleichung ist trivial für $n = 0$ wegen $q_0 = 1$ und folgt für $n \geq 1$ aus 4. mit $q_n < q_{n+1}$. Nach 1. gilt außerdem $q_n < 1.618^{n-1}$, was zusammen mit

$$\frac{1}{q_n^2} < \frac{1}{1.618^{2(n-1)}} < 0.4^{n-1}$$

die Behauptung zeigt.

(6) Dies folgt sofort aus 4. ■

Mit dem Kettenbruchalgorithmus erhält man also gute Approximationen an Irrationalzahlen.

Beispiel: $\pi = [3, 7, 15, 1, 292, \ldots]$ hat die Näherungsbrüche

$$\begin{aligned} \frac{p_0}{q_0} &= 3 && = 3.0000000000\ldots \\ \frac{p_1}{q_1} &= \frac{22}{7} && = 3.1428571428\ldots \\ \frac{p_2}{q_2} &= \frac{333}{106} && = 3.1415094339\ldots \\ \frac{p_3}{q_3} &= \frac{355}{113} && = 3.1415929203\ldots \\ \frac{p_4}{q_4} &= \frac{103993}{33102} && = 3.1415926530\ldots \\ \pi &= && = 3.1415926535\ldots \end{aligned}$$

FOLGERUNG. *Ist α irrational reell, so gibt es unendlich viele Brüche $\frac{p}{q}$ mit*

$$|\alpha - \frac{p}{q}| < \frac{1}{q^2}.$$

Bemerkungen:

(1) Ist α reell und q eine natürliche Zahl, so gibt es eine ganze Zahl m mit

$$\frac{m}{q} \leq \alpha < \frac{m+1}{q}.$$

Dann gilt

$$|\alpha - \frac{m}{q}| \leq \frac{1}{2q} \quad \text{oder} \quad |\alpha - \frac{m+1}{q}| \leq \frac{1}{2q}.$$

Setzt man dann $p = m$ bzw. $p = m + 1$, so erhält man

$$|\alpha - \frac{p}{q}| \leq \frac{1}{2q}.$$

Ist q eine 10-er Potenz, so hat man die Approximation durch Dezimalbrüche. Die Kettenbruchapproximation ist aber wesentlich besser.

(2) Ist $\alpha = \frac{a}{b}$ rational, so folgt aus

$$0 < |\alpha - \frac{p}{q}| < \frac{1}{q^2} \quad \text{wegen} \quad \alpha - \frac{p}{q} = \frac{aq - bp}{bq}$$

und $|aq - bp| \in \mathbf{N}$ sofort sofort $q < b$. Also können nur endlich viele $\frac{p}{q}$ die Ungleichung erfüllen.

Wir geben nun noch einige Verbesserungen der obigen Folgerung an.

SATZ. *Von zwei aufeinanderfolgenden Näherungsbrüchen $\frac{p}{q} = \frac{p_{n-1}}{q_{n-1}}$ oder $\frac{p}{q} = \frac{p_n}{q_n}$ an die reelle Zahl α genügt zumindest eine der Ungleichung*

$$|\alpha - \frac{p}{q}| < \frac{1}{2q^2}.$$

Beweis: Wir nehmen an,

$$|\alpha - \frac{p_{n-1}}{q_{n-1}}| \geq \frac{1}{2q_{n-1}^2} \quad \text{und} \quad |\alpha - \frac{p_n}{q_n}| \geq \frac{1}{2q_n^2}.$$

Da α zwischen $\frac{p_{n-1}}{q_{n-1}}$ und $\frac{p_n}{q_n}$ liegt, ist

$$|\frac{p_{n-1}}{q_{n-1}} - \frac{p_n}{q_n}| = |\alpha - \frac{p_{n-1}}{q_{n-1}}| + |\alpha - \frac{p_n}{q_n}|$$

und damit

$$\frac{1}{q_{n-1}q_n} = \frac{|p_{n-1}q_n - p_nq_{n-1}|}{q_{n-1}q_n} = |\frac{p_{n-1}}{q_{n-1}} - \frac{p_n}{q_n}| = |\alpha - \frac{p_{n-1}}{q_{n-1}}| + |\alpha - \frac{p_n}{q_n}| \geq \frac{1}{2q_{n-1}^2} + \frac{1}{2q_n^2} = \frac{q_{n-1}^2 + q_n^2}{2q_{n-1}^2q_n^2},$$

was sofort $2q_{n-1}q_n \geq q_{n-1}^2 + q_n^2$ und damit $0 \geq (q_{n-1} - q_n)^2$, also $q_n = q_{n-1}$ liefert. Dies kann nur im Fall $n = 1$ auftreten, außerdem werden dann aus den Ungleichungen Gleichungen, d.h.

$$\alpha - \frac{p_0}{q_0} = \alpha - a_0 = \frac{1}{2} \quad \text{und} \quad \frac{p_1}{q_1} - \alpha = a_0 + 1 - \alpha = \frac{1}{2}.$$

Nun hat aber die Zahl $\alpha = a_0 + \frac{1}{2}$ die Kettenbruchentwicklung $\alpha = [a_0, 2]$ mit den Näherungsbrüchen $\frac{p_0}{q_0} = a_0$ und $\frac{p_1}{q_1} = \frac{2a_0+1}{2}$, so daß auch dieser Fall nicht möglich ist. ■

Beispiel: Die Kettenbruchentwicklung von

$$\alpha = \frac{509473}{788342},$$

das zufällig gewählt wurde, illustriert den letzten Satz.

i	a_i	p_i	q_i	$\frac{p_i}{q_i}$	$q_i^2(\alpha - \frac{p_i}{q_i})$
0	0	0	1	0.0000000000	+0.646259
1	1	1	1	1.0000000000	-0.353741
2	1	1	2	0.5000000000	+0.585035
3	1	2	3	0.6666666667	-0.183670
4	4	9	14	0.6428571429	+0.666736
5	1	11	17	0.6470588235	-0.231190
6	3	42	65	0.6461538462	+0.443672
7	1	53	82	0.6463414634	-0.555444
8	1	95	147	0.6462585034	+0.007645
9	130	12403	19192	0.6462588579	-0.243448
10	4	49707	76915	0.6462588572	+0.097566
11	10	509473	788342	0.6462588572	+0.000000

SATZ. *Ist α irrational reell, so erfüllt mindestens einer von drei aufeinanderfolgenden Näherungsbrüchen*

$$\frac{p}{q} \in \{\frac{p_{n-1}}{q_{n-1}}, \frac{p_n}{q_n}, \frac{p_{n+1}}{q_{n+1}}\}$$

die Ungleichung

$$|\alpha - \frac{p}{q}| < \frac{1}{\sqrt{5}q^2}.$$

Beweis: Angenommen, dies wäre nicht der Fall. Dann gilt:

$$\frac{1}{q_n q_{n+1}} = |\alpha - \frac{p_n}{q_n}| + |\alpha - \frac{p_{n+1}}{q_{n+1}}| \geq \frac{1}{\sqrt{5}q_n^2} + \frac{1}{\sqrt{5}q_{n+1}^2},$$

da die beiden Näherungsbrüche auf verschiedenen Seiten von α liegen. Dies liefert

$$\sqrt{5} \cdot \frac{q_{n+1}}{q_n} \geq (\frac{q_{n+1}}{q_n})^2 + 1 \text{ und nach quadratischer Ergänzung } (\frac{q_{n+1}}{q_n} - \frac{1}{2}\sqrt{5})^2 \leq \frac{1}{4},$$

also

$$|\frac{q_{n+1}}{q_n} - \frac{1}{2}\sqrt{5}| \leq \frac{1}{2} \quad \text{und insbesondere} \quad \frac{q_{n+1}}{q_n} \leq \frac{1+\sqrt{5}}{2}.$$

Da $\frac{q_{n+1}}{q_n}$ rational ist, folgt

$$\frac{q_{n+1}}{q_n} < \frac{1+\sqrt{5}}{2}.$$

Genauso findet man

$$\frac{q_n}{q_{n-1}} < \frac{1+\sqrt{5}}{2}.$$

Nun ist

$$1 + \frac{q_{n-1}}{q_n} = \frac{q_n + q_{n-1}}{q_n} \leq \frac{a_{n+1}q_n + q_{n-1}}{q_n} = \frac{q_{n+1}}{q_n} < \frac{1+\sqrt{5}}{2},$$

also

$$\frac{q_{n-1}}{q_n} < \frac{\sqrt{5}-1}{2}.$$

Dies ergibt

$$1 = \frac{q_n}{q_{n-1}} \cdot \frac{q_{n-1}}{q_n} < \frac{\sqrt{5}+1}{2}\frac{\sqrt{5}-1}{2} = 1,$$

ein Widerspruch. Also ist die Annahme falsch und die Behauptung folgt. ■

FOLGERUNG. *Ist α eine irrationale reelle Zahl, so gibt es unendlich viele $\frac{p}{q} \in \mathbf{Q}$ mit*

$$|\alpha - \frac{p}{q}| < \frac{1}{\sqrt{5}q^2}.$$

Wir wollen jetzt an einem Beispiel zeigen, daß die Konstante $\sqrt{5}$ in der Folgerung nicht verbessert werden kann.

Beispiel: Die Zahl $\alpha = \frac{\sqrt{5}-1}{2}$ genügt der Gleichung $f = x^2 + x - 1 = 0$. Sei $\frac{b_n}{c_n}$ eine Folge rationaler Zahlen mit

$$|\alpha - \frac{b_n}{c_n}| < \frac{1}{Ac_n^2}$$

für alle n. Nach dem Mittelwertsatz der Differentialrechnung gibt es ein ξ_n zwischen α und $\frac{b_n}{c_n}$ mit

$$\frac{f(\alpha) - f(\frac{b_n}{c_n})}{\alpha - \frac{b_n}{c_n}} = f'(\xi_n),$$

also

$$\frac{1}{c_n^2} \leq |\alpha - \frac{b_n}{c_n}||2\xi_n + 1| \leq \frac{1}{Ac_n^2}|2\xi_n + 1|,$$

d.h.

$$A \leq |2\xi_n + 1|.$$

Wegen $\lim \xi_n = \alpha$ folgt

$$A \leq \sqrt{5},$$

was wir zeigen wollten.

Der folgende Satz gibt ein wichtiges (hinreichendes) Kriterium an, wann ein Bruch Näherungsbruch in der Kettenbruchentwicklung einer Zahl ist.

SATZ. *Ist* $\alpha \in \mathbf{R}$ *und* $\frac{p}{q} \in \mathbf{Q}$ *mit*

$$|\alpha - \frac{p}{q}| < \frac{1}{2q^2},$$

so ist $\frac{p}{q}$ *Näherungsbruch in der Kettenbruchentwicklung von* α.

Beweis:

- Wir können schreiben

$$\alpha - \frac{p}{q} = \frac{\varepsilon\theta}{2q^2} \text{ mit } \varepsilon = \pm 1 \text{ und } 0 < \theta < 1.$$

- Da wir zeigen wollen, daß $\frac{p}{q}$ Näherungsbruch für α ist, bilden wir die Kettenbruchentwicklung von $\frac{p}{q}$:

$$\frac{p}{q} = [a_0, a_1, \ldots, a_n].$$

  Indem wir eventuell a_n durch $a_n - 1, 1$ ersetzen und dadurch die Länge der Kettenbruchdarstellung von $\frac{p}{q}$ um 1 erhöhen, können wir erreichen, daß gilt

$$\varepsilon = (-1)^n.$$

  Seien $\frac{p_i}{q_i}$ die zugehörigen Näherungsbrüche.
- Da der Fall $\alpha = \frac{p}{q} = \frac{p_n}{q_n}$ trivial ist, können wir ihn ausschließen und somit definieren:

$$x = -\frac{\alpha q_{n-1} - p_{n-1}}{\alpha q_n - p_n}.$$

  Dann gilt

$$\alpha = \frac{x p_n + p_{n-1}}{x q_n + q_{n-1}}$$

  und mit den üblichen Formeln für Kettenbrüche

$$\alpha = [a_0, a_1, \ldots, a_n, x].$$

- Nun haben wir mit entsprechenden Kettenbruchformeln

$$\frac{\varepsilon\theta}{2q^2} = \frac{(-1)^n\theta}{2q_n^2} = \alpha - \frac{p}{q} = \alpha - \frac{p_n}{q_n} = \frac{(-1)^n}{q_n(xq_n + q_{n-1})}.$$

  Es folgt

$$\frac{\theta}{2q_n^2} = \frac{1}{q_n(xq_n + q_{n-1})}$$

  und damit $xq_n - q_{n-1} = \frac{2}{\theta}q_n$, also

$$x = \frac{2}{\theta} - \frac{q_{n-1}}{q_n}.$$

  Mit der Voraussetzung folgt $x > 1$, so daß wegen der Eindeutigkeit der Kettenbruchentwicklung $\alpha = [a_0, a_1, \ldots, a_n, x]$ tatsächlich der Anfang der Kettenbruchentwicklung von α ist. Also ist auch $\frac{p}{q}$ Näherungsbruch für α. ■

Das folgende Beispiel zeigt, daß man das Kriterium $|\alpha - \frac{p}{q}| < \frac{1}{2q^2}$ nicht durch eine Bedingung $|\alpha - \frac{p}{q}| < \frac{1}{(2-\varepsilon)q^2}$ ersetzen kann mit einem festen $\varepsilon > 0$.

Beispiel: Wir betrachten für $m \geq 1$ die rationale Zahl

$$\alpha = \frac{2m^2 + m + 2}{2m^3}.$$

Man rechnet leicht nach, daß gilt

$$[0, m-1, 2] < \alpha < [0, m-1, 3].$$

Für $x > 0$ ist die Funktion $x \mapsto [0, m-1, x]$ streng monoton steigend, also hat die Kettenbruchentwicklung von α die Gestalt

$$\alpha = [0, m-1, 2, \dots].$$

Nun gilt

$$|\alpha - \frac{1}{m}| = \alpha - \frac{1}{m} = \frac{1}{2m^2} + \frac{1}{m^3} = \frac{1}{(2 - \frac{4}{m+2})m^2} < \frac{1}{m^2} \quad \text{für} \quad m \geq 3,$$

aber $\frac{1}{m} = [0, m] = [0, m-1, 1]$ ist kein Näherungsbruch von α.

6. Äquivalente reelle Zahlen und der Satz von Serret

Wir nennen zwei reelle Zahlen α und β äquivalent, wenn gilt

$$\alpha = \frac{A\beta + B}{C\beta + D} \text{ mit Zahlen } A, B, C, D \in \mathbf{Z} \text{ und } AD - BC = \pm 1.$$

LEMMA. *Die eben definierte Relation ist eine Äquivalenzrelation.*
Beweis: Reflexivität: Wegen

$$\alpha = \frac{1 \cdot \alpha + 0}{0 \cdot \alpha + 1}$$

ist α zu sich selbst äquivalent.
Symmetrie: Ist α äquivalent zu β, d.h. gibt es $A, B, C, D \in \mathbf{Z}$ mit $AD - BC = \pm 1$ und

$$\alpha = \frac{A\beta + B}{C\beta + D},$$

so folgt sofort

$$\beta = \frac{(-D)\alpha + B}{C\alpha - A},$$

also ist auch β äquivalent zu α.
Transitivität: Aus

$$\alpha = \frac{A_1\beta + B_1}{C_1\beta + D_1} \text{ und } \beta = \frac{A_2\gamma + B_2}{C_2\gamma + D_2}$$

folgt

$$\alpha = \frac{(A_1A_2 + B_1C_2)\gamma + (A_1B_2 + B_1D_2)}{(C_1A_2 + D_1C_2)\gamma + (C_1B_2 + D_1D_2)},$$

was zusammen mit

$$(A_1A_2 + B_1C_2)(C_1B_2 + D_1D_2) - (A_1B_2 + B_1D_2)(C_1A_2 + D_1C_2) = (A_1D_1 - B_1C_1)(A_2D_2 - B_2C_2)$$

sofort die Transitivität der Relation zeigt. ■

Bemerkung: Wir geben alternativ eine algebraische Formulierung obiger Äquivalenzrelation: Die Matrizen

$$\mathrm{GL}_2(\mathbf{Z}) = \{\begin{pmatrix} A & B \\ C & D \end{pmatrix} : A, B, C, D \in \mathbf{Z}, AD - BC = \pm 1\}$$

bilden mit der Matrizenmultiplikation als Verknüpfung eine Gruppe. Die Abbildung

$$\mathrm{GL}_2(\mathbf{Z}) \times \mathbf{R} \to \mathbf{R}, \quad \begin{pmatrix} A & B \\ C & D \end{pmatrix} \cdot \alpha = \frac{A\alpha + B}{C\alpha + D}$$

definiert eine Operation der Gruppe $\mathrm{GL}_2(\mathbf{Z})$ auf $\mathbf{R}$. Zwei reelle Zahlen α und β sind genau dann äquivalent, wenn $g \in \mathrm{GL}_2(\mathbf{Z})$ existiert mit $\alpha = g \cdot \beta$, d.h. wenn α und β in der gleichen $\mathrm{GL}_2(\mathbf{Z})$-Bahn liegen.

Beispiele:

(1) Für $\alpha \in \mathbf{R}$ und $m \in \mathbf{Z}$ ist α wegen $-\alpha = \frac{1\cdot\alpha+0}{0\cdot\alpha-1}$ und $\alpha + m = \frac{1\cdot\alpha+m}{0\cdot\alpha+1}$ äquivalent zu $-\alpha$ und zu $\alpha + m$.

(2) Für $\alpha \in \mathbf{R}^*$ sind α und $\frac{1}{\alpha}$ wegen

$$\frac{1}{\alpha} = \frac{0 \cdot \alpha + 1}{1 \cdot \alpha + 0}$$

äquivalent.

LEMMA. *Die rationalen Zahlen* $\mathbf{Q}$ *bilden eine Äquivalenzklasse unter obiger Äquivalenzrelation.*

Beweis: Sei zunächst $\alpha = \frac{a}{b} \in \mathbf{Q}$ mit $a, b \in \mathbf{Z}$, $\mathrm{ggT}(a, b) = 1$. Mit dem erweiterten euklidischen Algorithmus findet man $C, D \in \mathbf{Z}$ mit $Ca + Db = 1$. Setzt man jetzt $A = b$, $B = -a$, so gilt $AD - BC = 1$ und

$$\frac{A\alpha + B}{C\alpha + D} = \frac{A\frac{a}{b} + B}{C\frac{a}{b} + D} = \frac{Aa + Bb}{Ca + Db} = 0,$$

d.h. α ist äquivalent zu 0. Da natürlich jede zu 0 äquivalente Zahl rational ist, folgt die Behauptung. ■

Der Satz von Serret charakterisiert an Hand der Kettenbruchentwicklungen, wann zwei reelle Zahlen äquivalent sind.

LEMMA. *Hat* $\alpha \in \mathbf{R} \setminus \mathbf{Q}$ *die Kettenbruchentwicklung* $\alpha = [a_0, a_1, a_2, \dots]$ *und ist* $\alpha_n = [a_n, a_{n+1}, a_{n+2}, \dots]$, *so ist* α *äquivalent zu* α_n.

Beweis: Wegen der Eindeutigkeit der Kettenbruchentwicklung gilt

$$\alpha = [a_0, a_1, \dots, a_{n-1}, \alpha_n].$$

Sind $\frac{p_i}{q_i}$ die Näherungsbrüche der Kettenbruchentwicklung von α, so gilt bekanntlich

$$\alpha = \frac{\alpha_n p_{n-1} + p_{n-2}}{\alpha_n q_{n-1} + q_{n-2}}.$$

Die Behauptung folgt nun aus $p_{n-1}q_{n-2} - p_{n-2}q_{n-1} = (-1)^n$. ■

FOLGERUNG. *Sind* $\alpha, \beta \in \mathbf{R} \setminus \mathbf{Q}$ *mit den Kettenbruchentwicklungen* $\alpha = [a_0, a_1, a_2, \dots]$ *und* $\beta = [b_0, b_1, b_2, \dots]$ *und existieren* m, n *mit*

$$a_m = b_n, \quad a_{m+1} = b_{n+1}, \quad a_{m+2} = b_{n+2}, \quad a_{m+3} = b_{n+3}, \quad \dots,$$

so sind α *und* β *äquivalent.*

Beweis: Das Lemma liefert, daß α äquivalent zu $\alpha_m = [a_m, a_{m+1}, a_{m+2}, \dots]$ und β äquivalent zu $\beta_n = [b_n, b_{n+1}, b_{n+2}, \dots]$ ist. Nach Voraussetzung gilt $\alpha_m = \beta_n$, was dann die Behauptung zeigt. ■

Wir wollen jetzt schrittweise die Umkehrung der Folgerung zeigen.

LEMMA. *Sei* $\alpha = [a_0, a_1, a_2, \dots] = [a_0, a_1, \dots, a_{n-1}, \alpha_n]$ *mit* $\alpha_n = [a_n, a_{n+1}, a_{n+2}, \dots]$. *Ist*

$$\beta = \frac{A\alpha + B}{C\alpha + D} \quad \textit{mit} \quad AD - BC = \pm 1 \quad \textit{und} \quad A, B, C, D \in \mathbf{Z},$$

so ist

$$\beta = \frac{A_n\alpha_n + B_n}{C_n\alpha_n + D_n}$$

mit

$$A_n = Ap_{n-1} + Bq_{n-1}, \qquad B_n = Ap_{n-2} + Bq_{n-2},$$
$$C_n = Cp_{n-1} + Dq_{n-1}, \qquad D_n = Cp_{n-2} + Dq_{n-2}$$

und $A_nD_n - B_nC_n = \pm 1$. *Ist (eventuell nach Ersetzen von* A, B, C, D *durch* $-A, -B, -C, -D$*) o.E.* $C\alpha + D > 0$, *so gilt für alle hinreichend großen* n *die Abschätzung*

$$0 < D_n < C_n \quad \textit{und} \quad \alpha_n > 1.$$

Beweis: Die ersten Aussagen folgen durch Einsetzen aus

$$\alpha = \frac{\alpha_n p_{n-1} + p_{n-2}}{\alpha_n q_{n-1} + q_{n-2}}$$

und $A_nD_n - B_nC_n = (p_{n-1}q_{n-2} - p_{n-2}q_{n-1})(AD - BC)$. Weiter gilt

$$C_n = Cp_{n-1} + Dq_{n-1} = q_{n-1}(C \cdot \frac{p_{n-1}}{q_{n-1}} + D),$$
$$D_n = Cp_{n-2} + Dq_{n-2} = q_{n-2}(C \cdot \frac{p_{n-2}}{q_{n-2}} + D).$$

Da $\frac{p_{n-1}}{q_{n-1}}$ und $\frac{p_{n-2}}{q_{n-2}}$ gegen α konvergieren und $q_{n-1} > q_{n-2}$ gilt, folgt auch die letzte Behauptung. Die Aussage $\alpha_n > 1$ gilt für alle $n \geq 1$. ■

LEMMA. *Gilt*

$$\beta = \frac{A\alpha + B}{C\alpha + D}$$

mit $A, B, C, D \in \mathbf{Z}$, $AD - BC = \pm 1$ *und* $\alpha > 1$, $C > D > 0$, *so ist* $\alpha = \beta_n$ *für ein* n.

Beweis: Wir entwickeln $\frac{A}{C}$ in einen Kettenbruch:

$$\frac{A}{C} = [a_0, a_1, \ldots, a_{n-1}] \quad \text{mit Näherungsbrüchen} \quad \frac{p_0}{q_0}, \ldots, \frac{p_{n-1}}{q_{n-1}}.$$

Wir können die Länge des Kettenbruchs so einrichten, daß

$$p_{n-1}q_{n-2} - p_{n-2}q_{n-1} = AD - BC = \pm 1$$

gilt, indem wir eventuell $[a_0, a_1, \ldots, a_{n-1}]$ durch $[a_0, a_1, \ldots, a_{n-1} - 1, 1]$ ersetzen. Nun ist $A = p_{n-1}$ und $C = q_{n-1}$. Wir erhalten also

$$AD - BC = Aq_{n-2} - Cp_{n-2} \quad \text{und damit} \quad A(D - q_{n-2}) = C(B - p_{n-2}).$$

Wegen $\mathrm{ggT}(A, C) = 1$ gibt es $m \in \mathbf{Z}$ mit

$$D - q_{n-2} = mC, \quad B - p_{n-2} = mA.$$

Nun ist $0 < q_{n-2} < q_{n-1} = C$. Die Voraussetzung $0 < D < C$ liefert dann $0 < q_{n-2} + mC < C$ wegen $0 < q_{n-2} < C$ sofort $m = 0$ und damit dann $D = q_{n-2}$ und $B = p_{n-2}$. Also erhalten wir

$$\beta = \frac{p_{n-1}\alpha + p_{n-2}}{q_{n-1}\alpha + q_{n-2}} = [a_0, a_1, \ldots, a_{n-1}, \alpha].$$

Wegen $\alpha > 1$ kann man hier die Kettenbruchentwicklung von α einsetzen und erhält mit der Eindeutigkeit der Kettenbruchentwicklung sofort $\alpha = \beta_n$. ■

SATZ (Serret). *Seien* α *und* β *reelle irrationale Zahlen mit den Kettenbruchentwicklungen* $[a_0, a_1, a_2, \ldots]$ *und* $[b_0, b_1, b_2, \ldots]$. *Genau dann sind* α *und* β *äquivalent, wenn es Zahlen* m *und* n *gibt mit*

$$a_m = b_n, \quad a_{m+1} = b_{n+1}, \quad a_{m+2} = b_{n+2}, \quad a_{m+3} = b_{n+3}, \quad a_{m+4} = b_{n+4}, \quad \ldots$$

Beweis: Gilt $a_{m+i} = b_{n+i}$ für alle $i \geq 0$, so haben wir bereits gesehen, daß α und β äquivalent sind. Wir zeigen die Umkehrung: Sei β äquivalent zu α. Ist $\alpha_m = [a_m, a_{m+1}, \ldots]$, so gibt es für genügend großes m eine Darstellung

$$\beta = \frac{A_m\alpha_m + B_m}{C_m\alpha_m + D_m}$$

mit $A_m, B_m, C_m, D_m \in \mathbf{Z}$, $A_mD_m - B_mC_m = 1$ und $0 < D_m < C_m$, $\alpha_m > 1$. Das letzte Lemma liefert dann die Existenz einer Zahl n mit $\alpha_m = \beta_n$ und damit $a_{m+i} = b_{n+i}$ für alle $i \geq 0$, was zu beweisen war. ■

7. Angriffe auf das RSA-Kryptosystem mit Kettenbrüchen

Bemerkungen:

(1) Beim RSA-Kryptosystem hat man einen öffentlichen Schlüssel (N, e) und dazu einen passenden privaten Schlüssel (N, d). Dabei ist $N = pq$ das Produkt zweier verschiedener ungerader Primzahlen, außerdem gilt

$$ed \equiv 1 \bmod \varphi(N) \quad (\text{und } \varphi(N) = \varphi(pq) = (p-1)(q-1)).$$

Daten werden in Zahlen modulo N übersetzt, die Verschlüsselungsabbildung ist dann

$$E_{(N,e)} : \mathbf{Z}/(N) \to \mathbf{Z}/(N), \quad x \mapsto x^e \bmod N$$

mit der Umkehrabbildung, der Entschlüsselungsabbildung

$$D_{(N,d)} : \mathbf{Z}/(N) \to \mathbf{Z}/(N), \quad y \mapsto y^d \bmod N.$$

(2) Für die Sicherheit des RSA-Verschlüsselungssystems ist es wesentlich, daß man aus dem (allgemein zugänglichen) öffentlichen Schlüssel (N, e) den privaten Schlüssel (N, d) praktisch nicht herleiten kann. Dies führt zu einigen unmittelbaren Konsequenzen:

(a) N sollte nicht faktorisiert werden können, da man wegen

$$d \equiv \frac{1}{d} \bmod (p-1)(q-1)$$

sonst sofort (N, d) hätte.

(b) d sollte nicht zu klein sein, so daß man durch Probieren von $d = 3, 5, 7, \ldots$ auf d schließen kann. (Man kann beispielsweise testen, ob $a^{ed} \equiv 2 \bmod N$ gilt für $a = 2, 3, 5, \ldots$. Ist es wahrscheinlich, daß man den privaten Schlüssel gefunden hat, kann man mit einem früher beschrieben Verfahren versuchen, N zu faktorisieren.)

(c) Im folgenden werden wir sehen, daß im Fall von $d \leq N^{\frac{1}{4}}$ (und ein paar Voraussetzungen an p und q) die Zahl N mit einem Kettenbruchverfahren faktorisiert werden kann.

Sei (N, e) ein öffentlicher RSA-Schlüssel, (N, d) der zugehörige private Schlüssel. Wegen $ed \equiv 1 \bmod \varphi(N)$ ist $k = \frac{ed-1}{\varphi(N)}$ eine natürliche Zahl und damit

$$ed - k\varphi(N) = 1,$$

also

$$\frac{e}{\varphi(N)} - \frac{k}{d} = \frac{1}{d\varphi(N)}.$$

Ist N groß, so wird also $\frac{e}{\varphi(N)} \approx \frac{k}{d}$ gelten.

Beispiel: Mit den obigen Bezeichnungen werden bei Vorgabe von p, q und e die weiteren Größen berechnet:

$$\begin{aligned}
p &= 19765462153461536357165341 3463 \\
q &= 33826349863248752637849652 8509 \\
N &= 66859343801179203617996040043272414073115678811065543916667 \\
\varphi(N) &= 66859343801179203617996040042736495952948575921115393974696 \\
e &= 3 \\
d &= 44572895867452802411997360028490997301965717280743595983131 \\
k &= 2 \\
\frac{k}{d} &= 0.44870317736308515175476686528315966148414206777661005714375164800333485 \cdot 10^{-58} \\
\frac{e}{\varphi(N)} &= 0.44870317736308515175476686528315966148414206777661005714375500357902444 \cdot 10^{-58}
\end{aligned}$$

Wir wollen jetzt zunächst eine Abschätzung für $\varphi(N)$ geben.

LEMMA. *Ist $N = pq$ Produkt zweier ungerader Primzahlen $p < q$, $s = p + q$, so ist $\varphi(N) = N + 1 - s$ und es gelten die Abschätzungen:*

(1)
$$\lfloor\sqrt{4N}\rfloor + 1 \leq s \leq \frac{1}{3}N + 3 \quad \textit{und} \quad \frac{2}{3}N - 2 \leq \varphi(N) \leq N - \lfloor\sqrt{4N}\rfloor.$$

(2) *Ist $p < q < 2p$, so gilt*
$$2\sqrt{N} < s < \frac{3}{2}\sqrt{2}\cdot\sqrt{N} \quad \textit{und} \quad N + 1 - 2\sqrt{N} < \varphi(N) < N + 1 - \frac{3}{2}\sqrt{2}\sqrt{N}.$$

(3) *Ist $p < q < 3.9p$, so gilt*
$$2\sqrt{N} < s < 2.48121125\sqrt{N} \quad \textit{und} \quad N + 1 - 2\sqrt{N} < \varphi(N) < N + 1 - 2.48121145\sqrt{N}.$$

Beweis:

(1) Es gilt
$$\varphi(N) = (p-1)(q-1) = pq - p - q + 1 = N + 1 - (p+q) = N + 1 - s.$$

(2) Wir schreiben $q = x\sqrt{N}$, $p = \frac{1}{x}\sqrt{N}$ mit $x > 1$. Dann ist
$$s = (x + \frac{1}{x})\sqrt{N}.$$
Bei festem N ist die Funktion s in Abhängigkeit von x streng monoton wachsend wegen $\frac{\partial s}{\partial x} = (1 - \frac{1}{x^2})\sqrt{N}$. Also erhalten wir
$$s > 2\sqrt{N} \quad \text{und} \quad s \geq \lfloor\sqrt{4N}\rfloor + 1.$$
Die obere Schranke kommt von einer möglichen Zerlegung $N = 3q$, also
$$s \leq 3 + \frac{1}{3}N.$$
Somit
$$2\sqrt{N} < \lfloor\sqrt{4N}\rfloor + 1 \leq s \leq 3 + \frac{1}{3}N.$$
Für die Eulersche φ-Funktion impliziert dies
$$N + 1 - 2\sqrt{N} > N - \lfloor\sqrt{4N}\rfloor \geq \varphi(N) \geq \frac{2}{3}N - 2.$$

(3) Die Bedingung $p < q < \mu p$ ist dann äquivalent mit $\frac{1}{x} < x < \frac{\mu}{x}$, also $1 < x < \sqrt{\mu}$. Da die Funktion
$$s = p + q = (x + \frac{1}{x})\sqrt{N}$$
(bei festem N) für $x > 0$ streng monoton steigend ist, folgt die Abschätzung
$$2\sqrt{N} < s < (\sqrt{\mu} + \frac{1}{\sqrt{\mu}})\sqrt{N}.$$
Wählt man $\mu = 2$ ergibt sich $s < \frac{3}{2}\sqrt{2}\sqrt{N}$, bei der Wahl von $\mu = 3.9$ ergibt sich $s < 2.48121145\sqrt{N}$. ■

Liegen die Primzahlen p und q eines RSA-Schlüssels nicht zu weit auseinander, erhalten wir also
$$\frac{k}{d} \approx \frac{e}{\varphi(N)} \approx \frac{e}{N} \approx \frac{e}{N - \lfloor\sqrt{4N}\rfloor}.$$

Wir betrachten ein Beispiel:

Beispiel: Wir nehmen die 20-stellige Zahl
$$N = 32758659611582346361 = 4072951217 \cdot 8042978633$$
mit
$$\varphi(N) = (p-1)(q-1) = 32758659599466416512.$$

1. Beispiel:

$d = 6231628522119231213 \approx N^{0.963}$, $e = 28554566343820204901 \approx N^{0.997}$, $k = 5431890444864056951$.

$$\begin{aligned}
\frac{k}{d} &= [0,1,6,1,3,1,4,3,1,12,7,2,\\
&\quad 2,17,1,2,12,7,88,1,1,3,1,6,104,1,1,1,1,1,1,3,1,1,2,8,1,3]\\
\frac{e}{\varphi(N)} &= [0,1,6,1,3,1,4,3,1,12,7,2,\\
&\quad 2,17,1,2,12,7,88,1,1,3,1,6,104,1,1,1,1,1,1,3,1,1,2,8,1,3,5]\\
\frac{e}{N} &= [0,1,6,1,3,1,4,3,1,12,5,1,\\
&\quad 15,4,2,1,12,2,1,4,2,9,6,3,1,37,1,9,1,1,3,1,1,1,1,1,2,1,2,3,1,22]\\
\frac{e}{N-\lfloor\sqrt{4N}\rfloor} &= [0,1,6,1,3,1,4,3,1,12,7,3,\\
&\quad 3,2,1,23,2,1,2,2,5,9,21,1,1,5,1,1,1,1,1,24,1,10,1,1,1,14,1,5,1,2]
\end{aligned}$$

Die betrachteten Kettenbruchentwicklungen fangen gleich an. $\frac{k}{d}$ ist Näherungsbruch von $\frac{e}{\varphi(N)}$.

2. Beispiel:

$$d = 5723518015 \approx N^{0.500}, \quad e = 2507877484912269663 9 \approx N^{0.994}, \quad k = 4381706132$$

$$\begin{aligned}
\frac{k}{d} &= [0,1,3,3,1,3,3,1,1,2,3,1,10,1,2,1,\\
&\quad 3,1,1,2,2,2,7,1,6,3]\\
\frac{e}{\varphi(N)} &= [0,1,3,3,1,3,3,1,1,2,3,1,10,1,2,1,\\
&\quad 3,1,1,2,2,2,7,1,6,2,1,5723518212]\\
\frac{e}{N} &= [0,1,3,3,1,3,3,1,1,2,3,1,9,1,2,1,\\
&\quad 1,1,21,1,5,16,7,10,1,143,1,1,4,1,2,2,1,1,1,7,3,2,2,5,1,3,2]\\
\frac{e}{N-\lfloor\sqrt{4N}\rfloor} &= [0,1,3,3,1,3,3,1,1,2,3,1,10,1,2,13,\\
&\quad 4,1,3,3,8,2,7,2,2,16,6,1,1,1,2,1,1,2,10,2,2,1,4,157]
\end{aligned}$$

$\frac{k}{d}$ ist Näherungsbruch von $\frac{e}{\varphi(N)}$. Für $\frac{k}{d}$ ist $\frac{e}{N-\lfloor\sqrt{4N}\rfloor}$ eine bessere Approximation als $\frac{e}{N}$.

3. Beispiel:

$$d = 118575 \approx N^{0.260}, \quad e = 30770071517687295567 \approx N^{0.999}, \quad k = 111377$$

$$\begin{aligned}
\frac{k}{d} &= [0,1,15,2,8,1,6,1,5,8]\\
\frac{e}{\varphi(N)} &= [0,1,15,2,8,1,6,1,5,7,1,276269530672286]\\
\frac{e}{N} &= [0,1,15,2,8,1,6,1,5,20,\\
&\quad 6,1,6,1,1,1,15,5,2,1,1,1,1,4,22,3,2,1,1,22,2,6,1,3,1,2,2,3,1,1,1,2]\\
\frac{e}{N-\lfloor\sqrt{4N}\rfloor} &= [0,1,15,2,8,1,6,1,5,8,\\
&\quad 3,1,1,2,2,2,4,3,1,7,6,1,4,1,1,3,1,1,1,2,4,1,1,1,2,2,1,1,4,1,1,1,2,1,8,2,28]
\end{aligned}$$

Wir sehen jetzt, daß $\frac{k}{d}$ Näherungsbruch in der Kettenbruchentwicklung von $\frac{e}{N-\lfloor\sqrt{4N}\rfloor}$ ist. Kennt man den öffentlichen Schlüssel (N,e), kann man also $\frac{k}{d}$ und damit den privaten Schlüssel (N,d) bestimmen.

Wir verallgemeinern das letzte Beispiel in einem Satz, dessen erste Version auf M. Wiener zurückgeht.

SATZ. *Sei $N = pq > 3000$ mit $p < q < 3.9p$, seien e,d natürliche Zahlen mit $1 < e,d < \varphi(N)$ und $ed \equiv 1 \bmod N$, sowie $k = \frac{ed-1}{\varphi(N)}$. Dann gilt die Implikation*

$$d \le N^{\frac{1}{4}} \quad \Longrightarrow \quad \left|\frac{e}{N-\lfloor\sqrt{4N}\rfloor} - \frac{k}{d}\right| < \frac{1}{2d^2},$$

ist also $d \leq N^{0.25}$, so kommt $\frac{k}{d}$ als Näherungsbruch in der Kettenbruchentwicklung von $\frac{e}{N-\lfloor\sqrt{4N}\rfloor}$ vor.

Beweis:

(1) Das vorangegangene Lemma liefert mit $s = p + q$ die Abschätzung

$$\lfloor\sqrt{4N}\rfloor + 1 \leq s < 2.48121145\sqrt{N}.$$

(2) Im Fall $s = \lfloor\sqrt{4N}\rfloor + 1$ ist $\varphi(N) = N + 1 - s = N - \lfloor\sqrt{4N}\rfloor$ und damit

$$|\frac{e}{N-\lfloor\sqrt{4N}\rfloor} - \frac{k}{d}| \quad = \quad |\frac{e}{\varphi(N)} - \frac{k}{d}| = |\frac{e}{\varphi(N)} - \frac{e-\frac{1}{d}}{\varphi(N)}| = \frac{1}{d\varphi(N)}$$

Für $N \geq 10$ ist $2N^{\frac{1}{4}} + 2\sqrt{N} < N$, also

$$2d \leq 2N^{\frac{1}{4}} < N - 2\sqrt{N} \leq N - \lfloor\sqrt{4N}\rfloor = \varphi(N)$$

und damit

$$|\frac{e}{N-\lfloor\sqrt{4N}\rfloor} - \frac{k}{d}| = \frac{1}{d\varphi(N)} < \frac{1}{2d^2},$$

was gezeigt werden sollte.

(3) Im Fall $s > \lfloor\sqrt{4N}\rfloor + 1$ gilt

$$\begin{aligned} k(N - \lfloor\sqrt{4N}\rfloor) - ed &= k(N - \lfloor\sqrt{4N}\rfloor) - (1 + k\varphi(N)) \geq \\ &\geq k(N - \lfloor\sqrt{4N}\rfloor) - k - k\varphi(N) = \\ &= k(N - \lfloor\sqrt{4N}\rfloor - 1 - (N + 1 - s)) = k(s - \lfloor\sqrt{4N}\rfloor - 2) \geq 0 \end{aligned}$$

und damit

$$\frac{k}{d} \geq \frac{e}{N - \lfloor\sqrt{4N}\rfloor}.$$

Dann ist

$$\begin{aligned} |\frac{e}{N-\lfloor\sqrt{4N}\rfloor} - \frac{k}{d}| &= \frac{k}{d} - \frac{e}{N-\lfloor\sqrt{4N}\rfloor} < \frac{k}{d} - \frac{e}{N+1-2\sqrt{N}} = \\ &= \frac{e-\frac{1}{d}}{\varphi(N)} - \frac{e}{N+1-2\sqrt{N}} < \frac{e}{\varphi(N)} - \frac{e}{N+1-2\sqrt{N}} = \\ &= \frac{e((N+1-2\sqrt{N}) - (N+1-s))}{\varphi(N)(N+1-2\sqrt{N})} < \frac{s-2\sqrt{N}}{N+1-2\sqrt{N}} < \\ &< \frac{(2.48121145-2)\sqrt{N}}{N+1-2\sqrt{N}} = \frac{0.48121145\sqrt{N}}{N+1-2\sqrt{N}} \end{aligned}$$

Nun gilt

$$\begin{aligned} \frac{0.48121145\sqrt{N}}{N+1-2\sqrt{N}} \leq \frac{1}{2\sqrt{N}} \quad &\iff \quad 0.9624229N \leq N + 1 - 2\sqrt{N} \\ &\iff \quad 2\sqrt{N} - 1 \leq 0.0375771N \quad \iff \quad N \geq 2780 \end{aligned}$$

Damit folgt für $N \geq 3000$ und $d \leq N^{\frac{1}{4}}$ zunächst $d^2 \leq \sqrt{N}$ und damit

$$|\frac{e}{N-\lfloor\sqrt{4N}\rfloor} - \frac{k}{d}| < \frac{1}{2d^2},$$

was gezeigt werden sollte. ■

Wir geben jetzt noch konkret einen Algorithmus an, der versucht aus der Kettenbruchentwicklung von $\frac{e}{N-\lfloor\sqrt{4N}\rfloor}$ den privaten Schlüssel (N, d) zu berechnen. Dazu brauchen wir ein Lemma:

LEMMA. *Sei $N = pq$ mit ungeraden Primzahlen $p < q$, seien e, d natürliche Zahlen mit $1 < e, d < \varphi(N)$ und $ed \equiv 1 \bmod \varphi(N)$.*

(1) *Setzt man nacheinander*

$$k = \frac{ed-1}{\varphi(N)}, \quad s = N+1-\frac{ed-1}{k}, \quad D = s^2 - 4N,$$

so gilt

$$ed \equiv 1 \bmod k, \quad D \text{ ist ein Quadrat} \quad \text{und} \quad p = \frac{s-\sqrt{D}}{2}, \quad q = \frac{s+\sqrt{D}}{2}.$$

(2) *Sind umgekehrt k', d' natürliche Zahlen, setzt man*

$$s' = N+1-\frac{ed'-1}{k'}, \quad D' = s'^2 - 4N,$$

gilt $ed' \equiv 1 \bmod k'$, und ist D' ein Quadrat, so ist

$$p = \frac{s'-\sqrt{D'}}{2} \quad \text{und} \quad q = \frac{s'+\sqrt{D'}}{2}.$$

Beweis: 1. Wir haben $ed-1 = k\varphi(N)$, was sofort $k|ed-1$ zeigt. Dann ist

$$s = N+1-\frac{ed-1}{k} = N+1-\varphi(N) = pq+1-(p-1)(q-1) = p+q,$$

also

$$D = s^2 - 4N = (p+q)^2 - 4pq = (q-p)^2$$

ein Quadrat und

$$\frac{s-\sqrt{D}}{2} = \frac{(p+q)-(q-p)}{2} = p \quad \text{und} \quad \frac{s+\sqrt{D}}{2} = \frac{(p+q)+(q-p)}{2} = q,$$

was die erste Behauptung beweist.
2. Wegen $ed' \equiv 1 \bmod k'$ sind s' und D' ganze Zahlen und damit wegen $D' \equiv s' \bmod 2$ auch

$$p' = \frac{s'-\sqrt{D'}}{2} \quad \text{und} \quad q' = \frac{s'+\sqrt{D'}}{2}.$$

Wegen $D' = s'^2 - 4N$ ist $D' > s'^2$, also $1 \le p' \le q'$. Aus

$$p'q' = \frac{s'-\sqrt{D'}}{2} \cdot \frac{s'+\sqrt{D'}}{2} = \frac{s'^2 - D'}{4} = N$$

sieht man, daß es nur die beiden Möglichkeiten

$$p' = 1, q' = N \quad \text{oder} \quad p' = p, q' = q$$

gibt. Wäre $p' = 1, q' = N$, so wäre $s' = p' + q' = N+1$ und damit $ed' = 1$, was der Voraussetzung $e > 1$ widerspricht. Also bleibt nur die zweite Möglichkeit, womit die Behauptung bewiesen ist. ■

Algorithmus: Hat man einen öffentlichen RSA-Schlüssel (N, e) gegeben, kann man nach folgendem Verfahren versuchen, die Faktorisierung $N = pq$ und den privaten Schlüssel d zu finden:

(1) Bestimme die Näherungsbrüche

$$\frac{k_0}{d_0}, \quad \frac{k_1}{d_1}, \quad \frac{k_2}{d_2}, \quad \ldots, \quad \frac{k_n}{d_n}$$

der Kettenbruchentwicklung von

$$\frac{e}{N - \lfloor\sqrt{4N}\rfloor}$$

und setze $i = 0$

(2) Setze $i := i+1$. Ist $i > n$ beendige ohne Erfolg.

(3) Ist $ed_i \not\equiv 1 \bmod k_i$, gehe zu 2.

(4) Setze

$$s_i = N+1-\frac{ed_i-1}{k_i} \quad \text{und} \quad D_i = s_i^2 - 4N.$$

(5) Ist D_i kein Quadrat, gehe zu 2.

(6) Ist D_i ein Quadrat, gib

$$p = \frac{s_i - \sqrt{D_i}}{2}, \quad q = \frac{s_i + \sqrt{D_i}}{2} \quad \text{und} \quad d \equiv \frac{1}{e} \bmod (p-1)(q-1)$$

als gesuchte Lösung aus.

Bemerkung: Die Maple-Funktion kb_rsa realisiert den dargestellten Algorithmus.

Beispiele: In den folgenden Beispielen ist $\frac{\ln(d)}{\ln(N)} \approx 0.25$. Manchmal ist der Algorithmus erfolgreich, manchmal nicht.

1. Beispiel:

$$\begin{aligned}
N &= 82257510389380447225013058328054960189274459591361757147434743211245 63\\
p &= 63642133071342787856743220237688643\\
q &= 12925008389830339920376935931753144 1\\
\frac{q}{p} &= 2.030889\\
e &= 65798253881547077561033458280194449501897084997263138289422141262433 3\\
d &= 1111235133951504677\\
\frac{\ln(d)}{\ln(N)} &= .258110\\
\frac{e}{N - \lfloor\sqrt{4N}\rfloor} &= [0, 12, 1, 1, 169, 4, 1, 7, 3, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 7, 1, 1, 6, 24, 1, 8, 1, 2, 1, 8, 3, 1, 2, 1, 1, 2,\\
&\quad 211, 1, 6, 4, 15, 5, 6, 14, 2, 1, 32, 1, 5, 1, 1, 7, 7, 17, 1, 1, 2, 1, 1, 2, 21, 1, 2, 1, 32, 2, 4, 3, 37, 4,\\
&\quad 6, 9, 2, 2, 1, 1, 3, 5, 1, 7, 1, 4, 8, 2, 2, 3, 1, 40, 2, 1, 7, 10, 1, 1, 54, 1, 24, 1, 1, 5, 1, 1, 4, 3, 3, 1, 10,\\
&\quad 1, 5, 1, 61, 1, 5, 3, 2, 1, 10, 1, 1, 4, 3, 2, 2, 2, 8, 1, 1, 1, 7, 1, 3, 2, 4]\\
\frac{k}{d} &= [0, 12, 1, 1, 169, 4, 1, 7, 3, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 7, 1, 1, 6, 24, 1, 8, 1, 2, 1, 8, 3, 1, 2, 1, 1, 2,\\
&\quad 212]
\end{aligned}$$

Der Algorithmus hat Erfolg.

2. Beispiel:

$$\begin{aligned}
N &= 16267828224447286944047076614006829226087184613526395634552214561790 97\\
p &= 24257763286132280790960960610552741\\
q &= 67062358687238516099099348034732517\\
\frac{q}{p} &= 2.764573\\
e &= 93287748061254136097346812144579989544445924143659894076836187483092 3\\
d &= 489467349331406467\\
\frac{\ln(d)}{\ln(N)} &= .255590\\
\frac{e}{N - \lfloor\sqrt{4N}\rfloor} &= [0, 1, 1, 2, 1, 9, 2, 1, 1, 2, 12, 63, 1, 6, 3, 1, 2, 2, 1, 2, 5, 5, 1, 1, 1, 104, 34, 1, 1, 2, 1, 18, 1, 1, 1, 7,\\
&\quad 24, 1, 6, 1, 13, 2, 9, 63, 4, 15, 2, 2, 1, 22, 3, 5, 3, 9, 3, 1, 2, 1, 2, 4, 1, 1, 54, 1, 1, 1, 4, 70, 1, 6, 2, 1,\\
&\quad 29, 1, 2, 1, 2, 42, 4, 4, 3, 9, 5, 2, 1, 2, 17, 1, 20, 1, 2, 2, 3, 8, 1, 2, 1, 4, 4, 3, 2, 2, 6, 5, 2, 1, 5, 1, 2,\\
&\quad 3, 8, 29, 1, 2, 1, 22, 2, 24, 1, 1, 4, 3, 1, 4, 6]\\
\frac{k}{d} &= [0, 1, 1, 2, 1, 9, 2, 1, 1, 2, 12, 63, 1, 6, 3, 1, 2, 2, 1, 2, 5, 5, 1, 1, 1, 104, 34, 1, 1, 2, 1, 18, 1, 1, 1, 6]
\end{aligned}$$

Der Algorithmus hat keinen Erfolg.

3. Beispiel:

$$
\begin{aligned}
N &= 584848245832818966184145525652796528482850386898810478316658617713489 9\\
p &= 46936341332614722240707748528878477\\
q &= 12460456636112466525058138834595628 7\\
\frac{q}{p} &= 2.654757\\
e &= 43142442550884520582247716941939330380060285181477518135142620108715 1\\
d &= 699303130395894943\\
\frac{\ln(d)}{\ln(N)} &= .255775\\
\frac{e}{N - \lfloor\sqrt{4N}\rfloor} &= [0, 13, 1, 1, 3, 1, 17, 1, 13, 1, 1, 34, 4, 21, 2, 3, 3, 1, 3, 1, 1, 1, 2, 1, 1, 2, 1, 11, 1, 2, 62, 1, 17, 2, 3, 1,\\
&\quad 7, 1, 16, 1, 4, 1, 8, 1, 1, 1, 4, 1, 2, 1, 1, 3, 3, 4, 6, 2, 1, 2, 2, 1, 1, 15, 1, 1, 1, 14, 1, 9, 1, 3, 2, 56,\\
&\quad 2, 49, 1, 6, 1, 3, 1, 2, 4, 2, 4, 4, 1, 938, 1, 1, 22, 4, 2, 1, 11, 10, 2, 1, 1, 6, 2, 1, 1, 15, 249, 1, 1,\\
&\quad 22, 1, 1, 4, 1, 1, 1, 1, 2, 2, 3, 1, 5, 2, 12, 22, 4, 4, 2, 1, 5, 1, 1, 12, 2, 3, 1, 2, 1, 16]\\
\frac{k}{d} &= [0, 13, 1, 1, 3, 1, 17, 1, 13, 1, 1, 34, 4, 21, 2, 3, 3, 1, 3, 1, 1, 1, 2, 1, 1, 2, 1, 11, 1, 2, 62, 1, 17, 2, 4]
\end{aligned}
$$

Der Algorithmus hat Erfolg.

4. Beispiel:

$$
\begin{aligned}
N &= 628132752154033093821256475001725389032356982878193434482485231932 7191\\
p &= 54625846065374370175746762876463151\\
q &= 11498819650359373865157379979463804 1\\
\frac{q}{p} &= 2.105014\\
e &= 26722347180210576408089314460079188881804027372274115107185929368163 41\\
d &= 885468195135947261\\
\frac{\ln(d)}{\ln(N)} &= .257130\\
\frac{e}{N - \lfloor\sqrt{4N}\rfloor} &= [0, 2, 2, 1, 5, 1, 3, 2, 1, 1, 3, 12, 1, 1, 1, 2, 3, 1, 4, 1, 1, 5, 19, 65, 168, 1, 3, 1, 1, 1, 5, 1, 4, 3, 1, 1,\\
&\quad 2, 1, 2, 2, 27, 3, 8, 1, 2, 1, 1, 1, 1, 3, 4, 3, 1, 1, 2, 1, 5, 2, 1, 2, 7, 3, 1, 3, 1, 1, 8, 1, 1, 2, 1, 2, 1, 2,\\
&\quad 2, 2, 4, 2, 3, 1, 2, 1, 37, 1, 3, 1, 2, 1, 5, 20, 15, 1, 3, 1, 117, 1, 1, 1, 1, 1, 2, 6, 1, 2, 19, 1, 3, 2, 1, 2,\\
&\quad 1, 2, 20, 1, 3, 2, 9548, 3, 8, 1, 27, 1, 1, 1, 19, 12, 2, 1, 18, 2, 9, 1, 1, 1, 1, 1, 7, 2, 2, 2, 2, 1, 1, 1, 2, 4]\\
\frac{k}{d} &= [0, 2, 2, 1, 5, 1, 3, 2, 1, 1, 3, 12, 1, 1, 1, 2, 3, 1, 4, 1, 1, 5, 19, 65, 168, 1, 3, 1, 1, 1, 5, 1, 4, 3, 1, 1,\\
&\quad 2, 1, 3]
\end{aligned}
$$

Der Algorithmus hat keinen Erfolg.

5. Beispiel:

$$
\begin{aligned}
N &= 44153262380073646954383015297853068889355406107977111244330514010805 63\\
p &= 46937605553258338686840832705710967\\
q &= 94067990600787247070603426159263589\\
\frac{q}{p} &= 2.004107\\
e &= 93987850860134983992362099290873380769008696834223250341768016254 55\\
d &= 7189884310724546759\\
\frac{\ln(d)}{\ln(N)} &= .270755\\
\frac{e}{N-\lfloor\sqrt{4N}\rfloor} &= [0, 469, 1, 3, 2, 7, 1, 3, 3, 1, 3, 1, 608, 58, 1, 60, 6, 36, 1, 1, 11, 2, 1, 3, 1, 2, 2, 2, 4, 1, 1, 6,\\
&\quad 1, 1, 1, 1, 1, 1, 95, 1, 7, 3, 2, 6, 2, 4, 2, 1, 8, 1, 1, 90, 1, 1, 1, 5, 1, 24, 4, 2, 13, 1, 2, 1, 5, 6, 6,\\
&\quad 1, 3, 21, 27, 1, 9, 1, 9, 216, 1, 1, 8, 6, 5, 1, 6, 1, 1, 3, 1, 2, 6, 2, 1, 1, 11, 4, 2, 1, 23, 1, 1, 1, 1,\\
&\quad 6, 1, 124, 7, 3, 6, 57, 4, 13, 2, 1, 13, 1, 74]\\
\frac{k}{d} &= [0, 469, 1, 3, 2, 7, 1, 3, 3, 1, 3, 1, 608, 58, 1, 60, 6, 36, 1, 1, 11, 2, 1, 3, 1, 2, 2, 2]
\end{aligned}
$$

Der Algorithmus hat Erfolg. (Dies ist ein interessantes Beispiel.)

6. Beispiel:

$$
\begin{aligned}
N &= 59814262581111495012520362949654103505409384361061312972576470421892 69\\
p &= 52855629701729781274727756693095853\\
q &= 11316535801134156545705016789726407 3\\
\frac{q}{p} &= 2.141028\\
e &= 86786843645279280511129693026396951865865848025826625695467692956779 1\\
d &= 775134741803520975\\
\frac{\ln(d)}{\ln(N)} &= .256380\\
\frac{e}{N-\lfloor\sqrt{4N}\rfloor} &= [0, 6, 1, 8, 3, 1, 2, 1, 20, 1, 26, 1, 2, 1, 37, 1, 1, 4, 2, 1, 4, 1, 1, 1, 3, 5, 1, 1, 134, 1, 3, 4, 6, 1, 25,\\
&\quad 1, 5, 11, 1, 1, 13, 41, 1, 3, 1, 3, 2, 3, 1, 9, 1, 1, 10, 1, 2, 1, 4, 3, 2, 5, 3, 1, 1, 2, 10, 1, 1, 1, 1, 1, 1,\\
&\quad 35, 2, 1, 3, 1, 1, 6, 1, 2, 1, 3, 1, 151, 5, 1, 1, 2, 1, 2, 13, 1, 1, 96, 2, 1, 1, 7, 1, 1, 3, 3, 7, 9, 2, 4, 1,\\
&\quad 1, 5, 2, 1, 1, 1, 9, 9, 14, 5, 5, 3, 1, 3, 3, 4, 517, 1, 1, 1, 2, 22, 3, 3, 1, 2, 1, 3, 1, 4, 4]\\
\frac{k}{d} &= [0, 6, 1, 8, 3, 1, 2, 1, 20, 1, 26, 1, 2, 1, 37, 1, 1, 4, 2, 1, 4, 1, 1, 1, 3, 5, 1, 1, 134, 1, 3, 4, 6, 1, 26]
\end{aligned}
$$

Der Algorithmus hat Erfolg.

Eine Verallgemeinerung des obigen Satzes gibt folgender Satz:

SATZ. *Die Bezeichnungen seien wie im letzten Satz. Sei* $\delta = |p-q| = N^\beta$, $d = N^\delta$. *Wir setzen voraus* $p, q \geq 23$ *und*

$$\beta + \delta \leq \frac{3}{4}.$$

Dann gilt:

$$\left|\frac{e}{N+1-2\sqrt{N}} - \frac{k}{d}\right| < \frac{1}{2d^2}.$$

Also kommt $\frac{k}{d}$ *als Näherungsbruch in der Kettenbruchentwicklung von* $\frac{e}{N+1-2\sqrt{N}}$ *vor.*

Betrachtet man aber Beispiele mit $q \gtrapprox 2p$, so ist $\beta \gtrapprox 0.5$, damit die Voraussetzung des letzten Satzes erfüllt ist, muß also $\delta \lessapprox 0.25$ gelten, der letzte Satz liefert dann in diesem Fall nichts Neues.

8. Übungen

Aufgabe 3.1: Charakterisiere die reellen Zahlen, deren Kettenbruchentwicklung die Gestalt $[a_0, a_1]$ hat.

Aufgabe 3.2: Bestimme den Anfang der Kettenbruchentwicklung $[a_0, a_1, a_2, \ldots, a_n, \ldots]$ von $\pi = 3.1415926535\ldots$ mindestens bis a_{100}.

Aufgabe 3.3:

(1) Bestimme die Kettenbruchentwicklungen von

$$\frac{1807}{350}, \quad -\frac{1807}{350}, \quad \frac{350}{1807}, \quad -\frac{350}{1807}.$$

(2) Vergleiche für ein $\alpha \in \mathbf{R}^*$ die Kettenbruchentwicklungen von

$$\alpha, \quad -\alpha, \quad \frac{1}{\alpha}, \quad -\frac{1}{\alpha}.$$

Aufgabe 3.4: Seien $\frac{p_n}{q_n}$ die Näherungsbrüche des Kettenbruchs $[a_0, a_1, a_2, \ldots]$. Dann gilt

$$p_n = \begin{vmatrix} a_0 & -1 & & & & \\ 1 & a_1 & -1 & & & \\ & 1 & a_2 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & a_{n-1} & -1 \\ & & & & 1 & a_n \end{vmatrix} \quad \text{und} \quad q_n = \begin{vmatrix} a_1 & -1 & & & \\ 1 & a_2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & a_{n-1} & -1 \\ & & & 1 & a_n \end{vmatrix}.$$

Aufgabe 3.5: Sei $\alpha = 1 + \sqrt{2} = 2.41421356\ldots$.

(1) Die Kettenbruchentwicklung von α ist $[2, 2, 2, 2, 2, \ldots]$.

(2) Für den Nenner q_n der Näherungsbrüche $\frac{p_n}{q_n}$ von α gilt für alle $n \geq 0$

$$q_n = \frac{2+\sqrt{2}}{4}\left(1+\sqrt{2}\right)^n + \frac{2-\sqrt{2}}{4}\left(1-\sqrt{2}\right)^n.$$

(3) Für alle $n \geq 0$ gilt $p_n = q_{n+1}$.

(4) Für $n \equiv 0 \bmod 2$ gilt $q_n > \alpha q_{n-1}$, für $n \equiv 1 \bmod 2$ gilt $q_n < \alpha q_{n-1}$.

(5) Für alle n gilt

$$|\alpha - \frac{p_n}{q_n}| = \frac{1}{q_n(\alpha q_n + q_{n-1})}.$$

(6) Genau dann gilt

$$|\alpha - \frac{p_n}{q_n}| < \frac{1}{2\sqrt{2}q_n^2},$$

wenn n ungerade ist. (Insbesondere hat die Ungleichung $|\alpha - \frac{p}{q}| < \frac{1}{2\sqrt{2}q^2}$ unendlich viele Lösungen in rationalen Zahlen $\frac{p}{q}$.)

(7) Zeige, daß für reelles $A > 2\sqrt{2}$ die Ungleichung

$$|\alpha - \frac{p}{q}| < \frac{1}{Aq^2}$$

nur endlich viele Lösungen in rationalen Zahlen $\frac{p}{q}$ hat.

Aufgabe 3.6: Bestimme alle rationalen Lösungen $\frac{p}{q}$ der Ungleichung

$$|\frac{12345}{23456} - \frac{p}{q}| < \frac{1}{q^2}.$$

Welche der Lösungen sind Näherungsbrüche der zugehörigen Kettenbruchentwicklung?

Aufgabe 3.7: Seien $\frac{p_n}{q_n}$ die Näherungsbrüche der Kettenbruchentwicklung einer reellen Zahl α und

$$f(n) = \lfloor \sqrt{5} q_n^2 |\alpha - \frac{p_n}{q_n}| \rfloor.$$

Zeige:

(1) $f(n) \in \{0, 1, 2\}$ für alle n.
(2) Es gibt kein n mit $f(n) = f(n+1) = f(n+2) = 1$.
(3) Es gibt kein n mit $f(n) = f(n+1) = 2$.

Aufgabe 3.8: Für die Näherungsbrüche des Kettenbruchs $[a_0, a_1, a_2, \dots]$ gilt

$$\frac{q_n}{q_{n-1}} = [a_n, a_{n-1}, \dots, a_1].$$

Aufgabe 3.9: Hat $\alpha \in \mathbf{R} \setminus \mathbf{Q}$ die Kettenbruchentwicklung $[a_0, a_1, a_2, \dots]$ mit den Näherungsbrüchen $\frac{p_n}{q_n}$, definiert man λ_n durch

$$|\alpha - \frac{p_n}{q_n}| = \frac{1}{\lambda_n q_n^2},$$

so gilt

$$\lambda_n = [a_{n+1}, a_{n+2}, a_{n+3}, \dots] + [0, a_n, a_{n-1}, a_{n-2}, \dots, a_1].$$

Aufgabe 3.10: Die reellen Zahlen

$$\sqrt{2}, \quad 2\sqrt{2}, \quad 3\sqrt{2}, \quad 4\sqrt{2}, \quad 5\sqrt{2}, \quad 6\sqrt{2}, \quad \dots$$

sind paarweise nicht äquivalent.

Aufgabe 3.11:

(1) Das Fermatsche Faktorisierungsverfahren bestimmt eine nichttriviale Faktorzerlegung einer ungeraden zusammengesetzten natürlichen Zahl nach folgendem Schema:
   (a) Berechne $x = \lfloor \sqrt{N} \rfloor$.
   (b) Berechne $y_2 = x^2 - N$.
   (c) Teste, ob y_2 ein Quadrat ist. Ist y_2 keine Quadratzahl, setze $x := x + 1$ und gehe zurück zu (b). Ist y_2 eine Quadratzahl, berechne $y \in \mathbf{N}$ mit $y_2 = y^2$. Dann hat man die Faktorisierung $N = (x + y)(x - y)$ und man beendet das Verfahren.
(2) Weshalb funktioniert das Verfahren?
(3) Wieviele Schritte/Durchläufe braucht man im schlimmsten Fall?
(4) Für welche Zahlen N ist das Verfahren gut geeignet?

Aufgabe 3.12: Seien p und q verschiedene ungerade Primzahlen und $N = pq$. Gilt

$$p < q < p + 2\sqrt{2}\sqrt{p} + 2,$$

so faktorisiert das Fermatsche Faktorisierungsverfahren die Zahl N bereits mit $x = \lfloor\sqrt{N}\rfloor + 1$.

Aufgabe 3.13: Seien p und q verschiedene ungerade Primzahlen und $N = pq$, wobei sich p und q höchstens in der letzten Hälfte der Dezimaldarstellung unterscheiden. Wie schnell findet das Fermatsche Faktorisierungsverfahren eine nichttriviale Zerlegung von N? Man gebe Beispiele für alle Möglichkeiten an.

Aufgabe 3.14: Drei öffentliche RSA-Schlüssel (N_i, e_i) sind wie folgt gegeben:

$$\begin{aligned}
N_1 &= 24408639508385557909682562153344178945282138354417396920489587412039227505018323\\
e_1 &= 11588592803013808684418635963716197903847813042918525522581062904580907432549823\\
N_2 &= 35087334881269946985209116612253251307740195653151750296805039601705762407891727\\
e_2 &= 5\\
N_3 &= 18723651913532093571887595276360418040702886847229097549396021601250552637843877\\
e_3 &= 3
\end{aligned}$$

Man faktorisiere mindestens zwei der N_i's.

KAPITEL 4

Periodische Kettenbrüche und reellquadratische Zahlen

1. Quadratische Zahlkörper

Sei $d \in \mathbf{Z}$ keine Quadratzahl, also

$$d \in \{-1, \pm 2, \pm 3, -4, \pm 5, \pm 6, \pm 7, \pm 8, -9, \pm 10, \dots\}.$$

Dann ist

$$K = \{x + y\sqrt{d} : x, y \in \mathbf{Q}\}$$

mit den Rechenregeln

$$\begin{aligned}
(x_1 + y_1\sqrt{d}) + (x_2 + y_2\sqrt{d}) &= (x_1 + x_2) + (y_1 + y_2)\sqrt{d}, \\
(x_1 + y_1\sqrt{d}) \cdot (x_2 + y_2\sqrt{d}) &= (x_1x_2 + dy_1y_2) + (x_1y_2 + y_1x_2)\sqrt{d}, \\
\frac{1}{x + y\sqrt{d}} &= \frac{x}{x^2 - dy^2} - \frac{y}{x^2 - dy^2}\sqrt{d}
\end{aligned}$$

ein Körper, der sogenannte quadratische Zahlkörper $\mathbf{Q}(\sqrt{d})$. Im Fall $d > 0$ nennt man $\mathbf{Q}(\sqrt{d})$ reellquadratisch, im Fall $d < 0$ imaginärquadratisch.

Ist $d' = m^2 d$, so liefert

$$\mathbf{Q}(\sqrt{d'}) \to \mathbf{Q}(\sqrt{d}), \quad x + y\sqrt{d'} \mapsto x + ym\sqrt{d}$$

einen Körperisomorphismus, d.h. man kann $\mathbf{Q}(\sqrt{d'})$ mit $\mathbf{Q}(\sqrt{d})$ identifizieren und sich daher auf quadratfreie $d \in \mathbf{Z}$ beschränken, also

$$d \in \{-1, \pm 2, \pm 3, \pm 5, \pm 6, \pm 7, \pm 10, \dots\}.$$

Im reellquadratischen Fall kann man $\mathbf{Q}(\sqrt{d})$ als Teilkörper von $\mathbf{R}$ betrachten. In diesem Fall wählen wir immer die positive Quadratwurzel. Entsprechend werden wir $\sqrt{m^2 d}$ immer mit $|m|\sqrt{d}$ identifizieren.

Ein nichttrivialer Körperautomorphismus von $\mathbf{Q}(\sqrt{d})$ wird durch

$$(x + y\sqrt{d})' = x - y\sqrt{d}$$

gegeben und als Konjugation bezeichnet. Weiter werden Spur und Norm durch

$$\mathrm{Sp}(\alpha) = \alpha + \alpha' \quad \text{und} \quad \mathrm{N}(\alpha) = \alpha\alpha'$$

definiert. Daher ist die Spur additiv, die Norm multiplikativ. Aus

$$\mathrm{Sp}(x + y\sqrt{d}) = 2x \quad \text{und} \quad \mathrm{N}(x + y\sqrt{d}) = x^2 - dy^2$$

sieht man, daß Spur und Norm Werte in $\mathbf{Q}$ haben.

Version vom 9.2.2002

2. Periodische Kettenbrüche

Wir nennen die Kettenbruchentwicklung einer reellen Zahl α rein periodisch, wenn für eine natürliche Zahl k gilt

$$\alpha = [u_0, u_1, \dots, u_{k-2}, u_{k-1}, u_0, u_1, \dots, u_{k-2}, u_{k-1}, u_0, u_1, \dots, u_{k-2}, u_{k-1}, \dots]$$

und schreiben dafür auch

$$\alpha = [\overline{u_0, u_1, \dots, u_{k-2}, u_{k-1}}].$$

Die Kettenbruchentwicklung wird periodisch genannt, wenn mit den Bezeichnungen von eben gilt

$$\alpha = [u_0, u_1, \dots, u_{r-1}, \overline{u_r, \dots, u_{r+k-1}}].$$

$u_0, u_1, \dots, u_{r-1}$ ist dann die Vorperiode der Länge r, $\overline{u_r, \dots, u_{r+k-1}}$ ist die Periode der Länge k.

SATZ. *Hat die reelle Zahl α die periodische Kettenbruchentwicklung*

$$\alpha = [u_0, u_1, \dots, u_{r-1}, \overline{u_r, \dots, u_{r+k-1}}],$$

so ist α reellquadratisch.

Beweis: Wir betrachten zunächst den Fall, daß die Kettenbruchentwicklung rein periodisch ist:

$$\alpha = [\overline{u_0, \dots, u_{k-1}}].$$

Sind $\frac{p_n}{q_n}$ wie üblich die zugehörigen Näherungsbrüche, so gilt

$$\alpha = [u_0, \dots, u_{k-1}, \alpha] = \frac{\alpha p_{k-1} + p_{k-2}}{\alpha q_{k-1} + q_{k-2}},$$

also

$$q_{k-1}\alpha^2 + (q_{k-2} - p_{k-1})\alpha - p_{k-2} = 0.$$

α kann nicht rational sein, da die Kettenbruchentwicklung von α unendlich ist, also ist α reellquadratisch, was die Behauptung in diesem Fall beweist.
Im allgemeinen ist

$$\alpha = [u_0, u_1, \dots, u_{r-1}, \overline{u_r, \dots, u_{r+k-1}}].$$

Dann ist nach der eben gezeigten Behauptung

$$\beta = [\overline{u_r, \dots, u_{r+k-1}}]$$

reellquadratisch und wegen

$$\alpha = [u_0, u_1, \dots, u_{r-1}, \beta] = \frac{\beta p_{r-1} + p_{r-2}}{\beta q_{r-1} + q_{r-2}}$$

ist $\alpha \in \mathbf{Q}(\beta)$, also auch reellquadratisch, da es nicht rational sein kann. ■

Beispiel: Der einfachste Fall einer periodischen Kettenbruchentwicklung ist der Fall

$$\alpha = [\overline{m}] = [m, m, m, \dots]$$

mit einer natürlichen Zahl m. Dann ist also $\alpha = m + \frac{1}{\alpha}$, was wegen $\alpha > 0$

$$\alpha = \frac{m + \sqrt{m^2 + 4}}{2}$$

ergibt. Für $m = 1, 2, 3, 4, 5, 6$ erhält man explizit

$$\frac{1+\sqrt{5}}{2}, \quad 1+\sqrt{2}, \quad \frac{3+\sqrt{13}}{2}, \quad 2+\sqrt{5}, \quad \frac{5+\sqrt{29}}{2}, \quad 3+\sqrt{10}.$$

Beispiel: Wir betrachten den Fall

$$\alpha = [\overline{a, b}] = a + \cfrac{1}{b + \cfrac{1}{\alpha}}$$

mit natürlichen Zahlen $a \neq b$. Dann ist $a + ab\alpha - b\alpha^2 = 0$ und damit (wegen $\alpha > 1$)

$$\alpha = \frac{ab + \sqrt{a^2b^2 + 4ab}}{2b}.$$

Ein paar explizite Beispiele:

a	1	2	1	3	1	2	3	4
b	2	1	3	1	4	3	2	1
$[\overline{a,b}]$	$\frac{1+\sqrt{3}}{2}$	$1+\sqrt{3}$	$\frac{3+\sqrt{21}}{6}$	$\frac{3+\sqrt{21}}{2}$	$\frac{1+\sqrt{2}}{2}$	$\frac{3+\sqrt{15}}{3}$	$\frac{3+\sqrt{15}}{2}$	$2+2\sqrt{2}$

Im folgenden wird (nach einigen Vorbereitungen) umgekehrt gezeigt werden, daß jede reellquadratische Zahl eine periodische Kettenbruchentwicklung besitzt.

3. Diskriminanten quadratischer Zahlen

Sei $d \in \mathbf{Z}$ kein Quadrat und quadratfrei, $K = \mathbf{Q}(\sqrt{d})$ ein quadratischer Zahlkörper.

Für $\alpha = x + y\sqrt{d} \in \mathbf{Q}(\sqrt{d}) \setminus \mathbf{Q}$ definiert man ein Polynom $f(t)$ durch

$$f(t) = (t-\alpha)(t-\alpha') \in \mathbf{Q}(\sqrt{d})[t].$$

Zunächst ist $f(\alpha) = 0$. Aus

$$f(t) = t^2 - (\alpha+\alpha')t + \alpha\alpha' = t^2 - \mathrm{Sp}(\alpha)t + \mathrm{N}(\alpha) \in \mathbf{Q}[t]$$

folgt, daß $f(t)$ rationale Koeffizienten hat. $f(t)$ ist das normierte Minimalpolynom von α.

Multipliziert man $f(t)$ mit dem gemeinsamen Nenner der Koeffizienten, so erhält man ein Polynom

$$g(t) = at^2 + bt + c \quad \text{mit} \quad a,b,c \in \mathbf{Z}, a > 0, \mathrm{ggT}(a,b,c) = 1 \quad \text{und} \quad g(\alpha) = 0.$$

Das Polynom $g(t)$ ist dadurch eindeutig bestimmt. Die Größe

$$D(\alpha) = b^2 - 4ac$$

heißt die Diskriminante von α.

Beispiele:

(1) Das Minimalpolynom von $\sqrt{d}$ ist $t^2 - d$, also $D(\sqrt{d}) = 4d$.

(2) Ist $d \equiv 1 \bmod 4$, so ist das Minimalpolynom von $\frac{1+\sqrt{d}}{2}$

$$f(t) = (t - \frac{1+\sqrt{d}}{2})(t - \sqrt{1-\sqrt{d}2}) = t^2 - t + \frac{1-d}{4}$$

ganzzahlig, also folgt $D(\frac{1+\sqrt{d}}{2}) = 1 - 4 \cdot \frac{1-d}{4} = d$, also

$$D(\frac{1+\sqrt{d}}{2}) = d.$$

(3) Die kleinsten reellquadratischen Diskriminanten sind also 5, 8, 12, 13, ..., zugehörige Elemente beispielsweise

$$D(\frac{1+\sqrt{5}}{2}) = 5, \quad D(\sqrt{2}) = 8, \quad D(\sqrt{3}) = 12, \quad D(\frac{1+\sqrt{13}}{2}) = 13.$$

LEMMA. *Jede quadratische Zahl α mit Diskriminante D läßt sich in der Form*

$$\alpha = \frac{b+\sqrt{D}}{2a} \quad \textit{mit} \quad 4a | D - b^2 \textit{ (oder } D = b^2 - 4ac)$$

schreiben. (Dabei ist $\mathrm{ggT}(a,b,c) = 1$*.) Die Darstellung ist eindeutig. Ist umgekehrt*

$$\alpha = \frac{b+\sqrt{D}}{2a} \quad \textit{mit} \quad D = b^2 - 4ac \textit{ und } \mathrm{ggT}(a,b,c) = 1,$$

so hat α Diskriminante D.

Beweis: Sei $A\alpha^2+B\alpha+C=0$ mit $A,B,C\in\mathbf{Z}$ und $\mathrm{ggT}(A,B,C)=1$. Dann ist $D=B^2-4AC$. Es folgt

$$\alpha=\frac{(-B)+\sqrt{D}}{2A}\quad\text{oder}\quad\alpha=\frac{-B-\sqrt{D}}{2A}=\frac{B+\sqrt{D}}{2(-A)},$$

die Wahl von $(a,b)=(A,-B)$ bzw. $(a,b)=(B,-A)$ liefert dann die Existenz einer gewünschten Darstellung. Zur Eindeutigkeit: Aus $\frac{b+\sqrt{D}}{2a}=\frac{\widetilde{b}+\sqrt{D}}{2\widetilde{a}}$ folgt $\frac{b}{2a}=\frac{\widetilde{b}}{2\widetilde{a}}$ und $\frac{1}{2a}=\frac{1}{2\widetilde{a}}$, was natürlich sofort $a=\widetilde{a}$ und $b=\widetilde{b}$ liefert.
Sei jetzt umgekehrt

$$\alpha=\frac{b+\sqrt{D}}{2a}\text{ mit }D=b^2-4ac\text{ und }\mathrm{ggT}(a,b,c)=1.$$

Dann ist

$$\alpha+\alpha'=\frac{b}{a},\quad\alpha\alpha'=\frac{b^2-D}{4a^2}=\frac{4ac}{4a^2}=\frac{c}{a},$$

also genügt α der Gleichung $at^2-bt+c=0$. Wegen $\mathrm{ggT}(a,b,c)=1$ ist die Diskriminante $D=b^2-4ac$, wie behauptet. ■

Beispiele: α habe Diskriminante D, d.h. $\alpha=\frac{b+\sqrt{D}}{2a}$ mit $D=b^2-4ac$ und $\mathrm{ggT}(a,b,c)=1$. Dann ist

$$\begin{aligned}
-\alpha &= \frac{b+\sqrt{D}}{2(-a)},\\
\frac{1}{\alpha} &= \frac{2a}{b+\sqrt{D}}\cdot\frac{b-\sqrt{D}}{b-\sqrt{D}}=\frac{2a(b-\sqrt{D}}{b^2-D}=\frac{2a(b-\sqrt{D}}{4ac}=\frac{b-\sqrt{D}}{2c}=\frac{-b+\sqrt{D}}{2(-c)},\\
\alpha' &= \frac{b-\sqrt{D}}{2a}=\frac{-b+\sqrt{D}}{2(-a)},
\end{aligned}$$

also haben auch die angegebenen Zahlen Diskriminante D.

Das folgende Lemma zeigt, wie man obige Normalform algorithmisch erreichen kann (Maple-Funktion qz_abD):

LEMMA. *Sei $\alpha=q_1+q_2\sqrt{d}$ gegeben mit $q_1,q_2\in\mathbf{Q}$, $q_2\neq0$. Schreibt man*

$$2q_1=\frac{b}{a},\quad q_1^2-dq_2^2=\frac{c}{a}\quad\textit{mit}\quad\mathrm{ggT}(a,b,c)=1\quad\textit{und}\quad aq_2>0,$$

(d.h. a ist bis auf das Vorzeichen der gemeinsame Nenner von $2q_1$ und $q_1^2-dq_2^2$), so gilt

$$D=D(\alpha)=b^2-4ac,\quad\alpha=\frac{b+\sqrt{D}}{2a},\quad\mathrm{ggT}(a,b,c)=1.$$

Beweis: α genügt der Gleichung $\alpha^2-2q_1\alpha+(q_1^2-dq_2^2)=0$, nach Konstruktion also auch der Gleichung $a\alpha^2-b\alpha+c=0$. Wegen $\mathrm{ggT}(a,b,c)=1$ folgt für die Diskriminante $D(\alpha)=b^2-4ac$. Mit der üblichen Formel erhält man

$$\alpha=\frac{b\pm\sqrt{D}}{2a},$$

nach Wahl von $aq_2>0$ haben $\frac{\sqrt{D}}{2a}$ und $q_2\sqrt{d}$ das gleiche Vorzeichen, also folgt $\alpha=\frac{b+\sqrt{D}}{2a}$, was gezeigt werden sollte. ■

LEMMA. *$K=\mathbf{Q}(\sqrt{d})$ ein quadratischer Zahlkörper und d quadratfrei. Für die Diskriminante $D(\alpha)$ eines Elements $\alpha\in\mathbf{Q}(\sqrt{d})\setminus\mathbf{Q}$ gilt dann*

$$D(\alpha)=m^2d\textit{ mit }m\in\mathbf{N}\quad\textit{und}\quad D(\alpha)\equiv0\textit{ oder }1\bmod4.$$

Beweis: Mit den obigen Bezeichnungen gilt $\frac{\sqrt{D(\alpha)}}{2a}=q_2\sqrt{d}$, also $D(\alpha)=(2aq_2)^2d$. Wir setzen $m=2aq_2$. Da d quadratfrei vorausgesetzt war, folgt $m\in\mathbf{Z}$, nach Wahl von a also $m\in\mathbf{N}$ und damit $D(\alpha)=m^2d$. Sei nun $\alpha=\frac{b+\sqrt{D}}{2a}$ mit $D=b^2-4ac$ und $\mathrm{ggT}(a,b,c)=1$. Gilt $b\equiv0\bmod2$, so folgt $D(\alpha)\equiv0\bmod4$, gilt $b\equiv1\bmod2$, so haben wir $D(\alpha)\equiv1\bmod4$. ■

4. Die Kettenbruchentwicklung reellquadratischer Zahlen

Für eine reellquadratische Zahl α definieren wir

$$\rho(\alpha) = \frac{1}{\alpha - \lfloor\alpha\rfloor},$$

also ist

$$\alpha = \lfloor\alpha\rfloor + \frac{1}{\rho(\alpha)},$$

d.h. $\rho(\alpha)$ ist der Nachfolger in der Kettenbruchentwicklung von α.

LEMMA. *α sei reellquadratisch mit Diskriminante D, also*

$$\alpha = \frac{b+\sqrt{D}}{2a} \quad \textit{mit} \quad D = b^2 - 4ac.$$

Definiert man (nacheinander)

$$\begin{aligned} u &= \lfloor\alpha\rfloor, \\ \widetilde{a} &= -au^2 + bu - c, \\ \widetilde{b} &= 2au - b, \\ \widetilde{c} &= -a, \end{aligned}$$

dann gilt $D = \widetilde{b}^2 - 4\widetilde{a}\widetilde{c}$,

$$\rho(\alpha) = \frac{1}{\alpha - \lfloor\alpha\rfloor} = \frac{\widetilde{b}+\sqrt{D}}{2\widetilde{a}}$$

und auch $\rho(\alpha)$ hat Diskriminante D.

Beweis: Die Beziehung $D = \widetilde{b}^2 - 4\widetilde{a}\widetilde{c}$ rechnet man sofort nach. Damit gilt dann

$$1 = \frac{\widetilde{b}^2 - D}{4\widetilde{a}\widetilde{c}} = \frac{(\widetilde{b}+\sqrt{D})(\widetilde{b}-\sqrt{D})}{4\widetilde{a}(-a)} = \frac{\widetilde{b}+\sqrt{D}}{2\widetilde{a}} \cdot \frac{b - 2au + \sqrt{D}}{2a} = \frac{\widetilde{b}+\sqrt{D}}{2\widetilde{a}} \cdot (\alpha - u),$$

woraus

$$\rho(\alpha) = \frac{1}{\alpha - u} = \frac{\widetilde{b}+\sqrt{D}}{2\widetilde{a}}$$

folgt. Gilt $\mathrm{ggT}(a,b,c) = 1$, so offensichtlich auch $\mathrm{ggT}(\widetilde{a},\widetilde{b},\widetilde{c}) = 1$, d.h. auch $\rho(\alpha)$ hat Diskriminante D. ■

Bemerkung: Man kann im Prinzip auf die Größen c und $\widetilde{c}$ verzichten, denn es gilt

$$\begin{aligned} \widetilde{b} &= 2au - b, \\ \widetilde{a} &= \frac{D - \widetilde{b}^2}{-4\widetilde{c}} = \frac{D - \widetilde{b}^2}{4a}. \end{aligned}$$

Das Lemma liefert sofort ein Verfahren um die Kettenbruchentwicklung einer reellquadratischen Zahl zu bestimmen:

SATZ. *Sei α reellquadratisch mit Diskriminante D, $\alpha = \frac{b+\sqrt{D}}{2a}$, $D = b^2 - 4ac$. Definiert man $\alpha_0 = \alpha$, $a_0 = a$, $b_0 = b$, $c_0 = c$ und dann rekursiv*

$$\begin{aligned} u_i &= \lfloor\alpha_i\rfloor, \\ a_{i+1} &= -a_i u_i^2 + b_i u_i - c_i, \\ b_{i+1} &= 2a_i u_i - b_i, \\ c_{i+1} &= -a_i, \\ \alpha_{i+1} &= \frac{b_{i+1}+\sqrt{D}}{2a_{i+1}}, \end{aligned}$$

so erhält man die Kettenbruchentwicklung von α:

$$\alpha = [u_0, u_1, u_2, \ldots, u_n, \alpha_{n+1}] = [u_0, u_1, u_2, \ldots].$$

Dabei kann man folgende Beziehung benutzen:

$$\lfloor \frac{b+\sqrt{D}}{2a} \rfloor = \begin{cases} \lfloor \frac{b+\lfloor \sqrt{D} \rfloor}{2a} \rfloor & \text{für } a > 0, \\ \lfloor \frac{b+\lceil \sqrt{D} \rceil}{2a} \rfloor & \text{für } a < 0. \end{cases}$$

Beweis: Zusammen mit obigem Lemma sieht man, daß die Größen so definiert sind, daß gilt

$$\alpha_i = \frac{b_i + \sqrt{D}}{2a_i}, \quad u_i = \lfloor \alpha_i \rfloor, \quad \alpha_i = u_i + \frac{1}{\alpha_{i+1}},$$

was die behauptete Kettenbruchentwicklung liefert.
Sei $u \in \mathbf{Z}$. Für $a > 0$ haben wir die Äquivalenzen

$$u \leq \frac{b+\sqrt{D}}{2a} \iff 2au - b \leq \sqrt{D} \iff 2au - b \leq \lfloor \sqrt{D} \rfloor \iff u \leq \frac{b + \lfloor \sqrt{D} \rfloor}{2a},$$

woraus die Behauptung folgt. Für $a < 0$ ergibt

$$u \leq \frac{b+\sqrt{D}}{2a} \iff 2au - b \geq \sqrt{D} \iff 2au - b \geq \lceil \sqrt{D} \rceil \iff u \leq \frac{b + \lceil \sqrt{D} \rceil}{2a}$$

die Behauptung. ■

Bemerkung: Obige Vorgehensweise haben wir in der Maple-Funktion kbe_uvwd_tex benutzt, die die Kettenbruchentwicklung von $\frac{u+v\sqrt{d}}{w}$ bestimmt. Mit kbe_uvwd_tex wurden die folgenden Beispiele erstellt.

Beispiel: $\sqrt{19}$ hat Diskriminante 76.

i	a_i	b_i	c_i	α_i	u_i	α_i	α_i'
0	1	0	-19	$\sqrt{19}$	4	4.359	-4.359
1	3	8	-1	$\frac{4+\sqrt{19}}{3}$	2	2.786	-.120
2	5	4	-3	$\frac{2+\sqrt{19}}{5}$	1	1.272	-.472
3	2	6	-5	$\frac{3+\sqrt{19}}{2}$	3	3.679	-.679
4	5	6	-2	$\frac{3+\sqrt{19}}{5}$	1	1.472	-.272
5	3	4	-5	$\frac{2+\sqrt{19}}{3}$	2	2.120	-.786
6	1	8	-3	$4+\sqrt{19}$	8	8.359	-.359
7	3	8	-1	$\frac{4+\sqrt{19}}{3}$	2	2.786	-.120

Es ist $\alpha_7 = \alpha_1$, also folgt $\alpha_8 = \alpha_2$ etc. Die Kettenbruchentwicklung wird also

$$\sqrt{19} = [4, \overline{2, 1, 3, 1, 2, 8}].$$

Beispiel: $4-3\sqrt{163}$ hat Diskriminante 5868. Wir bilden die Kettenbruchentwicklung:

i	a_i	b_i	c_i	α_i	u_i	α_i	α_i'
0	-1	-8	1451	$4-3\sqrt{163}$	-35	-34.301	42.301
1	54	78	1	$\frac{13+1\sqrt{163}}{18}$	1	1.432	.013
2	23	30	-54	$\frac{15+3\sqrt{163}}{23}$	2	2.317	-1.013
3	22	62	-23	$\frac{31+3\sqrt{163}}{22}$	3	3.150	-.332
4	11	70	-22	$\frac{35+3\sqrt{163}}{11}$	6	6.664	-.300
5	46	62	-11	$\frac{31+3\sqrt{163}}{46}$	1	1.507	-.159
6	27	30	-46	$\frac{5+1\sqrt{163}}{9}$	1	1.974	-.863
7	49	24	-27	$\frac{12+3\sqrt{163}}{49}$	1	1.027	-.537
8	2	74	-49	$\frac{37+3\sqrt{163}}{2}$	37	37.651	-.651
9	49	74	-2	$\frac{37+3\sqrt{163}}{49}$	1	1.537	-.027
10	27	24	-49	$\frac{4+1\sqrt{163}}{9}$	1	1.863	-.974
11	46	30	-27	$\frac{15+3\sqrt{163}}{46}$	1	1.159	-.507
12	11	62	-46	$\frac{31+3\sqrt{163}}{11}$	6	6.300	-.664
13	22	70	-11	$\frac{35+3\sqrt{163}}{22}$	3	3.332	-.150
14	23	62	-22	$\frac{31+3\sqrt{163}}{23}$	3	3.013	-.317
15	1	76	-23	$38+3\sqrt{163}$	76	76.301	-.301
16	23	76	-1	$\frac{38+3\sqrt{163}}{23}$	3	3.317	-.013
17	22	62	-23	$\frac{31+3\sqrt{163}}{22}$	3	3.150	-.332

Nun ist $\alpha_{17}=\alpha_3$, also $\alpha_{17+j}=\alpha_{3+j}$ für alle $j\geq 0$, die Kettenbruchentwicklung wird also periodisch:

$$4-3\sqrt{163}=[-35,1,2,\overline{3,6,1,1,1,37,1,1,1,6,3,3,76,3}].$$

Schaut man sich das numerische Verhalten von (α_i,α_i') in den vorangegangenen Beispielen an, kommt man auf folgende Feststellung:

LEMMA. *Sei α reellquadratisch und $\alpha=[u_0,u_1,\ldots,u_{n-1},\alpha_n]$ die Kettenbruchentwicklung. Dann gilt für alle hinreichend großen n:*

$$-1<\alpha_n'<0<1<\alpha_n.$$

Beweis: Nach Konstruktion der Kettenbruchentwicklung von α ist $\alpha_n>1$ für alle $n\geq 1$. Sind $\frac{p_i}{q_i}$ wie üblich die Näherungsbrüche, so gilt:

$$\alpha=[u_0,u_1,\ldots,u_{n-1},\alpha_n]=\frac{\alpha_n p_{n-1}+p_{n-2}}{\alpha_n q_{n-1}+q_{n-2}}.$$

Lösen wir dies nach α_n auf, so erhalten wir

$$\alpha_n=-\frac{q_{n-2}}{q_{n-1}}\frac{\alpha-\frac{p_{n-2}}{q_{n-2}}}{\alpha-\frac{p_{n-1}}{q_{n-1}}}.$$

Übergang zu den Konjugierten ergibt

$$\alpha_n'=-\frac{q_{n-2}}{q_{n-1}}\frac{\alpha'-\frac{p_{n-2}}{q_{n-2}}}{\alpha'-\frac{p_{n-1}}{q_{n-1}}}.$$

Wegen $\lim_{k\to\infty}\frac{p_k}{q_k}=\alpha$ ist

$$\lim_{n\to\infty}\frac{\alpha'-\frac{p_{n-2}}{q_{n-2}}}{\alpha'-\frac{p_{n-1}}{q_{n-1}}}=1.$$

Für hinreichend großes n ist also

$$\alpha_n'<0.$$

Weiter rechnet man nach

$$q_{n-1}(\alpha'_n + 1) - (q_{n-1} - q_{n-2}) = \frac{(-1)^n}{\alpha' q_{n-1} - p_{n-1}},$$

also

$$\alpha'_n + 1 = \frac{1}{q_n - 1}[(q_{n-1} - q_{n-2}) + \frac{(-1)^n}{q_{n-1}(\alpha' - \frac{p_{n-1}}{q_{n-1}})}].$$

Aus $\lim_{n\to\infty}(\alpha' - \frac{p_{n-1}}{q_{n-1}}) = \alpha' - \alpha$ und $q_{n-1} \geq n-1$ und $q_{n-1} > q_{n-2}$ folgt, daß die rechte Seite der Gleichung für großes n positiv ist, d.h. $\alpha'_n + 1 > 0$ und damit $\alpha'_n > -1$, wie behauptet. ■

DEFINITION. Ein reellquadratisches α heißt reduziert, falls gilt

$$-1 < \alpha' < 0 \text{ und } \alpha > 1.$$

Bemerkung: Das vorangegangene Lemma besagt also,daß für alle hinreichend großen n die reellquadratische Zahl $\alpha_n = \rho^n(\alpha)$ reduziert ist.

LEMMA. *Sei $\alpha = \frac{b+\sqrt{D}}{2a}$ mit $D = b^2 - 4ac$. Dann ist α genau dann reduziert, wenn gilt:*

$$1 \leq b < \sqrt{D} \quad \text{und} \quad \sqrt{D} - b < 2a < \sqrt{D} + b.$$

Insbesondere muß dann gelten $1 \leq a < \sqrt{D}$.

Beweis: Wir nehmen an, daß α reduziert ist, d.h. es gilt

$$-1 < \frac{b - \sqrt{D}}{2a} < 0 < 1 < \frac{b + \sqrt{D}}{2a}.$$

Aus $\frac{b-\sqrt{D}}{2a} < \frac{b+\sqrt{D}}{2a}$ folgt $a > 0$, da wir $\sqrt{D} > 0$ voraussetzen. Multiplikation der Ungleichungskette mit $2a$ ergibt

$$-2a < b - \sqrt{D} < 0 < 2a < b + \sqrt{D}.$$

Addition der Ungleichungen $-2a < b - \sqrt{D}$ und $2a < b + \sqrt{D}$ liefert $b > 0$ und damit $b \geq 1$. Die restlichen behaupteten Ungleichungen kann man nun sofort sehen.
Es gelte nun umgekehrt $1 \leq b < \sqrt{D}$ und $\sqrt{D} - b < 2a < \sqrt{D} + b$. Wegen $b < \sqrt{D}$ ist $0 < \sqrt{D} - b < 2a$, also $a \geq 1$. Dann folgt aus $2a < \sqrt{D} + b$ sofort $1 < \alpha$. Aus $\sqrt{D} - b < 2a$ folgt $-2a < b - \sqrt{D}$ und damit $-1 < \alpha'$. Wegen $b < \sqrt{D}$ gilt nun auch $\alpha' < 0$. Damit ist α reduziert. ■

FOLGERUNG. *Für $D \in \mathbf{N}$ ist die Menge*

$$\begin{aligned} \mathrm{Red}(D) &= \{\alpha \textit{ reellquadratisch mit Diskriminante } D \textit{ und reduziert}\} = \\ &= \{\alpha = \frac{b+\sqrt{D}}{2a} : 4a | b^2 - D, \mathrm{ggT}(a, b, \frac{D-b^2}{4a}) = 1, 1 \leq b < \sqrt{D}, \sqrt{D} - b < 2a < \sqrt{D} + b\} \end{aligned}$$

der reduzierten reellquadratischen Zahlen mit Diskriminante D endlich.

Bemerkung: Die Ungleichungen des letzten Lemmas kann man auch folgendermaßen ausdrücken:

$$1 \leq b \leq \lfloor\sqrt{D}\rfloor, \quad \lfloor\frac{\lfloor\sqrt{D}\rfloor - b + 2}{2}\rfloor \leq a \leq \lfloor\frac{\lfloor\sqrt{D}\rfloor + b}{2}\rfloor.$$

Wir haben damit eine Maple-Funktion $\mathrm{Red}(D)$ geschrieben, die die reduzierten reellquadratischen Zahlen mit Diskriminante D bestimmt.

Beispiele:

D	$\mathrm{Red}(D)$
5	$\{\frac{1+\sqrt{5}}{2}\}$
8	$\{1+\sqrt{2}\}$
12	$\{1+\sqrt{3}, \frac{1+\sqrt{3}}{2}\}$
13	$\{\frac{3+\sqrt{13}}{2}\}$
17	$\{\frac{1+\sqrt{17}}{4}, \frac{3+\sqrt{17}}{4}, \frac{3+\sqrt{17}}{2}\}$
20	$\{2+\sqrt{5}\}$
21	$\{\frac{3+\sqrt{21}}{6}, \frac{3+\sqrt{21}}{2}\}$

LEMMA. *Für die Nachfolgerabbildung ρ und die Menge der reduzierten reellquadratischen Zahlen mit Diskriminante D gilt:*

(1) $\rho(\mathrm{Red}(D)) \subseteq \mathrm{Red}(D)$.
(2) $\rho|_{\mathrm{Red}(D)} : \mathrm{Red}(D) \to \mathrm{Red}(D)$ *ist bijektiv.*
(3) *Die Umkehrabbildung kann explizit angegeben werden: Für $\alpha \in \mathrm{Red}(D)$ gilt:*

$$(\rho|_{\mathrm{Red}(D)})^{-1}(\alpha) = -1 - \lfloor \frac{1}{\alpha'} \rfloor + \frac{1}{\alpha}.$$

Beweis:

(1) Sei $\alpha \in \mathrm{Red}(D)$. Dann hat auch $\rho(\alpha) = \frac{1}{\alpha - \lfloor \alpha \rfloor}$ Diskriminante D. Außerdem sieht man, daß $\rho(\alpha) > 1$ gilt. Aus

$$\rho(\alpha)' = \frac{1}{\alpha' - \lfloor \alpha \rfloor}$$

und $-1 < \alpha' < 0$, $\lfloor \alpha \rfloor \geq 1$ sieht man, daß auch $-1 < \rho(\alpha)' < 0$ gilt. Also folgt $\rho(\alpha) \in \mathrm{Red}(D)$.

(2) Wir zeigen, daß die angegebene Abbildung injektiv ist. Seien also α und β reduziert mit $\rho(\alpha) = \rho(\beta)$. Dann ist

$$\frac{1}{\alpha - \lfloor \alpha \rfloor} = \rho(\alpha) = \rho(\beta) = \frac{1}{\beta - \lfloor \beta \rfloor},$$

also $\alpha - \beta = \lfloor \alpha \rfloor - \lfloor \beta \rfloor \in \mathbf{Z}$. Es folgt

$$\alpha' - \beta' = \lfloor \alpha \rfloor - \lfloor \beta \rfloor \in \mathbf{Z}.$$

Da α und β reduziert sind, gilt $-1 < \alpha' < 0$ und $0 < -\beta' < 1$, was durch Addition $-1 < \alpha' - \beta' < 1$ ergibt. Wegen $\alpha' - \beta' \in \mathbf{Z}$ folgt $\alpha' = \beta'$ und damit natürlich auch $\alpha = \beta$. Damit ist die angegebene Abbildung injektiv, da $\mathrm{Red}(D)$ endlich ist, auch bijektiv.

(3) Sei α reduziert und γ ein Vorgänger in der Kettenbruchentwicklung, d.h. $\gamma = c + \frac{1}{\alpha}$ und $c = \lfloor \gamma \rfloor$. Dann gilt:

$$\begin{aligned}
\gamma \text{ reduziert} &\iff -1 < \gamma' < 0 \quad \text{und } 1 < \gamma \\
&\iff -1 < c + \frac{1}{\alpha'} < 0 \quad \text{und} \quad c \geq 1 \\
&\iff -1 = c + \lfloor \frac{1}{\alpha'} \rfloor \quad \text{und} \quad c \geq 1 \\
&\iff c = -1 - \lfloor \frac{1}{\alpha'} \rfloor \quad \text{und} \quad c \geq 1 \\
&\iff c = -1 - \lfloor \frac{1}{\alpha'} \rfloor.
\end{aligned}$$

Also ist

$$\gamma = -1 - \lfloor \frac{1}{\alpha'} \rfloor + \frac{1}{\alpha}$$

reduziert und hat α als Nachfolger in der Kettenbruchentwicklung. ■

SATZ. *Die Kettenbruchentwicklung einer reellen Zahl α ist genau dann rein periodisch, wenn α reellquadratisch und reduziert ist.*

Beweis:

(1) Gilt $\alpha = [\overline{u_0, u_1, \ldots, u_{k-1}}]$, so haben wir bereits gesehen, daß α reellquadratisch ist. Aus

$$\alpha = [u_0, u_1, \ldots, u_{k-1}, \alpha] = [u_0, \ldots, u_{k-1}, u_0, \ldots, u_{k-1}, \alpha] = \ldots$$

folgt $\alpha = \alpha_k = \alpha_{2k} = \ldots$. Da für alle hinreichend großen n die Zahl α_n reduziert ist, folgt, daß α selbst reduziert ist.

(2) Sei umgekehrt α reduziert und $\alpha = [u_0, u_1, \ldots, u_{n-1}, \alpha_n]$. Dann ist $\alpha_n = \rho^n(\alpha) \in \mathrm{Red}(D)$ für alle n. Da die Menge $\mathrm{Red}(D)$ endlich ist, muß es Indizes $m \geq 0$ und $l > 0$ geben mit $\rho^m(\alpha) = \rho^{m+l}(\alpha)$, wobei wir l als minimal voraussetzen können. Da ρ auf $\mathrm{Red}(D)$ bijektiv ist, folgt $\alpha = \rho^l(\alpha) = \alpha_l$ und die Behauptung folgt. ■

Ist α reellquadratisch, so haben wir gesehen, daß für große n die Zahl α_n reduziert ist, also hat α_n eine rein periodische Kettenbruchentwicklung. Damit folgt insgesamt:

FOLGERUNG. *Genau die reellquadratischen Zahlen haben eine periodische Kettenbruchentwicklung.*

Beispiel: Für die Diskriminante $D = 101$ gilt

$$\mathrm{Red}(101) = \{\frac{9+\sqrt{101}}{2}, \frac{9+\sqrt{101}}{10}, \frac{1+\sqrt{101}}{10}\}.$$

Die Kettenbruchentwicklungen sind

$$\frac{9+\sqrt{101}}{2} = [\overline{9,1,1}], \quad \frac{9+\sqrt{101}}{10} = [\overline{1,1,9}], \quad \frac{1+\sqrt{101}}{10} = [\overline{1,9,1}].$$

Beispiel: Hier sind die reduzierten Elemente mit Diskriminante $D = 148$ zusammen mit ihrer Kettenbruchentwicklung:

$$\begin{aligned}
6+\sqrt{37} &= [\overline{12}],\\
\frac{4+\sqrt{37}}{7} &= [\overline{1,2,3}],\\
\frac{3+\sqrt{37}}{4} &= [\overline{2,3,1}],\\
\frac{5+\sqrt{37}}{3} &= [\overline{3,1,2}],\\
\frac{3+\sqrt{37}}{7} &= [\overline{1,3,2}],\\
\frac{4+\sqrt{37}}{3} &= [\overline{3,2,1}],\\
\frac{5+\sqrt{37}}{4} &= [\overline{2,1,3}].
\end{aligned}$$

LEMMA. *Ist α reduziert mit Diskriminante D, so auch $-\frac{1}{\alpha'}$. Gilt*

$$\alpha = \frac{b+\sqrt{D}}{2a} \quad \textit{mit} \quad D = b^2 - 4ac,$$

so folgt

$$-\frac{1}{\alpha'} = \frac{b+\sqrt{D}}{2(-c)}.$$

Beweis: Da α reduziert ist, gilt $\alpha > 1$ und $-1 < \alpha' < 0$. Daher ergeben sich

$$\alpha > 1 \quad \Longrightarrow \quad 0 < \frac{1}{\alpha} < 1 \quad \Longrightarrow \quad -1 < -\frac{1}{\alpha} < 0$$

und

$$-1 < \alpha' < 0 \quad \Longrightarrow \quad 0 < -\alpha' < 1 \quad \Longrightarrow \quad -\frac{1}{\alpha'} > 1,$$

also ist $-\frac{1}{\alpha'}$ wegen $\alpha'' = \alpha$ reduziert. Nun haben wir

$$-\frac{1}{\alpha'} = -\frac{2a}{b-\sqrt{D}} = \frac{2a}{-b+\sqrt{D}} \cdot \frac{b+\sqrt{D}}{b+\sqrt{D}} = \frac{2a(b+\sqrt{D}}{-4ac} = \frac{b+\sqrt{D}}{2(-c)}.$$

Die Behauptung folgt nun aus $D = b^2 - 4(-c)(-a)$. ■

SATZ. *Ist α reduziert reellquadratisch mit der Kettenbruchentwicklung*

$$\alpha = [\overline{u_0, u_1, \ldots, u_{k-2}, u_{k-1}}],$$

so hat $-\frac{1}{\alpha'}$ die Kettenbruchentwicklung

$$-\frac{1}{\alpha'} = [\overline{u_{k-1}, u_{k-2}, \ldots, u_1, u_0}].$$

Beweis: Wir schreiben wie üblich $\alpha_0 = \alpha$ und rekursiv $\alpha_{i+1} = \rho(\alpha_i)$. Dann ist also $\alpha_i = u_i + \frac{1}{\alpha_{i+1}}$. Es folgt

$$-\frac{1}{\alpha_{i+1}} = u_i - \alpha_i \quad \text{und damit} \quad -\frac{1}{\alpha'_{i+1}} = u_i + \frac{1}{-\frac{1}{\alpha'_i}}.$$

Wir definieren für $0 \leq i \leq k$

$$\beta_i = -\frac{1}{\alpha'_{k-i}}.$$

Nach dem Lemma ist β_i reduziert. Es gilt

$$\beta_i = -\frac{1}{\alpha'_{k-i}} = -\frac{1}{\alpha'_{(k-1-i)+1}} = u_{k-1-i} + \frac{1}{-\frac{1}{\alpha'_{k-1-i}}} = u_{k-1-i} + \frac{1}{\beta_{i+1}}.$$

Dies liefert die Kettenbruchentwicklung

$$\begin{aligned} -\frac{1}{\alpha'} &= -\frac{1}{\alpha'_k} = \beta_0 = [u_{k-1}, \beta_1] = [u_{k-1}, u_{k-2}, \beta_2] = [u_{k-1}, u_{k-2}, u_{k-3}, \beta_3] = \dots \\ &= [u_{k-1}, u_{k-2}, \dots, u_1, \beta_{k-1}] = [u_{k-1}, u_{k-2}, \dots, u_1, u_0, \beta_k] = [u_{k-1}, u_{k-2}, \dots, u_1, u_0, -\frac{1}{\alpha'_0}] = \\ &= [u_{k-1}, u_{k-2}, \dots, u_1, u_0, -\frac{1}{\alpha'}], \end{aligned}$$

was die Behauptung zeigt. ■

5. Die Kettenbruchentwicklung von $\sqrt{d}$

Sei $d \in \mathbf{N}$ keine Quadratzahl. Wir wollen die Kettenbruchentwicklung von $\sqrt{d}$ bestimmen. Wir beginnen mit einigen Beispielen:

Beispiele:

$\sqrt{2} = [1, \overline{2}]$ $\sqrt{3} = [1, \overline{1, 2}]$ $\sqrt{5} = [2, \overline{4}]$

$\sqrt{6} = [2, \overline{2, 4}]$ $\sqrt{7} = [2, \overline{1, 1, 1, 4}]$ $\sqrt{8} = [2, \overline{1, 4}]$

$\sqrt{10} = [3, \overline{6}]$ $\sqrt{11} = [3, \overline{3, 6}]$ $\sqrt{12} = [3, \overline{2, 6}]$

$\sqrt{13} = [3, \overline{1, 1, 1, 1, 6}]$ $\sqrt{14} = [3, \overline{1, 2, 1, 6}]$ $\sqrt{15} = [3, \overline{1, 6}]$

$\sqrt{17} = [4, \overline{8}]$ $\sqrt{18} = [4, \overline{4, 8}]$ $\sqrt{19} = [4, \overline{2, 1, 3, 1, 2, 8}]$

$\sqrt{20} = [4, \overline{2, 8}]$ $\sqrt{21} = [4, \overline{1, 1, 2, 1, 1, 8}]$ $\sqrt{22} = [4, \overline{1, 2, 4, 2, 1, 8}]$

$\sqrt{23} = [4, \overline{1, 3, 1, 8}]$ $\sqrt{24} = [4, \overline{1, 8}]$ $\sqrt{26} = [5, \overline{10}]$

$\sqrt{27} = [5, \overline{5, 10}]$ $\sqrt{28} = [5, \overline{3, 2, 3, 10}]$ $\sqrt{29} = [5, \overline{2, 1, 1, 2, 10}]$

$\sqrt{30} = [5, \overline{2, 10}]$ $\sqrt{31} = [5, \overline{1, 1, 3, 5, 3, 1, 1, 10}]$ $\sqrt{32} = [5, \overline{1, 1, 1, 10}]$

$\sqrt{33} = [5, \overline{1, 2, 1, 10}]$ $\sqrt{34} = [5, \overline{1, 4, 1, 10}]$ $\sqrt{35} = [5, \overline{1, 10}]$

$\sqrt{37} = [6, \overline{12}]$ $\sqrt{38} = [6, \overline{6, 12}]$ $\sqrt{39} = [6, \overline{4, 12}]$

$\sqrt{40} = [6, \overline{3, 12}]$ $\sqrt{41} = [6, \overline{2, 2, 12}]$ $\sqrt{42} = [6, \overline{2, 12}]$

$\sqrt{43} = [6, \overline{1, 1, 3, 1, 5, 1, 3, 1, 1, 12}]$ $\sqrt{44} = [6, \overline{1, 1, 1, 2, 1, 1, 1, 12}]$ $\sqrt{45} = [6, \overline{1, 2, 2, 2, 1, 12}]$

$\sqrt{46} = [6, \overline{1, 3, 1, 1, 2, 6, 2, 1, 1, 3, 1, 12}]$ $\sqrt{47} = [6, \overline{1, 5, 1, 12}]$ $\sqrt{48} = [6, \overline{1, 12}]$

Überlegungen:

- $\alpha = \alpha_0 = \sqrt{d}$ hat das Minimalpolynom $t^2 - d$, also ist die Diskriminante $D = D(\sqrt{d}) = 4d$. Für die Kettenbruchentwicklung haben wir rekursiv $\alpha_{i+1} = \rho(\alpha_i)$ mit $\alpha_i = u_i + \frac{1}{\alpha_{i+1}}$. Wir schreiben

$$\alpha_i = \frac{b_i + \sqrt{D}}{2a_i} \quad \text{mit} \quad D = b_i^2 - 4a_ic_i.$$

  Wegen $D = 4d$ folgt $b_i \equiv 0 \bmod 2$, also können wir schreiben $b_i = 2v_i$ mit einem $v_i \in \mathbf{Z}$. Damit wird

$$\alpha_i = \frac{v_i + \sqrt{d}}{a_i} \quad \text{mit} \quad d = v_i^2 - a_ic_i.$$

- Wir betrachten die Rekursionsbedingungen

$$u_i = \lfloor \alpha_i \rfloor, \quad a_{i+1} = -a_iu_i^2 + b_iu_i - c_i, \quad b_{i+1} = 2a_iu_i - b_i, \quad c_{i+1} = -a_i$$

  und

$$\alpha_{i+1} = \frac{b_{i+1} + \sqrt{D}}{2a_i} = \frac{v_{i+1} + \sqrt{d}}{a_{i+1}}.$$

  Wegen

$$4d = D = b_i^2 - 4a_ic_i = 4(v_i^2 - a_ic_i)$$

  folgt

$$a_{i+1} = \frac{d - v_{i+1}^2}{-c_{i+1}} = \frac{d - v_{i+1}^2}{a_i}.$$

  Wir können daher auf c_i verzichten. Es bleiben die Rekursionsformeln

$$u_i = \lfloor \alpha_i \rfloor, \quad v_{i+1} = a_iu_i - v_i, \quad a_{i+1} = \frac{d - v_{i+1}^2}{a_i}, \quad \alpha_{i+1} = \frac{v_{i+1} + \sqrt{d}}{a_{i+1}}.$$

- α_0 ist wegen $\alpha_0' = -\sqrt{d} < -1$ nicht reduziert. Aber α_1 ist wegen

$$\alpha_1 = \frac{1}{\sqrt{d} - \lfloor \sqrt{d} \rfloor} \quad \text{und} \quad \alpha_1' = \frac{1}{-\sqrt{d} - \lfloor \sqrt{d} \rfloor}$$

  und $\alpha_1 > 1$ und $-1 < \alpha_1' < 0$ offensichtlich reduziert. Daher wiederholt sich die Kettenbruchentwicklung, wenn man wieder auf α_1 stößt. Schreiben wir

$$\alpha = [u_0, \overline{u_1, u_2, \ldots, u_k}],$$

  so gilt also $\alpha_{k+1} = \alpha_1$. Da $\alpha_i \in \mathrm{Red}(D)$, folgt mit der früheren Formel $(\rho_{\mathrm{Red}(D)})^{-1}(\alpha) = -1 - \lfloor \frac{1}{\alpha'} \rfloor + \frac{1}{\alpha}$:

$$\begin{aligned} \alpha_k &= (\rho_{\mathrm{Red}(D)})^{-1}(\alpha_{k+1}) = (\rho_{\mathrm{Red}(D)})^{-1}(\alpha_1) = (\rho_{\mathrm{Red}(D)})^{-1}\left(\frac{1}{\sqrt{d} - \lfloor \sqrt{d} \rfloor}\right) = \\ &= -1 - \lfloor -\sqrt{d} - \lfloor \sqrt{d} \rfloor \rfloor + (\sqrt{d} - \lfloor \sqrt{d} \rfloor) = -1 - (-1 - 2\lfloor \sqrt{d} \rfloor) + \sqrt{d} - \lfloor \sqrt{d} \rfloor = \\ &= \lfloor \sqrt{d} \rfloor + \sqrt{d}. \end{aligned}$$

  Insbesondere erhalten wir jetzt aus $\alpha_0 = \sqrt{d}$ und $\alpha_k = \lfloor \sqrt{d} \rfloor + \sqrt{d}$ die Aussagen

$$u_0 = \lfloor \sqrt{d} \rfloor \quad \text{und} \quad u_k = 2\lfloor \sqrt{d} \rfloor = 2u_0.$$

- Wir betrachten jetzt die Reduziertheitsbedingungen für $\alpha_i = \frac{v_i + \sqrt{d}}{a_i}$: Sie lauten

$$1 \le 2v_i < 2\sqrt{d} \quad \text{und} \quad 2\sqrt{d} - 2v_i < 2a_i < 2\sqrt{d} + 2v_i,$$

  was gleichwertig mit

$$1 \le v_i < \sqrt{d} \quad \text{und} \quad \sqrt{d} - v_i < a_i < \sqrt{d} + v_i$$

  ist.
  Man sieht an dieser Darstellung sofort, daß die Bedingung $a_i = 1$ zu $0 < \sqrt{d} - v_i < 1$, also $v_i = \lfloor \sqrt{d} \rfloor$ und damit $\alpha_i = \lfloor \sqrt{d} \rfloor + \sqrt{d}$ führt. Das erste $i \ge 1$ mit $a_i = 1$ ist also unser gesuchtes $i = k$.

- α_1 hat die Kettenbruchentwicklung $\alpha_1 = [\overline{u_1, u_2, \ldots, u_k}]$. Nach einem früheren Lemma hat man dann
$$-\frac{1}{\alpha_1'} = [\overline{u_k, u_{k-1}, \ldots, u_1}].$$
Nun ist aber
$$-\frac{1}{\alpha_1'} = \lfloor\sqrt{d}\rfloor + \sqrt{d} = [\overline{2u_0, u_1, u_2, \ldots, u_{k-1}}].$$
Also folgt wegen der Eindeutigkeit der Kettenbruchentwicklung
$$u_1 = u_{k-1}, \quad u_2 = u_{k-2}, \quad \ldots, \quad u_{k-1} = u_1.$$

Diese Überlegungen führen zu folgendem Satz:

SATZ. *Sei $d \in \mathbf{N}$ keine Quadratzahl.*

(1) *Sei $v_0 = 0$, $a_0 = 1$. Rekursiv berechnet man*
$$u_i = \lfloor\frac{v_i + \lfloor\sqrt{d}\rfloor}{a_i}\rfloor, \quad v_{i+1} = a_i u_i - v_i, \quad a_{i+1} = \frac{d - v_{i+1}^2}{a_i},$$
bis man ein $i \geq 1$ mit $a_i = 1$ erhält. Dann setzt man $k = i$ und $u_k = 2\lfloor\sqrt{d}\rfloor$. Die Kettenbruchentwicklung von $\sqrt{d}$ ist nun
$$\sqrt{d} = [u_0, \overline{u_1, u_2, \ldots, u_k}].$$

(2) *Es gilt*
$$u_0 = \lfloor\sqrt{d}\rfloor, \quad u_k = 2u_0 = 2\lfloor\sqrt{d}\rfloor \quad \textit{und} \quad u_1 = u_{k-1}, \quad u_2 = u_{k-2}, \quad \ldots u_{k-1} = u_1.$$

(3) *Für die Nachfolger in der Kettenbruchentwicklung von $\sqrt{d}$ erhält man für $i \geq 1$*
$$\rho^i(\sqrt{d}) = \frac{v_i + \sqrt{d}}{a_i} \quad \textit{mit} \quad 1 \leq v_i < \sqrt{d} \quad \textit{und} \quad \sqrt{d} - v_i < a_i < \sqrt{d} + v_i.$$

(4) *Für die Näherungsbrüche $\frac{p_n}{q_n}$ von $\sqrt{d}$, die man rekursiv aus $p_{-2} = 0$, $q_{-2} = 1$, $p_{-1} = 1$, $q_{-1} = 0$ und*
$$p_{n+1} = u_{n+1}p_n + p_{n-1}, \quad q_{n+1} = u_{n+1}q_n + q_{n-1}$$
berechnen kann, gilt
$$p_n^2 - dq_n^2 = (-1)^{n+1}a_{n+1}.$$

Beweis: Wir müssen nur noch die Aussage über die Näherungsbrüche beweisen. Die allgemeinen Formeln liefern
$$\sqrt{d} = [u_0, u_1, \ldots, u_n, \alpha_{n+1}] = \frac{\alpha_{n+1}p_n + p_{n-1}}{\alpha_{n+1}q_n + q_{n-1}}.$$
Setzt man hier $\alpha_{n+1} = \frac{v_{n+1}+\sqrt{d}}{a_{n+1}}$ ein und macht dann Koeffizientenvergleich bzgl. $\sqrt{d}$, so erhält man:
$$p_n = v_{n+1}q_n + a_{n+1}q_{n-1} \quad \text{und} \quad dq_n = v_{n+1}p_n + a_{n+1}p_{n-1},$$
also
$$\begin{aligned} p_n^2 - dq_n^2 &= p_n(v_{n+1}q_n + a_{n+1}q_{n-1}) - q_n(v_{n+1}p_n + a_{n+1}p_{n-1}) = a_{n+1}(p_nq_{n-1} - p_{n-1}q_n) = \\ &= a_{n+1}(-1)^{n+1}, \end{aligned}$$
wie behauptet. ■

Bemerkung: Die Maple-Funktion kbe_wd bestimmt die Kettenbruchentwicklung von $\sqrt{d}$. Damit wurden die folgenden Beispiele gerechnet.

Beispiel: Für $m \in \mathbf{N}$ ist die Kettenbruchentwicklung von $\sqrt{m^2+1}$
$$\sqrt{m^2+1} = [m, \overline{2m}].$$

Beispiele: Im folgenden geben wir die Zahlen $d \leq 500$ an, deren Kettenbruchperiode ≥ 30 ist:

$$\begin{aligned}
\sqrt{331} &= [18, \overline{5, 5, 1, 6, 2, 3, 1, 1, 2, 1, 2, 1, 11, 2, 1, 1, 17, 1, 1, 2, 11, 1, 2, 1, 2, 1, 1, 3, 2, 6, 1, 5, 5, 36}] \\
\sqrt{379} &= [19, \overline{2, 7, 3, 2, 2, 6, 12, 1, 4, 1, 1, 1, 3, 4, 19, 4, 3, 1, 1, 1, 4, 1, 12, 6, 2, 2, 3, 7, 2, 38}] \\
\sqrt{421} &= [20, \overline{1, 1, 13, 5, 1, 3, 1, 2, 1, 1, 1, 2, 9, 1, 7, 3, 3, 2, 2, 3, 3, 7, 1, 9, 2, 1, 1, 1, 2, 1, 3, 1, 5, 13, 1, 1, 40}] \\
\sqrt{436} &= [20, \overline{1, 7, 2, 1, 1, 1, 13, 3, 2, 2, 5, 1, 1, 4, 10, 4, 1, 1, 5, 2, 2, 3, 13, 1, 1, 1, 2, 7, 1, 40}] \\
\sqrt{454} &= [21, \overline{3, 3, 1, 13, 2, 3, 2, 1, 1, 4, 6, 1, 7, 1, 1, 1, 20, 1, 1, 1, 7, 1, 6, 4, 1, 1, 2, 3, 2, 13, 1, 3, 3, 42}] \\
\sqrt{463} &= [21, \overline{1, 1, 13, 1, 5, 4, 1, 1, 1, 1, 2, 2, 6, 1, 3, 21, 3, 1, 6, 2, 2, 1, 1, 1, 1, 4, 5, 1, 13, 1, 1, 42}] \\
\sqrt{478} &= [21, \overline{1, 6, 3, 4, 1, 1, 5, 1, 2, 3, 1, 1, 1, 1, 1, 13, 1, 20, 1, 13, 1, 1, 1, 1, 1, 3, 2, 1, 5, 1, 1, 4, 3, 6, 1, 42}]
\end{aligned}$$

6. Die Pellsche Gleichung $x^2 - dy^2 = 1$

Sei $d \in \mathbf{N}$ keine Quadratzahl. Bei der Pellschen Gleichung sucht man nach ganzzahligen Lösungen der Gleichung $x^2 - dy^2 = 1$. Natürlich hat man immer die Lösungen $(x, y) = (\pm 1, 0)$. Außerdem ist mit (x, y) auch $(\pm x, \pm y)$ eine Lösung. Man kann sich daher auf die Suche nach Lösungen mit $x > 0$, $y > 0$ beschränken.

Beispiele: Für die angegebenen d's bestimmen wir alle Lösungen von $x^2 - dy^2 = 1$ mit $1 \leq y \leq 1000$ und $x \geq 1$. (Dazu läßt man y im angegebenen Bereich laufen und testet dann, ob $1 + dy^2$ ein Quadrat

ist.)

d	(x, y) mit $x^2 - dy^2 = 1$
2	(3,2), (17,12), (99,70), (577,408)
3	(2,1), (7,4), (26,15), (97,56), (362,209), (1351,780)
5	(9,4), (161,72)
6	(5,2), (49,20), (485,198)
7	(8,3), (127,48), (2024,765)
8	(3,1), (17,6), (99,35), (577,204)
10	(19,6), (721,228)
11	(10,3), (199,60)
12	(7,2), (97,28), (1351,390)
13	(649,180)
14	(15,4), (449,120)
15	(4,1), (31,8), (244,63), (1921,496)
17	(33,8), (2177,528)
18	(17,4), (577,136)
19	(170,39)
20	(9,2), (161,36), (2889,646)
21	(55,12)
22	(197,42)
23	(24,5), (1151,240)
24	(5,1), (49,10), (485,99), (4801,980)
26	(51,10)
27	(26,5), (1351,260)
28	(127,24)
29	
30	(11,2), (241,44), (5291,966)
31	(1520,273)
32	(17,3), (577,102)
33	(23,4), (1057,184)
34	(35,6), (2449,420)
35	(6,1), (71,12), (846,143)
37	(73,12)
38	(37,6), (2737,444)
39	(25,4), (1249,200)
40	(19,3), (721,114)
41	(2049,320)
42	(13,2), (337,52)
43	(3482,531)
44	(199,30)
45	(161,24)
46	
47	(48,7), (4607,672)
48	(7,1), (97,14), (1351,195)
50	(99,14)

Es fällt auf, daß für $d = 29$ und $d = 46$ keine Lösung gefunden wurde.

Wir wollen zunächst Kettenbrüche ins Spiel bringen, was mit Hilfe des folgenden Lemmas möglich ist. Wir betrachten dabei die etwas allgemeinere Gleichung $x^2 - dy^2 = \pm 1$.

LEMMA. *Sind $x, y \in \mathbf{N}$ mit $x^2 - dy^2 = \pm 1$, so gilt*

$$|\sqrt{d} - \frac{x}{y}| < \frac{1}{2y^2}.$$

Also ist $\frac{x}{y}$ ein Näherungsbruch in der Kettenbruchentwicklung von $\sqrt{d}$.

Beweis: Wir zeigen zunächst, daß $2y < x+\sqrt{d}y$ gilt. Wir nehmen das Gegenteil an, also $2y \geq x+\sqrt{d}y$, und wollen einen Widerspruch herleiten. Nur die Fälle $d = 2$ und $d = 3$ sind möglich. Dann folgt $x \leq (2-\sqrt{d})y$ und damit

$$x^2 - dy^2 \leq (2-\sqrt{d})^2y^2 - dy^2 = (4-4\sqrt{d})y^2 < -y^2 \leq -1,$$

was der Voraussetzung widerspricht. Also muß $2y < x + \sqrt{d}y$ gelten.
Mit $1 = |x^2 - dy^2| = |x - \sqrt{d}y||x + \sqrt{d}y|$ erhalten wir nun

$$|\sqrt{d} - \frac{x}{y}| = \frac{1}{y(x+\sqrt{d}y)} < \frac{1}{y \cdot 2y} = \frac{1}{2y^2},$$

wie behauptet. ■

Damit können wir jetzt folgenden Satz beweisen:

SATZ. *Die Kettenbruchentwicklung von $\sqrt{d}$ habe Periode k, d.h. $\sqrt{d} = [u_0, \overline{u_1, \ldots, u_k}]$, $\frac{p_i}{q_i}$ seien die zugehörigen Näherungsbrüche. Sind $x, y \in \mathbf{N}$ mit $x^2 - dy^2 = \pm 1$, so gibt es ein $l \geq 1$ mit*

$$x = p_{kl-1}, \quad y = q_{kl-1}.$$

Für das Vorzeichen bestimmt sich durch

$$p_{kl-1}^2 - dq_{kl-1}^2 = (-1)^{kl}.$$

Beweis: Das letzte Lemma zeigt, daß $\frac{x}{y}$ ein Näherungsbruch von $\sqrt{d}$ ist, d.h. es gibt ein $n \geq 1$ mit $\frac{x}{y} = \frac{p_{n-1}}{q_{n-1}}$, was wegen $\mathrm{ggT}(x, y) = \mathrm{ggT}(p_{n-1}, q_{n-1}) = 1$ sofort $x = p_{n-1}$ und $y = q_{n-1}$ liefert. Nun hatten wir die Formel

$$p_{n-1}^2 - dq_{n-1}^2 = (-1)^n a_n \quad \text{mit} \quad \rho^n(\sqrt{d}) = \frac{v_n + \sqrt{d}}{a_n}.$$

Also muß $a_n = 1$ gelten. k war minimal ≥ 1 mit $a_k = 1$. Wegen der Periodizität $a_k = a_{2k} = a_{3k} = \cdots = 1$ gibt es ein $l \geq 1$ mit $n = kl$, was dann die Behauptung beweist. ■

Algebraische Interpretation: Die Gleichung $x^2 - dy^2 = \pm 1$ läßt sich auch als $N(x + y\sqrt{d}) = \pm 1$ schreiben. Die Lösungen stehen also in Bijektion zu den Elementen der Menge

$$\mathrm{P}(d) = \{x + y\sqrt{d} \in \mathbf{Q}(\sqrt{d}) : x, y \in \mathbf{Z}, N(x + y\sqrt{d}) = \pm 1\}.$$

Die Menge $\mathrm{P}(d)$ ist nun sogar eine Untergruppe der multiplikativen Gruppe von $\mathbf{Q}(\sqrt{d})$, denn für $x_1 + y_1\sqrt{d}, x_2 + y_2\sqrt{d} \in \mathrm{P}(d)$ gilt

$$x_3 + y_3\sqrt{d} = (x_1 + y_1\sqrt{d})(x_2 + y_2\sqrt{d}) = (x_1x_2 + dy_1y_2) + (x_1y_2 + x_2y_1)\sqrt{d}$$

und $N(x_3 + y_3\sqrt{d}) = N(x_1 + y_1\sqrt{d})N(x_1 + y_1\sqrt{d}) = \pm 1$, also $x_3 + y_3\sqrt{d} \in \mathrm{P}(d)$. Außerdem ist für $x + y\sqrt{d} \in \mathrm{P}(d)$ wegen $(x + y\sqrt{d})(x - y\sqrt{d}) = \pm 1$ das Inverse

$$\frac{1}{x + y\sqrt{d}} = \pm(x - y\sqrt{d}) \in \mathrm{P}(d).$$

Beispiel: Für $d = 30$ fanden wir die Lösungen $(11, 2)$, $(241, 44)$, $(5291, 966)$ der Pellschen Gleichung. Nun findet man

$$(11+2\sqrt{30})^2 = 241+44\sqrt{30}, \quad (11+2\sqrt{30})^3 = 5291+966\sqrt{30}, \quad (11+2\sqrt{30})^4 = 116161+21208\sqrt{30}, \quad \ldots$$

LEMMA. *Wir betrachten die Abbildung*

$$\ell : \mathrm{P}(d) \to \mathbf{R}, \quad x + y\sqrt{d} \mapsto \ln|x + y\sqrt{d}|.$$

(1) *ℓ ist ein Gruppenhomomorphismus von $\mathrm{P}(d)$ in die additive Gruppe von $\mathbf{R}$.*
(2) *Für den Kern gilt:* $\mathrm{Kern}(\ell) = \{\pm 1\}$.
(3) *Ist $\ell(x + y\sqrt{d}) > 0$, so gilt $x, y \geq 1$ oder $-x, -y \geq 1$.*

Beweis:

(1) Die Gruppenhomomorphieeigenschaft folgt aus

$$\ell(\alpha_1\alpha_2) = \ln|\alpha_1\alpha_2| = \ln|\alpha_1| + \ln|\alpha_2| = \ell(\alpha_1) + \ell(\alpha_2).$$

(2) Für $x + y\sqrt{d} \in \mathrm{P}(d)$ gilt

$$x + y\sqrt{d} \in \mathrm{Kern}(\ell) \iff \ln|x + y\sqrt{d}| = 0 \iff |x + y\sqrt{d}| = 1 \iff x + y\sqrt{d} = \pm 1,$$

was $\mathrm{Kern}(\ell) = \{\pm 1\}$ zeigt.

(3) Aus $\ell(x + y\sqrt{d}) > 0$ folgt $|x + y\sqrt{d}| > 1$. Wegen $1 = |x - y\sqrt{d}||x + y\sqrt{d}|$ folgt $|x - y\sqrt{d}| < 1$. Daher können x und y nicht verschiedenes Vorzeichen haben. ■

Mit diesen Vorbereitungen können wir jetzt folgenden Satz beweisen:

SATZ. *Die Kettenbruchentwicklung von $\sqrt{d}$ habe Periode k, d.h. $\sqrt{d} = [u_0, \overline{u_1, \ldots, u_k}]$, $\frac{p_i}{q_i}$ seien die zugehörigen Näherungsbrüche. Dann gilt*

$$p_{kl-1} + q_{kl-1}\sqrt{d} = (p_{k-1} + q_{k-1}\sqrt{d})^l$$

für alle $l \geq 1$.

Beweis:

(1) Offensichtlich ist (p_{k-1}, q_{k-1}) die kleinste Lösung der Gleichung $x^2 - dy^2 = \pm 1$ mit $x, y \geq 1$.

(2) Wir betrachten $l \geq 1$ und setzen

$$\mu(l) = \frac{\ell(p_{kl-1} + q_{kl-1}\sqrt{d})}{\ell(p_{k-1} + q_{k-1}\sqrt{d})} \quad \text{und} \quad m = \lfloor \mu(l) \rfloor.$$

Sei

$$x + y\sqrt{d} = \frac{p_{kl-1} + q_{kl-1}\sqrt{d}}{(p_{k-1} + q_{k-1}\sqrt{d})^m}.$$

Dann gilt

$$0 \leq \ell(x + y\sqrt{d}) < \ell(p_{k-1} + q_{k-1}\sqrt{d}).$$

Wäre $\mu(l) \neq m$, wäre $x + y\sqrt{d}$ eine kleinere Lösung als $p_{k-1} + q_{k-1}\sqrt{d}$, was nicht sein kann. Also ist $\mu(l) \in \mathbf{N}$.

(3) Wir haben also

$$p_{kl-1} + q_{kl-1}\sqrt{d} = (p_{k-1} + q_{k-1}\sqrt{d})^{\mu(l)}.$$

Da aber auch alle Potenzen von $p_{k-1} + q_{k-1}\sqrt{d}$ als $p_{kl-1} + q_{kl-1}\sqrt{d}$ auftreten, folgt sofort $\mu(l) = l$, was wir zeigen wollten. ■

Damit erhalten wir schließlich folgenden Satz:

SATZ. *Die Kettenbruchentwicklung von $\sqrt{d}$ habe Periode k, d.h. $\sqrt{d} = [u_0, \overline{u_1, \ldots, u_k}]$, $\frac{p_i}{q_i}$ seien die zugehörigen Näherungsbrüche. Dann ist*

$$(x, y) = \begin{cases} (p_{k-1}, q_{k-1}) & \text{im Fall } k \equiv 0 \bmod 2, \\ (p_{k-1}^2 + dq_{k-1}^2, 2p_{k-1}q_{k-1}) & \text{im Fall } k \equiv 1 \bmod 2. \end{cases}$$

die kleinste nichttriviale Lösung der Pellschen Gleichung mit $x, y \geq 1$.

Beweis: Im Fall $k \equiv 0 \bmod 2$ ist $p_{k-1}^2 - dq_{k-1}^2 = 1$, also hat man eine Lösung, im Fall $k \equiv 1 \bmod 2$ hat man $p_{k-1}^2 - dq_{k-1}^2 = -1$ und berechnet damit

$$(p_{k-1}^2 + dq_{k-1}^2)^2 - d(2p_{k-1}q_{k-1})^2 = (p_{k-1}^2 - dq_{k-1}^2)^2 = 1,$$

was alles zeigt. ■

Bemerkung: Die Maple-Funktion pell(d) berechnet die oben angegebene Lösung der Pellschen Gleichung.

Beispiele: Wir geben einige d's mit $d \leq 500$ an, für die die oben angegebene Lösung groß ausfällt.

d	(x, y)
29	(9801,1820)
46	(24335,3588)
53	(66249,9100)
61	(1766319049,226153980)
109	(158070671986249,15140424455100)
181	(2469645423824185801,183567298683461940)
277	(15915007379898047584 9,9562401173878027020)
397	(838721786045180184649,42094239791738433660)
409	(250529772730924279860 49,12387899986472185821 60)
421	(387947404591492687946821 7167061449,189073995951839020880499780706260)

Beispiel: Sei (x_d, y_d) die kleinste Lösung der Pellschen Gleichung $x^2 - dy^2 = 1$. In der folgenden Tabelle sind jeweils alle d's mit $d \leq 100000$ aufgelistet, bei denen x_d größer als alle vorhergehenden ist.

d	$\ln x_d$	$\frac{\ln x_d}{\ln d \cdot \sqrt{d}}$	d	$\ln x_d$	$\frac{\ln x_d}{\ln d \cdot \sqrt{d}}$
2	1.098612	1.120738	5581	333.194689	.516983
5	2.197225	.610542	6301	352.606873	.507755
10	2.944439	.404377	6829	371.907675	.509740
13	6.475433	.700194	8269	390.332699	.475871
29	9.190240	.506812	8941	463.761565	.539059
46	10.099671	.388941	9949	486.702039	.530077
53	11.101176	.384068	12541	541.682384	.512573
61	21.292164	.663164	13381	588.193009	.535155
109	32.694063	.667510	16069	600.962312	.489518
181	42.350606	.605540	17341	639.918834	.497853
277	46.516379	.496958	24049	645.409898	.412561
397	48.178411	.404083	24229	696.618331	.443310
409	51.575280	.424070	25309	753.313438	.467032
421	77.341008	.623796	29269	759.171110	.431481
541	84.203409	.575234	30781	829.079969	.457256
661	85.691664	.513265	32341	905.264674	.484764
1021	108.908245	.491933	36061	910.360291	.456874
1069	110.226925	.483378	39901	1031.654752	.487502
1381	149.583449	.556691	40429	1037.913362	.486642
1549	162.751796	.562972	43261	1099.613858	.495249
1621	174.534130	.586540	56149	1173.758520	.452959
2389	199.141836	.523783	58909	1207.755225	.453041
3061	239.586900	.539517	60589	1273.930351	.469989
3469	259.821988	.541165	63781	1294.406524	.463280
4621	285.391008	.497524	74869	1378.013809	.448719
4789	296.050801	.504837	82021	1570.235560	.484572
4909	306.439803	.514624	92821	1666.981532	.478345

7. Übungen

Aufgabe 4.1: Ist α eine quadratische Zahl, $a, b, c, d \in \mathbf{Z}$ mit $ad - bc = \pm 1$ und

$$\beta = \frac{a\alpha + b}{c\alpha + d},$$

so haben α und β die gleiche Diskriminante.

Aufgabe 4.2: Sei D eine reellquadratische Diskriminante.

(1) Zeige für

$$S(D) = \{\frac{b+\sqrt{D}}{2a} : 1 \leq b \leq \lfloor\sqrt{D}\rfloor, \lfloor\frac{\lfloor\sqrt{D}\rfloor - b}{2}\rfloor + 1 \leq a \leq \lfloor\frac{\lfloor\sqrt{D}\rfloor + b}{2}\rfloor, b \equiv D \bmod 2\}$$

die Abschätzung

$$\#S(D) < \frac{1}{4}(D + 2\sqrt{D} + 1)$$

und folgere aus $\mathrm{Red}(D) \subseteq S(D)$ die Abschätzung

$$\#\mathrm{Red}(D) < \frac{1}{4}(D + 2\sqrt{D} + 1).$$

(2) Untersuche $\#\mathrm{Red}(D)$ experimentell.

Aufgabe 4.3: Sei D eine reellquadratische Diskriminante und

$$s(D) = \sum_{\alpha \in \mathrm{Red}(D)} \alpha.$$

(1) Beweise die Abschätzung

$$s(D) < D.$$

(Hinweis: Für $\alpha = \frac{b+\sqrt{D}}{2a} \in \mathrm{Red}(D)$ gilt $\alpha < \frac{\sqrt{D}}{a}$. Zu gegebenem a gibt es höchstens a reduzierte $\alpha \in \mathrm{Red}(D)$ der Gestalt $\alpha = \frac{b+\sqrt{D}}{2a}$.)

(2) Untersuche $s(S)$ experimentell.

(Insbesondere folgt dann auch $\#\mathrm{Red}(D) < D$.)

Aufgabe 4.4: Man untersuche experimentell, wie die Kettenbruchentwicklungen der Zahlen

$$\sqrt{m^2+1}, \quad \sqrt{m^2-1}, \quad \sqrt{m^2+2}, \quad \sqrt{m^2-2}$$

aussehen und versuche, die entdeckten Gesetzmäßigkeiten zu beweisen.

Aufgabe 4.5: Wie sehen die reduzierten reellquadratischen Zahlen $\alpha = [\overline{u_0, u_1, \ldots, u_{k-1}}]$ aus, bei denen die Periode $[\overline{u_0, u_1, \ldots, u_{k-1}}]$ periodisch ist, d.h. wobei $u_i = u_{k-1-i}$ gilt?

Aufgabe 4.6: Für ein Nichtquadrat $d \in \mathbf{N}$ bezeichne $\ell(d)$ die Periodenlänge der Kettenbruchentwicklung von $\sqrt{d}$.

(1) Charakterisiere die d's mit $\ell(d) = 1$.

(2) Charakterisiere die d's mit $\ell(d) = 2$.

(3) Definiere eine Folge d_i rekursiv wie folgt, wobei mit $d_1 = 2$ zu beginnen ist:

$$d_i = \min\{d \in \mathbf{N} : d \text{ Nichtquadrat und } \ell(d) > \ell(d_{i-1})\}.$$

Berechne $(d_1, \ell(d_1))$, $(d_2, \ell(d_2))$, $\ldots$, $(d_{100}, \ell(d_{100}))$.

(4) Versuche experimentell eine Vermutung aufzustellen, wie man $\ell(d)$ nach oben abschätzen kann.

KAPITEL 5

Faktorisierung mit Kettenbrüchen (CFRAC)

1. Einführung

Will man eine natürliche Zahl N faktorisieren, geht man meist nach folgendem Schema vor:

(1) Teile aus N alle kleinen Primteiler heraus, z.B. alle Primteiler $\leq 10^6$.
(2) Teste mit einem Primzahltest, ob N zusammengesetzt oder wahrscheinlich prim ist, z.B. mit einem Fermat-Test: Ist $2^{N-1} \not\equiv 1 \bmod N$, so ist N zusammengesetzt, andernfalls wahrscheinlich prim.
(3) Ist N zusammengesetzt, bestimme einen nichttrivialen Teiler von N, d.h. eine nichttriviale Faktorisierung $N = N_1 N_2$ und beginne mit N_1 und N_2 von vorne mit 2.
(4) Ist N wahrscheinlich prim, beweise, daß N prim ist, oder gib dich zufrieden mit der Aussage, daß N 'wahrscheinlich prim' ist.

Wir interessieren uns hier für den 3. Punkt, der Bestimmung eines nichttrivialen Teilers einer (zusammengesetzten) natürlichen Zahl (ohne kleine Teiler). Dies ist ein schwieriges Problem.

Bemerkung: Ist x irgendeine ganze Zahl mit $1 \leq x \leq N-1$, so gilt

$$1 \leq \mathrm{ggT}(x, N) < N \quad \text{und} \quad \mathrm{ggT}(x, N) | N.$$

Gilt $\mathrm{ggT}(x, N) > 1$, so hat man also sofort einen nichttrivialen Teiler von N gefunden und man kann aufhören. Da man den ggT mit dem euklidischen Algorithmus sehr schnell berechnen kann, kann man in den folgenden Anwendungen oft o.E. $\mathrm{ggT}(x, N) = 1$ annehmen.

LEMMA. *Sei N eine ungerade natürliche Zahl.*

(1) *Sind x und y ganze Zahlen mit*

$$\mathrm{ggT}(x, N) = \mathrm{ggT}(y, N) = 1 \quad \textit{und} \quad x^2 \equiv y^2 \bmod N,$$

so gilt

$$N = \mathrm{ggT}(x+y, N) \cdot \mathrm{ggT}(x-y, N).$$

Außerdem ist die Faktorisierung genau dann trivial, wenn $x \equiv \pm y \bmod N$ gilt, d.h.

$$\{\mathrm{ggT}(x+y, N), \mathrm{ggT}(x-y, N)\} = \{1, N\} \quad \Longleftrightarrow \quad x \equiv \pm y \bmod N.$$

(2) *Ist $N = p_1^{e_1} \dots p_r^{e_r}$ mit r verschiedenen Primteilern p_i, so gilt*

$$\frac{\#\{(x,y) \in \mathbf{Z}/(N)^* \times \mathbf{Z}/(N)^* : x^2 \equiv y^2 \bmod N \textit{ und } 1 < \mathrm{ggT}(x+y, N) < N\}}{\#\{(x,y) \in \mathbf{Z}/(N)^* \times \mathbf{Z}/(N)^* : x^2 \equiv y^2 \bmod N\}} = 1 - \frac{1}{2^{r-1}}.$$

Beweis: 1. Wegen $\mathrm{ggT}(2, N) = 1$, $\mathrm{ggT}(x, N) = 1$ und

$$\mathrm{ggT}(x+y, x-y, N) = \mathrm{ggT}(x+y, 2x, N) = \mathrm{ggT}(x+y, x, N) = 1$$

sind die Zahlen $\mathrm{ggT}(x+y, N)$ und $\mathrm{ggT}(x-y, N)$ teilerfremd, so daß man nachfolgendes Produkt auseinanderziehen kann unter Verwendung von $N | x^2 - y^2$:

$$N = \mathrm{ggT}(x^2 - y^2, N) = \mathrm{ggT}((x+y)(x-y), N) = \mathrm{ggT}(x+y, N)\mathrm{ggT}(x-y, N).$$

Weiter gilt

$$\mathrm{ggT}(x \pm y, N) = N \quad \Longleftrightarrow \quad N | x \pm y \quad \Longleftrightarrow \quad x \equiv \mp y \bmod N.$$

2. Die Menge

$$G = \{(x, y) \in \mathbf{Z}/(N)^* \times \mathbf{Z}/(N)^* : x^2 \equiv y^2 \bmod N\}$$

wird mit komponentenweiser Multiplikation zu einer Gruppe, genauso wie

$$H = \{(z_1, \ldots, z_r) \in \mathbf{Z}/(p_1)^* \times \cdots \times \mathbf{Z}/(p_r)^* : z_i \equiv \pm 1 \bmod p_i\}.$$

Mit dem chinesischen Restsatz sieht man, daß

$$\psi : G \to H : \quad (x, y) \mapsto (\frac{x}{y} \bmod p_1, \ldots, \frac{x}{y} \bmod p_r)$$

ein surjektiver Gruppenhomomorphismus ist. Nun gilt für $(x, y) \in G$

$$x \equiv \pm y \bmod N \quad \Longleftrightarrow \quad \psi((x, y)) \in \{(1, \ldots, 1), (-1, \ldots, -1)\},$$

woraus sich folgende Gleichungen ergeben:

$$\begin{aligned} & \frac{\#\{(x, y) \in \mathbf{Z}/(N)^* \times \mathbf{Z}/(N)^* : x^2 \equiv y^2 \bmod N \text{ und } \mathrm{ggT}(x + y, N) \in \{1, N\}\}}{\#\{(x, y) \in \mathbf{Z}/(N)^* \times \mathbf{Z}/(N)^* : x^2 \equiv y^2 \bmod N\}} = \\ = & \frac{\#\{(x, y) \in \mathbf{Z}/(N)^* \times \mathbf{Z}/(N)^* : x^2 \equiv y^2 \bmod N \text{ und } x \equiv \pm y \bmod N\}}{\#\{(x, y) \in \mathbf{Z}/(N)^* \times \mathbf{Z}/(N)^* : x^2 \equiv y^2 \bmod N\}} = \\ = & \frac{\#\psi^{-1}(\{(1, \ldots, 1), (-1, \ldots, -1)\})}{\#\psi^{-1}(H)} = \frac{\#\{(1, \ldots, 1), (-1, \ldots, -1)\}}{\#H} = \frac{2}{2^r} = \frac{1}{2^{r-1}}, \end{aligned}$$

was die Behauptung beweist. (Dabei wurde benutzt, daß für surjektive Gruppenhomomorphismen alle Urbilder $\psi^{-1}(h)$ gleichmächtig sind.) ■

Bemerkungen: Lösungen der Kongruenz $x^2 \equiv y^2 \bmod N$ liefern also Faktorisierungen von N. Das macht folgenden Ansatz plausibel:

- Suche nach 'zufälligen' Lösungen der Kongruenz
$$x^2 \equiv y^2 \bmod N.$$
- Teste, ob
$$\mathrm{ggT}(x + y, N) \quad \text{oder} \quad \mathrm{ggT}(x - y, N)$$
nichttriviale Teiler von N sind.
- Findet man 'zufällig' Lösungen der Gleichung $x^2 \equiv y^2 \bmod N$, so sollte (wegen des vorangegangenen Lemmas) in mindestens 50% der Fälle $\mathrm{ggT}(x+y, N)$ oder $\mathrm{ggT}(x-y, N)$ ein nichttrivialer Teiler sein.

Die Frage ist nun: Wie findet man x und y mit $x^2 \equiv y^2 \bmod N$?

2. Eine Idee zur Konstruktion von Kongruenzen $x^2 \equiv y^2 \bmod N$

Wir geben die grundlegende Idee zur Konstruktion von Kongruenzen $x^2 \equiv y^2 \bmod N$ an:

(1) Wir wählen endlich viele (kleine) Primzahlen $p_1, p_2, \ldots, p_{n-1}$ (und $p_0 = -1$), die sogenannte Faktorbasis.

(2) Wir suchen ganze Zahlen x_i und y_i mit
$$x_i^2 \equiv y_i \bmod N,$$
sodaß die Primfaktorzerlegung von y_i nur mit $p_0, \ldots, p_{n-1}$ auskommt, d.h.
$$y_i = (-1)^{b_0} \cdot p_1^{b_1} \cdot \cdots \cdot p_{n-1}^{b_{n-1}}$$
mit Zahlen $b_0, b_1, \ldots, b_{n-1} \in \mathbf{N}_0$. Neben x_i und y_i merkt man sich nur
$$c_i = (b_0 \bmod 2, b_1 \bmod 2, \ldots, b_{n-1} \bmod 2)$$
Dann ist also (mit $c_i = (c_{i0}, c_{i1}, \ldots, c_{i,n-1})$)
$$y_i = (-1)^{c_{i0}} p_1^{c_{i1}} \cdot \cdots \cdot p_{n-1}^{c_{i,n-1}} \cdot \text{Quadrat}.$$

(3) Wir nehmen jetzt an, daß wir eine Anzahl von Relationen $x_i^2 \equiv y_i \bmod N$ gefunden haben. Findet man jetzt Indizes $i_1, \ldots, i_l$, so daß

$$c_{i_1} + c_{i_2} + \cdots + c_{i_l} \equiv 0 \bmod 2$$

gilt, so ist

$$y_{i_1} \cdot y_{i_2} \cdot \cdots \cdot y_{i_l} = (-1)^{c_{i_1 0} + \cdots + c_{i_l 0}} \cdot p_1^{c_{i_1 1} + \cdots + c_{i_l 1}} \cdot \cdots \cdot p_{n-1}^{c_{i_1, n-1} + \cdots + c_{i_l, n-1}} \cdot \text{Quadrat}$$

ein Quadrat in $\mathbf{Z}$, da der Exponent bei allen p_i's gerade ist. Also erhält man durch Aufmultiplizieren

$$(x_{i_1} \ldots x_{i_l})^2 \equiv (\sqrt{y_{i_1} \ldots y_{i_l}})^2 \bmod N,$$

und wir probieren, ob

$$\text{ggT}(x_{i_1} x_{i_2} \ldots x_{i_l} + \sqrt{y_{i_1} \ldots y_{i_l}}, N) \quad \text{oder} \quad \text{ggT}(x_{i_1} x_{i_2} \ldots x_{i_l} - \sqrt{y_{i_1} \ldots y_{i_l}}, N)$$

ein nichttrivialer Teiler von N ist.

Beispiel: Wir betrachten $N = 7201970021$ und wählen als Faktorbasis $p_0 = -1$, $p_1 = 2$, $p_2 = 5$, $p_3 = 7$. Für die nachfolgenden x_i und y_i gilt $x_i^2 \equiv y_i \bmod N$.

$x_0 = 1742268380$	$y_0 = -42875 = -5^3 \cdot 7^3 = -5 \cdot 7 \cdot \text{Quadrat}$	$c_0 = (1,0,1,1)$
$x_1 = 4108098645$	$y_1 = -100 = -2^2 \cdot 5^2 = -\text{Quadrat}$	$c_1 = (1,0,0,0)$
$x_2 = 756791010$	$y_2 = 6860 = 2^2 \cdot 5 \cdot 7^3 = 5 \cdot 7 \cdot \text{Quadrat}$	$c_2 = (0,0,1,1)$
$x_3 = 7201969971$	$y_3 = 2500 = 2^2 \cdot 5^4 = \text{Quadrat}$	$c_3 = (0,0,0,0)$
$x_4 = 5378321218$	$y_4 = -87500 = -2^2 \cdot 5^5 \cdot 7 = -5 \cdot 7 \cdot \text{Quadrat}$	$c_4 = (1,0,1,1)$

Aus $c_3 = 0$ folgt, daß y_3 ein Quadrat ist, aber aus

$$\text{ggT}(x_3 + \sqrt{y_3}, N) = N, \quad \text{ggT}(x_3 - \sqrt{y_3}, N) = 1$$

erhält man keinen nichttrivialen Teiler von N.
Nun ist $c_1 + c_2 + c_4 \equiv 0 \bmod 2$, also ist $y_1 y_2 y_4$ ein Quadrat, nämlich $y_1 y_2 y_4 = 245000^2 = (2^3 \cdot 5^4 \cdot 7^2)^2$. Mit

$$\text{ggT}(x_1 x_2 x_4 + \sqrt{y_1 y_2 y_4}, N) = 80021 \quad \text{und} \quad \text{ggT}(x_1 x_2 x_4 - \sqrt{y_1 y_2 y_4}, N) = 90001$$

erhält man die Faktorisierung $N = 80021 \cdot 90001$.

Es stellen sich jetzt zwei Fragen:

(1) Wie findet man Relationen $x_i^2 \equiv y_i \bmod N$, sodaß in der Primfaktorzerlegung von y_i nur Primzahlen der Faktorbasis vorkommen?
(2) Hat man eine Anzahl von gewünschten Relationen $x_i^2 \equiv y_i \bmod N$, $i = 0, \ldots, r$ gefunden, wie findet man Indizes $i_1, \ldots, i_l$, so daß $y_{i_1} \ldots y_{i_l}$ ein Quadrat ist bzw. $c_{i_1} + \cdots + c_{i_l} \equiv 0 \bmod 2$ gilt?

Wir behandeln zuerst die zweite Frage.

3. Etwas Lineare Algebra

Seien jetzt Kongruenzen $x_i^2 \equiv y_i \bmod N$, $0 \le i \le r$ gegeben, sodaß in der Primfaktorzerlegung von y_i nur Primzahlen der Faktorbasis $\{p_1, \ldots, p_{n-1}\}$ vorkommen.
Wir betrachten Matrizen $E = (e_{ij})$ mit einer Begleitmatrix $W = (w_{ij})$, die die folgenden Bedingungen erfüllen:

- E und W haben Einträgen aus $\{0, 1\}$.
- Die Matrix E hat Zeilenindizes von 0 bis m, Spaltenindizes von 0 bis $n-1$, die Matrix W hat Zeilenindizes von 0 bis m und Spaltenindizes von 0 bis r, also

$$E = \begin{pmatrix} e_{00} & e_{01} & \ldots & e_{0,n-1} \\ e_{10} & e_{11} & \ldots & e_{1,n-1} \\ \vdots & \vdots & & \vdots \\ e_{m0} & e_{m1} & \ldots & e_{m,n-1} \end{pmatrix} \quad \text{und} \quad W = \begin{pmatrix} w_{00} & w_{01} & \ldots & w_{0r} \\ w_{10} & w_{11} & \ldots & w_{1r} \\ \vdots & \vdots & & \vdots \\ w_{m0} & w_{m1} & \ldots & w_{mr} \end{pmatrix}.$$

- Für alle Zeilenindizes i gilt:

$$\prod_{\substack{0\le j\le r\\ w_{ij}=1}} y_j = \prod_{0\le j\le r} y_j^{w_{ij}} = (-1)^{e_{i0}}\cdot p_1^{e_{i1}}\cdot p_2^{e_{i2}}\cdot\ldots\cdot p_{n-1}^{e_{i,n-1}}\cdot \text{Quadrat} \qquad (*).$$

(Auf der linken Seite multipliziert man also alle y_j's auf, für die $w_{ij}=1$ gilt.)

Beispiel: Haben wir die Relationen $x_i^2 \equiv y_i \bmod N$, $0\le i\le r$ mit

$$y_i = (-1)^{c_{i0}}\cdot p_1^{c_{i1}}\cdot\ldots\cdot p_{n-1}^{c_{i,n-1}}\cdot\text{Quadrat} \quad \text{mit} \quad c_{ij}\in\{0,1\},$$

so setzen wir $m=r$,

$$E=\begin{pmatrix} c_{00} & c_{01} & \ldots & c_{0,n-1}\\ c_{10} & c_{11} & \ldots & c_{1,n-1}\\ \vdots & \vdots & & \vdots\\ c_{r0} & c_{r1} & \ldots & c_{r,n-1}\end{pmatrix} \quad \text{und} \quad W=\begin{pmatrix} 1 & 0 & \ldots & 0\\ 0 & 1 & \ldots & 0\\ \vdots & \vdots & & \vdots\\ 0 & 0 & \ldots & 1\end{pmatrix}.$$

Offensichtlich ist die Bedingung $(*)$ an E und W erfüllt.

Die Matrizen E und W sind so konstruiert, daß man sofort folgende Aussage erhält:

FOLGERUNG. *Ist eine Zeile (mit dem Index i) der Matrix E identisch 0, so ist*

$$\prod_{\substack{0\le j\le 1\\ w_{ij}=1}} y_j \text{ ein Quadrat}, \quad \Big(\prod_{\substack{0\le j\le r\\ w_{mj}=1}} x_j\Big)^2 \equiv \Big(\sqrt{\prod_{\substack{0\le j\le r\\ w_{mj}=1}} y_j}\Big)^2 \bmod N,$$

und

$$\text{ggT}\Big(\prod_{\substack{0\le j\le r\\ w_{mj}=1}} x_j + \sqrt{\prod_{\substack{0\le j\le r\\ w_{mj}=1}} y_j}, N\Big) \quad \textit{und} \quad \text{ggT}\Big(\prod_{\substack{0\le j\le r\\ w_{mj}=1}} x_j - \sqrt{\prod_{\substack{0\le j\le r\\ w_{mj}=1}} y_j}, N\Big)$$

sind (eventuell triviale) Teiler von N.

Um Teiler von N zu finden, muß man also nach Zeilen von E suchen, die identisch 0 sind. Hier hilft nun das folgende Lemma wesentlich weiter:

LEMMA. *Die Bedingung $(*)$ an die Matrizen E und W bleibt erhalten, wenn man folgende elementare Zeilenoperationen bei beiden Matrizen gleichzeitig durchführt:*

- *Vertauschen der Zeilen i_1 und i_2.*
- *Addition (modulo 2) der Zeile i_1 zur Zeile i_2 (mit $i_1\ne i_2$).*

Beweis: Sei also Indizes i_1 und i_2 gegeben. Die Bedingung $(*)$ liefert dann

$$\begin{aligned}\prod_j y_j^{w_{i_1 j}} &= (-1)^{e_{i_1 0}}\cdot p_1^{e_{i_1 1}}\cdot p_2^{e_{i_1 2}}\cdot\ldots\cdot p_{n-1}^{e_{i_1,n-1}}\cdot\text{Quadrat},\\ \prod_j y_j^{w_{i_2 j}} &= (-1)^{e_{i_2 0}}\cdot p_1^{e_{i_2 1}}\cdot p_2^{e_{i_2 2}}\cdot\ldots\cdot p_{n-1}^{e_{i_2,n-1}}\cdot\text{Quadrat}.\end{aligned}$$

Es ist klar, daß Zeilenvertauschung die Bedingung $(*)$ nicht ändert. Durch Multiplikation der Gleichungen erhält man

$$\prod_j y_j^{w_{i_1 j}+w_{i_2 j}} = (-1)^{e_{i_1 0}+e_{i_2 0}}\cdot p_1^{e_{i_1 1}+e_{i_2 1}}\cdot\ldots\cdot p_{n-1}^{e_{i_1,n-1}+e_{i_2,n-1}}\cdot\text{Quadrat}.$$

Da man quadratische Anteile von p_j in das Quadrat stecken kann, kann man $p_j^{e_{i_1 j}+e_{i_2 j}}$ durch $p_j^{(e_{i_1 j}+e_{i_2 j}) \bmod 2}$ ersetzen. Ist $w_{i_1 j}=w_{i_2 j}=1$, so teilt y_j^2 die linke Seite, also auch die rechte. Daher kann man y_j^2 aus dem Quadrat der rechten Seite herausteilen und somit $y_j^{w_{i_1 j}+w_{i_2 j}}$ durch $y_j^{(w_{i_1 j}+w_{i_2 j}) \bmod 2}$ ersetzen. Also erhält man

$$\prod_j y_j^{(w_{i_1 j}+w_{i_2 j}) \bmod 2} = (-1)^{(e_{i_1 0}+e_{i_2 0}) \bmod 2}\cdot p_1^{(e_{i_1 1}+e_{i_2 1}) \bmod 2}\cdot\ldots\cdot p_{n-1}^{(e_{i_1,n-1}+e_{i_2,n-1}) \bmod 2}\cdot\text{Quadrat},$$

was identisch mit

$$\prod_{\substack{0\le j\le r\\ (w_{i_1 j}+w_{i_2 j}) \bmod 2=1}} y_j = (-1)^{(e_{i_1 0}+e_{i_2 0}) \bmod 2} \cdot p_1^{(e_{i_1 1}+e_{i_2 1}) \bmod 2} \cdot \ldots \cdot p_{n-1}^{(e_{i_1,n-1}+e_{i_2,n-1}) \bmod 2} \cdot \text{Quadrat}$$

ist. Diese Gleichung entspricht der Addition der Zeilen i_1 und i_2 in den Matrizen E und W. Damit ist also die Bedingung $(*)$ erfüllt, wenn man eine Zeile zu einer anderen (modulo 2) addiert. ■

Bemerkungen:

(1) Mit den im Lemma beschriebenen elementaren Zeilenumformungen kann man die Matrix E auf Zeilenstufenform (modulo 2) bringen. Eventuell vorhandene 0-Zeilen sind dabei unten zu finden.

(2) Die Matrix E hat $m+1$ Zeilen (Indizes von 0 bis m) und n Spalten (Indizes von 0 bis $n-1$). Ist also $m \ge n$ und ist E in Zeilenstufenform, so ist die letzte Zeile der Matrix E identisch 0 und man kann testen, ob sich damit ein nichttrivialer Teiler von N finden läßt.

Wir skizzieren jetzt eine Möglichkeit, wie man neue Kongruenzen $x_i^2 \equiv y_i \bmod N$ gleich in die Matrizen E und W einarbeiten kann.

(1) Die aktuellen Matrizen sind jeweils

$$E = \begin{pmatrix} e_{00} & e_{01} & \ldots & e_{0,n-1} \\ e_{10} & e_{11} & \ldots & e_{1,n-1} \\ \vdots & \vdots & & \vdots \\ e_{m0} & e_{m1} & \ldots & e_{m,n-1} \end{pmatrix} \quad \text{und} \quad W = \begin{pmatrix} w_{00} & w_{01} & \ldots & w_{0r} \\ w_{10} & w_{11} & \ldots & w_{1r} \\ \vdots & \vdots & & \vdots \\ w_{m0} & w_{m1} & \ldots & w_{mr} \end{pmatrix},$$

wobei E bereits Zeilenstufenform hat.

(2) Zu Beginn setzen wir $m = -1$, $r = -1$.

(3) Finden wir eine neue Kongruenz, so setzen wir $m := m+1$, $r := r+1$ und nennen die gefundene Kongruenz $x_r^2 \equiv y_r \bmod N$. Die noch nicht definierte Zeile m der Matrix E wird durch die Exponenten $e_{mj} \in \{0,1\}$ aus der Gleichung

$$y_r = (-1)^{e_{m0}} \cdot p_1^{e_{m1}} \cdot \ldots \cdot p_{n-1}^{e_{m,n-1}} \cdot \text{Quadrat}$$

gegeben. Bei der um eine Spalte und eine Zeile vergrößerten Matrix W werden alle neuen Einträge 0 gesetzt, bis auf $w_{mr} = 1$. Also

$$E = \begin{pmatrix} e_{00} & e_{01} & \ldots & e_{0,n-1} \\ e_{10} & e_{11} & \ldots & e_{1,n-1} \\ \vdots & \vdots & & \vdots \\ e_{m-1,0} & e_{m-1,1} & \ldots & e_{m-1,n-1} \\ e_{m0} & e_{m1} & \ldots & e_{m,n-1} \end{pmatrix}, \quad W = \begin{pmatrix} w_{00} & w_{01} & \ldots & w_{0,r-1} & 0 \\ w_{10} & w_{11} & \ldots & w_{1,r-1} & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ w_{m-1,0} & w_{m-1,1} & \ldots & w_{m-1,r-1} & 0 \\ 0 & 0 & \ldots & 0 & 1 \end{pmatrix}.$$

(E und W erfüllen $(*)$.)

(4) Wir bringen jetzt E durch gleichzeitige elementare Zeilenumformungen von E und W auf Zeilenstufenform (modulo 2), wobei wir wissen, daß die Zeilen i mit $0 \le i \le m-1$ bereits Zeilenstufenform haben.
Algorithmisch kann man dies so machen:
Beginne mit $i := 0$ und $j := 0$. Führe den nachfolgenden Schritt aus, solange $i < m$ gilt, mit folgender Fallunterscheidung:
Fall $e_{ij} \equiv 1 \bmod 2$, $e_{mj} \equiv 0 \bmod 2$**:** Setze $i := i+1$ und $j := j+1$.
Fall $e_{ij} \equiv 1 \bmod 2$, $e_{mj} \equiv 1 \bmod 2$**:** Addiere die i-te Zeile zur m-ten Zeile (in E und W), setze dann $i := i+1$ und $j := j+1$.
Fall $e_{ij} \equiv 0 \bmod 2$, $e_{mj} \equiv 1 \bmod 2$**:** Vertausche die i-te Zeile mit der m-ten Zeile (in E und W), setze dann $i := i+1$, $j := j+1$.
Fall $e_{ij} \equiv 0 \bmod 2$, $e_{mj} \equiv 0 \bmod 2$**:** Setze $j := j+1$.

(5) Ist die letzte Zeile von E (mit dem Index m) identisch 0, so ist

$$\prod_{\substack{0\le j\le r\\ w_{mj}=1}} y_j$$

ein Quadrat und wir testen, ob

$$\mathrm{ggT}(\prod_{\substack{0\le j\le r\\ w_{mj}=1}} x_j + \sqrt{\prod_{\substack{0\le j\le r\\ w_{mj}=1}} y_j}, N) \quad \text{oder} \quad \mathrm{ggT}(\prod_{\substack{0\le j\le r\\ w_{mj}=1}} x_j - \sqrt{\prod_{\substack{0\le j\le r\\ w_{mj}=1}} y_j}, N)$$

ein nichttrivialer Teiler von N ist.
Haben wir einen nichttrivialen Teiler gefunden, beenden wir das Verfahren.
Im andern Fall setzen wir $m := m - 1$, d.h. wir streichen die letzte Zeile von E und W, da die enthaltene Information nutzlos ist.

(6) Nun gehen wir zurück zu Schritt 3.

Beispiel: Wir betrachten $N = 30001600021$ mit der Faktorbasis $p_0 = -1$, $p_1 = 2$, $p_2 = 3$, $p_3 = 5$, $p_4 = 7$. Nacheinander geben wir Kongruenzen $x_i^2 \equiv y_i \bmod N$ an, die wie folgt verarbeitet werden:

- $x_0 = 1732097$, $y_0 = 15309 = 3 \cdot 7 \cdot$ (Quadrat) liefert die Ausgangsmatrizen
$$E = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 \end{pmatrix} \text{ und } W = \begin{pmatrix} 1 \end{pmatrix}.$$
- $x_1 = 28258749633$, $y_1 = -5292 = -3 \cdot$ (Quadrat) liefert nun
$$E = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \end{pmatrix} \text{ und } W = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$
Zeilenvertauschung liefert
$$E = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix} \text{ und } W = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$
- $x_2 = 13519593710$, $y_2 = -2268 = -7 \cdot$ (Quadrat) liefert nun
$$E = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix} \text{ und } W = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$
Die ersten beiden Zeilen werden zur letzten addiert:
$$E = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \text{ und } W = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}.$$
Die letzte Zeile von E ist 0. Daher erhält man ein Quadrat: $y_0 y_1 y_2 = 428652^2$.
$$\mathrm{ggT}(x_0 x_1 x_2 + \sqrt{y_0 y_1 y_2}, N) = 30001600021, \quad \mathrm{ggT}(x_0 x_1 x_2 - \sqrt{y_0 y_1 y_2}, N) = 1.$$
Die Faktorisierung ist trivial. Daher entfernen wir die letzte Zeile von E und W:
$$E = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix} \text{ und } W = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}.$$
- $x_3 = 17777827814$, $y_3 = 4116 = 3 \cdot 7 \cdot$ (Quadrat) liefert
$$E = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix} \text{ und } W = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$
Addition der beiden letzten Zeilen ergibt
$$E = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \text{ und } W = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}.$$
Daher ist $y_0 y_3$ ein Quadrat, nämlich 7938^2. Wieder ist die zugehörige Faktorisierung trivial. Daher wird die letzte Zeile entfernt:
$$E = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix} \text{ und } W = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$

- $x_4 = 19596918126$, $y_4 = -151263 = -7 \cdot$ (Quadrat) liefert

$$E = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix} \text{ und } W = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

  Addition aller Zeilen zur letzten

$$E = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \text{ und } W = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

  $y_0 y_1 y_4 = 3500658^2$. Triviale Faktorisierung. Daher wird die letzte Zeile entfernt:

$$E = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix} \text{ und } W = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

- $x_5 = 13558239728$, $y_5 = -128625 = -3 \cdot 5 \cdot 7 \cdot$ (Quadrat)

$$E = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 \end{pmatrix} \text{ und } W = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

  Addition der ersten zur letzten Zeile ergibt:

$$E = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} \text{ und } W = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

- $x_6 = 1444310552$, $y_6 = -500 = -5 \cdot$ (Quadrat) liefert

$$E = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix} \text{ und } W = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

  Addition aller Zeilen zur letzten liefert:

$$E = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \text{ und } W = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

  Die letzte Zeile von E ist 0. Daher ist $y_0 y_5 y_6 = 992250^2$. Allerdings liefert dies nur die triviale Faktorisierung. Wir streichen die letzte Zeile:

$$E = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} \text{ und } W = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

- $x_7 = 20061265988$, $y_7 = 165375 = 3 \cdot 5 \cdot$ (Quadrat) liefert

$$E = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix} \text{ und } W = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

  Addition von Zeilen 2 und 3 zur letzten ergibt

$$E = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \text{ und } W = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

Die letzte Zeile von E ist 0. Daher ist $y_0y_1y_5y_7 = 1312746750^2$. Allerdings ergibt sich wieder nur die triviale Faktorisierung. Wir streichen daher die letzte Zeile:

$$E = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} \text{ und } W = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

- $x_8 = 10770303153$, $y_8 = 70875 = 5 \cdot 7 \cdot$ (Quadrat) liefert

$$E = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} \text{ und } W = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Addition der vorletzten zur letzten Zeile:

$$E = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \text{ und } W = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}.$$

Die letzte Zeile von E ist 0. Daher ist $y_1y_5y_8 = 6945750^2$. Aber wieder erhält man nur die triviale Faktorisierung. Wir entfernen die letzte Zeile:

$$E = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} \text{ und } W = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}.$$

- $x_9 = 30001599771$, $y_9 = 62500 =$ (Quadrat).

$$E = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \text{ und } W = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Die letzte Zeile von E ist 0. Daher ist $y_9 = 250^2$. Aber nur die triviale Faktorisierung.

$$E = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} \text{ und } W = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

- $x_{10} = 16920901760$, $y_{10} = -55125 = -5 \cdot$ (Quadrat) liefert

$$E = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix} \text{ und } W = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Wir addieren alle Zeilen zur letzten und erhalten:

$$E = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \text{ und } W = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Die letzte Zeile von E ist 0. Also erhalten wir ein Quadrat: $y_0y_5y_{10} = 10418625^2$.

$\operatorname{ggT}(x_0x_5x_{10} + \sqrt{y_0y_5y_{10}}, N) = 100003$ und $\operatorname{ggT}(x_0x_5x_{10} - \sqrt{y_0y_5y_{10}}, N) = 300007$,

was die Faktorisierung $N = 100003 \cdot 300007$ liefert.

Bemerkungen:

(1) Bei der zuletzt beschriebenen Methode werden neugewonnene Kongruenzen $x_i^2 \equiv y_i \bmod N$ gleich in die Matrizen E und W eingefügt, die Matrizen dann normalisiert und getestet, ob man einen nichttrivialen Teiler findet.

(2) Hat man ein größeres Problem und will man parallelisieren, wird man erst nach vielen Kongruenzen $x_i^2 \equiv y_i \bmod N$ suchen, was man unabhängig auf vielen Rechner machen kann, schließlich setzt man alles zusammen und erhält große Matrizen E und W, die man dann zentral weiterverarbeitet.

(3) Die beschriebene allgemeine Idee wird von verschiedenen Faktorisierungsmethoden benutzt, die sich im wesentlichen dadurch unterscheiden, wie man an geeignete Relationen $x_i^2 \equiv y_i \bmod N$ kommt, so daß sich y_i mit den Primzahlen der gewählten Faktorbasis faktorisieren läßt:
- CFRAC (nach Morrison-Brillhart), das Kettenbrüche benutzt und weiter unten beschrieben wird,
- QS (quadratic sieve) - quadratisches Sieb,
- MPQS (multiple polynomial quadratic sieve),
- NFS (number field sieve) - Zahlkörpersieb.

4. Das Morrision-Brillhart-Faktorisierungsverfahren CFRAC

Wir erinnern an die Kettenbruchentwicklung von $\sqrt{d}$:

(1) Sei $v_0 = 0$, $a_0 = 1$. Rekursiv berechnet man

$$u_i = \lfloor \frac{v_i + \lfloor\sqrt{d}\rfloor}{a_i} \rfloor, \quad v_{i+1} = a_i u_i - v_i, \quad a_{i+1} = \frac{d - v_{i+1}^2}{a_i} \text{ (exakte Division)},$$

bis man ein $i \geq 1$ mit $a_i = 1$ erhält. Dann setzt man $k = i$ und $u_k = 2\lfloor\sqrt{d}\rfloor$. Die Kettenbruchentwicklung von $\sqrt{d}$ ist nun

$$\sqrt{d} = [u_0, \overline{u_1, u_2, \ldots, u_k}],$$

wobei gilt

$$u_0 = \lfloor\sqrt{d}\rfloor, \quad u_k = 2u_0 = 2\lfloor\sqrt{d}\rfloor \quad \text{und} \quad u_1 = u_{k-1}, \quad u_2 = u_{k-2}, \quad \ldots u_{k-1} = u_1.$$

Außerdem hat man für $i \geq 1$ die Abschätzungen

$$1 \leq v_i < \sqrt{d} \quad \text{und} \quad \sqrt{d} - v_i < a_i < \sqrt{d} + v_i.$$

(2) Für die Näherungsbrüche $\frac{P_i}{Q_i}$ von $\sqrt{d}$, die man rekursiv aus $P_{-2} = 0$, $Q_{-2} = 1$, $P_{-1} = 1$, $Q_{-1} = 0$ und

$$P_i = u_i P_{i-1} + P_{i-2}, \quad Q_i = u_i Q_{i-1} + Q_{i-2}$$

berechnen kann, gilt

$$P_i^2 - dQ_i^2 = (-1)^{i+1} a_{i+1}.$$

(3) Wählt man $d = N$, so hat man $P_i^2 - NQ_i^2 = (-1)^{i+1} a_{i+1}$ und damit

$$P_i^2 \equiv (-1)^{i+1} a_{i+1} \bmod N \quad \text{mit der Abschätzung} \quad 1 \leq a_{i+1} < 2\sqrt{N}.$$

Kommen also in a_{i+1} nur Primzahlen der gewählten Faktorbasis vor, haben wir eine gesuchte Kongruenz gefunden. Dies ergibt sofort das Faktorisierungsverfahren CFRAC von Morrision-Brillhart.

Morrison-Brillhart-Faktorisierungsverfahren CFRAC: Sei N eine (ungerade zusammengesetzte) natürliche Zahl.

(1) Sei

$$P_{-2} = 0, \quad P_{-1} = 1, \quad v_0 = 0, \quad a_0 = 1, \quad d := N, \quad i := 0, \quad m := -1, \quad r := -1.$$

(2) Berechne

$$u_i = \lfloor \frac{v_i + \lfloor\sqrt{d}\rfloor}{a_i} \rfloor, \quad P_i = (u_i P_{i-1} + P_{i-2}) \bmod N, \quad v_{i+1} = a_i u_i - v_i, \quad a_{i+1} = \frac{d - v_{i+1}^2}{a_i},$$

(3) Teste, ob sich a_{i+1} mit $p_1, p_2, \ldots, p_{n-1}$ vollständig faktorisieren läßt. Wenn nein, gehe zu 4. Andernfalls führt man folgende Schritte aus:

(a) Setze $r := r + 1$ und $m := m + 1$. Dann hat man mit

$$x_r = P_i \quad \text{und} \quad y_r = (-1)^{i+1} a_{i+1}$$

eine Relation

$$x_r^2 \equiv y_r \bmod N.$$

Gilt

$$y_r = (-1)^{b_0} p_1^{b_1} \cdots p_{n-1}^{b_{n-1}},$$

so erweitert man E (wie früher beschrieben) um eine Zeile e_m und W um eine Zeile und eine Spalte, indem man setzt

$$e_m = (b_0 \bmod 2, b_1 \bmod 2, \ldots, b_{n-1} \bmod 2)$$

und

$$w_{mr} = 1, \quad w_{mj} = 0 \text{ für } 0 \le j < r, \quad w_{jr} = 0 \text{ für } 0 \le j < m.$$

(b) Durch gleichzeitige elementare Zeilenumformungen (modulo 2) von E und W bringt man E auf Zeilenstufenform.

(c) Ist die letzte Zeile e_m von E identisch 0, so ist

$$\prod_{\substack{0 \le j \le r \\ w_{mj}=1}} y_j$$

ein Quadrat und man testet, ob

$$\text{ggT}\Big(\prod_{\substack{0 \le j \le r \\ w_{mj}=1}} x_j + \sqrt{\prod_{\substack{0 \le j \le r \\ w_{mj}=1}} y_j}, N\Big) \quad \text{oder} \quad \text{ggT}\Big(\prod_{\substack{0 \le j \le r \\ w_{mj}=1}} x_j - \sqrt{\prod_{\substack{0 \le j \le r \\ w_{mj}=1}} y_j}, N\Big)$$

ein nichttrivialer Teiler von N ist.

Hat man einen nichttrivialen Teiler gefunden, ist man fertig. Ansonsten streicht man die letzte Zeile von E und W, d.h. man setzt $m := m - 1$.

(4) Setze $i := i + 1$ und gehe zu 2.

Bemerkung: Wir haben zum vorgestellten CFRAC-Faktorisierungsverfahren mehrere Programme geschrieben, mit denen die nachfolgenden Beispiele erstellt wurden.

Beispiel: Wir wollen $N = 1622623$ faktorisieren und wählen als Faktorbasis $-1, 2, 3, 7, 19$. Wir beginnen zunächst mit der Kettenbruchentwicklung von $\sqrt{N}$:

$u_0 = 1273$	$P_0 = 1273$	$v_1 = 1273$	$a_1 = 2094$	$(-1)^1 a_1 = -2 \cdot 3 \cdot 349$	
$u_1 = 1$	$P_1 = 1274$	$v_2 = 821$	$a_2 = 453$	$(-1)^2 a_2 = 3 \cdot 151$	
$u_2 = 4$	$P_2 = 6369$	$v_3 = 991$	$a_3 = 1414$	$(-1)^3 a_3 = -2 \cdot 7 \cdot 101$	
$u_3 = 1$	$P_3 = 7643$	$v_4 = 423$	$a_4 = 1021$	$(-1)^4 a_4 = 1021$	
$u_4 = 1$	$P_4 = 14012$	$v_5 = 598$	$a_5 = 1239$	$(-1)^5 a_5 = -3 \cdot 7 \cdot 59$	
$u_5 = 1$	$P_5 = 21655$	$v_6 = 641$	$a_6 = 978$	$(-1)^6 a_6 = 2 \cdot 3 \cdot 163$	
$u_6 = 1$	$P_6 = 35667$	$v_7 = 337$	$a_7 = 1543$	$(-1)^7 a_7 = -1543$	
$u_7 = 1$	$P_7 = 57322$	$v_8 = 1206$	$a_8 = 109$	$(-1)^8 a_8 = 109$	
$u_8 = 22$	$P_8 = 1296751$	$v_9 = 1192$	$a_9 = 1851$	$(-1)^9 a_9 = -3 \cdot 617$	
$u_9 = 1$	$P_9 = 1354073$	$v_{10} = 659$	$a_{10} = 642$	$(-1)^{10} a_{10} = 2 \cdot 3 \cdot 107$	
$u_{10} = 3$	$P_{10} = 491101$	$v_{11} = 1267$	$a_{11} = 27$	$(-1)^{11} a_{11} = -3^3$	$*$
$u_{11} = 94$	$P_{11} = 461500$	$v_{12} = 1271$	$a_{12} = 266$	$(-1)^{12} a_{12} = 2 \cdot 7 \cdot 19$	$*$
$u_{12} = 9$	$P_{12} = 1399355$	$v_{13} = 1123$	$a_{13} = 1359$	$(-1)^{13} a_{13} = -3^2 \cdot 151$	
$u_{13} = 1$	$P_{13} = 238232$	$v_{14} = 236$	$a_{14} = 1153$	$(-1)^{14} a_{14} = 1153$	
$u_{14} = 1$	$P_{14} = 14964$	$v_{15} = 917$	$a_{15} = 678$	$(-1)^{15} a_{15} = -2 \cdot 3 \cdot 113$	
$u_{15} = 3$	$P_{15} = 283124$	$v_{16} = 1117$	$a_{16} = 553$	$(-1)^{16} a_{16} = 7 \cdot 79$	
$u_{16} = 4$	$P_{16} = 1147460$	$v_{17} = 1095$	$a_{17} = 766$	$(-1)^{17} a_{17} = -2 \cdot 383$	
$u_{17} = 3$	$P_{17} = 480258$	$v_{18} = 1203$	$a_{18} = 229$	$(-1)^{18} a_{18} = 229$	
$u_{18} = 10$	$P_{18} = 1082171$	$v_{19} = 1087$	$a_{19} = 1926$	$(-1)^{19} a_{19} = -2 \cdot 3^2 \cdot 107$	
$u_{19} = 1$	$P_{19} = 1562429$	$v_{20} = 839$	$a_{20} = 477$	$(-1)^{20} a_{20} = 3^2 \cdot 53$	
$u_{20} = 4$	$P_{20} = 841395$	$v_{21} = 1069$	$a_{21} = 1006$	$(-1)^{21} a_{21} = -2 \cdot 503$	
$u_{21} = 2$	$P_{21} = 1622596$	$v_{22} = 943$	$a_{22} = 729$	$(-1)^{22} a_{22} = 3^6$	$*$
$u_{22} = 3$	$P_{22} = 841314$	$v_{23} = 1244$	$a_{23} = 103$	$(-1)^{23} a_{23} = -103$	
$u_{23} = 24$	$P_{23} = 720033$	$v_{24} = 1228$	$a_{24} = 1113$	$(-1)^{24} a_{24} = 3 \cdot 7 \cdot 53$	
$u_{24} = 2$	$P_{24} = 658757$	$v_{25} = 998$	$a_{25} = 563$	$(-1)^{25} a_{25} = -563$	
$u_{25} = 4$	$P_{25} = 109815$	$v_{26} = 1254$	$a_{26} = 89$	$(-1)^{26} a_{26} = 89$	
$u_{26} = 28$	$P_{26} = 488331$	$v_{27} = 1238$	$a_{27} = 1011$	$(-1)^{27} a_{27} = -3 \cdot 337$	
$u_{27} = 2$	$P_{27} = 1086477$	$v_{28} = 784$	$a_{28} = 997$	$(-1)^{28} a_{28} = 997$	
$u_{28} = 2$	$P_{28} = 1038662$	$v_{29} = 1210$	$a_{29} = 159$	$(-1)^{29} a_{29} = -3 \cdot 53$	
$u_{29} = 15$	$P_{29} = 440177$	$v_{30} = 1175$	$a_{30} = 1522$	$(-1)^{30} a_{30} = 2 \cdot 761$	
$u_{30} = 1$	$P_{30} = 1478839$	$v_{31} = 347$	$a_{31} = 987$	$(-1)^{31} a_{31} = -3 \cdot 7 \cdot 47$	
$u_{31} = 1$	$P_{31} = 296393$	$v_{32} = 640$	$a_{32} = 1229$	$(-1)^{32} a_{32} = 1229$	
$u_{32} = 1$	$P_{32} = 152609$	$v_{33} = 589$	$a_{33} = 1038$	$(-1)^{33} a_{33} = -2 \cdot 3 \cdot 173$	
$u_{33} = 1$	$P_{33} = 449002$	$v_{34} = 449$	$a_{34} = 1369$	$(-1)^{34} a_{34} = 1369$	
$u_{34} = 1$	$P_{34} = 601611$	$v_{35} = 920$	$a_{35} = 567$	$(-1)^{35} a_{35} = -3^4 \cdot 7$	$*$
$u_{35} = 3$	$P_{35} = 631212$	$v_{36} = 781$	$a_{36} = 1786$	$(-1)^{36} a_{36} = 2 \cdot 19 \cdot 47$	
$u_{36} = 1$	$P_{36} = 1232823$	$v_{37} = 1005$	$a_{37} = 343$	$(-1)^{37} a_{37} = -7^3$	$*$
$u_{37} = 6$	$P_{37} = 1537658$	$v_{38} = 1053$	$a_{38} = 1498$	$(-1)^{38} a_{38} = 2 \cdot 7 \cdot 107$	
$u_{38} = 1$	$P_{38} = 1147858$	$v_{39} = 445$	$a_{39} = 951$	$(-1)^{39} a_{39} = -3 \cdot 317$	
$u_{39} = 1$	$P_{39} = 1062893$	$v_{40} = 506$	$a_{40} = 1437$	$(-1)^{40} a_{40} = 3 \cdot 479$	
$u_{40} = 1$	$P_{40} = 588128$	$v_{41} = 931$	$a_{41} = 526$	$(-1)^{41} a_{41} = -2 \cdot 263$	
$u_{41} = 4$	$P_{41} = 170159$	$v_{42} = 1173$	$a_{42} = 469$	$(-1)^{42} a_{42} = 7 \cdot 67$	
$u_{42} = 5$	$P_{42} = 1438923$	$v_{43} = 1172$	$a_{43} = 531$	$(-1)^{43} a_{43} = -3^2 \cdot 59$	
$u_{43} = 4$	$P_{43} = 1057982$	$v_{44} = 952$	$a_{44} = 1349$	$(-1)^{44} a_{44} = 19 \cdot 71$	
$u_{44} = 1$	$P_{44} = 874282$	$v_{45} = 397$	$a_{45} = 1086$	$(-1)^{45} a_{45} = -2 \cdot 3 \cdot 181$	
$u_{45} = 1$	$P_{45} = 309641$	$v_{46} = 689$	$a_{46} = 1057$	$(-1)^{46} a_{46} = 7 \cdot 151$	
$u_{46} = 1$	$P_{46} = 1183923$	$v_{47} = 368$	$a_{47} = 1407$	$(-1)^{47} a_{47} = -3 \cdot 7 \cdot 67$	
$u_{47} = 1$	$P_{47} = 1493564$	$v_{48} = 1039$	$a_{48} = 386$	$(-1)^{48} a_{48} = 2 \cdot 193$	
$u_{48} = 5$	$P_{48} = 538628$	$v_{49} = 891$	$a_{49} = 2147$	$(-1)^{49} a_{49} = -19 \cdot 113$	
$u_{49} = 1$	$P_{49} = 409569$	$v_{50} = 1256$	$a_{50} = 21$	$(-1)^{50} a_{50} = 3 \cdot 7$	$*$

Die Zeilen, bei denen sich a_{i+1} mit der Faktorbasis faktorisieren läßt, sind mit $*$ gekennzeichnet. Sie liefern über $x_r = P_i \bmod N$, $y_r = (-1)^{i+1} a_{i+1}$ die folgenden Kongruenzen $x_r^2 \equiv y_r \bmod N$:

$x_0 = 491101$	$y_0 = -27 = (-1) \cdot 3 \cdot$ Quadrat	$c_0 = (1,0,1,0,0)$
$x_1 = 461500$	$y_1 = 266 = 2 \cdot 7 \cdot 19 \cdot$ Quadrat	$c_1 = (0,1,0,1,1)$
$x_2 = 1622596$	$y_2 = 729 =$ Quadrat	$c_2 = (0,0,0,0,0)$
$x_3 = 601611$	$y_3 = -567 = (-1) \cdot 7 \cdot$ Quadrat	$c_3 = (1,0,0,1,0)$
$x_4 = 1232823$	$y_4 = -343 = (-1) \cdot 7 \cdot$ Quadrat	$c_4 = (1,0,0,1,0)$
$x_5 = 409569$	$y_5 = 21 = 3 \cdot 7 \cdot$ Quadrat	$c_5 = (0,0,1,1,0)$

Dabei geben die Einträge von c_i die Exponenten modulo 2 von y_i bzgl. $p_0, \ldots, p_4$ an.
Wir erstellen nun sukzessiv die Matrizen E und W:

- $x_0^2 \equiv y_0 \bmod N$ führt zu

$$E = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \end{pmatrix} \quad \text{und} \quad W = \begin{pmatrix} 1 \end{pmatrix}.$$

- $x_1^2 \equiv y_1 \bmod N$ führt zu

$$E = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix} \quad \text{und} \quad W = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

- $x_2^2 \equiv y_2 \bmod N$ führt zu

$$E = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad \text{und} \quad W = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

  y_2 ist also Quadrat, allerdings erhält man wegen

$$\mathrm{ggT}(x_2 + \sqrt{y_2}, N) = N, \quad \mathrm{ggT}(x_2 - \sqrt{y_2}, N) = 1$$

  nur die triviale Faktorisierung. Also wird die letzte Zeile entfernt:

$$E = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix} \quad \text{und} \quad W = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

- $x_3^2 \equiv y_3 \bmod N$ liefert

$$E = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix} \quad \text{und} \quad W = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

  Addition von Zeile 0 zu Zeile 2 ergibt Zeilenstufenform:

$$E = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix} \quad \text{und} \quad W = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix}.$$

- $x_4^2 \equiv y_4 \bmod N$ liefert

$$E = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix} \quad \text{und} \quad W = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

  Addition von Zeile 0 und Zeile 2 zu Zeile 3 liefert

$$E = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad \text{und} \quad W = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

  Also ist $y_3 y_4$ ein Quadrat, und tatsächlich findet man

$$\mathrm{ggT}(x_3 x_4 + \sqrt{y_3 y_4}, N) = 907, \quad \mathrm{ggT}(x_3 x_4 - \sqrt{y_3 y_4}, N) = 1789,$$

was die Faktorisierung $N = 907 \cdot 1789$ liefert. (Natürlich sieht man sofort, daß y_2 und y_3y_4 Quadrate sind, hätte diese Kandidaten also gleich ausprobieren können. Wir wollten aber den systematischen Weg wählen.)

Frage: Wie sollte die Faktorbasis $-1, p_1, p_2, \ldots, p_{n-1}$ gewählt werden?

(1) Wir starten immer mit $p_0 = -1$ und $p_1 = 2$.

(2) Gesuchte Relationen erhalten wir aus der Kettenbruchgleichung

$$P_i^2 - NQ_i^2 = (-1)^{i+1}a_{i+1}.$$

Ist p eine ungerade Primzahl mit $p|a_{i+1}$, so folgt $P_i^2 \equiv NQ_i^2 \bmod p$. Wegen $\mathrm{ggT}(P_i, Q_i) = 1$ ist $Q_i \not\equiv 0 \bmod p$ und damit

$$N \equiv (\frac{P_i}{Q_i})^2 \bmod p,$$

d.h. N ist ein Quadrat modulo p. Dann gilt $N \equiv 0 \bmod p$ oder $N \not\equiv 0 \bmod p$ und damit (mit dem kleinen Satz von Fermat)

$$N^{\frac{p-1}{2}} \equiv (\frac{P_i}{Q_i})^{p-1} \equiv 1 \bmod p.$$

Also sind nur solche Primzahlen p in der Faktorbasis sinnvoll, für die

$$N^{\frac{p-1}{2}} \equiv 0 \text{ oder } 1 \bmod N$$

gilt, was sich schnell testen läßt. (In der Sprache des Legendre-Symbols: Nur ungerade Primzahlen p mit $\left(\frac{N}{p}\right) = 0$ oder 1 sind sinnvoll.)

(3) Wie groß sollte man die Faktorbasis wählen? — Wählt man die Faktorbasis sehr klein, findet man vielleicht keine oder zu wenige Relationen. Wählt man eine große Faktorbasis, so erhält man schneller Relationen, dafür braucht man mehr Relationen, außerdem werden die Matrizen E und W größer.

Um ein gewissen Gefühl für das skizzierte Faktorisierungsverfahren zu gewinnen, haben wir einige Zahlen faktorisiert und dabei ein Anzahltripel (n_i, n_r, n_t) bestimmt. Dabei gilt:

- n_i ist die Anzahl der Gleichungen $P_i^2 \equiv (-1)^{i+1}a_{i+1} \bmod N$, für die getestet wurde, ob sich a_{i+1} mit der Faktorbasis vollständig faktorisieren läßt. .
- n_r ist dabei die Anzahl der a_{i+1}'s die sich vollständig mit der Faktorbasis faktorisieren lassen.
- n_t zählt, wie oft tatsächlich ggT's gebildet wurden um einen Teiler von N zu finden.

Beispiel: Die Tripel folgender Tabelle sind (n_i, n_r, n_t), die Zahl n gibt die Größe der Faktorbasis $\{p_0, p_1, \ldots, p_{n-1}\}$ an. Es wurden 20-stellige Zahlen N gewählt.

N	$n = 10$	$n = 50$	$n = 100$	$n = 200$
48918721545363495263	(108162,8,2)	(10521,54,9)	(5599,90,6)	(4296,160,14)
39508907647178976011	(512320,25,17)	(5552,45,2)	(4022,91,4)	(2875,140,6)
58455891741562488449	(152789,19,10)	(5672,46,3)	(3956,82,4)	(2920,160,10)
31521634445818659997	(198050,15,7)	(6416,71,29)	(3824,103,19)	(3383,163,21)
18683756432742845551	(107345,11,4)	(2280,52,9)	(966,54,2)	(966,89,2)
17976491140236713597	(678661,24,16)	(7776,53,8)	(3061,89,7)	(2844,144,8)
21562256780339489707	(294757,17,9)	(2408,49,5)	(1801,81,3)	(1556,132,6)
18376241448087408323	(145966,10,2)	(5165,52,6)	(3365,98,19)	(2535,145,11)
18158694282388471213	(169919,13,5)	(2346,34,3)	(2346,59,6)	(2346,122,7)
30781405526327662231	(41775,8,2)	(2862,52,6)	(1938,87,2)	(1473,128,2)

Die zugehörigen Faktorisierungen sind:

$$\begin{aligned}
48918721545363495263 &= 7519954963 \cdot 6505188101\\
39508907647178976011 &= 6582086519 \cdot 6002489869\\
58455891741562488449 &= 6894087209 \cdot 8479134361\\
31521634445818659997 &= 5221791149 \cdot 6036555953\\
18683756432742845551 &= 2080777241 \cdot 8979219911\\
17976491140236713597 &= 2961404561 \cdot 6070258477\\
21562256780339489707 &= 2472337631 \cdot 8721404597\\
18376241448087408323 &= 6013929029 \cdot 3055613287\\
18158694282388471213 &= 6329051777 \cdot 2869101869\\
30781405526327662231 &= 8562969781 \cdot 3594711451
\end{aligned}$$

Nun folgen 30-stellige Zahlen:

N	$n = 100$	$n = 300$	$n = 500$	$n = 700$
664611256030932029991149705201	(235909,93,1)	(84604,259,3)	(68468,409,2)	(61064,526,3)
256112604031810931418893914253	(429712,95,2)	(126346,256,6)	(89849,371,1)	(82737,525,5)
136470722644397386869660777467	(208067,96,2)	(71678,257,7)	(60271,388,3)	(52351,506,2)
142139171560267204905441854563	(270623,94,3)	(105125,267,7)	(79216,400,8)	(65860,512,5)
163800353595967788634862839859	(358740,94,2)	(105801,256,3)	(85305,395,5)	(73348,540,6)
282717497436976106174925433613	(290780,98,5)	(88832,284,9)	(70243,417,4)	(58364,525,3)
121679999974103241606449716747	(288266,98,3)	(93340,251,1)	(73058,383,3)	(68038,521,1)
566947825319448965440309861471	(152414,97,3)	(56672,251,2)	(48273,372,4)	(43372,482,4)
344309275412326467725006433833	(308122,93,1)	(116875,267,4)	(87144,410,3)	(77149,525,5)
603886949839961668745797442683	(445198,99,5)	(126953,271,3)	(92662,428,3)	(80700,553,3)

mit den Faktorisierungen

$$\begin{aligned}
664611256030932029991149705201 &= 922537519955053 \cdot 720416505188117\\
256112604031810931418893914253 &= 413651631798463 \cdot 619150474321331\\
136470722644397386869660777467 &= 649586894087171 \cdot 210088479134377\\
142139171560267204905441854563 &= 267005221791157 \cdot 532346036555959\\
163800353595967788634862839859 &= 686912080777243 \cdot 238458979219913\\
282717497436976106174925433613 &= 672982961404573 \cdot 420096070258481\\
121679999974103241606449716747 &= 721542472337627 \cdot 168638721404561\\
566947825319448965440309861471 &= 768896013929033 \cdot 737353055613287\\
344309275412326467725006433833 &= 503006329051757 \cdot 684502869101869\\
603886949839961668745797442683 &= 879070244946137 \cdot 686960971903859
\end{aligned}$$

Für die praktische Umsetzung des Faktorisierungsverfahrens gibt es noch ein Problem:

Problem: Ist a eine natürliche Zahl, sind $p_1, \ldots, p_{n-1}$ verschiedene Primzahlen, wie testet man, ob in der Primfaktorzerlegung von a nur die Primzahlen $p_1, \ldots, p_{n-1}$ vorkommen?
Wir geben zwei einfache Ideen an:

(1) Man teilt $p_1, \ldots, p_{n-1}$ solange aus a heraus, bis es nicht mehr geht:
- Setze $\widetilde{a} := a$.
- Führe für $i = 0, 1, \ldots, n-1$ folgenden Schritt durch: Solange $\widetilde{a} \equiv 0 \bmod p_i$ gilt, setze $\widetilde{a} := \frac{\widetilde{a}}{p_i}$.

- Ist nun $\widetilde{a} = 1$, so läßt sich a mit $p_1, \ldots, p_{n-1}$ faktorisieren, andernfalls nicht.

(2) Hier wird der euklidische Algorithmus benutzt:
- Setze $\widetilde{p} := p_1 p_2 \ldots p_{n-1}$, $\widetilde{a} := a$.
- Führe folgenden Schritt aus, bis $\mathrm{ggT}(\widetilde{p}, \widetilde{a}) = 1$ gilt: $\widetilde{a} := \frac{\widetilde{a}}{\mathrm{ggT}(\widetilde{p}, \widetilde{a})}$.
- Genau läßt sich a mit $p_1, \ldots, p_{n-1}$ faktorisieren, wenn $\widetilde{a} = 1$ gilt.

Zugehörig haben wir zwei NTL-Funktionen cfrac_1 und cfrac_2, die sich nur in diesem Punkt unterscheiden. Das folgende Beispiel testet die unterschiedliche Ausführungsdauer.

Beispiel: Wir betrachten zwei 40-stellige Zahlen:

$$\begin{aligned} N_1 &= 1519466596648096389347287996048270032547 = \\ &= 28629413651631798493 \cdot 53073619150474321279, \\ N_2 &= 708079746578928884362571476578114177362 3 = \\ &= 58600008631982804471 \cdot 120832703460127480513. \end{aligned}$$

Die linke Tabelle gehört zu N_1, die rechte zu N_2. Die Abkürzung 1 steht für die NTL-Funktion cfrac_1, entsprechend 2 für cfrac_2.

n	(n_i, n_r, n_t)	1	2
100	(21672381,100,2)	24:01	19:13
200	(5972093,192,2)	10:45	6:42
300	(3314744,286,2)	8:33	4:37
400	(2215414,377,1)	7:22	3:44
500	(1886679,478,7)	7:43	3:47
600	(1619363,554,4)	7:53	3:49
700	(1396524,632,4)	7:56	3:50
800	(1261735,703,2)	8:13	3:59
900	(1171794,793,3)	8:40	4:16
1000	(1088360,883,2)	9:05	4:34

n	(n_i, n_r, n_t)	1	2
100	(72112236,94,1)	1:19:06	58:50
200	(14734533,198,2)	26:22	15:01
300	(7364765,291,3)	18:59	9:07
400	(4867012,387,5)	16:11	7:12
500	(3682568,470,1)	15:01	6:25
600	(3028868,552,1)	14:51	6:07
700	(2616327,648,1)	14:43	6:05
800	(2295969,733,1)	14:45	6:07
900	(2135317,820,1)	15:28	6:28
1000	(1969617,903,1)	15:58	6:47

Man sieht also, daß das Verfahren mit dem euklidischen Algorithmus deutlich schneller ist. Außerdem sieht man, daß die Größe der Faktorbasis eine entscheidende Rolle spielt.

Die Konsequenz aus dem vorangegangenen Beispiel ist, daß man zum Test, ob a vollständig mit $p_1, \ldots, p_{n-1}$ faktorisierbar ist, das Verfahren mit dem euklidischen Algorithmus wählen sollte. (Dies entscheidet allerdings nichts über andere mögliche Varianten.) Zugehörig haben wir auch eine gmp-Funktion cfrac.

In den folgenden Beispielen geht es darum, die Faktorisierungszeiten mit cfrac_2 (NTL), cfrac (gmp) und maple zu vergleichen.

Beispiel: Wir faktorisieren die folgenden 30-stelligen Zahlen.

$$\begin{aligned}
N_1 = 664611256030932029991149705201 &= 922537519955053 \cdot 720416505188117 \\
N_2 = 256112604031810931418893914253 &= 413651631798463 \cdot 619150474321331 \\
N_3 = 136470722644397386869660777467 &= 649586894087171 \cdot 210088479134377 \\
N_4 = 142139171560267204905441854563 &= 267005221791157 \cdot 532346036555959 \\
N_5 = 163800353595967788634862839859 &= 686912080777243 \cdot 238458979219913 \\
N_6 = 282717497436976106174925433613 &= 672982961404573 \cdot 420096070258481 \\
N_7 = 121679999974103241606449716747 &= 721542472337627 \cdot 168638721404561 \\
N_8 = 566947825319448965440309861471 &= 768896013929033 \cdot 737353055613287 \\
N_9 = 344309275412326467725006433833 &= 503006329051757 \cdot 684502869101869 \\
N_{10} = 603886949839961668745797442683 &= 879070244946137 \cdot 686960971903859
\end{aligned}$$

N	(n_i, n_r, n_t) (200)	ntl (200)	gmp (200)	(n_i, n_r, n_t) (500)	gmp (500)	maple
N_1	(113031,185,1)	9 sec	5 sec	(68468,409,2)	12 sec	12 sec
N_2	(183583,182,1)	12 sec	7 sec	(89849,371,1)	11 sec	18 sec
N_3	(85128,184,5)	5 sec	3 sec	(60271,388,3)	10 sec	10 sec
N_4	(128074,173,4)	8 sec	5 sec	(79216,400,8)	12 sec	14 sec
N_5	(147057,181,2)	9 sec	5 sec	(85305,395,5)	11 sec	16 sec
N_6	(121898,184,6)	8 sec	4 sec	(70243,417,4)	10 sec	14 sec
N_7	(130955,179,3)	8 sec	5 sec	(73058,383,3)	10 sec	14 sec
N_8	(77142,184,5)	5 sec	3 sec	(48273,372,4)	8 sec	9 sec
N_9	(158139,195,3)	12 sec	7 sec	(87144,410,3)	13 sec	17 sec
N_{10}	(165103,182,1)	10 sec	6 sec	(92662,428,3)	13 sec	17 sec

Beispiel: Wir wählen 40-stellige Zahlen und jeweils Faktorbasen mit 500 Elementen.

$$N_1 = 1350086321707056825250765608973898564467 = 75593922537519955067 \cdot 17859720416505188201$$
$$N_2 = 151946659664809638934728799604827003254 7 = 28629413651631798493 \cdot 53073619150474321279$$
$$N_3 = 17075135803390629233212811711852486923 43 = 89085450136582086547 \cdot 19167143206002489869$$
$$N_4 = 39565321207567248175176995919536975670 11 = 480222670052217911 41 \cdot 823895323460365560 71$$
$$N_5 = 23269637711150191025943548021183396463 47 = 35314686912080777273 \cdot 65892238458979219939$$
$$N_6 = 27903661987146058807945138508071787381 09 = 51574672982961404581 \cdot 54103420096070258489$$
$$N_7 = 19755748830820257273371765856969928242 41 = 60934721542472337649 \cdot 32421168638721404609$$
$$N_8 = 22224103553081010072898261666767846817 49 = 38241768896013929021 \cdot 58114737353055613369$$
$$N_9 = 33303821172923421583304825307930622079 39 = 91479053858562969827 \cdot 36405952803594711457$$
$$N_{10} = 45911874723019164201274878334017327863 87 = 83272284729390500321 \cdot 55134640381513175347$$

N	(n_i, n_r, n_t) (500)	gmp (500)	maple
N_1	(1776863,452,1)	2:40	6:16
N_2	(1886679,478,7)	2:49	6:54
N_3	(4170342,471,2)	5:22	15:15
N_4	(1605421,455,1)	2:15	6:21
N_5	(2753458,461,2)	3:37	10:27
N_6	(2570465,459,1)	3:53	8:42
N_7	(1100754,452,2)	2:04	4:05
N_8	(1240503,455,2)	1:59	4:38
N_9	(2969850,456,1)	4:08	10:42
N_{10}	(1898833,480,8)	2:36	7:24

5. CFRAC mit Multiplikator

Morrison und Brillhart demonstrierten das CFRAC-Verfahren mit der Faktorisierung der 39-stelligen 7. Fermatzahl

$$F_7 = 2^{2^7} + 1 = 340282366920938463463374607431768211457.$$

Starten wir unser Programm cfrac_ntl.c (mit der Funktione cfrac_2) mit einer Faktorbasis der Größe $n = 500$, so bricht es allerdings nach kurzer Zeit mit der Fehlermeldung 'segmentation fault' (Speicherprobleme) ab. Wir schauen uns die erzeugten Relationen $x_i^2 \equiv y_i \bmod N$ an:

```
x[0]=18446744073709551616 y[0]=1
x[1]=340282366920938463463374607431768211456 y[1]=1
x[2]=340282366920938463444927863358058659841 y[2]=1
x[3]=1 y[3]=1
x[4]=18446744073709551616 y[4]=1
x[5]=340282366920938463463374607431768211456 y[5]=1
x[6]=340282366920938463444927863358058659841 y[6]=1
```

```
x[7]=1 y[7]=1
x[8]=18446744073709551616 y[8]=1
x[9]=340282366920938463463374607431768211456 y[9]=1
x[10]=340282366920938463444927863358058659841 y[10]=1
x[11]=1 y[11]=1
```

Die Folge (x_i, y_i) scheint also periodisch mit Periode 4 zu sein. Wie ist das zu erklären?
Die Zahl F_7 hat die Gestalt $N = m^2 + 1$ mit $m = 2^{2^6}$. Für solche Zahlen gilt folgender (einfach zu beweisender) Satz:

SATZ. *Ist $N = m^2 + 1$, so gilt für die Kettenbruchentwicklung der Zahl $\sqrt{N}$:*

$$\sqrt{N} = [m, \overline{2m}] \quad \textit{mit} \quad m = \lfloor \sqrt{N} \rfloor.$$

Mit den üblichen Bezeichnungen $\rho^i(\sqrt{N}) = \frac{v_i + \sqrt{N}}{a_i}$ gilt $v_i = m$, $a_i = 1$ für alle $i \geq 1$. Für die Näherungsbrüche $\frac{P_i}{Q_i}$ gilt

$$P_0 = m, \quad P_1 = 2m^2 + 1, \quad P_{2+4j} = -m, \quad P_{3+4j} = 1, \quad P_{4+4j} = m, \quad P_{5+4j} = -1,$$

$$Q_{0+4j} = 1 + 4j, \quad Q_{1+4j} = (2 + 4j)m, \quad Q_{2+4j} = -(3 + 4j), \quad Q_{3+4j} = -(4 + 4j)m$$

für alle $j \geq 0$. Modulo N erhält man damit

$$P_{0+4j} = m, \quad P_{1+4j} \equiv m^2 \bmod N, \quad P_{2+4j} \equiv m^2 - m + 1 \bmod N, \quad P_{3+4j} \equiv 1 \bmod N \quad \textit{für alle } j \geq 0.$$

Anwendung: Was liefert $P_i^2 \equiv (-1)^{i+1} a_{i+1} \bmod N$ für $N = m^2 + 1$?
Wir haben immer $a_{i+1} = 1$ und unterscheiden nach $P_i \bmod N$:

- $i = 0 + 4j$: $P_i \equiv m$ liefert die Relation $m^2 \equiv -1 \bmod N$.
- $i = 1 + 4j$: $P_i \equiv m^2 \equiv -1 \bmod N$ liefert $(-1)^2 \equiv 1 \bmod N$.
- $i = 2 + 4j$: $P_i \equiv -m \bmod N$ liefert $m^2 \equiv -1 \bmod N$.
- $i = 3 + 4j$: $P_i \equiv 1 \bmod N$ liefert $1^2 \equiv 1 \bmod N$.

Alle Relationen sind trivial, also kann man so sicher nichts für die Faktorisierung von $N = m^2 + 1$ gewinnen.

Wir wollen untersuchen, was man über die Lösungen der Kongruenz $x^2 \equiv y \bmod N$ sagen kann, wenn y klein ist. Wir bemerken zuvor, daß sich jede ganze Zahl $\widetilde{z} \neq 0$ eindeutig schreiben läßt als $\widetilde{z} = z^2 k$ mit k quadratfrei und $z \geq 1$.

LEMMA. *Gilt $x^2 \equiv y \bmod N$ mit $x \geq 0$, $x^2 \neq y$ und*

$$|y| < \frac{1}{2}\sqrt{N},$$

schreibt man $x^2 = y + z^2 kN$ mit $z \geq 1$ und k quadratfrei, so ist $k \geq 1$ und es gilt

$$|\sqrt{kN} - \frac{x}{z}| < \frac{1}{2z^2},$$

d.h. $\frac{x}{z}$ kommt als Näherungsbruch in der Kettenbruchentwicklung von $\sqrt{kN}$ vor. Schreibt man

$$\rho^i(\sqrt{kN}) = \frac{v_i + \sqrt{kN}}{a_i},$$

und sind $\frac{P_i}{Q_i}$ die Näherungsbrüche, so gibt es einen Index i und eine natürliche Zahl $l \geq 1$ mit

$$x = lP_i \quad \textit{und } y = l^2 \cdot (-1)^{i+1} a_{i+1}.$$

Beweis: Wäre $k < 0$, würde folgen

$$x^2 = y + z^2 kN = y - z^2 |k| N \leq y - N < \frac{1}{2}\sqrt{N} - N < 0,$$

ein Widerspruch. Da $k = 0$ wegen $x^2 \neq y$ ausgeschlossen ist, folgt $k \geq 1$. Nun gilt

$$y = x^2 - z^2 \cdot kN = (x + z\sqrt{kN})(x - z\sqrt{kN})$$

und damit

$$|\sqrt{kN} - \frac{x}{z}| = \frac{|x - z\sqrt{kN}|}{z} = \frac{|x - z\sqrt{kN}||x + z\sqrt{kN}|}{z|x + z\sqrt{kN}|} = \frac{|x^2 - z^2 \cdot kN|}{z(x + z\sqrt{kN})} = \frac{|y|}{z(x + z\sqrt{kN})} \leq$$
$$\leq \frac{|y|}{z^2\sqrt{kN}} \leq \frac{|y|}{z^2\sqrt{N}} < \frac{1}{2z^2},$$

insbesondere ist $\frac{x}{z}$ Näherungsbruch in der Kettenbruchentwicklung von $\sqrt{kN}$, d.h. es gibt einen Index i mit $\frac{x}{z} = \frac{P_i}{Q_i}$, also $x = lP_i$, $z = lQ_i$ und $l \geq 1$. Mit $P_i^2 - kNQ_i^2 = (-1)^{i+1}a_{i+1}$ folgt

$$y = x^2 - z^2kN = l^2P_i^2 - l^2Q_i^2kN = l^2(-1)^{i+1}a_{i+1},$$

was wir zeigen wollten. ■

Überlegungen: Sei N eine natürliche Zahl, die faktorisiert werden soll.

(1) Wähle $k \geq 1$ (klein, quadratfrei). Bilde die Kettenbruchentwicklung von $\sqrt{kN}$ mit $\rho^i(\sqrt{kN}) = \frac{v_i + \sqrt{kN}}{a_i}$ und Näherungsbrüchen $\frac{P_i}{Q_i}$. Dann gilt

$$P_i^2 - kNQ_i^2 = (-1)^{i+1}a_{i+1}.$$

Modulo N folgt

$$P_i^2 \equiv (-1)^{i+1}a_{i+1} \bmod N,$$

wie im Fall $k = 1$. Wir erhalten also wieder Relationen der gewünschten Form.

(2) Ist p eine Primzahl mit $p|a_{i+1}$, so folgt wegen $\mathrm{ggT}(P_i, Q_i) = 1$ auch $\mathrm{ggT}(p, Q_i) = 1$ und damit

$$(\frac{P_i}{Q_i})^2 \equiv kN \bmod p,$$

d.h. kN ist ein Quadrat modulo p. Für die Faktorbasis sind also nur solche Primzahlen interessant, für die $\left(\frac{kN}{p}\right) = 0$ oder $\left(\frac{kN}{p}\right) = 1$ gilt.

Diese Überlegungen liefern ein modifiziertes CFRAC-Faktorisierungsverfahren:

Das CFRAC-Faktorisierungsverfahren mit Multiplikator:

- Wähle eine (kleine) quadratfreie Zahl $k \geq 1$.
- Wähle eine Faktorbasis $p_0 = -1$, $p_1 = 2$, $p_2, \ldots, p_{n-1}$, so daß für $i \geq 2$ die Zahl p_i prim ist und für das Legendre-Symbol gilt $\left(\frac{kN}{p_i}\right) = 0$ oder $\left(\frac{kN}{p_i}\right) = 1$.
- Wir bilden die Kettenbruchentwicklung von $\sqrt{kN}$ mit den üblichen Formeln

$$d = kN, \quad P_{-2} = 0, \quad P_{-1} = 1, \quad v_0 = 0, \quad a_0 = 1$$

und für $i \geq 0$

$$u_i = \lfloor \frac{v_i + \lfloor\sqrt{d}\rfloor}{a_i} \rfloor, \quad P_i = (u_iP_{i-1} + P_{i-2}) \bmod N, \quad v_{i+1} = a_iu_i - v_i, \quad a_{i+1} = \frac{d - v_{i+1}^2}{a_i}$$

und testen jeweils, ob sich a_{i+1} mit $p_1, p_2, \ldots, p_{n-1}$ vollständig faktorisieren läßt. Wenn ja, haben wir eine Relation

$$P_i^2 \equiv (-1)^{i+1}a_{i+1} \bmod N$$

gefunden.

Man muß also in dem früheren Algorithmus nur die Auswahl der Faktorbasis von $\left(\frac{N}{p}\right) \neq -1$ zu $\left(\frac{kN}{p}\right) \neq -1$ abändern und dann bei der Kettenbruchentwicklung $d = N$ durch $d = kN$ ersetzen. Wir haben das Programm die gmp-Funktion cfrac zu cfrac_mul erweitert und damit die folgenden Beispiele gerechnet.

Beispiel: Wir faktorisieren jetzt $F_7 = 2^{2^7} + 1$:

$$N = 340282366920938463463374607431768211457 = 59649589127497217 \cdot 5704689200685129054721$$

Wir wählten jeweils eine Faktorbasis mit $n = 500$ Elementen und probierten verschiedene Multiplikatoren k:

k	p_{n-1}	Dezimalstellen von $\prod_{i=1}^{n-1} p_i$	(n_i, n_r, n_t)	Zeit
2	7853	1687	(1769372,465,2)	2:39
3	7451	1670	(1000383,461,1)	1:27
5	7121	1665	(941872,464,6)	1:35
6	8423	1704	(3776393,472,3)	4:53
7	7841	1682	(2227529,461,2)	2:59
10	7873	1685	(2400628,462,5)	3:10
11	7573	1685	(2485791,472,2)	3:36
13	8731	1712	(3227417,461,5)	4:56
14	8089	1691	(2019588,461,3)	3:01
15	7723	1685	(2687375,469,2)	3:31
17	7883	1693	(1158913,447,3)	2:21
19	8233	1697	(2215997,452,1)	2:57
21	7759	1684	(3114347,468,1)	4:33
22	7673	1688	(2749267,479,1)	3:43
23	7681	1686	(2445413,468,1)	3:43
26	8243	1700	(2026951,454,1)	3:05
29	7877	1697	(3492469,472,4)	5:35
30	7687	1667	(1838438,462,1)	2:27
257	8311	1696	(1619873,444,2)	3:19

(Die Zahl p_{n-1} und die Dezimalstellenzahl von $\prod_{i=1}^{n-1} p_i$ wurden angegeben, um eventuell eine Auswirkung auf die Rechenzeit zu sehen. Allerdings sieht man nichts.) Morrison und Brillhart wählten den Multiplikator $k = 257$.

6. The Early Abort Strategy

Betrachtet man die vorangegangenen Beispiele, so sieht man, daß im Zahlentripel (n_i, n_r, n_t) die Zahl n_i deutlich größer ist als n_r, d.h. in den meisten Fällen versucht man vergeblich in der Kongruenz $P_i^2 \equiv (-1)^{i+1} a_{i+1}$ die Zahl a_{i+1} mit den Elementen der Faktorbasis zu faktorisieren. Das kostet Zeit. Die Frage ist daher, ob man eventuell schneller feststellen kann, ob a_{i+1} mit $p_1, \ldots, p_{n-1}$ faktorisierbar ist oder nicht.

Die Situation ist also, daß wir eine Zahl a_{i+1} haben mit $1 \leq a_{i+1} < 2\sqrt{N}$ und testen wollen, ob sich a_{i+1} allein mit der Faktorbasis $p_1, \ldots, p_{n-1}$ faktorisieren läßt. Zerlegen wir

$$a_{i+1} = p_1^{b_1} \cdot p_2^{b_2} \cdot \cdots \cdot p_{n-1}^{b_{n-1}} \cdot \widetilde{a}_{i+1},$$

so ist also die Frage, ob $\widetilde{a}_{i+1} = 1$ gilt.
Die 'early abort strategy' beantwortet diese Frage nicht, hilft aber praktisch, schnell solche a_{i+1}'s wegzulassen, für die vielleicht $\widetilde{a}_{i+1} > 1$ ist. Wir teilen auf

$$a_{i+1} = (p_1^{b_1} \ldots p_{15}^{b_{15}}) \cdot (p_{16}^{b_{16}} \ldots p_{95}^{b_{95}}) \cdot (p_{96}^{b_{96}} \ldots p_{n-1}^{b_{n-1}}) \cdot \widetilde{a}_{i+1}.$$

Pomerance und Wagstaff haben experimentiert und dann folgende Empfehlungen ausgesprochen (für Zahlen N zwischen 10^{40} und 10^{54} mit $n = 960$):

- Man dividiert zunächst die Potenzen von $p_1, \ldots, p_{15}$ aus a_{i+1} heraus. Gilt für den Rest
$$(p_{16}^{b_{16}} \ldots p_{95}^{b_{95}}) \cdot (p_{96}^{b_{96}} \ldots p_{n-1}^{b_{n-1}}) \cdot \widetilde{a}_{i+1} > \frac{1}{500}\sqrt{N},$$
so testet man diese Zahl nicht weiter, sondern setzt $i := i + 1$ und fängt von vorne an.
- Erfüllt a_{i+1} die letzte Bedingung, so dividiert man die Potenzen von $p_{16}, \ldots, p_{95}$ heraus. Gilt für den Rest
$$(p_{96}^{b_{96}} \ldots p_{n-1}^{b_{n-1}}) \cdot \widetilde{a}_{i+1} > \frac{1}{2 \cdot 10^7}\sqrt{N},$$
so testet man nicht weiter, sondern setzt $i := i + 1$ und fängt von vorne an.

- Besteht a_{i+1} die vorangegangenen Tests, so teilt man alle Potenzen von $p_{96}, \ldots, p_{n-1}$ heraus, erhält also $\widetilde{a}_{i+1}$, und testet, ob $\widetilde{a}_{i+1} = 1$ gilt.

Wir haben die frühreren NTL-Funktionen cfrac_1, cfrac_2 entsprechend zu cfrac_1_eas und cfrac_2_eas abgewandelt und analog die gmp-Funktion cfrac zu cfrac_mul. Die folgenden Beispiele zeigen, daß die 'early abort strategy' die Faktorisierung mit CFRAC deutlich beschleunigt.

Beispiele: Es wurden jeweils Faktorbasen mit $n = 500$ Elementen gewählt und die Funktionen cfrac_1_eas (NTL), cfrac_2_eas (NTL) und cfrac_eas (gmp) benutzt.

N	(n_i, n_r, n_t)	1	2	gmp
135008632170705682525076560897389856446 7	(2506079,444,2)	1:50	2:03	1:06
151946659664809638934728799604827003254 7	(2647841,451,5)	1:57	2:11	1:11
170751358033906292332128117118524869234 3	(6728726,453,1)	4:04	4:51	2:15
395653212075672481751769959195369756701 1	(2397336,447,2)	1:49	1:56	1:07
232696377111501910259435480211833964634 7	(4840388,454,2)	3:04	3:30	1:44
279036619871460588079451385080717873810 9	(3950017,453,4)	2:41	3:09	1:33
197557488308202572733717658569699282424 1	(1460538,439,1)	1:12	1:24	0:46
222241035530810100728982616667678468174 9	(1506745,432,1)	1:18	1:20	0:50
333038211729234215833048253079306220793 9	(4176674,438,1)	2:52	3:12	1:40
459118747230191642012748783340173278638 7	(2890094,453,2)	2:05	2:16	1:15

In der folgenden Tabelle wurden Faktorbasen mit $n = 1000$ Elementen benutzt.

N	(n_i, n_r, n_t)	1	2
135008632170705682525076560897389856446 7	(1785305,798,2)	2:06	2:10
151946659664809638934728799604827003254 7	(1763330,808,1)	2:07	2:10
170751358033906292332128117118524869234 3	(4294255,828,3)	3:27	3:46
395653212075672481751769959195369756701 1	(1620892,790,2)	2:01	2:00
232696377111501910259435480211833964634 7	(3105807,835,2)	2:51	3:01
279036619871460588079451385080717873810 9	(2540342,809,4)	2:33	2:44
197557488308202572733717658569699282424 1	(1017352,779,2)	1:37	1:40
222241035530810100728982616667678468174 9	(1074757,798,3)	1:43	1:38
333038211729234215833048253079306220793 9	(2763002,815,2)	2:49	2:55
459118747230191642012748783340173278638 7	(1994981,823,6)	2:19	2:20

Bemerkung: Natürlich kann es sein, daß durch die Pomerance-Wagstaff-Bedingungen Relationen $P_i^2 \equiv (-1)^{i+1} a_{i+1} \bmod N$ ausgeschlossen werden, bei denen sich a_{i+1} doch mit der Faktorbasis faktorisieren läßt.

Beispiele: In der folgenden Tabelle bezeichnet n_{pwb} die Anzahl der n_r Relationen, die tatsächlich die Pomerance-Wagstaff-Bedingungen erfüllen. (Faktorbasen mit $n = 500$ Elementen)

N	(n_i, n_r, n_t)	n_{pwb}
135008632170705682525076560897389856446 7	(1776863,452,1)	313
151946659664809638934728799604827003254 7	(1886679,478,7)	313
170751358033906292332128117118524869234 3	(4170342,471,2)	276
395653212075672481751769959195369756701 1	(1605421,455,1)	300
232696377111501910259435480211833964634 7	(2753458,461,2)	253
279036619871460588079451385080717873810 9	(2570465,459,1)	297
197557488308202572733717658569699282424 1	(1100754,452,2)	335
222241035530810100728982616667678468174 9	(1240503,455,2)	353
333038211729234215833048253079306220793 9	(2969850,456,1)	306
459118747230191642012748783340173278638 7	(1898833,480,8)	296

7. Large Prime Variation

Um Relationen $x^2 \equiv y \bmod N$ zu erhalten, wo y sich multiplikativ aus Elementen der Faktorbasis $p_0, p_1, \ldots, p_{n-1}$ zusammensetzt, betrachtet man $P_i^2 \equiv (-1)^{i+1} a_{i+1} \bmod N$ und zerlegt a_{i+1}:

$$\begin{aligned} (-1)^{i+1} a_{i+1} &= a'_{i+1} \cdot \widetilde{a}_{i+1}, \\ a'_{i+1} &= p_0^{b_{i0}} \cdot p_1^{b_{i1}} \cdot \cdots \cdot p_{n-1}^{b_{i,n-1}}, \end{aligned}$$

wobei $\widetilde{a}_{i+1}$ keinen Primteiler $\leq p_{n-1}$ hat.
Gilt $\widetilde{a}_{i+1} = 1$, so hat man eine gewünschte Relation.

Wir betrachten jetzt den Fall

$$1 < \widetilde{a}_{i+1} < p_{n-1}^2.$$

Hier ist $\widetilde{a}_{i+1}$ eine Primzahl, da sonst ein nichttrivialer Teiler $\leq \sqrt{\widetilde{a}_{i+1}} < p_{n-1}$ existieren würde, was ausgeschlossen ist.
Findet man jetzt $i_1 \neq i_2$ mit

$$\widetilde{a}_{i_1+1} = \widetilde{a}_{i_2+1} \quad \text{und} \quad 1 < \widetilde{a}_{i_1+1} = \widetilde{a}_{i_2+1} < p_{n-1}^2,$$

so ergibt sich mit $\widetilde{p} = \widetilde{a}_{i_1+1} = \widetilde{a}_{i_2+1}$

$$\begin{aligned} P_{i_1}^2 &\equiv (-1)^{i_1+1} a_{i_1+1} = a'_{i_1+1} \widetilde{p} \bmod N, \\ P_{i_2}^2 &\equiv (-1)^{i_2+1} a_{i_2+1} = a'_{i_2+1} \widetilde{p} \bmod N, \\ (P_{i_1} P_{i_2})^2 &\equiv a'_{i_1+1} a'_{i_2+1} \cdot \widetilde{p}^2 \bmod N, \\ (\frac{P_{i_1} P_{i_2}}{\widetilde{p}})^2 &\equiv a'_{i_1+1} a_{i_2+1} \bmod N \end{aligned}$$

Da sich $a'_{i_1+1} a'_{i_2+1}$ aus Elementen der Faktorbasis multiplikativ zusammensetzt, hat man wieder eine gewünschte Relation gefunden.
Will man diese Idee praktisch auswerten, muß man die entsprechenden $\widetilde{a}_{i+1}$'s speichern zusammen mit P_i und $(-1)^{i+1} a_{i+1}$ und testen, ob Indizes $i_1 \neq i_2$ existieren mit $\widetilde{a}_{i_1+1} = \widetilde{a}_{i_2+1}$.

Beispiel: Wir betrachten die 30-stellige Zahl $N = 868719445232927625514336417981$ und wählen eine Faktorbasis mit 100 Elementen:

```
-1 2 3 5 7 19 31 37 71 79 89 97 107 109 113 131 137 139 151 167 173 179
181 199 211 223 229 233 241 251 257 263 269 283 293 317 347 349 353 359
367 373 383 421 439 463 487 503 521 523 541 547 563 569 587 593 601 613
617 619 661 673 677 701 727 733 739 757 797 809 839 853 859 863 877 883
887 929 937 953 967 977 991 997 1009 1013 1031 1033 1051 1069 1087 1091
1093 1151 1163 1171 1181 1193 1201 1213
```

Die Faktorisierung erhalten wir mit $(n_i, n_r, n_t) = (252631, 92, 3)$ als $N = 1313896666115183 \cdot 661177905109907$.
Wir sammeln dabei alle Kongruenzen $P_i^2 \equiv (-1)^{i+1} a_{i+1} \bmod N$, bei denen a_{i+1} nach Herausdividieren der Elemente der Faktorbasis > 1 und $< p_{n-1}^2$ ist. Es gibt 3729 Fälle. 846 davon treten mehrfach auf. Aus der Fülle der Fälle geben wir ein Beispiel: Für $i_1 = 140543$ und $i_2 = 55814$ gilt

$$\begin{aligned} P_{i_1} &= 449473233177902922284905276464, \\ P_{i_2} &= 799089603778258432488761878660, \\ a_{i_1+1} &= 140292935830148 = 2^2 \cdot 97 \cdot 521 \cdot 661 \cdot 1049941, \\ a_{i_2+1} &= 1267203884209060 = 2^2 \cdot 5 \cdot 7 \cdot 109 \cdot 139 \cdot 569 \cdot 1049941. \end{aligned}$$

Nach Herausdividieren der Basisprimzahlen bleibt also bei a_{i_1+1} und a_{i_2+1} jeweils die Primzahl $\widetilde{p} = 1049941$ übrig. Aus den Relationen

$$P_{i_1}^2 \equiv (-1)^{i_1+1} a_{i_1+1} \bmod N \quad \text{und} \quad P_{i_2}^2 \equiv (-1)^{i_2+1} a_{i_2+1} \bmod N$$

erhalten wir daher die Relation

$$(\frac{P_{i_1} P_{i_2}}{\widetilde{p}})^2 \equiv \frac{(-1)^{i_1+1} a_{i_1+1} (-1)^{i_2+1} a_{i_2+1}}{\widetilde{p}^2} \bmod N$$

mit

$$\begin{aligned}\frac{P_{i_1}P_{i_2}}{\widetilde{p}} &= 542558596238462054462521413 83,\\ \frac{(-1)^{i_1+1}a_{i_1+1}(-1)^{i_2+1}a_{i_2+1}}{\widetilde{p}^2} &= -161269599957470480 = -2^4\cdot 5\cdot 7\cdot 97\cdot 109\cdot 139\cdot 521\cdot 569\cdot 661.\end{aligned}$$

8. Glatte Zahlen

Wir wollen ein paar heuristische Überlegungen zum Funktionieren von CFRAC anstellen. Wird N mit der Faktorbasis $p_0, p_1, \ldots, p_{n-1}$ faktorisiert, so hatten wir dabei ein Zahlentripel (n_i, n_r, n_t) angegeben. Dabei zählt (mit den früheren Bezeichnungen)

- n_i, wie oft ein a_{i+1} berechnet wurde,
- n_r, wieviele der a_{i+1}'s sich mit der Faktorbasis faktorisieren lassen,
- n_t, wie oft versuchsweise ein ggT gebildet wurde um einen nichttrivialen Teiler von N zu finden.

Größenordnungsmäßig haben wir

$$n_r \approx n \quad \text{und} \quad n_t \approx 1.$$

Für die Laufzeit ist die Anzahl n_i wichtig, die angibt, wieviele Durchgänge man braucht um genügend Relationen zu erhalten.
Eine auftretende Zahl a_{i+1} ist dann interessant, wenn sich a_{i+1} mit den Primzahlen der Faktorbasis faktorisieren läßt, oder anders ausgedrückt, wenn alle Primteiler von a_{i+1} die Ungleichung $\le p_{n-1}$ erfüllen. Die Zahlen a_{i+1} liegen zwischen 1 und $2\sqrt{N}$.

DEFINITION. Eine natürliche Zahl n heißt z-glatt, wenn für jeden Primteiler p von n gilt $p \le z$, d.h. in der Primfaktorzerlegung von n treten nur Primzahlen $\le z$ auf. Die Anzahl der z-glatten natürlichen Zahlen $\le x$ wird mit $\psi(x,z)$ bezeichnet, also

$$\psi(x,z) = \#\{n \in \mathbf{N} : n \le x, n \text{ ist } z\text{-glatt}\}.$$

Beispiel: Was ist $\psi(100,10)$? Die Menge der 10-glatten Zahlen ≤ 100, d.h. der Zahlen, bei denen nur $2,3,5,7$ in der Primfaktorzerlegung vorkommt, ist

$$\{1,2,3,4,5,6,7,8,9,10,12,14,15,16,18,20,21,24,25,27,28,30,32,35,36,40,42,45,48,49,50,$$
$$54,56,60,63,64,70,72,75,80,81,84,90,96,98,100\},$$

und damit $\psi(100,10) = 46$.

Heuristische Überlegung (Riesel): Wir wollen die Anzahl $\psi(x,z)$ der z-glatten Zahlen $\le x$ abschätzen.

- Der sogenannte Primzahlsatz besagt, daß für die Primzahlzählfunktion $\pi(n) = \#\{p \le n\}$ gilt
$$\lim_{n\to\infty} \pi(n)\cdot\frac{\ln n}{n} = 1.$$
In diesem Umfeld hat man dann
$$\sum_{p\le z} \ln p \approx z \quad \text{und} \quad \pi(z) \approx \frac{z}{\ln z}.$$
- Für den Durchschnittswert von $\ln p$ über alle Primzahlen $p \le z$ gilt also
$$\overline{\ln p} = \frac{\sum_{p\le z}\ln p}{\pi(z)} \approx \frac{z}{z/\ln z} = \ln z.$$
- Wir wollen jetzt die Anzahl der z-glatten Zahlen $\le x$ abschätzen. Ist $\prod_i p_i$ eine z-glatte Zahl $\approx x$, wobei nicht alle p_i verschieden sein müssen, so gilt also $\prod_i p_i \approx x$, d.h. $\sum_i \ln p_i \approx \ln x$. Wegen $\overline{\ln p} \approx \ln z$ erwarten wir ungefähr $u = \frac{\ln x}{\ln z}$ Summanden.
- Eine z-glatte Zahl $\approx x$ wird also ungefähr $u = \frac{\ln x}{\ln z}$ Primfaktoren haben. Aus $\pi(z)$ möglichen Primzahlen muß man also u Stück auswählen, so daß man erhält
$$\psi(x,z) \approx \binom{\pi(z)}{u} = \frac{\pi(z)(\pi(z)-1)\ldots(\pi(z)-(u-1))}{u!} \approx \frac{\pi(z)^u}{u!}.$$

- Die Stirlingsche Formel $u! \approx u \ln u - u$ liefert

$$\begin{aligned} \ln \psi(x) &\approx u \ln \pi(z) - \ln u! \approx (u \ln \frac{z}{\ln z}) - (u \ln u - u) = \frac{\ln x}{\ln z}(\ln z - \ln \ln z) - u \ln u + u = \\ &= (\ln x - u \ln u) + u(-\ln \ln z + 1) \approx \ln x - u \ln u. \end{aligned}$$

Also

$$\psi(x, z) \approx x u^{-u}.$$

Wir werden diese Abschätzung weiter unten benutzen.

Es gibt verschiedene Sätze von Canfield/Erdös/Pomerance, die Aussagen über die Funktion $\psi(x, z)$ machen. Wir erwähnen einen derartigen Satz:

SATZ. *Es gibt eine (absolute) Konstante C, so daß für alle x, z mit $z < x^{1/3}$ gilt:*

$$\ln \psi(x, z) \geq \ln x - u \left(\ln u + (1 + \frac{1}{\ln u})(\ln \ln u - 1) + C(\frac{\ln \ln u}{\ln u})^2 \right).$$

Die Abschätzung der heuristischen Überlegung stimmt zumindest mit den ersten beiden Termen der Abschätzung des Satzes überein.

Anwendung: Wir wählen zufällig natürliche Zahlen a_i zwischen 1 und x. Die Wahrscheinlichkeit, daß a_i z-glatt ist, ist dann

$$\frac{\psi(x, z)}{x} \approx u^{-u} = \frac{1}{u^u} \quad \text{mit} \quad u = \frac{\ln x}{\ln z}.$$

Also kommt auf u^u Zahlen a_i eine z-glatte Zahl.

Beispiel: Wir wählen $x = 10^{30}$ und $z = 1000 = 10^3$. Dann ist $u = \frac{\ln x}{\ln z} = 10$, also kommt ungefähr auf 10^{10} Zahlen eine 1000-glatte Zahl.

Anwendung: Beim CFRAC-Faktorisierungsverfahren zur Faktorisierung der natürlichen Zahl N mit Faktorbasis $p_0, p_1, \ldots, p_{n-1}$ werden Zahlen a_{i+1} mit $1 \leq a_{i+1} < 2\sqrt{N}$ produziert. Interessant sind solche, die p_{n-1}-glatt sind. Wir nehmen an, daß die Zahlenfolge a_{i+1} einigermaßen zufällig verteilt ist. Dann kommen auf u^u Zahlen a_{i+1} eine, die p_{n-1}-glatt ist, wobei

$$u = \frac{\ln 2\sqrt{N}}{\ln p_{n-1}}$$

gilt. Da wir $n_r \approx n$ solcher Zahlen brauchen, erhalten wir

$$n_i \approx n_r \cdot u^u \approx n \cdot u^u.$$

Beispiel: Wir betrachten die 40-stellige Zahl

$$N = 1519466596648096389347287996048270032547$$

und eine Faktorbasis mit $n = 500$ Elementen. Dann ist $p_{499} = 7759$.
Wir wählen $x = 2\sqrt{N}$, $z = p_{499}$ und erhalten

$$u \approx 5.11, \quad u^u \approx 4211.42.$$

Alle 4211 stößt man also auf eine geeignete Zahl. Wir brauchen 500 Relationen, also brauchen wir $4211 \cdot 500 = 2105500$ Durchläufe. Tatsächlich haben wir 1 886679 Durchläufe gebraucht.

9. Übungen

Aufgabe 5.1: Sei N eine ungerade zusammengesetzte natürliche Zahl, x, y ganze Zahlen mit $x^2 \equiv y^2 \bmod N$. Beweise oder widerlege die folgenden Implikationen:

$$\begin{aligned} \mathrm{ggT}(x-y,N) = 1 &\implies \mathrm{ggT}(x+y) = N, \\ \mathrm{ggT}(x-y,N) = N &\implies \mathrm{ggT}(x+y) = 1. \end{aligned}$$

Aufgabe 5.2: Wie testet man schnell, ob eine Zahl ein Quadrat ist oder nicht?

(1) Sind m und x natürliche Zahlen, ist x ein Quadrat in $\mathbf{Z}$, so ist x auch ein Quadrat modulo m. Man beschreibe (für festes m) ein Verfahren 'isquadmodm', das testet, ob x ein Quadrat modulo m ist.
(2) Wie wahrscheinlich ist es, daß x ein Quadrat ist, wenn x den Test isquadmodm besteht?
(3) Man schreibe einen Test 'isquad', der testet, ob eine natürliche Zahl ein Quadrat ist oder nicht und eventuell die Quadratwurzel ausgibt. Man benutze als ersten Schritt die Funktion isquadmodm und versuche durch Probieren ein geeignetes m zu finden.

Aufgabe 5.3: Sind $\frac{P_i}{Q_i}$ die Näherungsbrüche der Kettenbruchentwicklung von $\sqrt{N}$ und $\rho^i(\sqrt{N}) = \frac{v_i+\sqrt{N}}{a_i}$, so gilt

$$P_i^2 - NQ_i^2 = (-1)^{i+1} a_{i+1}.$$

Ist $(-1)^{i+1}a_{i+1}$ ein Quadrat, d.h. $(-1)^{i+1}a_{i+1} = m^2$, so gilt $P_i^2 \equiv m^2 \bmod N$ und man kann untersuchen, ob $\mathrm{ggT}(P_i - m, N)$ oder $\mathrm{ggT}(P_i + m, N)$ ein nichttrivialer Faktor von N ist. Man schreibe ein Programm und vergleiche es mit CFRAC.

Aufgabe 5.4: Sei $N = 72089009$ und die Bezeichnungen aus Aufgabe 5.3: Bestimme die i's, für die $(-1)^{i+1}a_{i+1}$ ein Quadrat ist.

Aufgabe 5.5: Bestimme die Periodenlänge der Kettenbruchentwicklung von $\sqrt{8009 \cdot 9001 \cdot k}$ für $k = 1, 2, 3, \ldots$.

Aufgabe 5.6: Seien $\frac{P_i}{Q_i}$ die Näherungsbrüche der Kettenbruchentwicklung von $\sqrt{N}$. Untersuche das Periodizitätsverhalten der Folge $(P_i \bmod N, Q_i \bmod N)$.

Aufgabe 5.7: Skizziere die Funktion $z \mapsto \psi(10^4, z)$ und vergleiche mit der angegebenen heuristischen Abschätzung für $\psi(x, z)$.

KAPITEL 6

Endlich erzeugte Z-Moduln in quadratischen Zahlkörpern

1. Normalformen

Sei $K = \mathbf{Q}(\sqrt{d})$ ein quadratischer Zahlkörper, $d \in \mathbf{Z}$ kein Quadrat, aber d nicht notwendig quadratfrei. Ein endlich erzeugter $\mathbf{Z}$-Modul in K ist eine Teilmenge der Gestalt

$$M = \mathbf{Z}\alpha_1 + \mathbf{Z}\alpha_2 + \cdots + \mathbf{Z}\alpha_r = \{m_1\alpha_1 + m_2\alpha_2 + \cdots + m_r\alpha_r \in K : m_1, m_2, \ldots, m_r \in \mathbf{Z}\}$$

mit endlich vielen Zahlen $\alpha_1, \alpha_2, \ldots, \alpha_r \in K$, die ein Erzeugendensystem von M genannt werden. Offensichtlich ist M bzgl. Addition eine abelsche Gruppe, also ein $\mathbf{Z}$-Modul.

Ein Erzeugendensystem $\alpha_1, \ldots, \alpha_r$ eines endlich erzeugten $\mathbf{Z}$-Moduls M ist nicht eindeutig bestimmt, so kann man z.B. das Paar α_i, α_j durch $\alpha_i' = \alpha_i, \alpha_j' = \alpha_j + m\alpha_i$ für jedes beliebige $m \in \mathbf{Z}$ ersetzen, was man sofort aus den Gleichungen

$$\begin{pmatrix} \alpha_i' \\ \alpha_j' \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ m & 1 \end{pmatrix} \begin{pmatrix} \alpha_i \\ \alpha_j \end{pmatrix} \quad \text{und} \quad \begin{pmatrix} \alpha_i \\ \alpha_j \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ -m & 1 \end{pmatrix} \begin{pmatrix} \alpha_i' \\ \alpha_j' \end{pmatrix}$$

ersieht.
Natürlich kann man auch ein α_j aus dem Erzeugendensystem weglassen, wenn es einen Index $i \neq j$ und eine ganze Zahl m gibt mit $\alpha_j = m\alpha_i$.

Wir wollen jetzt eine Normalform für endlich erzeugte $\mathbf{Z}$-Moduln in $K = \mathbf{Q}(\sqrt{d})$ herleiten. Das folgende Lemma erledigt dies konstruktiv:

LEMMA. *Ein endlich erzeugter $\mathbf{Z}$-Modul $M \subseteq \mathbf{Q}(\sqrt{d})$ sei durch ein Erzeugendensystem $\alpha_1, \ldots, \alpha_r \in \mathbf{Q}(\sqrt{d})$ gegeben.*
Schreibt man

$$\alpha_i = \frac{v_i + w_i\sqrt{d}}{n} \text{ für } i = 1, \ldots, r \text{ mit } v_i, w_i \in \mathbf{Z}, n \in \mathbf{N},$$

berechnet man mit dem erweiterten euklidischen Algorithmus $x_1, \ldots, x_r, v, w \in \mathbf{Z}$ mit $w \geq 0$ und

$$w = \mathrm{ggT}(w_1, \ldots, w_r) = x_1w_1 + \cdots + x_rw_r, \quad v = x_1v_1 + \cdots + x_rv_r,$$

setzt man

$$u_i = \begin{cases} v_i - \frac{w_i}{w}v, & \text{falls } w > 0, \\ v_i, & \text{falls } w = 0 \end{cases}$$

berechnet man

$$u = \mathrm{ggT}(u_1, \ldots, u_r),$$

ersetzt man im Fall $u > 0$ die Zahl v durch $v \bmod u$, so gilt

$$M = \mathbf{Z} \cdot \frac{u}{n} + \mathbf{Z} \cdot (\frac{v}{n} + \frac{w}{n}\sqrt{d})$$

mit folgenden Eigenschaften:

(1) *Ist $u \neq 0$, $w \neq 0$, so ist*

$$M = \mathbf{Z} \cdot \frac{u}{n} + \mathbf{Z} \cdot (\frac{v}{n} + \frac{w}{n}\sqrt{d}) \quad \text{mit} \quad 0 \leq \frac{v}{n} < \frac{u}{n} \quad \text{und} \quad \frac{w}{n} > 0.$$

(2) *Ist $u \neq 0$, $w = 0$, so ist*

$$M = \mathbf{Z} \cdot \frac{u}{n} \quad \text{mit} \quad \frac{u}{n} > 0 \quad \text{und} \quad \frac{v}{n} = \frac{w}{n} = 0.$$

Version vom 11.2.2002

(3) *Ist* $u = 0$, $w \neq 0$, *so ist*

$$M = \mathbf{Z} \cdot (\frac{v}{n} + \frac{w}{n}\sqrt{d}) \quad \textit{mit} \quad \frac{w}{n} > 0 \quad \textit{und} \quad \frac{u}{n} = 0.$$

(4) *Ist* $u = 0$, $w = 0$, *so ist*

$$M = 0 \quad \textit{und} \quad \frac{u}{n} = \frac{v}{n} = \frac{w}{n} = 0.$$

Beweis: Wir berechnen

$$\beta = x_1\alpha_1 + \cdots + x_r\alpha_r = \frac{(\sum x_i v_i) + (\sum x_i w_i)\sqrt{d}}{n} = \frac{v + w\sqrt{d}}{n}.$$

Dann ist auch $\alpha_1, \ldots, \alpha_r, \beta$ ein Erzeugendensystem von M.
Ist $w = 0$, setzen wir $\beta_i = \alpha_i$, im Fall $w > 0$ setzen wir

$$\beta_i = \alpha_i - \frac{w_i}{w}\beta = \frac{v_i + w_i\sqrt{d}}{n} - \frac{w_i}{w}\frac{v + w\sqrt{d}}{n} = \frac{v_i - \frac{w_i}{w}v}{n} = \frac{u_i}{n}.$$

Nun ist auch $\beta_1, \ldots, \beta_r, \beta$ ein Erzeugendensystem von M. Mit dem erweiterten euklidischen Algorithmus findet man $y_1, \ldots, y_r \in \mathbf{Z}$ mit $u = y_1u_1 + \cdots + y_ru_r$. Wir setzen

$$\alpha = y_1\alpha_1 + \cdots + y_r\alpha_r = \frac{y_1u_1 + \cdots + y_ru_r}{n} = \frac{u}{n}.$$

Natürlich ist auch $\beta_1, \ldots, \beta_r, \beta, \alpha$ ein Erzeugendensystem von M. Nun sind aber die β_i's Vielfache von α, also ist auch α, β ein Erzeugendensystem von M, was gezeigt werden sollte.
Wir betrachten die gewünschten Eigenschaften.
Im Fall $u, w \neq 0$ ist nach Konstruktion $w \geq 1$ und $0 \leq v < u$.
Im Fall $u \neq 0$, $w = 0$ gilt für alle Indizes $w_i = 0$, also ist $u_i = v_i$, also gilt $v_i \equiv 0 \bmod u$ und damit auch $v \equiv 0 \bmod u$, was aufgrund der Normierung $v = 0$ liefert.
Der Rest ist klar. ■

Bemerkung: Betrachten wir mit den Bezeichnungen des Lemmas den von M aufgespannten $\mathbf{Q}$-Vektorraum QM in $\mathbf{Q}(\sqrt{d})$, so ist

$$\mathbf{Q}M = \mathbf{Q} \cdot \frac{u}{n} + \mathbf{Q} \cdot (\frac{v}{n} + \frac{w}{n}\sqrt{d})$$

und man erhält folgende Fallunterscheidung:

(1) Ist $u \neq 0$, $w \neq 0$, so ist $\dim_{\mathbf{Q}} \mathbf{Q}M = 2$.
(2) Ist $u \neq 0$, $w = 0$, so ist $\dim_{\mathbf{Q}} \mathbf{Q}M = 1$.
(3) Ist $u = 0$, $w \neq 0$, so ist $\dim_{\mathbf{Q}} \mathbf{Q}M = 1$.
(4) Ist $u = 0$, $w = 0$, so ist $\dim_{\mathbf{Q}} \mathbf{Q}M = 0$.

Man nennt (hier) die Dimension $\dim_{\mathbf{Q}} \mathbf{Q}M$ den Rang des Moduls M.

Beispiele: Wir betrachten Wir betrachten

$$\alpha_1 = \frac{1}{2} + \frac{1}{3}\sqrt{d}, \quad \alpha_2 = \frac{2}{5} - \frac{3}{2}\sqrt{d}, \quad \alpha_3 = -2 + \frac{7}{3}\sqrt{d}, \quad \alpha_4 = -\frac{1}{2} + \frac{1}{2}\sqrt{d}.$$

Mit dem obigen Verfahren findet man

$$\begin{aligned}
\mathbf{Z}\alpha_1 + \mathbf{Z}\alpha_2 + \mathbf{Z}\alpha_3 + \mathbf{Z}\alpha_4 &= \mathbf{Z} \cdot \frac{1}{10} + \mathbf{Z} \cdot \frac{1}{6}\sqrt{d}, \\
\mathbf{Z}\alpha_1 + \mathbf{Z}\alpha_2 + \mathbf{Z}\alpha_3 &= \mathbf{Z} \cdot \frac{1}{10} + \mathbf{Z} \cdot \frac{1}{6}\sqrt{d}, \\
\mathbf{Z}\alpha_1 + \mathbf{Z}\alpha_2 &= \mathbf{Z} \cdot \frac{53}{10} + \mathbf{Z} \cdot (\frac{29}{10} + \frac{1}{6}\sqrt{d}), \\
\mathbf{Z}\alpha_1 &= \mathbf{Z} \cdot (\frac{1}{2} + \frac{1}{3}\sqrt{d}).
\end{aligned}$$

Wir geben noch eine andere Formulierung des obigen Lemmas an:

SATZ. *Ist $M \subseteq \mathbf{Q}(\sqrt{d})$ ein endlich erzeugter $\mathbf{Z}$-Modul, so gibt es $q_1, q_2, q_3 \in \mathbf{Q}$ mit*

$$M = \mathbf{Z} \cdot q_1 + \mathbf{Z} \cdot (q_2 + q_3\sqrt{d})$$

und den Eigenschaften:

(1) *Ist $q_1 \neq 0$, $q_3 \neq 0$, so ist $M = \mathbf{Z} \cdot q_1 + \mathbf{Z} \cdot (q_2 + q_3\sqrt{d})$ mit $0 \leq q_2 < q_1$ und $q_3 > 0$.*
(2) *Ist $q_1 \neq 0$, $q_3 = 0$, so ist $M = \mathbf{Z} \cdot q_1$ mit $q_2 = 0$ und $q_1 > 0$.*
(3) *Ist $q_1 = 0$, $q_3 \neq 0$, so ist $M = \mathbf{Z} \cdot (q_2 + q_3\sqrt{d})$ mit $q_3 > 0$.*
(4) *Ist $q_1 = 0$, $q_3 = 0$, so ist $M = 0$ und $q_2 = 0$.*

Die Darstellung ist dann eindeutig und wird Hermitesche Normalform von M genannt.

Beweis: Die Existenzaussage wurde bereits bewiesen. Wir müssen noch die Eindeutigkeit von q_1, q_2, q_3 zeigen. Nun sieht man sofort

$$\begin{aligned} \{y \in \mathbf{Q} : x + y\sqrt{d} \in M\} &= \mathbf{Z} \cdot q_3, \\ \{x \in \mathbf{Q} : x \in M\} &= \mathbf{Z} \cdot q_1, \end{aligned}$$

sodaß q_1 und q_3 wegen der Normierungen $q_1 \geq 0$, $q_3 \geq 0$ eindeutig bestimmt sind. Wir müssen noch die Eindeutigkeit von q_2 zeigen und können dafür $q_3 > 0$ voraussetzen. Sei also

$$M = \mathbf{Z} \cdot q_1 + \mathbf{Z} \cdot (q_2 + q_3\sqrt{d}) = \mathbf{Z} \cdot q_1 + \mathbf{Z} \cdot (\widetilde{q_2} + q_3\sqrt{d}).$$

Es gibt $x, y \in \mathbf{Z}$ mit

$$\widetilde{q_2} + q_3\sqrt{d} = xq_1 + y(q_2 + q_3\sqrt{d}) = (xq_1 + yq_2) + yq_3\sqrt{d},$$

was zunächst $y = 1$ und damit $\widetilde{q_2} = xq_1 + q_2$ liefert. Ist $q_1 = 0$, so folgt die Behauptung. Im Fall $q_1 > 0$ haben wir die Normierung $0 \leq q_2 < q_1$, $0 \leq \widetilde{q_2} = xq_1 + q_2 < q_1$, was sofort $x = 0$ und damit $\widetilde{q_2} = q_2$ liefert. ■

Bemerkung: Die Maple-Funktion 'modul_hnf' bestimmt die im Satz angegebene Hermitesche Normalform.

Für spätere Anwendungen geben wir noch einen weiteren Satz an:

SATZ. *Sei $K = \mathbf{Q}(\sqrt{d})$ ein quadratischer Zahlkörper, seien $\alpha_1, \alpha_2 \in K$ linear unabhängig über $\mathbf{Q}$. Dann gilt für $\beta_1, \beta_2 \in K$:*

$$\mathbf{Z} \cdot \alpha_1 + \mathbf{Z} \cdot \alpha_2 = \mathbf{Z} \cdot \beta_1 + \mathbf{Z} \cdot \beta_2 \iff \text{es gibt } A, B, C, D \in \mathbf{Z} \text{ mit } AD - BC = 1 \text{ und } \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix}.$$

Vor dem Beweis stellen wir noch ein Lemma bereit:

LEMMA. *Sei*

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \quad \text{mit} \quad A, B, C, D \in \mathbf{Z} \quad \text{und} \quad \det(M) \neq 0.$$

Dann gilt:

$$M^{-1} \text{ hat Einträge aus } \mathbf{Z} \iff AD - BC = \pm 1.$$

Beweis: Haben M und M^{-1} Einträge in $\mathbf{Z}$, so gilt $\det(M) \in \mathbf{Z}$ und $\det(M^{-1}) \in \mathbf{Z}$. Aus $1 = \det(M) \cdot \det(M^{-1})$ folgt dann $\det(M) = AD - BC = \pm 1$.
Zur Umkehrung: Aus

$$M^{-1} = \frac{1}{AD - BC} \begin{pmatrix} D & -B \\ -C & A \end{pmatrix}$$

folgt sofort die Behauptung. ■

Beweis des Satzes: Wir können annehmen, daß auch β_1, β_2 linear unabhängig über $\mathbf{Q}$ sind. Dann gibt es eine 2×2-Matrix M mit Einträgen aus $\mathbf{Q}$ und

$$\begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} = M \cdot \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix}, \quad \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} = M^{-1} \cdot \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix}.$$

Nun gilt

$$\begin{aligned}
\mathbf{Z}\beta_1 + \mathbf{Z}\beta_2 \subseteq \mathbf{Z}\alpha_1 + \mathbf{Z}\alpha_2 \quad &\Longleftrightarrow \quad \beta_1, \beta_2 \in \mathbf{Z}\alpha_1 + \mathbf{Z}\alpha_2 \\
&\Longleftrightarrow \quad \text{es gibt } x,y,z,t \in \mathbf{Z} \text{ mit } \beta_1 = x\alpha_1 + y\alpha_2, \beta_2 = z\alpha_1 + t\alpha_2 \\
&\Longleftrightarrow \quad \text{es gibt } x,y,z,t \in \mathbf{Z} \text{ mit } \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} = \begin{pmatrix} x & y \\ z & t \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix} \\
&\Longleftrightarrow \quad M \text{ hat Einträge aus } \mathbf{Z}.
\end{aligned}$$

Analog ist $\mathbf{Z}\alpha_1 + \mathbf{Z}\alpha_2 \subseteq \mathbf{Z}\beta_1 + \mathbf{Z}\beta$ äquivalent damit, daß M^{-1} Einträge in $\mathbf{Z}$ hat. Mit dem letzten Lemma erhalten wir dann:

$$\begin{aligned}
\mathbf{Z}\beta_1 + \mathbf{Z}\beta_2 = \mathbf{Z}\alpha_1 + \mathbf{Z}\alpha_2 \quad &\Longleftrightarrow \quad \mathbf{Z}\beta_1 + \mathbf{Z}\beta_2 \subseteq \mathbf{Z}\alpha_1 + \mathbf{Z}\alpha_2 \text{ und } \mathbf{Z}\alpha_1 + \mathbf{Z}\alpha_2 \subseteq \mathbf{Z}\beta_1 + \mathbf{Z}\beta_2 \\
&\Longleftrightarrow \quad M \text{ und } M^{-1} \text{ haben Einträge aus } \mathbf{Z} \\
&\Longleftrightarrow \quad \det(M) = \pm 1,
\end{aligned}$$

was die Behauptung beweist. ■

Wir wollen noch eine weitere Normalform für endlich erzeugte Moduln vom Rang 2 herleiten: Ist $M = \mathbf{Z}q_1 + \mathbf{Z}(q_2 + q_3\sqrt{d})$, so können wir schreiben

$$M = \mathbf{Z}q_1 + \mathbf{Z}(q_2 + q_3\sqrt{d}) = q_1(\mathbf{Z} + \mathbf{Z}(\frac{q_2}{q_1} + \frac{q_3}{q_1}\sqrt{d})).$$

Nun wissen wir, daß sich die Zahl $\frac{q_2}{q_1} + \frac{q_3}{q_1}\sqrt{d}$ eindeutig als

$$\frac{q_2}{q_1} + \frac{q_3}{q_1}\sqrt{d} = \frac{b + \sqrt{D}}{2a} \text{ mit } D = b^2 - 4ac, \operatorname{ggT}(a,b,c) = 1$$

schreiben läßt. Damit wird:

$$M = q_1(\mathbf{Z} + \mathbf{Z}\frac{b + \sqrt{D}}{2a}) = \frac{q_1}{a}(\mathbf{Z} \cdot a + \mathbf{Z} \cdot \frac{b + \sqrt{D}}{2}).$$

Dies beweist bereits folgenden Satz:

SATZ. *Jeder endliche erzeugte* $\mathbf{Z}$*-Modul vom Rang* 2 *in* $K = \mathbf{Q}(\sqrt{d})$ *läßt sich schreiben als*

$$M = A(\mathbf{Z} \cdot a + \mathbf{Z} \cdot \frac{b + \sqrt{D}}{2}) \text{ mit } D = b^2 - 4ac, \operatorname{ggT}(a,b,c) = 1 \text{ und } A \in \mathbf{Q} \setminus \{0\}.$$

Bemerkung: Die Darstellung $M = A(\mathbf{Z}a + \mathbf{Z}\frac{b+\sqrt{D}}{2})$ mit $D = b^2 - 4ac$, $\operatorname{ggT}(a,b,c) = 1$, $A \in \mathbf{Q}$ ist durch diese Forderungen offensichtlich noch nicht eindeutig bestimmt:

- Man kann A durch $-A$ ersetzen.
- Man kann a durch $-a$ ersetzen, wobei dann c durch $-c$ ersetzt wird.
- Für $m \in \mathbf{Z}$ ist

  $$M = A(\mathbf{Z}a + \mathbf{Z}\frac{b + \sqrt{D}}{2}) = A(\mathbf{Z}a + \mathbf{Z}(am + \frac{b + \sqrt{D}}{2})) = A(\mathbf{Z}a + \mathbf{Z}\frac{(b + 2am) + \sqrt{D}}{2}),$$

  also kann man b durch $b + 2am$ ersetzen. Wegen

  $$D = b^2 - 4ac = (b + 2am)^2 - 4a(c + bm + am^2)$$

  geht dann c über in $c + bm + am^2$.

Eine Eindeutigkeitsaussage macht der folgende Satz:

SATZ. *Gilt*

$$M = A_1(\mathbf{Z}a_1 + \mathbf{Z}\frac{b_1 + \sqrt{D_1}}{2}) = A_2(\mathbf{Z}a_2 + \mathbf{Z}\frac{b_2 + \sqrt{D_2}}{2})$$

mit

$$D_1 = b_1^2 - 4a_1c_1, \quad \operatorname{ggT}(a_1,b_1,c_1) = 1, \quad D_2 = b_2^2 - 4a_2c_2, \quad \operatorname{ggT}(a_2,b_2,c_2) = 1,$$

so folgt

$$A_2 = \pm A_1, \quad a_2 = \pm a_1, \quad b_2 \equiv b_1 \bmod 2|a_1|, \quad D_2 = D_1.$$

Die Zahl $D = D_1 = D_2$ ist also durch M eindeutig bestimmt und wird als Diskriminante des Moduls bezeichnet.

Beweis: Sei $D_1 = m_1^2 d$, $D_2 = m_2^2 d$ mit $m_1, m_2 \geq 1$. O.E. können wir $A_i > 0$, $a_i > 0$ und $0 \leq b_i < 2a_i$ annehmen. Wir haben

$$M = A_i(\mathbf{Z}a_i + \mathbf{Z}\frac{b_i + m_i\sqrt{d}}{2}) = \mathbf{Z} \cdot A_i a_i + \mathbf{Z} \cdot (\frac{1}{2}A_i b_i + \frac{1}{2}A_i m_i\sqrt{d})$$

mit

$$A_i a_i > 0, \quad 0 \leq \frac{1}{2}A_i b_i < A_i a_i, \quad \frac{1}{2}A_i m_i > 0.$$

Die früher gezeigte Eindeutigkeitsaussage liefert also

$$A_1 a_1 = A_2 a_2, \quad A_1 b_1 = A_2 b_2, \quad A_1 m_1 = A_2 m_2.$$

Es folgt

$$\frac{A_1}{A_2} = \frac{a_2}{a_1} = \frac{b_2}{b_1} = \frac{m_2}{m_1},$$

schreiben wir gekürzt $\frac{A_1}{A_2} = \frac{n_2}{n_1}$ mit $n_1, n_2 \in \mathbf{N}$, $\mathrm{ggT}(n_1, n_2) = 1$, so gibt es also $a, b, m \in \mathbf{N}_0$ mit

$$a_i = a n_i, \quad b_i = b n_i, \quad m_i = m n_i.$$

Für das zugehörige c_i gilt

$$c_i = \frac{b_i^2 - D}{4a_i} = \frac{n_i^2 b^2 - n_i^2 m^2 d}{4 n_i a} = \frac{b^2 - m^2 d}{4a} n_i.$$

Wegen $c_1, c_2 \in \mathbf{Z}$ und $\mathrm{ggT}(n_1, n_2) = 1$ folgt $\frac{b^2 - m^2 d}{4a} \in \mathbf{Z}$, also setzen wir $c = \frac{b^2 - m^2 d}{4a}$ und haben dann $c_i = c n_i$. Es folgt $\mathrm{ggT}(a_i, b_i, c_i) = \mathrm{ggT}(a, b, c) \cdot n_i$, was wegen $\mathrm{ggT}(a_i, b_i, c_i) = 1$ sofort $n_1 = n_2 = 1$ liefert. Es folgt $a_1 = a_2$, $b_1 = b_2$, $c_1 = c_2$, $m_1 = m_2$, $D_1 = D_2$ und damit auch $A_1 = A_2$, was zu zeigen war. ■

Bemerkung: Die Maple-Funktion 'modul_nf' liefert zu

$$M = \mathbf{Z}\alpha_1 + \ldots \mathbf{Z}\alpha_r \subseteq \mathbf{Q}(\sqrt{d})$$

eine Normalformdarstellung

$$M = A(\mathbf{Z} \cdot a + \mathbf{Z} \cdot \frac{b + \sqrt{D}}{2}) \quad \text{mit} \quad D = b^2 - 4ac, \quad \mathrm{ggT}(a, b, c) = 1,$$

falls M Rang 2 hat.

Beispiele:

(1) Wir betrachten

$$\alpha_1 = \frac{1}{2} + \frac{1}{3}\sqrt{7}, \quad \alpha_2 = \frac{2}{5} - \frac{3}{2}\sqrt{7}, \quad \alpha_3 = -2 + \frac{7}{3}\sqrt{7}, \quad \alpha_4 = -\frac{1}{2} + \frac{1}{2}\sqrt{7}$$

und erhalten dann die Normalformen

$$\begin{aligned}
\mathbf{Z}\alpha_1 + \mathbf{Z}\alpha_2 + \mathbf{Z}\alpha_3 + \mathbf{Z}\alpha_4 &= \frac{1}{90}\left(\mathbf{Z} \cdot 9 + \mathbf{Z} \cdot \frac{0 + \sqrt{6300}}{2}\right) \\
\mathbf{Z}\alpha_1 + \mathbf{Z}\alpha_2 + \mathbf{Z}\alpha_3 &= \frac{1}{90}\left(\mathbf{Z} \cdot 9 + \mathbf{Z} \cdot \frac{0 + \sqrt{6300}}{2}\right) \\
\mathbf{Z}\alpha_1 + \mathbf{Z}\alpha_2 &= \frac{1}{4770}\left(\mathbf{Z} \cdot 25281 + \mathbf{Z} \cdot \frac{-22896 + \sqrt{17696700}}{2}\right)
\end{aligned}$$

mit den Diskriminanten

$$6300 = 2^2 \cdot 3^2 \cdot 5^2 \cdot 7 \quad \text{und} \quad 17696700 = 2^2 \cdot 3^2 \cdot 5^2 \cdot 7 \cdot 53^2.$$

(2) Für den Modul

$$M = \mathbf{Z}(2+11\sqrt{2}) + \mathbf{Z}(25+7\sqrt{2}) = \mathbf{Z}\cdot 261 + \mathbf{Z}(190+\sqrt{2})$$

ergibt sich die Normalform

$$M = \frac{1}{261}\left(\mathbf{Z}\cdot 68121 + \mathbf{Z}\cdot\frac{99180+\sqrt{544968}}{2}\right)$$

mit Diskriminante $544968 = 2^3\cdot 3^4\cdot 29^2$.

Die Menge der Moduln mit Diskriminante D bezeichnen wir mit $\mathrm{Mod}(D)$, also

$$\mathrm{Mod}(D) = \{A(\mathbf{Z}\cdot a + \mathbf{Z}\cdot\frac{b+\sqrt{D}}{2}) : A\in\mathbf{Q}^*, a,b,c\in\mathbf{Z}, \mathrm{ggT}(a,b,c)=1, D=b^2-4ac\}.$$

Ein Modul $M\in\mathrm{Mod}(D)$ wird primitiv genannt, wenn er sich als $M=\mathbf{Z}a+\mathbf{Z}\frac{b+\sqrt{D}}{2}$ mit $D=b^2-4ac$, $\mathrm{ggT}(a,b,c)=1$ schreiben läßt.

Wir wollen nun noch betrachten, welche Diskriminanten in $\mathbf{Q}(\sqrt{d})$ auftreten.

SATZ. *Sei $d\in\mathbf{Z}$ kein Quadrat und quadratfrei, $K=\mathbf{Q}(\sqrt{d})$. Wir definieren die Fundamentaldiskriminante des Körpers durch*

$$D_{\mathrm{fundamental}} = \begin{cases} d & \text{im Fall } d\equiv 1 \bmod 4,\\ 4d & \text{im Fall } d\equiv 2 \text{ oder } 3 \bmod 4.\end{cases}$$

Dann gilt:

(1) *Ist M ein endlich erzeugter $\mathbf{Z}$-Modul vom Rang 2 in $\mathbf{Q}(\sqrt{d})$, so gibt es $f\in\mathbf{N}$, so daß für die Diskriminante D von M gilt:*

$$D = f^2 D_{\mathrm{fundamental}}.$$

(2) *Für jedes $f\in\mathbf{N}$ gibt es einen Modul $M\subseteq\mathbf{Q}(\sqrt{d})$ mit Diskriminante $D=f^2D_{\mathrm{fundamental}}$, z.B.*

$$M = \mathbf{Z} + \mathbf{Z}\frac{(D \bmod 2)+\sqrt{D}}{2}.$$

Beweis: 1. Ist $M=A(\mathbf{Z}a+\mathbf{Z}\frac{b+\sqrt{D}}{2})$ mit $D=b^2-4ac$, $\mathrm{ggT}(a,b,c)=1$, so gibt es $m\in\mathbf{N}$ mit $D=m^2d$. Es folgt $m^2d=D\equiv b^2 \bmod 4$. Wegen $b^2\equiv 0$ oder $1 \bmod 4$ unterscheiden wir zwei Fälle:

- $d\equiv 1 \bmod 4$. Dann ist $D=m^2D_{\mathrm{fundamental}}$. Mit $f=m$ folgt die Behauptung.
- $d\equiv 2,3 \bmod 4$. Wäre $m\equiv 1 \bmod 2$, so würde $d\equiv b^2 \bmod 4$ folgen, was nicht sein kann. Also ist $m\equiv 0 \bmod 2$, d.h. $m=2f$, und $D=m^2d=f^2D_{\mathrm{fundamental}}$, wie behauptet.

2. Aus $D=f^2D_{\mathrm{fundamental}}$ folgt $D\equiv 0$ oder $1 \bmod 4$, also kann man mit $b_0\equiv D \bmod 2$ schreiben $D=b_0+4m=b_0^2-4\cdot 1\cdot(-m)$. Aus $\mathrm{ggT}(1,b_0,-m)=1$ folgt dann, daß $M=\mathbf{Z}+\mathbf{Z}\frac{b_0+\sqrt{D}}{2}$ ein Modul mit Diskriminante D ist. ■

Beispiele: Wir geben die (betragsmäßig) kleinsten Diskriminanten an:

$$5,\quad 8,\quad 12,\quad 13,\quad 17,\quad 20,\quad 21,\quad 24,\quad 28,\quad 29,\quad \ldots$$

bzw.

$$-3,\quad -4,\quad -7,\quad -8,\quad -11,\quad -12,\quad -15,\quad -16,\quad -19,\quad -20,\quad -23,\quad -24,\quad -27,\quad -28,\quad \ldots$$

Das folgende Lemma charakterisiert die Inklusion von Moduln in $\mathrm{Mod}(D)$:

LEMMA. *Für zwei Moduln*

$$M_1 = \mathbf{Z}a_1 + \mathbf{Z}\frac{b_1+\sqrt{D}}{2},\quad M_2 = \mathbf{Z}a_2 + \mathbf{Z}\frac{b_2+\sqrt{D}}{2}$$

aus $\mathrm{Mod}(D)$ *gilt:*

$$M_2\subseteq M_1 \quad\Longleftrightarrow\quad a_1|a_2,\quad b_2\equiv b_1 \bmod 2a_1.$$

Beweis:

$$\begin{aligned} M_2 \subseteq M_1 &\iff a_2 \in M_1 \text{ und } \frac{b_2+\sqrt{D}}{2} \in M \\ &\iff \text{es gibt } x,y,z \in \mathbf{Z} \text{ mit } a_2 = xa_1 \text{ und} \\ &\quad \frac{b_2+\sqrt{D}}{2} = ya_1 + z\frac{b_1+\sqrt{D}}{2} = \frac{(2ya_1+zb_1)+z\sqrt{D}}{2} \\ &\iff a_1|a_2 \text{ und es gibt } y \in \mathbf{Z} \text{ mit } b_2 = 2ya_1 + b_1 \\ &\iff a_1|a_2 \text{ und } b_2 \equiv b_1 \bmod 2a_1, \end{aligned}$$

was zu zeigen war. ■

LEMMA. *Für zwei Moduln $M_1, M_2 \in \mathrm{Mod}(D)$ mit*

$$M_1 = A_1(\mathbf{Z}a_1 + \mathbf{Z}\frac{b_1+\sqrt{D}}{2}) \quad \textit{und} \quad M_2 = A_2(\mathbf{Z}a_2 + \mathbf{Z}\frac{b_2+\sqrt{D}}{2})$$

gilt

$$M_2 \subseteq M_1 \iff z = \frac{A_2}{A_1} \in \mathbf{Z}, \quad a_2 z \equiv 0 \bmod a_1, \quad b_2 z \equiv b_1 z \bmod 2a_1.$$

Beweis:

$$\begin{aligned} M_2 \subseteq M_1 &\iff \text{es gibt } x,y,z \in \mathbf{Z} \text{ mit } A_2a_2 = A_1a_1x \text{ und } A_2\frac{b_2+\sqrt{D}}{2} = A_1a_1y + A_1\frac{b_1+\sqrt{D}}{2}z \\ &\iff A_2a_2 = A_1a_1x, \frac{A_2b_2+A_2\sqrt{D}}{2} = \frac{(2A_1a_1y+A_1b_1z)+A_1z\sqrt{D}}{2} \text{ mit } x,y,z \in \mathbf{Z} \\ &\iff A_2a_2 = A_1a_1x, A_2b_2 = 2A_1a_1y + A_1b_1z, A_2 = A_1z \text{ mit } x,y,z \in \mathbf{Z} \\ &\iff z = \frac{A_2}{A_1} \in \mathbf{Z}, A_1a_2z = A_1a_1x, A_1b_2z = 2A_1a_1y + A_1b_1z \text{ mit } x,y \in \mathbf{Z} \\ &\iff z = \frac{A_2}{A_1} \in \mathbf{Z}, a_2z = a_1x, b_2z = 2a_1y + b_1z \text{ mit } x,y \in \mathbf{Z} \\ &\iff z = \frac{A_2}{A_1} \in \mathbf{Z}, a_2z \equiv 0 \bmod a_1, b_2z \equiv b_1z \bmod 2a_1, \end{aligned}$$

was zu zeigen war. ■

2. Endomorphismenringe

Für einen endlich erzeugten $\mathbf{Z}$-Modul M vom Rang 2 in $K = \mathbf{Q}(\sqrt{d})$ definieren wir den Endomorphismenring durch

$$\mathrm{End}(M) = \{\lambda \in \mathbf{Q}(\sqrt{d}) : \lambda M \subseteq M\}.$$

Nach Konstruktion ist klar, daß $\mathrm{End}(M)$ ein Unterring von $\mathbf{Q}(\sqrt{d})$ ist, d.h. $1 \in \mathrm{End}(M)$ und $\lambda_1, \lambda_2 \in \mathrm{End}(M)$ impliziert $\lambda_1 + \lambda_2 \in \mathrm{End}(M)$, $\lambda_1\lambda_2 \in \mathrm{End}(M)$. Den Endomorphismenring eines Moduls kann man leicht ausrechnen:

SATZ. *Für einen Modul $M \in \mathrm{Mod}(D)$ mit*

$$M = A(\mathbf{Z}a + \mathbf{Z}\frac{b+\sqrt{D}}{2}), \quad D = b^2 - 4ac, \quad \mathrm{ggT}(a,b,c) = 1$$

gilt

$$\mathrm{End}(M) = \mathbf{Z} + \mathbf{Z}\frac{b+\sqrt{D}}{2} = \mathbf{Z} + \mathbf{Z}\frac{D+\sqrt{D}}{2} = \mathbf{Z} + \mathbf{Z}\frac{(D \bmod 2)+\sqrt{D}}{2}.$$

Beweis: Abkürzend schreiben wir

$$\alpha = \frac{b+\sqrt{D}}{2a}$$

und haben dann die Gleichung $a\alpha^2 - b\alpha + c = 0$ und damit auch

$$\alpha^2 = -\frac{c}{a} + \frac{b}{a}\alpha.$$

Für $\lambda \in \mathbf{Q}(\sqrt{d})$ gilt:

$$\begin{aligned}
\lambda \in \mathrm{End}(M) \quad &\Longleftrightarrow \quad \lambda M \subseteq M \quad \Longleftrightarrow \quad \lambda \cdot A(\mathbf{Z}a + \mathbf{Z}\frac{b+\sqrt{D}}{2}) \subseteq A(\mathbf{Z}a + \mathbf{Z}\frac{b+\sqrt{D}}{2}) \quad \Longleftrightarrow \\
&\Longleftrightarrow \quad \lambda(\mathbf{Z} + \mathbf{Z}\frac{b+\sqrt{D}}{2a}) \subseteq \mathbf{Z} + \mathbf{Z}\frac{b+\sqrt{D}}{2a} \quad \Longleftrightarrow \quad \lambda(\mathbf{Z} + \mathbf{Z}\alpha) \subseteq \mathbf{Z} + \mathbf{Z}\alpha \quad \Longleftrightarrow \\
&\Longleftrightarrow \quad \lambda \cdot 1 \in \mathbf{Z} + \mathbf{Z}\alpha \text{ und } \lambda \cdot \alpha \in \mathbf{Z} + \mathbf{Z}\alpha \\
&\Longleftrightarrow \quad \text{es gibt } x, y \in \mathbf{Z} \text{ mit } \lambda = x + y\alpha \text{ und} \\
&\qquad \lambda\alpha = x\alpha + y\alpha^2 = x\alpha + y(-\frac{c}{a} + \frac{b}{a}\alpha) = -\frac{cy}{a} + (x + \frac{by}{a})\alpha \in \mathbf{Z} + \mathbf{Z}\alpha \\
&\Longleftrightarrow \quad \text{es gibt } x, y \in \mathbf{Z} \text{ mit } \lambda = x + y\alpha \text{ und } -\frac{cy}{a} \in \mathbf{Z}, x + \frac{by}{a} \in \mathbf{Z}, \\
&\qquad \text{was wegen } \mathrm{ggT}(a,b,c) = 1 \text{ sofort die nächste Zeile liefert} \\
&\Longleftrightarrow \quad \text{es gibt } x, y \in \mathbf{Z} \text{ mit } \lambda = x + y\alpha \text{ und } a|y \\
&\Longleftrightarrow \quad \lambda \in \mathbf{Z} + \mathbf{Z}a\alpha = \mathbf{Z} + \mathbf{Z}\frac{b+\sqrt{D}}{2},
\end{aligned}$$

was wir zeigen wollten. ■

SATZ. *Ist D eine quadratische Diskriminante, so ist*

$$R_D = \mathbf{Z} + \mathbf{Z} \cdot \frac{D + \sqrt{D}}{2} = \mathbf{Z} + \mathbf{Z} \cdot \frac{(D \bmod 2) + \sqrt{D}}{2}$$

ein Ring und

$$\mathrm{End}(R_D) = R_D, \quad R_D \in \mathrm{Mod}(D).$$

Beweis: Ist $D = 4m$, so ist $D = 0^2 - 4 \cdot 1 \cdot (-m)$, also hat $R_D = \mathbf{Z} + \mathbf{Z}\frac{0+\sqrt{D}}{2}$ Diskriminante D. Ist $D = 4m + 1$, so ist $D = 1^2 - 4 \cdot 1 \cdot (-m)$, so hat $R_D = \mathbf{Z} + \mathbf{Z}\frac{1+\sqrt{D}}{2}$ Diskriminante D. Nach dem letzten Satz ist damit $R_D = \mathrm{End}(R_D)$, was die Behauptung zeigt. ■

Ein Unterring von $\mathbf{Q}(\sqrt{d})$, der gleichzeitig ein endlich erzeugter $\mathbf{Z}$-Modul vom Rang 2 ist, wird Ordnung genannt.

SATZ. *Sei $d \in \mathbf{Z}$ kein Quadrat und quadratfrei,*

$$D_1 = \begin{cases} d & \text{für } d \equiv 1 \bmod 4, \\ 4d & \text{für } d \equiv 2, 3 \bmod 4, \end{cases}, \quad \omega = \begin{cases} \frac{1+\sqrt{d}}{2} & \text{für } d \equiv 1 \bmod 4, \\ \sqrt{d} & \text{für } d \equiv 2, 3 \bmod 4. \end{cases}$$

Dann sind die Ordnungen in $\mathbf{Q}(\sqrt{d})$

$$R_{D_f} = \mathbf{Z} + \mathbf{Z} \cdot f\omega = \mathbf{Z} + \mathbf{Z} \cdot \frac{(D_f \bmod 2) + \sqrt{D_f}}{2} \quad \text{mit Diskriminante} \quad D_f = f^2 D_1,$$

wo f alle natürlichen Zahlen durchläuft. Es gilt:

$$R_{D_{f_2}} \subseteq R_{D_{f_1}} \quad \Longleftrightarrow \quad f_1 | f_2.$$

Insbesondere sind alle Ordnungen in der Ordnung

$$R_{D_1} = \mathbf{Z} + \mathbf{Z}\frac{(D_1 \bmod 2) + \sqrt{D_1}}{2} = \mathbf{Z} + \mathbf{Z}\omega$$

enthalten, die auch Maximalordnung genannt wird.

Beweis: Ist R eine Ordnung mit Diskriminante D, so folgt mit dem letzten Satz aus $R = \mathrm{End}(R)$ sofort $R = R_D$. Umgekehrt gibt es zu jeder Diskriminante D die Ordnung $R_D = \mathbf{Z} + \mathbf{Z}\frac{(D \bmod 2)+\sqrt{D}}{2}$. Indem man die Fälle $d \equiv 1 \bmod 4$ und $d \equiv 2, 3 \bmod 4$ unterscheidet, sieht man sofort, daß

$$\omega = \frac{(D_1 \bmod 2) + \sqrt{D_1}}{2}$$

gilt. Ist nun $D = f^2 D_1$ mit $f \geq 1$, so folgt wegen $(f^2 D_1 \bmod 2) - f(D_1 \bmod 2) \in 2\mathbf{Z}$ sofort

$$R_D = \mathbf{Z} + \mathbf{Z}\frac{(f^2 D_1 \bmod 2) + f\sqrt{D_1}}{2} = \mathbf{Z} + \mathbf{Z}f\frac{(D_1 \bmod 2) + \sqrt{D_1}}{2} = \mathbf{Z} + \mathbf{Z}f\omega.$$

Wir müssen nun nur noch die Inklusionsbeziehung zeigen.

$$\begin{aligned} \mathbf{Z} + \mathbf{Z}f_2\omega \subseteq \mathbf{Z} + \mathbf{Z}f_1\omega \quad &\iff \quad f_2\omega \in \mathbf{Z} + \mathbf{Z}f_1\omega \quad \iff \quad f_2\omega = x \cdot f_1\omega \text{ mit } x \in \mathbf{Z} \quad \iff \\ &\iff \quad f_1 | f_2, \end{aligned}$$

was zu zeigen war. ■

3. Konjugierte Moduln

In $\mathbf{Q}(\sqrt{d})$ hatten wir den nichttrivialen Automorphismus $x + y\sqrt{d} \mapsto x - y\sqrt{d}$ als $'$ bezeichnet, d.h. $(x + y\sqrt{d})' = x - y\sqrt{d}$. Ist M ein endlich erzeugter Modul in $\mathbf{Q}(\sqrt{d})$, so auch

$$M' = \{\alpha' : \alpha \in M\}.$$

Die Aussagen des folgenden Satzes sind sofort klar:

SATZ. *Für einen Modul $M \in \mathrm{Mod}(D)$ gilt:*

(1)
$$M = A(\mathbf{Z}a + \mathbf{Z}\frac{b + \sqrt{D}}{2}) \quad \Longrightarrow \quad M' = A(\mathbf{Z}a + \mathbf{Z}\frac{-b + \sqrt{D}}{2}),$$
insbesondere hat auch M' Diskriminante D.

(2) $M'' = M$.

Wir stellen noch ein paar einfache Aussagen zusammen.

LEMMA. *Sei $M = \mathbf{Z}a + \mathbf{Z}\frac{b+\sqrt{D}}{2} \in \mathrm{Mod}(D)$ und o.E. $-a < b \leq a$. Dann gilt:*

$$M = M' \quad \iff \quad M = \mathbf{Z}a + \mathbf{Z}\frac{0 + \sqrt{D}}{2} \quad \textit{oder} \quad M = \mathbf{Z}a + \mathbf{Z}\frac{a + \sqrt{D}}{2}.$$

Für die Diskriminante gilt dann $D = -4ac$ bzw. $D = a(a - 4c)$.

Beweis: $M = M'$ ist gleichwertig mit $-b \equiv b \bmod 2a$, also mit $a|b$. Wegen $-a < b \leq a$ bleiben nur die beiden Fälle $b = 0$ oder $b = a$. ■

LEMMA. *Sei $N = pq$ mit Primzahlen p, q und $N \equiv 3 \bmod 4$. Die primitiven Moduln M in $\mathrm{Mod}(-N)$ mit $M' = M$ sind genau die Moduln*

$$\mathbf{Z} + \mathbf{Z}\frac{1 + \sqrt{-N}}{2}, \quad \mathbf{Z}p + \mathbf{Z}\frac{p + \sqrt{-N}}{2}, \quad \mathbf{Z}q + \mathbf{Z}\frac{q + \sqrt{-N}}{2}, \quad \mathbf{Z}N + \mathbf{Z}\frac{N + \sqrt{-N}}{2}.$$

Beweis: Die Diskriminante ist $D = -N \equiv 1 \bmod 4$. Für primitive Moduln mit $M' = M$ können wir aufgrund des vorangegangenen Lemmas ansetzen

$$M = \mathbf{Z}a + \mathbf{Z}\frac{a + \sqrt{D}}{2} \text{ mit } -pq = -N = D = a^2 - 4ac = a(a - 4c) \text{ und o.E. } a > 0.$$

Für a gibt es also 4 Möglichkeiten: 1, p, q, N, die auch tatsächlich realisiert werden, wenn wir für das zugehörige c setzen $c = \frac{N+1}{4}$, $c = \frac{p+q}{4}$, $c = \frac{p+q}{4}$, $c = \frac{N+1}{4}$. ■

Bemerkung: Hat man also eine natürliche Zahl $N = pq \equiv 3 \bmod 4$, findet man 'zufällig' einen primitiven Modul $M = \mathbf{Z} \cdot a + \mathbf{Z}\frac{b+\sqrt{-N}}{2} \in \mathrm{Mod}(-N)$ mit $M' = M$, so kann man N faktorisieren, falls $1 < a < N$ gilt, denn a ist ein Teiler von N.

4. Die Norm von Moduln

Sei $M \in \mathrm{Mod}(D)$ gegeben und $R = R_D = \mathrm{End}(M)$. Schreibt man $R = \mathbf{Z}\alpha_1 + \mathbf{Z}\alpha_2$, $M = \mathbf{Z}\beta_1 + \mathbf{Z}\beta_2$, so gibt es rationale Zahlen q_{ij} mit

$$\beta_1 = q_{11}\alpha_1 + q_{12}\alpha_2, \quad \beta_2 = q_{21}\alpha_1 + q_{22}\alpha_2.$$

Dann heißt

$$\mathrm{N}(M) = \left| \det \begin{pmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{pmatrix} \right|$$

die Norm des Moduls M.

LEMMA. *Die Norm eines Moduls M ist wohldefiniert, d.h. unabhängig von der Wahl der Basen α_1, α_2 bzw. β_1, β_2.*

Beweis: Gilt $R = \mathbf{Z}\widetilde{\alpha_1} + \mathbf{Z}\widetilde{\alpha_2}$, $M = \mathbf{Z}\widetilde{\beta_1} + \mathbf{Z}\widetilde{\beta_2}$, so gibt es 2×2 Matrizen M, N mit ganzzahligen Einträgen und $\det(M) = \pm 1$, $\det(N) = \pm 1$ mit

$$\begin{pmatrix} \widetilde{\alpha_1} \\ \widetilde{\alpha_2} \end{pmatrix} = M \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix}, \quad \begin{pmatrix} \widetilde{\beta_1} \\ \widetilde{\beta_2} \end{pmatrix} = N \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix}.$$

Ist

$$\begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} = Q \begin{pmatrix} \alpha_1 \\ \alpha_2 \end{pmatrix},$$

so ist

$$\begin{pmatrix} \widetilde{\beta_1} \\ \widetilde{\beta_2} \end{pmatrix} = NQM^{-1} \begin{pmatrix} \widetilde{\alpha_1} \\ \widetilde{\alpha_2} \end{pmatrix}.$$

Nach Definition war $\mathrm{N}(M) = |\det(Q)|$, was aber mit $|\det(NQM^{-1})|$ wegen $|\det(M)| = |\det(N)| = 1$ identisch ist. ■

SATZ. *$D = b^2 - 4ac$ mit $\mathrm{ggT}(a,b,c) = 1$ und $A \in \mathbf{Q}$. Dann gilt*

$$\mathrm{N}(A(\mathbf{Z}a + \mathbf{Z}\frac{b+\sqrt{D}}{2})) = A^2|a|.$$

Beweis: Wir schreiben

$$\mathrm{End}(M) = \mathbf{Z} + \mathbf{Z}\frac{b+\sqrt{D}}{2}$$

und erhalten dann als Übergangsmatrix

$$\begin{pmatrix} Aa & 0 \\ 0 & A \end{pmatrix},$$

die die Determinante A^2a hat, was sofort die Behauptung ergibt. ■

Bemerkung: Die Norm $\mathrm{N}(M)$ eines Moduls $M \in \mathrm{Mod}(D)$ ist eine wichtige Invariante von M. Ein primitiver Modul M mit Diskriminante D und Norm a hat die Gestalt

$$M = \mathbf{Z} \cdot a + \mathbf{Z} \cdot \frac{b+\sqrt{D}}{2}, \quad D = b^2 - 4ac, \quad \mathrm{ggT}(a,b,c) = 1, \quad \text{o.E. } -a < b \leq a.$$

Bei Vorgabe von D und a kann man also im Prinzip alle primitiven Moduln mit Diskriminante D und Norm a aufstellen, indem man alle Zahlen

$$b \in \{-a+1, -a+2, \ldots, a\} \quad \text{mit} \quad D \equiv b^2 \bmod 4a, \quad \mathrm{ggT}(a, b, \frac{b^2-D}{4a}) = 1$$

bestimmt. Wir haben dazu eine Maple-Funktion 'modul_kon' geschrieben. Dies ist natürlich nur für kleine a's sinnvoll.

Beispiel: Wir wählen $D = 21$ und geben alle primitiven Moduln mit Norm zwischen 1 und 50 an:

$$\mathbf{Z}\cdot 1+\mathbf{Z}\cdot\frac{1+\sqrt{21}}{2},\quad \mathbf{Z}\cdot 3+\mathbf{Z}\cdot\frac{3+\sqrt{21}}{2},\quad \mathbf{Z}\cdot 5+\mathbf{Z}\cdot\frac{1+\sqrt{21}}{2},\quad \mathbf{Z}\cdot 5+\mathbf{Z}\cdot\frac{-1+\sqrt{21}}{2},$$
$$\mathbf{Z}\cdot 7+\mathbf{Z}\cdot\frac{7+\sqrt{21}}{2},\quad \mathbf{Z}\cdot 15+\mathbf{Z}\cdot\frac{9+\sqrt{21}}{2},\quad \mathbf{Z}\cdot 15+\mathbf{Z}\cdot\frac{-9+\sqrt{21}}{2},\quad \mathbf{Z}\cdot 17+\mathbf{Z}\cdot\frac{15+\sqrt{21}}{2}$$
$$\mathbf{Z}\cdot 17+\mathbf{Z}\cdot\frac{-15+\sqrt{21}}{2},\quad \mathbf{Z}\cdot 21+\mathbf{Z}\cdot\frac{21+\sqrt{21}}{2},\quad \mathbf{Z}\cdot 25+\mathbf{Z}\cdot\frac{11+\sqrt{21}}{2},\quad \mathbf{Z}\cdot 25+\mathbf{Z}\cdot\frac{-11+\sqrt{21}}{2}$$
$$\mathbf{Z}\cdot 35+\mathbf{Z}\cdot\frac{21+\sqrt{21}}{2},\quad \mathbf{Z}\cdot 35+\mathbf{Z}\cdot\frac{-21+\sqrt{21}}{2},\quad \mathbf{Z}\cdot 37+\mathbf{Z}\cdot\frac{13+\sqrt{21}}{2},\quad \mathbf{Z}\cdot 37+\mathbf{Z}\cdot\frac{-13+\sqrt{21}}{2}$$
$$\mathbf{Z}\cdot 41+\mathbf{Z}\cdot\frac{29+\sqrt{21}}{2},\quad \mathbf{Z}\cdot 41+\mathbf{Z}\cdot\frac{-29+\sqrt{21}}{2},\quad \mathbf{Z}\cdot 43+\mathbf{Z}\cdot\frac{35+\sqrt{21}}{2},\quad \mathbf{Z}\cdot 43+\mathbf{Z}\cdot\frac{-35+\sqrt{21}}{2},$$
$$\mathbf{Z}\cdot 47+\mathbf{Z}\cdot\frac{31+\sqrt{21}}{2},\quad \mathbf{Z}\cdot 47+\mathbf{Z}\cdot\frac{-31+\sqrt{21}}{2}.$$

5. Multiplikation von Moduln

Für irgendwelche Moduln M_1 und M_2 in $\mathbf{Q}(\sqrt{d})$ definieren wir das Produkt durch

$$M_1M_2 = \{\alpha_1\beta_1 + \cdots + \alpha_r\beta_r : \alpha_i \in M_1, \beta_i \in M_2, r \in \mathbf{N}\}.$$

Das Produkt besteht also aus allen möglichen Linearkombinationen von Produkten $\alpha\beta$ mit $\alpha \in M_1$, $\beta \in M_2$.

LEMMA. *Ist* $M_1 = \mathbf{Z}\alpha_1 + \cdots + \mathbf{Z}\alpha_r$ *und* $M_2 = \mathbf{Z}\beta_1 + \cdots + \mathbf{Z}\beta_s$, *so gilt*

$$M_1M_2 = \sum_{i=1}^{r}\sum_{j=1}^{s}\mathbf{Z}\alpha_i\beta_j,$$

also ist auch M_1M_2 *ein endlich erzeugter* $\mathbf{Z}$*-Modul.*

Beweis: Ist $\alpha \in M_1$, $\beta \in M_2$, so gibt es $x_i, y_j \in \mathbf{Z}$ mit $\alpha = \sum x_i\alpha_i$, $\beta = \sum y_j\beta_j$ und somit

$$\alpha\beta = \sum_{i=1}^{r}\sum_{j=1}^{s}x_iy_j\alpha_i\beta_j.$$

Jedes Produkt $\alpha\beta$ ist also Linearkombination von $\alpha_i\beta_j$, damit natürlich auch jede Summe von Produkten, was die Behauptung zeigt. ■

Beweis: Wir multiplizieren in $\mathbf{Q}(\sqrt{5})$ die Moduln

$$M_1 = \mathbf{Z}\cdot 11+\mathbf{Z}\cdot\frac{8+\sqrt{20}}{2}\quad\text{und}\quad M_2 = \mathbf{Z}\cdot 19+\mathbf{Z}\cdot\frac{11+\sqrt{45}}{2}.$$

$$\begin{aligned}
M_1M_2 &= (\mathbf{Z}\cdot 11+\mathbf{Z}\cdot\frac{8+2\sqrt{5}}{2})(\mathbf{Z}\cdot 19+\mathbf{Z}\cdot\frac{11+3\sqrt{5}}{2}) = \\
&= \mathbf{Z}\cdot 209+\mathbf{Z}\cdot\frac{121+33\sqrt{5}}{2}+\mathbf{Z}\cdot\frac{152+38\sqrt{5}}{2}+\mathbf{Z}\cdot\frac{59+23\sqrt{5}}{2} = \\
&= \mathbf{Z}\cdot 209+\mathbf{Z}\cdot\frac{121+33\sqrt{5}}{2}+\mathbf{Z}\cdot\frac{152+38\sqrt{5}}{2}+\mathbf{Z}\cdot\frac{59+23\sqrt{5}}{2}+ \\
&\quad +\mathbf{Z}(15\cdot\frac{121+33\sqrt{5}}{2}-13\cdot\frac{152+38\sqrt{5}}{2}) \\
&= \mathbf{Z}\cdot 209+\mathbf{Z}\cdot\frac{121+33\sqrt{5}}{2}+\mathbf{Z}\cdot\frac{152+38\sqrt{5}}{2}+\mathbf{Z}\cdot\frac{59+23\sqrt{5}}{2}+\mathbf{Z}\cdot\frac{-161+\sqrt{5}}{2} \\
&= \mathbf{Z}\cdot 209+\mathbf{Z}\cdot 2717+\mathbf{Z}\cdot 3135+\mathbf{Z}\cdot 1881+\mathbf{Z}\cdot\frac{-161+\sqrt{5}}{2} \\
&= \mathbf{Z}\cdot 209+\mathbf{Z}\cdot\frac{-161+\sqrt{5}}{2}
\end{aligned}$$

Bemerkung: Die Maple-Funktion 'modul_mult' multipliziert in Normalform vorgegebene Moduln eines quadratischen Zahlkörpers.

SATZ. (1) *Die Multiplikation von Moduln ist assoziativ und kommutativ.*
(2) $(M_1M_2)' = M_1'M_2'$.
(3) $\mathrm{End}(M) \cdot M = M$.

Beweis: Die ersten beiden Aussagen sind sofort klar. Wir zeigen die 3. Behauptung:
$\subseteq$: Für $\alpha \in \mathrm{End}(M)$, $\beta \in M$ gilt nach Definition $\alpha\beta \in M$. Da $\mathrm{End}(M)M$ von Produkten $\alpha\beta$ erzeugt wird, folgt $\mathrm{End}(M)M \subseteq M$.
$\supseteq$: Wegen $1 \in \mathrm{End}(M)$ gilt für $\alpha \in M$ trivialerweise $\alpha = 1 \cdot \alpha \in \mathrm{End}(M) \cdot M$, was die Behauptung beweist. ■

Der folgende Satz spielt eine wichtige Rolle:

SATZ. *Für $M \in \mathrm{Mod}(D)$ gilt*

$$MM' = \mathrm{N}(M)\mathrm{End}(M).$$

Beweis: Sei $M = A(\mathbf{Z}a + \mathbf{Z}\frac{b+\sqrt{D}}{2})$ und $D = b^2 - 4ac$, $\mathrm{ggT}(a,b,c) = 1$. Dann ist

$$\begin{aligned} MM' &= A(\mathbf{Z}a + \mathbf{Z}\frac{b+\sqrt{D}}{2}) \cdot A(\mathbf{Z}a + \mathbf{Z}\frac{b-\sqrt{D}}{2}) = \\ &= A^2(\mathbf{Z}a^2 + \mathbf{Z}a\frac{b+\sqrt{D}}{2} + \mathbf{Z}a\frac{b-\sqrt{D}}{2} + \mathbf{Z}\frac{b^2-D}{4}) = \\ &= A^2(\mathbf{Z}a^2 + \mathbf{Z}a\frac{b+\sqrt{D}}{2} + \mathbf{Z}ab + \mathbf{Z}ac) = A^2a(\mathbf{Z}a + \mathbf{Z}b + \mathbf{Z}c + \mathbf{Z}\frac{b+\sqrt{D}}{2}) = \\ &= A^2a(\mathbf{Z} + \mathbf{Z}\frac{b+\sqrt{D}}{2}) = \mathrm{N}(M)\mathrm{End}(M), \end{aligned}$$

was zu zeigen war. ■

SATZ. *Für $M_1, M_2 \in \mathrm{Mod}(D)$ gilt $M_1M_2 \in \mathrm{Mod}(D)$ und $\mathrm{N}(M_1M_2) = \mathrm{N}(M_1)\mathrm{N}(M_2)$.*

Beweis: Sei

$$\mathrm{End}(M_1) = \mathrm{End}(M_2) = \mathbf{Z} + \mathbf{Z}\frac{D+\sqrt{D}}{2} \quad \text{und} \quad \mathrm{End}(M_1M_2) = \mathbf{Z} + \mathbf{Z}\frac{\widetilde{D}+\sqrt{\widetilde{D}}}{2}.$$

Dann erhalten wir mit dem letzten Satz

$$\begin{aligned} \mathrm{N}(M_1M_2)(\mathbf{Z} + \mathbf{Z}\frac{\widetilde{D}+\sqrt{\widetilde{D}}}{2}) &= \mathrm{N}(M_1M_2)\mathrm{End}(M_1M_2) = (M_1M_2)(M_1M_2)' = M_1M_2M_1'M_2' = \\ &= (M_1M_1')(M_2M_2') = \mathrm{N}(M_1)\mathrm{End}(M_1) \cdot \mathrm{N}(M_2)\mathrm{End}(M_2) = \\ &= \mathrm{N}(M_1)\mathrm{N}(M_2)\mathrm{End}(M_1)\mathrm{End}(M_2) = \mathrm{N}(M_1)\mathrm{N}(M_2)\mathrm{End}(M_1) = \\ &= \mathrm{N}(M_1)\mathrm{N}(M_2)(\mathbf{Z} + \mathbf{Z}\frac{D+\sqrt{D}}{2}) \end{aligned}$$

Daraus folgt $\mathrm{End}(M_1M_2) = \mathrm{End}(M_1)$ und $\mathrm{N}(M_1M_2) = \mathrm{N}(M_1)\mathrm{N}(M_2)$, insbesondere muß auch M_1M_2 Diskriminante D haben, was dann die Behauptung beweist. ■

SATZ. $\mathrm{Mod}(D)$ *wird mit der Modulmultiplikation eine abelsche Gruppe mit $R_D = \mathbf{Z} + \mathbf{Z}\frac{D+\sqrt{D}}{2}$ als neutralem Element und mit $\frac{1}{\mathrm{N}(M)}M'$ als Inversem zu M.*

Beweis: Der letzte Satz zeigt, daß $\mathrm{Mod}(D)$ abgeschlossen gegen Multiplikation ist. Aus $M\mathrm{End}(M) = M$ folgt, daß $\mathrm{End}(M)$ das neutrale Element ist. Schließlich liefert $MM' = \mathrm{N}(M)\mathrm{End}(M)$ das inverse Element. ■

Wir wollen jetzt Moduln in $\mathrm{Mod}(D)$ explizit multiplizieren.

LEMMA. *Seien*

$$M_1 = A_1(\mathbf{Z}a_1 + \mathbf{Z}\frac{b_1+\sqrt{D}}{2}), \quad M_2 = A_2(\mathbf{Z}a_2 + \mathbf{Z}\frac{b_2+\sqrt{D}}{2}) \in \mathrm{Mod}(D)$$

gegeben. Bestimmt man A, x, y, z, a, b wie folgt

$$\begin{aligned} A &= \mathrm{ggT}(a_1, a_2, \frac{b_1+b_2}{2}) = a_1x + a_2y + \frac{b_1+b_2}{2}z, \\ a &= \frac{a_1a_2}{A^2}, \\ b &= \frac{1}{A}(a_1b_2x + a_2b_1y + \frac{b_1b_2+D}{2}z), \end{aligned}$$

so gilt:

$$M_1M_2 = A(\mathbf{Z}\cdot a + \mathbf{Z}\cdot\frac{b+\sqrt{D}}{2}).$$

Beweis: Natürlich können wir zunächst $A_1 = A_2 = 1$ voraussetzen, also

$$M_1 = \mathbf{Z}a_1 + \mathbf{Z}\frac{b_1+\sqrt{D}}{2} \quad \text{und} \quad M_2 = \mathbf{Z}a_2 + \mathbf{Z}\frac{b_2+\sqrt{D}}{2}.$$

Wir setzen an

$$M_1M_2 = A(\mathbf{Z}a + \mathbf{Z}\frac{b+\sqrt{D}}{2})$$

mit $D = b^2 - 4ac$, $\mathrm{ggT}(a,b,c) = 1$. Die Normgleichung liefert

$$|a_1a_2| = A^2|a|.$$

Ein Erzeugendensystem von M_1M_2 ist

$$a_1a_2, \quad \frac{a_1b_2 + a_1\sqrt{D}}{2}, \quad \frac{a_2b_1 + a_2\sqrt{D}}{2}, \quad \frac{\frac{b_1b_2+D}{2} + \frac{b_1+b_2}{2}\sqrt{D}}{2}.$$

Offensichtlich ist

$$A = \mathrm{ggT}(a_1, a_2, \frac{b_1+b_2}{2}).$$

Wir wählen $x, y, z \in \mathbf{Z}$ mit

$$A = \mathrm{ggT}(a_1, a_2, \frac{b_1+b_2}{2}) = a_1x + a_2y + \frac{b_1+b_2}{2}z.$$

Dann können wir wählen

$$\begin{aligned} A\frac{b+\sqrt{D}}{2} &= \frac{a_1b_2 + a_1\sqrt{D}}{2}x + \frac{a_2b_1 + a_2\sqrt{D}}{2}y + \frac{\frac{b_1b_2+D}{2} + \frac{b_1+b_2}{2}\sqrt{D}}{2}z = \\ &= \frac{(a_1b_2x + a_2b_1y + \frac{b_1b_2+D}{2}z) + (a_1x + a_2y + \frac{b_1+b_2}{2}z)\sqrt{D}}{2}, \end{aligned}$$

also

$$b = \frac{a_1b_2x + a_2b_1y + \frac{b_1b_2+D}{2}z}{A},$$

was zu zeigen war. ■

Bemerkung: Das Verfahren des obigen Lemmas zur Multiplikation von Moduln mit gleicher Diskriminante haben wir in der Maple-Funktion 'modul_multD' benutzt.

Beispiele:

(1) Wegen $5 = 5^2 - 4 \cdot 5 \cdot 1$ ist

$$M = \mathbf{Z} \cdot 5 + \mathbf{Z}\frac{5+\sqrt{5}}{2}$$

ein Modul in Mod(5). Wir berechnen M^2 naiv:

$$\begin{aligned} M^2 &= (\mathbf{Z} \cdot 5 + \mathbf{Z}\frac{5+\sqrt{5}}{2})(\mathbf{Z} \cdot 5 + \mathbf{Z}\frac{5+\sqrt{5}}{2}) = \mathbf{Z} \cdot 25 + \mathbf{Z}\frac{25+5\sqrt{5}}{2} + \mathbf{Z}\frac{30+10\sqrt{5}}{4} = \\ &= \mathbf{Z} \cdot 25 + \mathbf{Z}\frac{25+5\sqrt{5}}{2} + \mathbf{Z}\frac{15+5\sqrt{5}}{2} = \mathbf{Z} \cdot 25 + \mathbf{Z} \cdot 5 + \mathbf{Z}\frac{15+5\sqrt{5}}{2} = \\ &= \mathbf{Z} \cdot 5 + \mathbf{Z}\frac{15+5\sqrt{5}}{2} = \mathbf{Z} \cdot 5 + \mathbf{Z}\frac{5+5\sqrt{5}}{2} = 5(\mathbf{Z} + \mathbf{Z}\frac{1+\sqrt{5}}{2}). \end{aligned}$$

(2) Wir berechnen (naiv) Potenzen des Moduls $M = \mathbf{Z} \cdot 7 + \mathbf{Z} \cdot \frac{5+\sqrt{-3}}{2}$:

$$\begin{aligned} M^0 &= \mathbf{Z} + \mathbf{Z} \cdot \frac{1+\sqrt{-3}}{2}, \\ M^1 &= \mathbf{Z} \cdot 7 + \mathbf{Z} \cdot \frac{5+\sqrt{-3}}{2}, \\ M^2 &= \mathbf{Z} \cdot 49 + \mathbf{Z} \cdot \frac{-37+\sqrt{-3}}{2}, \\ M^3 &= \mathbf{Z} \cdot 343 + \mathbf{Z} \cdot \frac{-37+\sqrt{-3}}{2}, \\ M^4 &= \mathbf{Z} \cdot 2401 + \mathbf{Z} \cdot \frac{-2095+\sqrt{-3}}{2}, \\ M^5 &= \mathbf{Z} \cdot 16807 + \mathbf{Z} \cdot \frac{2707+\sqrt{-3}}{2}, \\ M^6 &= \mathbf{Z} \cdot 117649 + \mathbf{Z} \cdot \frac{69935+\sqrt{-3}}{2}, \\ M^7 &= \mathbf{Z} \cdot 823543 + \mathbf{Z} \cdot \frac{-165363+\sqrt{-3}}{2}, \\ M^8 &= \mathbf{Z} \cdot 5764801 + \mathbf{Z} \cdot \frac{4775895+\sqrt{-3}}{2}, \\ M^9 &= \mathbf{Z} \cdot 40353607 + \mathbf{Z} \cdot \frac{-29812911+\sqrt{-3}}{2}, \\ M^{10} &= \mathbf{Z} \cdot 282475249 + \mathbf{Z} \cdot \frac{-271934553+\sqrt{-3}}{2}. \end{aligned}$$

6. R_D-Moduln

Sei $R_D = \mathbf{Z} + \mathbf{Z}\frac{(D \bmod 2)+\sqrt{D}}{2}$ gegeben. Ein endlich erzeugter $\mathbf{Z}$-Modul $M \subseteq \mathbf{Q}(\sqrt{D})$ ist ein R_D-Modul, falls gilt $R_D M \subseteq M$.

LEMMA.

$$\{M \subseteq \mathbf{Q}(\sqrt{d}) : M \textit{ ist endlich erzeugter } R_D\textit{-Modul}\} = \cup_{\widetilde{D}|D}\mathrm{Mod}(\widetilde{D}).$$

Beweis: Sei M ein endlich erzeugter R_D-Modul. Ist $M \in \mathrm{Mod}(D_1)$, so ist $\mathrm{End}(M) = R_{D_1}$, aus $R_D \subseteq \mathrm{End}(M) = R_{D_1}$ folgt $D_1|D$, was die Inklusion zeigt. Die Umkehrung sieht man genauso. ■

Ein R_D-Ideal ist ein in R_D enthaltener R_D-Modul.
Wir beschäftigen uns mit $\mathrm{Mod}(D)$, da diese Menge eine Gruppenstruktur besitzt.

7. Zur Struktur von $\mathrm{Mod}(D)$

LEMMA. *Sei* $M = \mathbf{Z}a + \mathbf{Z}\frac{b+\sqrt{D}}{2} \in \mathrm{Mod}(D)$. *Dann gilt*

$$\{M_1 \in \mathrm{Mod}(D) : M \subseteq M_1, M_1 \textit{ primitiv}\} = \{\mathbf{Z}a_1 + \mathbf{Z}\frac{b+\sqrt{D}}{2} : a = a_1 a_2, \mathrm{ggT}(a_1, a_2, D) = 1\},$$

d.h. die M enthaltenden primitiven Moduln entsprechen genau den Teilern a_1 von $\mathrm{N}(M)$, für die die Gleichung $\mathrm{ggT}(a_1, \frac{\mathrm{N}(M)}{a_1}, D) = 1$ gilt.

Beweis: Wir schreiben $D = b^2 - 4ac$ mit $\mathrm{ggT}(a,b,c) = 1$. Wir setzen an $M_1 = \mathbf{Z}a_1 + \mathbf{Z}\frac{b_1+\sqrt{D}}{2} \in \mathrm{Mod}(D)$ mit $D = b_1^2 - 4a_1c_1$ und $\mathrm{ggT}(a_1, b_1, c_1) = 1$. Dann gilt:

$$M \subseteq M_1 \quad \Longleftrightarrow \quad a_1|a \text{ und } b \equiv b_1 \bmod 2a_1.$$

Im Fall $a_1|a$ gilt dann aber

$$M_1 = \mathbf{Z}a_1 + \mathbf{Z}\frac{b_1+\sqrt{D}}{2} = \mathbf{Z}a_1 + \mathbf{Z}\frac{b+\sqrt{D}}{2},$$

d.h. man kann o.E. $b_1 = b$ wählen. Wir schreiben $a = a_1a_2$ und haben dann wegen $D = b^2 - 4ac = b^2 - 4a_1c_1$ die Gleichung $c_1 = a_2c$. Jetzt muß nur noch die Bedingung $\mathrm{ggT}(a_1, b, c_1) = \mathrm{ggT}(a_1, b, a_2c) = 1$ erfüllt sein. Es gelten die Äquivalenzen unter Verwendung von $\mathrm{ggT}(a,b,c) = 1$:

$$\begin{aligned} p|\mathrm{ggT}(a_1,b,a_2c) &\iff p|a_1, p|b, p|a_2c \iff p|a_1, p|b, p|a_2 \iff p|a_1, p|a_2, p|b, p|D \\ &\iff p|a_1, p|a_2, p|D \iff p|\mathrm{ggT}(a_1,a_2,D), \end{aligned}$$

woraus die Behauptung folgt. ■

Beispiel: Der Modul $M = \mathbf{Z}\cdot 28 + \mathbf{Z}\cdot\frac{18+\sqrt{-12}}{2}$ ist wegen $-12 = 18^2 - 4\cdot 28\cdot 3$ nur in folgenden primitiven Moduln mit Diskriminante -12 echt enthalten:

$$\mathbf{Z}\cdot 4 + \mathbf{Z}\cdot\frac{2+\sqrt{-12}}{2}, \quad \mathbf{Z}\cdot 7 + \mathbf{Z}\cdot\frac{4+\sqrt{-12}}{2}.$$

LEMMA. *Ist $M = \mathbf{Z}a + \mathbf{Z}\frac{b+\sqrt{D}}{2} \in \mathrm{Mod}(D)$, ist $a = a_1a_2$ mit $\mathrm{ggT}(a_1,a_2,D) = 1$, so gilt*

$$M_1 = \mathbf{Z}a_1 + \mathbf{Z}\frac{b+\sqrt{D}}{2}, \quad M_2 = \mathbf{Z}a_2 + \mathbf{Z}\frac{b+\sqrt{D}}{2} \in \mathrm{Mod}(D)$$

und

$$M = M_1M_2.$$

Die Zerlegung ist eindeutig, wenn man $\mathrm{N}(M_1) = a_1$ und $\mathrm{N}(M_2) = a_2$ fordert.

Beweis: Gilt $M = \widetilde{M_1}\widetilde{M_2}$ mit $\widetilde{M_1}, \widetilde{M_2} \in \mathrm{Mod}(D)$ und $\mathrm{N}(\widetilde{M_1}) = a_1$, $\mathrm{N}(\widetilde{M_2}) = a_2$, so ist wegen $\widetilde{M_i} \subseteq \mathrm{End}(M) = \mathbf{Z} + \mathbf{Z}\frac{(D \bmod 2)+\sqrt{D}}{2}$

$$M = \widetilde{M_1}\widetilde{M_2} \subseteq \widetilde{M_i}\mathrm{End}(M) = \widetilde{M_i},$$

also folgt aus dem letzten Lemma $\widetilde{M_i} = M_i \in \mathrm{Mod}(D)$, d.h. die Zerlegung ist eindeutig. Es bleibt also nur noch zu zeigen, daß tatsächlich $M_1M_2 = M$ gilt. Mit $b^2 + D = 2b^2 - 4ac = 2b^2 - 4ac$ und $\mathrm{ggT}(a_1,a_2,b) = 1$ erhält man

$$\begin{aligned} M_1M_2 &= (\mathbf{Z}a_1 + \mathbf{Z}\frac{b+\sqrt{D}}{2})(\mathbf{Z}a_2 + \mathbf{Z}\frac{b+\sqrt{D}}{2}) = \\ &= \mathbf{Z}a_1a_2 + \mathbf{Z}a_1\frac{b+\sqrt{D}}{2} + \mathbf{Z}a_2\frac{b+\sqrt{D}}{2} + \mathbf{Z}\frac{\frac{b^2+D}{2} + b\sqrt{D}}{2} = \\ &= \mathbf{Z}a + \mathbf{Z}a_1\frac{b+\sqrt{D}}{2} + \mathbf{Z}a_2\frac{b+\sqrt{D}}{2} + \mathbf{Z}\frac{b^2 - 2ac + b\sqrt{D}}{2} = \\ &= \mathbf{Z}a + \mathbf{Z}a_1\frac{b+\sqrt{D}}{2} + \mathbf{Z}a_2\frac{b+\sqrt{D}}{2} + \mathbf{Z}\frac{b^2 + b\sqrt{D}}{2} = \\ &= \mathbf{Z}a + \mathbf{Z}a_1\frac{b+\sqrt{D}}{2} + \mathbf{Z}a_2\frac{b+\sqrt{D}}{2} + \mathbf{Z}b\frac{b+\sqrt{D}}{2} = \\ &= \mathbf{Z}\cdot a + \mathbf{Z}\cdot\mathrm{ggT}(a_1,a_2,b)\cdot\frac{b+\sqrt{D}}{2} = \mathbf{Z}\cdot a + \mathbf{Z}\cdot\frac{b+\sqrt{D}}{2} = M, \end{aligned}$$

was zu zeigen war. ■

Das Lemma liefert jetzt sofort folgenden Zerlegungssatz:

SATZ. *Sei*

$$M = \mathbf{Z}a + \mathbf{Z}\frac{b+\sqrt{D}}{2} \in \mathrm{Mod}(D).$$

Ist dann

$$|a| = \prod_i p_i^{e_i}$$

die Primfaktorzerlegung von $|a| = \mathrm{N}(M)$*, so hat man die (eindeutige) Zerlegung*

$$M = \prod_{p_i | D} (\mathbf{Z}p_i^{e_i} + \mathbf{Z}\frac{b+\sqrt{D}}{2}) \cdot \prod_{p_i \nmid D} (\mathbf{Z}p_i + \mathbf{Z}\frac{b+\sqrt{D}}{2})^{e_i}.$$

Beispiel: Der Modul $M = \mathbf{Z} \cdot 28 + \mathbf{Z} \cdot \frac{18+\sqrt{-12}}{2}$ hat die Zerlegung

$$\mathbf{Z} \cdot 28 + \mathbf{Z} \cdot \frac{18+\sqrt{-12}}{2} = (\mathbf{Z} \cdot 4 + \mathbf{Z} \cdot \frac{2+\sqrt{-12}}{2}) \cdot (\mathbf{Z} \cdot 7 + \mathbf{Z} \cdot \frac{4+\sqrt{-12}}{2}).$$

Um die primitiven Moduln zu untersuchen, kann man sich also auf solche mit p-Potenz-Norm zurückziehen. Dabei deutet der Satz darauf hin, daß man zwischen $p \mid D$ und $p \nmid D$ unterscheiden sollte. Wir werden die Theorie nicht wesentlich weiter entwickeln, sondern uns beispielhaft nur um die Existenz von primitven Moduln mit Norm p kümmern.

8. Primitive Moduln mit Norm p

Wir wollen uns im folgenden darauf beschränken, zu untersuchen, wann es primitive Moduln mit Norm p gibt. Wir fragen also nach Moduln der Gestalt

$$M = \mathbf{Z} \cdot p + \mathbf{Z} \cdot \frac{b+\sqrt{D}}{2}, \quad D = b^2 - 4pc, \quad \mathrm{ggT}(p,b,c) = 1, \quad \text{o.E. } -p < b \leq p.$$

Es folgt sofort $D \equiv b^2 \bmod p$, d.h. D ist ein Quadrat modulo p. Für das weitere Vorgehen stellen wir ein Lemma bereit:

LEMMA. *Ist* p *ungerade mit* $\left(\frac{D}{p}\right) = 1$*, so gibt es genau ein* $x \in \mathbf{Z}$ *mit*

$$D \equiv x^2 \bmod p, \quad D \equiv x \bmod 2 \quad \textit{und} \quad 0 < x < p.$$

Dieses x *wird im folgenden mit* b_p *bezeichnet.*

Beweis: Existenz: Wegen $\left(\frac{D}{p}\right) = 1$ gibt es ein $y \in \mathbf{Z}$ mit $D \equiv y^2 \bmod p$. O.E. können wir $0 < y < p$ wählen. Ist $D \equiv y \bmod 2$, setzen wir $x = y$, ist $D \not\equiv y \bmod 2$, so hat $x = p - y$ die gewünschten Eigenschaften.
Eindeutigkeit: Angenommen, wir haben zwei $x_1, x_2 \in \mathbf{Z}$, die den Bedingungen genügen. Es folgt $x_1^2 - x_2^2 \equiv 0 \bmod p$, also $x_2 \equiv \pm x_1 \bmod p$. Es ist also $x_2 = x_1 + kp$ oder $x_2 = -x_1 + kp$ mit einem $k \in \mathbf{Z}$. Wegen $0 < x_i < p$ folgt $x_2 = x_1$ oder $x_2 = p - x_1$. Wäre $x_2 = p - x_1$, so hätte man $x_2 \equiv 1 + x_1 \bmod 2$, also kann nur eines der x_i's die Gleichugn $x_i \equiv D \bmod 2$ erfüllen. Somit bleibt nur $x_2 = x_1$. ■

Bemerkung: Es gibt einen Algorithmus, mit dem man b_p schnell berechnen kann. In Maple gibt es die Funktion 'numtheory[msqrt](a,p)', mit der man $x^2 \equiv a \bmod p$ schnell lösen kann, falls eine Lösung existiert.

SATZ. *Sei* p *eine ungerade Primzahl mit* $p \nmid D$*. Ist* $\left(\frac{D}{p}\right) = 1$*, so gibt es genau zwei primitive Moduln in* $\mathrm{Mod}(D)$ *mit Norm* p*, nämlich*

$$M_p = \mathbf{Z}p + \mathbf{Z}\frac{b_p + \sqrt{D}}{2} \quad \textit{und} \quad M_p' = \mathbf{Z}p + \mathbf{Z}\frac{-b_p + \sqrt{D}}{2}.$$

Ist $\left(\frac{D}{p}\right) = -1$*, so gibt es keinen primitiven Modul mit Norm* p *und Diskriminante* D*.*

Beweis: Existiert ein primitiver Modul mit Norm p, so haben wir bereits gesehen, daß D ein Quadrat modulo p ist, d.h. im Fall $\left(\frac{D}{p}\right) = -1$ gibt es keinen primitiven Modul mit Norm p. Sei nun $\left(\frac{D}{p}\right) = 1$. Mit dem im Lemma definierten b_p haben wir $D \equiv b_p^2 \bmod p$ und $D \equiv b_p^2 \bmod 4$, also $D \equiv b_p^2 \bmod 4p$, d.h. $D = b_p^2 - 4pc$ mit einem $c \in \mathbf{Z}$, wobei natürlich $\mathrm{ggT}(p, b_p, c) = 1$ gilt wegen $\mathrm{ggT}(b_p, p) = 1$. Also ist

$$M_p = \mathbf{Z}p + \mathbf{Z}\frac{b_p + \sqrt{D}}{2}$$

ein primitiver Modul mit Norm p und Diskriminante D. Das gleiche gilt dann auch für M_p'.
Wäre $M_p = M_p'$, so hätte man $b_p \equiv -b_p \bmod 2p$, also $2p | 2b_p$, was $0 < b_p < p$ widerspricht. Damit folgt $M_p \neq M_p'$.
Wir nehmen nun an, wir haben einen primitiven Modul mit Norm p:

$$M = \mathbf{Z}p + \mathbf{Z}\frac{b + \sqrt{D}}{2}, \quad D = b^2 - 4pc, \quad \mathrm{ggT}(p, b, c) = 1.$$

O.E. können wir $-p < b < p$ annehmen. Ist nun $b \geq 0$, so folgt wegen der Eindeutigkeit von b_p sofort $b = b_p$ und damit $M = M_p$. Ist $b < 0$, so erfüllt $-b$ die Aussagen des letzten Lemmas, also folgt $M = M_p'$. ■

Bemerkung: Mit Hilfe des quadratischen Reziprozitätsgesetzes kann man folgenden Satz beweisen:

SATZ. *Sind p und q ungerade Primzahlen mit $\mathrm{ggT}(p, D) = \mathrm{ggT}(q, D) = 1$, so gilt die Implikation*

$$p \equiv q \bmod D \quad \Longrightarrow \quad \left(\frac{D}{p}\right) = \left(\frac{D}{q}\right).$$

Interpretation: Die Frage, ob es primitive Moduln mit Norm p gibt, hängt also nur von der Restklasse $p \bmod |D|$ ab.

Wir betrachten nun die Primzahl $p = 2$ im Fall $D \not\equiv 0 \bmod 2$, also für $D \equiv 1 \bmod 4$.

SATZ. *Ist $D \equiv 1 \bmod 8$, so gibt es genau zwei primitivie Moduln in $\mathrm{Mod}(D)$ mit Norm 2, nämlich*

$$M_2 = \mathbf{Z} \cdot 2 + \mathbf{Z} \cdot \frac{1 + \sqrt{D}}{2} \quad \textit{und} \quad M_2' = \mathbf{Z} \cdot 2 + \mathbf{Z} \cdot \frac{-1 + \sqrt{D}}{2}.$$

Ist $D \equiv 5 \bmod 8$, so gibt es keinen primitiven Modul in $\mathrm{Mod}(D)$ mit Norm 2.

Beweis: Wir machen den Ansatz

$$M = \mathbf{Z} \cdot 2 + \mathbf{Z} \cdot \frac{b + \sqrt{D}}{2}, \quad D = b^2 - 8c, \quad \mathrm{ggT}(2, b, c) = 1, \quad \text{o.E.} -2 < b \leq 2.$$

Wegen $b \equiv D \bmod 2$ bleibt nur $b = \pm 1$ und damit

$$D = b^2 - 8c = 1 - 8c \equiv 1 \bmod 8.$$

Ist also $D \equiv 5 \bmod 8$, so kann es keinen primitiven Modul mit Norm 2 geben. Ist $D \equiv 1 \bmod 8$, so schreiben wir $D = 1 - 8c = 1^2 - 4 \cdot 2 \cdot c$ mit $c \in \mathbf{Z}$, also ist klar, daß die im Satz angegebenen Moduln M_2, M_2' in $\mathrm{Mod}(D)$ sind. Wegen $-1 \not\equiv 1 \bmod 2 \cdot 2$ sind sie auch verschieden. Ist $M = \mathbf{Z}2 + \mathbf{Z}\frac{b+\sqrt{D}}{2}$ irgendein primitiver Modul mit Norm 2, so kann man $-2 < b \leq 2$ annehmen, also folgt $b = 1$ oder $b = -1$ und damit $M = M_2$ oder $M = M_2'$. ■

Wir kommen nun zum Fall $p | D$.

SATZ. *Sei p ein Primteiler der Diskriminante D. Genau dann gibt es einen primitiven Modul in* $\mathrm{Mod}(D)$ *mit Norm p, wenn gilt*

$$D \equiv 8 \text{ oder } 12 \bmod 16 \text{ im Fall } p = 2 \text{ bzw. } D \not\equiv 0 \bmod p^2 \text{ im Fall } p \neq 2.$$

Dieser ist dann eindeutig bestimmt, nämlich

$$M_p = \begin{cases} \mathbf{Z} \cdot 2 + \mathbf{Z} \cdot \frac{0+\sqrt{D}}{2} & \text{für } D \equiv 8 \bmod 16, \\ \mathbf{Z} \cdot 2 + \mathbf{Z} \cdot \frac{2+\sqrt{D}}{2} & \text{für } D \equiv 12 \bmod 16, \\ \mathbf{Z} \cdot p + \mathbf{Z} \cdot \frac{0+\sqrt{D}}{2} & \text{für } p \equiv 1 \bmod 2, D \equiv 0 \bmod 4, \\ \mathbf{Z} \cdot p + \mathbf{Z} \cdot \frac{p+\sqrt{D}}{2} & \text{für } p \equiv 1 \bmod 2, D \equiv 1 \bmod 4. \end{cases}$$

Beweis:

(1) Wir untersuchen, wann es für $D \equiv 0 \bmod 4$ einen primitiven Modul mit Norm 2 gibt. Wir setzen an

$$M = \mathbf{Z} \cdot 2 + \frac{b+\sqrt{D}}{2}, \quad D = b^2 - 8c, \quad \mathrm{ggT}(2,b,c) = 1, \quad -2 < b \leq 2.$$

Wegen $2|D$ folgt $2|b$, also bleiben nur die Fälle $b = 0$ und $b = 2$.
Fall $b = 0$: Dann ist $D = -8c$ und es muß gelten $\mathrm{ggT}(2,c) = 1$. Dies ist genau der Fall $D \equiv 8 \bmod 16$. Dann ist

$$M = \mathbf{Z} \cdot 2 + \mathbf{Z}\frac{\sqrt{D}}{2}$$

ein Modul mit Norm 2.
Fall $b = 2$: Dann ist $D = 4 - 8c$, c ungerade, also $c = 2e - 1$ und damit

$$D = 4 - 8(2e-1) = 12 - 16e.$$

Das ist genau die Bedingung $D \equiv 12 \bmod 16$. In diesem Fall ist

$$M = \mathbf{Z} \cdot 2 + \mathbf{Z}\frac{2+\sqrt{D}}{2}$$

ein Modul mit Norm 2.

(2) Wir untersuchen, wann es für $p \neq 2$ und $D \equiv 0 \bmod p$ einen primitiven Modul mit Norm p gibt. Wir können ansetzen

$$M = \mathbf{Z}p + \mathbf{Z}\frac{b+\sqrt{D}}{2}, \quad D = b^2 - 4pc, \quad \mathrm{ggT}(p,b,c) = 1, \quad -p < b \leq p.$$

Aus $p|D$ folgt $p|b$, also bleiben nur die Fälle $b = 0$ oder $b = p$.
Fall $b = 0$: Dann ist $D = -4pc$ mit $\mathrm{ggT}(p,c) = 1$, also folgt $D \not\equiv 0 \bmod p^2$ und $D \equiv 0 \bmod 4$. Man sieht auch sofort, daß in diesem Fall

$$M_p = \mathbf{Z} \cdot p + \mathbf{Z}\frac{0+\sqrt{D}}{2}$$

ein primitiver Modul mit Norm p ist.
Fall $b = p$: Dann ist $D = p^2 - 4pc = p(p-4c)$ mit $\mathrm{ggT}(p,c) = 1$. Es folgt $D \not\equiv 0 \bmod p^2$ und $D \equiv 1 \bmod 4$. Ist umgekehrt $D \not\equiv 0 \bmod p^2$ und $D \equiv 1 \bmod 4$, so kann man schreiben $D = p(p-m)$ mit $m \not\equiv 0 \bmod p$, was modulo 4 zu $1 \equiv D \equiv p^2 - pm \equiv 1 - pm \bmod 4$ führt, also $m = 4c$ und $D = p^2 - 4pc$ mit $\mathrm{ggT}(p,c) = 1$. Dann ist

$$M_p = \mathbf{Z} \cdot p + \mathbf{Z} \cdot \frac{p+\sqrt{D}}{2}$$

ein primitiver Modul mit Norm p. ■

Bemerkung: Mit den vorangegangenen Sätzen haben wir eine Maple-Funktion 'modul_kon_p' geschrieben, die zu gegebener Diskriminante D und Primzahl p alle primitiven Moduln in $\mathrm{Mod}(D)$ mit Norm p (schnell) bestimmt.

Beispiel: Wir bestimmen die ersten 10 Primzahlen $p > 10^{30}$, zu denen es einen primitiven Modul mit Diskriminante $D = 1009$ und Norm p gibt:

$$\mathbf{Z}\cdot 1000000000000000000000000000099+\mathbf{Z}\cdot\frac{\pm2759245177242199442911650698 63+\sqrt{1009}}{2}$$

$$\mathbf{Z}\cdot 1000000000000000000000000000271+\mathbf{Z}\cdot\frac{\pm3952033424681425387065281309 89+\sqrt{1009}}{2}$$

$$\mathbf{Z}\cdot 1000000000000000000000000000469+\mathbf{Z}\cdot\frac{\pm4227747953987139391734636308 21+\sqrt{1009}}{2}$$

$$\mathbf{Z}\cdot 1000000000000000000000000000529+\mathbf{Z}\cdot\frac{\pm9616283299398521936779531575 01+\sqrt{1009}}{2}$$

$$\mathbf{Z}\cdot 1000000000000000000000000000751+\mathbf{Z}\cdot\frac{\pm4529812939164777682450973491 51+\sqrt{1009}}{2}$$

$$\mathbf{Z}\cdot 1000000000000000000000000001069+\mathbf{Z}\cdot\frac{\pm6624459360016564615165348743 3+\sqrt{1009}}{2}$$

$$\mathbf{Z}\cdot 1000000000000000000000000001137+\mathbf{Z}\cdot\frac{\pm3082073090277066594818802602 51+\sqrt{1009}}{2}$$

$$\mathbf{Z}\cdot 1000000000000000000000000001521+\mathbf{Z}\cdot\frac{\pm8213553425886258880038764131 09+\sqrt{1009}}{2}$$

$$\mathbf{Z}\cdot 1000000000000000000000000001543+\mathbf{Z}\cdot\frac{\pm1264571120988810693356000794 63+\sqrt{1009}}{2}$$

$$\mathbf{Z}\cdot 1000000000000000000000000001629+\mathbf{Z}\cdot\frac{\pm3175307637609212662431771566 51+\sqrt{1009}}{2}$$

9. Einheiten

Sei D eine quadratische Diskriminante, $\delta = D \bmod 2$, $\delta \in \{0,1\}$ und $\omega = \frac{\delta+\sqrt{D}}{2}$. Dann ist

$$R_D = \mathbf{Z} + \mathbf{Z}\omega \quad \text{mit} \quad \omega+\omega' = \delta, \quad \omega\omega' = \frac{\delta - D}{4}.$$

Für $\alpha = x + y\omega \in R_D$ gilt

$$\mathrm{N}(x+y\omega) = (x+y\omega)(x+y\omega') = x^2 + \delta xy + \frac{\delta - D}{4}y^2 \in \mathbf{Z}.$$

Die Einheitengruppe von R_D ist

$$R_D^* = \{\alpha \in R_D : \text{ es gibt } \beta \in R_D \text{ mit } \alpha\beta = 1\}.$$

Die Einheiten in R_D lassen sich wie folgt charakterisieren:

LEMMA. *Für $\alpha \in R_D$ gilt:*

$$\alpha \in R_D^* \quad \Longleftrightarrow \quad \mathrm{N}(\alpha) = \pm 1.$$

Beweis: Ist $\alpha \in R_D^*$, so gibt es $\beta \in R_D$ mit $\alpha\beta = 1$. Normbildung liefert $\mathrm{N}(\alpha)\mathrm{N}(\beta) = 1$. Wegen $\mathrm{N}(\alpha), \mathrm{N}(\beta) \in \mathbf{Z}$ folgt $\mathrm{N}(\alpha) = \pm 1$. Ist umgekehrt $\mathrm{N}(\alpha) = \pm 1$, so hat man $\alpha\alpha' = \pm 1$, also $\alpha(\pm\alpha') = 1$, d.h. α ist Einheit. ■

Wir können damit also schreiben:

$$R_D^* = \{x + y\omega : x, y \in \mathbf{Z}, x^2 + \delta xy + \frac{\delta - D}{4}y^2 = \pm 1\}.$$

Für negative Diskriminanten gilt folgender Satz:

SATZ. *Für $D<0$ gilt*

$$R_D^* = \begin{cases} \{\pm 1, \frac{\pm 1 \pm \sqrt{-3}}{2}\} & \text{für } D=-3, \\ \{\pm 1, \pm\sqrt{-1}\} & \text{für } D=-4, \\ \{\pm 1\} & \text{für } D<-4. \end{cases}$$

Beweis: Wir müssen die ganzzahligen Lösungen der Gleichung

$$(x+\frac{\delta}{2}y)^2 + \frac{|D|}{4}y^2 = x^2 + \delta xy + \frac{\delta+|D|}{4}y^2 = \pm 1$$

bestimmen. Für $D<-4$ folgt $y=0$, $x=\pm 1$, für $D=-4$ gibt es die Lösungen $(x,y)=(\pm 1,0),(0,\pm 1)$, für $D=-3$ die Lösungen $(x,y)=(\pm 1,0),(0,\pm 1),(1,-1),(-1,1)$, was die Behauptung zeigt. ■

Wir kommen nun zu positiven Diskriminanten. Natürlich gibt es immer die Einheiten ± 1.

LEMMA. *Sei $D>0$ und $\alpha \in \mathrm{Red}(D)$ mit Kettenbruchentwicklung $[\overline{u_0,u_1,\ldots,u_{k-1}}]$ und Näherungsbrüchen $\frac{p_n}{q_n}$. Ist ε eine Einheit in R_D mit $\varepsilon>1$, dann gibt es ein $l\geq 1$ mit*

$$\varepsilon = q_{kl-2} + q_{kl-1}\alpha.$$

Beweis: Wir schreiben

$$\alpha = \frac{b+\sqrt{D}}{2a} \quad \text{mit} \quad D=b^2-4ac, \quad \mathrm{ggT}(a,b,c)=1.$$

Wegen

$$R_D = \mathbf{Z} + \mathbf{Z}\cdot\frac{b+\sqrt{D}}{2} = \mathbf{Z} + \mathbf{Z}\cdot a\alpha$$

gibt es $x,y,z\in\mathbf{Z}$ mit $\varepsilon = x+y\alpha$ und $y=az$. Dann ist $\varepsilon' = x+y\alpha'$. Aus $\pm 1 = \varepsilon\varepsilon'$ und $\varepsilon>1$ folgt $-1<\varepsilon'<1$. Aus $\varepsilon'<\varepsilon$ folgt $y\alpha'<y\alpha$, also $y\geq 1$ und damit $z\geq 1$. Wegen $\alpha'<0$ folgt aus $-1<\varepsilon' = x+y\alpha' < x+y\cdot 0$ und Ungleichung $x\geq 0$. Wir unterscheiden nun ein paar Fälle:

- Wäre $x\geq y+1$, so hätte man
$$\varepsilon' = x+y\alpha' \geq 1+y+y\alpha' = 1+y(1+\alpha') > 1+y(1+(-1)) = 1,$$
ein Widerspruch. Dieser Fall kann also nicht eintreten, es gilt immer $x\leq y$.
- Wir behandeln jetzt den Fall $x=y=az$. Dann ist
$$\varepsilon = az(1+\alpha), \quad \varepsilon' = az(1+\alpha') > 0.$$
Es folgt
$$\begin{aligned} 1 &= \varepsilon\varepsilon' = z^2(a+\frac{b+\sqrt{D}}{2})(a+\frac{b-\sqrt{D}}{2}) = \frac{1}{4}z^2(2a+b+\sqrt{D})(2a+b-\sqrt{D}) = \\ &= \frac{1}{4}z^2((2a+b)^2-(b^2-4ac)) = \frac{1}{4}z^2(4a^2+4ab+b^2-b^2+4ac) = z^2a(a+b+c). \end{aligned}$$
Dies zeigt zunächst $a=z=1$, dann folgt aus $a+b+c=1$ die Beziehung $c=-b$. Dann ist
$$D = b^2-4ac = b^2-4\cdot 1\cdot(-b) = b^2+4b.$$
Also
$$\alpha = \frac{b+\sqrt{b^2+4b}}{2}, \quad \alpha' = \frac{b-\sqrt{b^2+4b}}{2}.$$
Es gilt
$$b^2+4b < b^2+4b+4 = (b+2)^2 \Longrightarrow \sqrt{D} < b+2 \Longrightarrow -2 < b-\sqrt{D} \Longrightarrow -1<\alpha',$$
woraus man schnell sieht, daß α reduziert ist. Dann ist
$$\varepsilon = x+y\alpha = 1+\alpha = \frac{(b+2)+\sqrt{D}}{2}$$
tatsächlich eine Einheit. $\alpha = [\overline{b,1}]$ erfüllt die Gleichung $\alpha^2 - b\alpha - b = 0$, also
$$\alpha = \frac{b+\sqrt{b^2+4b}}{2}.$$

Dann ist

$$\frac{p_0}{q_0} = [b] = \frac{b}{1}, \quad \frac{p_1}{q_1} = [b,1] = b+1 = \frac{b+1}{1},$$

also $q_0 = q_1 = 1$ und damit wie behauptet $\varepsilon = q_0 + q_1\alpha$.

- Was passiert im Fall $x = 0$? Dann ist

$$\varepsilon = y\alpha = za\alpha = z\frac{b+\sqrt{D}}{2},$$

$$\varepsilon\varepsilon' = z^2\frac{b^2-(b^2-4ac)}{4} = z^2 ac,$$

was sofort $z = 1$, $a = 1$, $c = \pm 1$ liefert. Aus $\alpha' = \frac{b-\sqrt{b^2-4c}}{2} < 0$ folgt $c = -1$, also $D = b^2+4$ und

$$\alpha = \frac{b+\sqrt{b^2+4}}{2}.$$

Die Kettenbruchentwicklung von α ist $\alpha = \overline{[b]}$ mit der Gleichung $\alpha^2 - b\alpha - 1 = 0$. Mit $q_{-1} = 0$ und $q_0 = 1$ folgt $\varepsilon = q_0\alpha + q_{-1}$.

- Es bleibt der Fall $1 \le x < y$. Wir schreiben $\varepsilon\alpha = u + v\alpha$ und erhalten damit

$$\alpha = \frac{\varepsilon\alpha}{\varepsilon} = \frac{v\alpha+u}{y\alpha+x}.$$

Es folgt

$$y\alpha + x = v + u\frac{1}{\alpha}$$

und damit

$$\frac{b+\sqrt{D}}{2a}y + x = v + \frac{b-\sqrt{D}}{2c}u,$$

$$(\frac{b}{2a}y + x) + \frac{1}{2a}y\sqrt{D} = (v + \frac{b}{2c}u) - \frac{1}{2c}u\sqrt{D},$$

was durch Koeffizientenvergleich

$$\frac{1}{2a}y = -\frac{1}{2c}u, \quad \frac{b}{2a}y + x = v - \frac{b}{2a}y,$$

also

$$u = -\frac{c}{a}y \quad \text{und} \quad v = \frac{b}{a}y + x$$

ergibt. Nun folgt

$$\begin{aligned} vx - uy &= (\frac{b}{a}y + x)x - (-\frac{c}{a}y)y = x^2 + \frac{b}{a}xy + \frac{c}{a}y^2 = x^2 + (\alpha+\alpha')xy + \alpha\alpha' y^2 = \\ &= (x+y\alpha)(x+y\alpha') = \varepsilon\varepsilon' = \pm 1. \end{aligned}$$

Wegen $1 \le x < y$ gibt es nun nach einem (beim Satz von Serret bewiesenen) Lemma $n \in \mathbf{N}$ mit

$$\alpha = \alpha_n, \quad v = p_{n-1}, \quad u = p_{n-2}, \quad y = q_{n-1}, \quad x = q_{n-2}.$$

Die die Kettenbruchentwicklung von α Periode k hat, gibt es wegen $\alpha = \alpha_n$ ein l mit $n = kl$, also $x = q_{kl-2}$, $y = q_{kl-1}$, wie behauptet. ■

LEMMA. *Sei α reduziert mit Diskriminante $D > 0$ und Kettenbruchentwicklung $\alpha = \overline{[u_0, u_1, \ldots, u_{k-1}]}$ und Näherungsbrüchen $\frac{p_n}{q_n}$. Dann ist*

$$q_{k-2} + q_{k-1}\alpha \in R_D^* \quad \textit{mit} \quad \mathrm{N}(q_{k-2} + q_{k-1}\alpha) = (-1)^k.$$

Beweis: Wir schreiben

$$\alpha = \frac{b+\sqrt{D}}{2a} \quad \text{mit} \quad D = b^2 - 4ac, \quad \mathrm{ggT}(a,b,c) = 1.$$

Es gilt

$$\alpha = \overline{[u_0, u_1, \ldots, u_{k-1}]} = [u_0, u_1, \ldots, u_{k-1}, \alpha] = \frac{\widetilde{p_k}}{\widetilde{q_k}} = \frac{\alpha p_{k-1} + p_{k-2}}{\alpha q_{k-1} + q_{k-2}},$$

was

$$q_{k-2} + q_{k-1}\alpha = p_{k-1} + p_{k-2}\frac{1}{\alpha}$$

und mit

$$\frac{1}{\alpha} = \frac{b - \sqrt{D}}{2c}$$

dann

$$(\frac{b}{2a}q_{k-1} + q_{k-2}) + \frac{1}{2a}q_{k-1}\sqrt{D} = (p_{k-1} + \frac{b}{2c}p_{k-2}) - \frac{1}{2c}p_{k-2}\sqrt{D}$$

liefert. Koeffizientenvergleich ergibt

$$\frac{b}{2a}q_{k-1} + q_{k-2} = p_{k-1} + \frac{b}{2c}p_{k-2} \quad \text{und} \quad \frac{1}{2a}q_{k-1} = -\frac{1}{2c}p_{k-2},$$

daher

$$\begin{aligned} p_{k-2} &= -\frac{c}{a}q_{k-1}, \\ p_{k-1} &= \frac{b}{2a}q_{k-1} + q_{k-2} - \frac{b}{2c}p_{k-2} = \frac{b}{2a}q_{k-1} + q_{k-2} + \frac{b}{2a}q_{k-1} = \frac{b}{a}q_{k-1} + q_{k-2}. \end{aligned}$$

Wegen $\mathrm{ggT}(a,b,c) = 1$ ergibt sich sofort $a|q_{k-1}$. Nun gilt

$$\begin{aligned} (-1)^k &= p_{k-1}q_{k-2} - p_{k-2}q_{k-1} = (\frac{b}{a}q_{k-1} + q_{k-2})q_{k-2} + \frac{c}{a}q_{k-1}^2 = \\ &= q_{k-2}^2 + \frac{b}{a}q_{k-2}q_{k-1} + \frac{c}{a}q_{k-1}^2 \\ \mathrm{N}(q_{k-2} + q_{k-1}\alpha) &= (q_{k-2} + q_{k-1}\alpha)(q_{k-2} + q_{k-1}\alpha') = q_{k-2}^2 + q_{k-2}q_{k-1}(\alpha + \alpha') + q_{k-1}^2\alpha\alpha' = \\ &= q_{k-2}^2 + q_{k-2}q_{k-1}\frac{b}{a} + q_{k-1}^2\frac{c}{a} = (-1)^k \end{aligned}$$

Wegen $a|q_{k-1}$ ist $q_{k-2} + q_{k-1}\alpha \in R_D$. ■

SATZ. *Sei $D > 0$ und $\alpha \in \mathrm{Red}(D)$ mit Kettenbruchentwicklung $\alpha = [\overline{u_0, u_1, \ldots, u_{k-1}}]$ und Näherungsbrüchen $\frac{p_n}{q_n}$. Dann gilt*

$$R_D^* = \{\pm(q_{k-2} + q_{k-1}\alpha)^n : n \in \mathbf{Z}\}.$$

Beweis: Wir müssen nur noch zeigen, daß sich tatsächlich jede Einheit ε in der Form $\pm(q_{k-2} + q_{k-1}\alpha)^n$ mit einem $n \in \mathbf{Z}$ schreiben läßt. Wir nehmen an, dies ist nicht der Fall für ε. Durch Übergang zu $\frac{1}{\varepsilon}$ oder $-\varepsilon$ oder $-\frac{1}{\varepsilon}$ können wir $\varepsilon > 1$ annehmen. Dann gibt es ein $n \geq 0$ mit

$$(q_{k-2} + q_{k-1}\alpha)^n < \varepsilon < (q_{k-2} + q_{k-1}\alpha)^{n+1}.$$

Offensichtlich ist nun

$$\widetilde{\varepsilon} = \varepsilon \cdot (q_{k-2} + q_{k-1}\alpha)^{-n}$$

eine Einheit mit

$$1 < \widetilde{\varepsilon} < q_{k-2} + q_{k-1}\alpha.$$

Nach einem vorangegangenen Satz gibt es ein $l \geq 1$ mit

$$\widetilde{\varepsilon} = q_{kl-2} + q_{kl-1}\alpha,$$

was aber offensichtlich unmöglich ist. Damit haben wir einen Widerspruch, der den Satz schließlich beweist. ■

Bemerkung: Wir haben eine Maple-Funktion 'rq_einheit' geschrieben, die zu einer positiven Diskriminante D die Grundeinheit in der Form $x + y\frac{(D \bmod 2) + \sqrt{D}}{2}$ ausgibt.

Beispiele: In der folgenden Tabelle liefert $x + y\frac{(D \mod 2)+\sqrt{D}}{2}$ die Grundeinheit des Ringes R_D.

D	x	y
5	0	1
8	1	1
12	2	1
13	1	1
17	3	2
20	2	1
21	2	1
24	5	2
28	8	3
29	2	1
32	3	1
33	19	8
37	5	2
40	3	1
41	27	10
44	10	3
45	3	1
48	7	2
52	18	5
53	3	1

D	x	y
56	15	4
57	131	40
60	4	1
61	17	5
65	7	2
68	4	1
69	11	3
72	17	4
73	943	250
76	170	39
77	4	1
80	9	2
84	55	12
85	4	1
88	197	42
89	447	106
92	24	5
93	13	3
96	5	1
97	5035	1138

D	x	y
101	9	2
104	5	1
105	37	8
108	26	5
109	118	25
112	127	24
113	703	146
116	70	13
117	5	1
120	11	2
124	1520	273
125	5	1
128	17	3
129	15371	2968
132	23	4
133	79	15
136	35	6
137	1595	298
140	6	1
141	87	16

D	x	y
145	11	2
148	6	1
149	28	5
152	37	6
153	2001	352
156	25	4
157	98	17
160	19	3
161	10847	1856
164	32	5
165	6	1
168	13	2
172	3482	531
173	6	1
176	199	30
177	57731	9384
180	161	24
181	604	97
184	24335	3588
185	63	10

D	x	y
2001	19301573127	882709936
2004	11242731902975	502288218432
2005	197	9
2008	3832352837	171046278
2009	137866468650241	6292130833920
2012	24648	1099
2013	329	15
2016	449	20
2017	1041436030172223769440529 63	4743392230761615119582962
2020	809	36
2021	22	1
2024	45	2
2028	1351	60
2029	22	1
2032	44757606858751	1985797689600
2033	60424559051	2741038872
2036	395727950	17540333
2037	331	15
2040	271	12
2041	15040127574888277	680896478514216

10. Modulklassen

Ist $M \in \mathrm{Mod}(D)$ und $\alpha \in K^*$, so gilt $\mathrm{End}(\alpha M) = \mathrm{End}(M) = R_D$, also hat auch αM Diskriminante D, d.h. $\alpha M \in \mathrm{Mod}(D)$. Wir können daher eine Relation auf $\mathrm{Mod}(D)$ definieren:

$$M_1 \sim M_2 \quad \Longleftrightarrow \quad M_1 = \alpha M_2 \text{ für ein } \alpha \in K^*.$$

Offensichtlich ist $\sim$ eine Äquivalenzrelation. Die Äquivalenzklassen von M heißt die Modulklasse $\overline{M}$ von M. Da die Relation auch mit der Multiplikation verträglich ist, d.h.

$$M_1 \sim M_2, \quad N_1 \sim N_2 \quad \Longrightarrow \quad M_1N_1 \sim M_2N_2,$$

induziert die Modulmultiplikation eine Multiplikation der Modulklassen durch die Vorschrift

$$\overline{M} \cdot \overline{N} = \overline{MN}.$$

Die Menge der Modulklassen wird dann zur sogenannten Klassengruppe $\mathrm{H}(D)$ zur Determinante D:

$$\mathrm{H}(D) = \{\overline{M} : M \in \mathrm{Mod}(D)\}.$$

LEMMA. *Das neutrale Element in* $\mathrm{H}(D)$ *ist* $\overline{R_D}$*, invers zu* $\overline{M}$ *ist* $\overline{M'}$*.*

Beweis: Die erste Aussage folgt aus $R_D M = M$ für $M \in \mathrm{Mod}(D)$, die zweite aus der Beziehung $MM' = \mathrm{N}(M)R_D \sim R_D$. ■

Bemerkung: Es ist zunächst nicht klar, wie man Moduln $M_1, M_2 \in \mathrm{Mod}(D)$ ansehen kann, ob sie äquivalent sind oder nicht.

SATZ. *Seien* $M_1, M_2 \in \mathrm{Mod}(D)$ *mit*

$$M_1 = A_1(\mathbf{Z} \cdot a_1 + \mathbf{Z} \cdot \frac{b_1 + \sqrt{D}}{2}), \quad M_2 = A_2(\mathbf{Z} \cdot a_2 + \mathbf{Z} \cdot \frac{b_2 + \sqrt{D}}{2}).$$

Dann gilt mit

$$\alpha_1 = \frac{b_1 + \sqrt{D}}{2a_1}, \quad \alpha_2 = \frac{b_2 + \sqrt{D}}{2a_2}$$

die Äquivalenz

$$M_1 \sim M_2 \quad \Longleftrightarrow \quad \textit{es gibt } A, B, C, D \in \mathbf{Z} \textit{ mit } AD - BC = \pm 1 \textit{ und } \alpha_2 = \frac{A\alpha_1 + B}{C\alpha_1 + D}.$$

Beweis: Es gilt

$$\begin{aligned}
M_1 \sim M_2 \quad &\Longleftrightarrow \quad A_1(\mathbf{Z} \cdot a_1 + \mathbf{Z} \cdot \frac{b_1 + \sqrt{D}}{2}) \sim A_2(\mathbf{Z} \cdot a_2 + \mathbf{Z} \cdot \frac{b_2 + \sqrt{D}}{2}) \quad \Longleftrightarrow \\
&\Longleftrightarrow \quad \mathbf{Z} + \mathbf{Z}\alpha_1 = \mathbf{Z} + \mathbf{Z} \cdot \frac{b_1 + \sqrt{D}}{2a_1} \sim \mathbf{Z} + \mathbf{Z} \cdot \frac{b_2 + \sqrt{D}}{2a_2} = \mathbf{Z} + \mathbf{Z}\alpha_2 \\
&\Longleftrightarrow \quad \mathbf{Z} + \mathbf{Z}\alpha_1 = \alpha(\mathbf{Z} + \mathbf{Z}\alpha_2) = \mathbf{Z}\alpha + \mathbf{Z}\alpha\alpha_2 \text{ für ein } \alpha \in K \\
&\Longleftrightarrow \quad \text{es gibt } A, B, C, D \in \mathbf{Z} \text{ mit } AD - BC = \pm 1 \text{ und } \alpha \in K \text{ mit} \\
&\qquad \alpha = C\alpha_1 + D, \quad \alpha\alpha_2 = A\alpha_1 + B \\
&\Longleftrightarrow \quad \text{es gibt } A, B, C, D \in \mathbf{Z} \text{ mit } AD - BC = \pm 1 \text{ und } \alpha_2 = \frac{A\alpha_1 + B}{C\alpha_1 + D},
\end{aligned}$$

was die Behauptung beweist. ■

Bemerkung: Der letzte Satz wird sich für Diskriminanten $D > 0$ als hilfreich erweisen. Um Klassengruppen explizit zu berechnen, werden wir die Fälle $D > 0$ und $D < 0$ unterscheiden.

11. Hauptideale

Sei D eine quadratische Diskriminante, $\delta = D \mod 2$, $\delta \in \{0,1\}$ und $\omega = \frac{\delta+\sqrt{D}}{2}$. Dann ist $R_D = \mathbf{Z} + \mathbf{Z}\omega$ und

$$\mathrm{S}(\omega) = \omega + \omega' = \delta, \quad \mathrm{N}(\omega) = \omega\omega' = \frac{\delta - D}{4}, \quad \omega^2 - \mathrm{S}(\omega)\omega + \mathrm{N}(\omega) = 0, \quad \text{also} \quad \omega^2 = \frac{D-\delta}{4} + \delta\omega.$$

Ist $\alpha \in K^*$, so gilt für $\alpha R_D = \mathbf{Z}\alpha + \mathbf{Z}\alpha\omega$ die Beziehung $\mathrm{End}(\alpha R_D) = \mathrm{End}(R_D) = R_D$, also hat auch αR_D Diskriminante D, d.h. $\alpha R_D \in \mathrm{Mod}(D)$.

Definition. Ein Modul $M \in \mathrm{Mod}(D)$ heißt ein Hauptideal/Hauptmodul, wenn es ein $\alpha \in K^*$ gibt mit $M = \alpha R_D$. Die Menge der Hauptideale bezeichnen wir mit $\mathrm{HMod}(D)$, d.h. $\mathrm{HMod}(D) = \{\alpha R_D : \alpha \in K^*\}$.

Das erzeugende Element α eines Hauptmoduls αR_D ist nur bis auf eine Einheit bestimmt, wie folgendes Lemma zeigt:

Lemma. *Für $\alpha, \beta \in K^*$ gilt:*

$$\alpha R_D = \beta R_D \quad \Longleftrightarrow \quad \frac{\alpha}{\beta} \in R_D^*.$$

Beweis:

$$\begin{aligned} \alpha R_D = \beta R_D \quad &\Longleftrightarrow \quad \alpha \in \beta R_D \text{ und } \beta \in \alpha R_D \quad \Longleftrightarrow \quad \alpha = \beta u \text{ und } \beta = \alpha v \text{ mit } u, v \in R_D \\ &\Longleftrightarrow \quad \frac{\alpha}{\beta} \in R_D \text{ und } \frac{\beta}{\alpha} \in R_D \quad \Longleftrightarrow \quad \frac{\alpha}{\beta} \in R_D^*, \end{aligned}$$

was alles beweist. ■

Natürlich ist $\mathrm{HMod}(D)$ eine Untergruppe von $\mathrm{Mod}(D)$. Mit dem letzten Lemma erhält man dann sofort folgende gruppentheoretische Charakterisierung:

Satz. *Die Zuordnung $K^* \to \mathrm{Mod}(D)$, $\alpha \mapsto \alpha R_D$ liefert einen Gruppenisomorphismus*

$$K^*/R_D^* \simeq \mathrm{HMod}(D).$$

Der folgende Satz gibt eine weitere gruppentheoretische Beschreibung der Klassengruppe $\mathrm{H}(D)$:

Satz. *Die Äquivalenzklassenabbildung $\mathrm{Mod}(D) \to \mathrm{H}(D)$, $M \mapsto \overline{M}$ induziert einen Gruppenisomorphismus*

$$\mathrm{Mod}(D)/\mathrm{HMod}(D) \simeq \mathrm{H}(D).$$

Beweis: Die Abbildung

$$f : \mathrm{Mod}(D) \to \mathrm{H}(D), \quad M \mapsto \overline{M}$$

ist offensichtlich ein surjektiver Gruppenhomomorphismus. Es gilt

$$M \in \mathrm{Kern}(f) \quad \Longleftrightarrow \quad \overline{M} = \overline{R_D} \quad \Longleftrightarrow \quad M = \alpha R_D \text{ für ein } \alpha \in K \quad \Longleftrightarrow \quad M \in \mathrm{HMod}(D),$$

was dann die Behauptung beweist. ■

Die entscheidende Beziehung zwischen Elementnorm $\mathrm{N}(\alpha) = \alpha\alpha'$ und Modulnorm $\mathrm{N}(\alpha R_D)$ liefert der folgende Satz:

Satz. *Für $\alpha \in K^*$ gilt:*

$$\mathrm{N}(\alpha R_D) = |\mathrm{N}(\alpha)|,$$

d.h. Idealnorm und Elementnorm stimmen bis aufs Vorzeichen überein.

Beweis: Wegen $R'_D = R_D$ gilt

$$\mathrm{N}(\alpha R_D)R_D = (\alpha R_D)(\alpha R_D)' = \alpha\alpha' R_D = \mathrm{N}(\alpha)R_D,$$

was durch Schnitt mit $\mathbf{Q}$ die Beziehung

$$\mathbf{Z}\cdot\mathrm{N}(\alpha R_D) = \mathbf{Q}\cap(\mathrm{N}(\alpha R_D)R_D) = \mathbf{Q}\cap(\mathrm{N}(\alpha)R_D) = \mathbf{Z}\cdot\mathrm{N}(\alpha)$$

und damit $\mathrm{N}(\alpha R_D) = \pm\mathrm{N}(\alpha)$ liefert. ■

Alternativer Beweis: Schreiben wir $\alpha = x + y\omega$, so gilt

$$\alpha R_D = \alpha(\mathbf{Z}+\mathbf{Z}\omega) = \mathbf{Z}(x+y\omega) + \mathbf{Z}(x\omega + y\frac{D-\delta}{4} + y\delta\omega) = \mathbf{Z}(x+y\omega) + \mathbf{Z}(y\frac{D-\delta}{4} + (x+y\delta)\omega).$$

Die Übergangsmatrix zur Berechnung der Norm von αR_D ist dann

$$\begin{pmatrix} x & y \\ y\frac{D-\delta}{4} & x+y\delta \end{pmatrix}$$

mit Determinante $x^2 + \delta xy + \frac{\delta - D}{4}y^2$, was wegen

$$\mathrm{N}(x+y\omega) = (x+y\omega)(x+y\omega') = x^2 + (\omega+\omega')xy + \omega\omega' y^2 = x^2 + \delta xy + \frac{\delta - D}{4}y^2$$

mit der Norm von α übereinstimmt und damit die Behauptung beweist. ■

Bemerkung: Mit den Bezeichnungen des letzten Beweises gilt für $\alpha = x + y\omega$

$$\alpha R_D = \mathbf{Z}(x+y\omega) + \mathbf{Z}(y\frac{D-\delta}{4} + (x+y\delta)\omega).$$

Bestimmt man die zugehörige Normalform, so erhält man αR_D in der Form $A(\mathbf{Z}a + \mathbf{Z}\frac{b+\sqrt{D}}{2})$. Die Maple-Funktion 'hauptideal' geht so vor. Eine alternative Möglichkeit gibt das folgende Lemma:

LEMMA. *Ist $\alpha = A(x+y\omega) \in K^*$ mit $A \in \mathbf{Q}$, $x, y \in \mathbf{Z}$ mit $\mathrm{ggT}(x,y) = 1$, wählt man $u, v, b, a \in \mathbf{Z}$ mit*

$$ux + vy = 1, \quad b = (-\delta x + \frac{D-\delta}{2}y)u + (2x+\delta y)v, \quad a = |x^2 + \delta xy + \frac{\delta - D}{4}y^2|,$$

so gilt

$$\alpha R_D = A(\mathbf{Z}\cdot a + \mathbf{Z}\cdot\frac{b+\sqrt{D}}{2}).$$

Beweis: Natürlich können wir o.E. $A = 1$ annehmen. Dann ist $\alpha = x + y\omega$ mit $x, y \in \mathbf{Z}$, $\mathrm{ggT}(x,y) = 1$ und

$$\begin{aligned}
\alpha R_D &= \alpha(\mathbf{Z}+\mathbf{Z}\omega) = \mathbf{Z}(x+y\omega) + \mathbf{Z}(x\omega + y\frac{D-\delta}{4} + y\delta\omega) = \\
&= \mathbf{Z}(x+y\omega) + \mathbf{Z}(-x\delta + y\frac{D-\delta}{4} + x\omega) = \\
&= \mathbf{Z}(x + y\frac{\delta+\sqrt{D}}{2}) + \mathbf{Z}(-x\delta + y\frac{D-\delta}{4} + x\frac{\delta+\sqrt{D}}{2}) = \\
&= \mathbf{Z}\cdot\frac{2x+\delta y + y\sqrt{D}}{2} + \mathbf{Z}\cdot\frac{-\delta x + \frac{D-\delta}{2}y + x\sqrt{D}}{2}.
\end{aligned}$$

Wegen $\mathrm{ggT}(x,y) = 1$ ist die Normalform offensichtlich $\mathbf{Z}\cdot a + \mathbf{Z}\cdot\frac{b+\sqrt{D}}{2}$, wo a die Norm ist. Für $\frac{b+\sqrt{D}}{2}$ kann irgendein Element aus αR_D gewählt werden, dessen Koeffizient bei $\sqrt{D}$ die Zahl $\frac{1}{2}$ ist, also z.B. das Element mit dem im Satz angegebenen b. ■

Beispiel: $D = -19$. Wir bestimmen alle primitiven Moduln mit Norm ≤ 20:

$$\mathbf{Z} + \mathbf{Z} \cdot \frac{1+\sqrt{-19}}{2}, \quad \mathbf{Z} \cdot 5 + \mathbf{Z} \cdot \frac{1+\sqrt{-19}}{2}, \quad \mathbf{Z} \cdot 5 + \mathbf{Z} \cdot \frac{-1+\sqrt{-19}}{2}, \quad \mathbf{Z} \cdot 7 + \mathbf{Z} \cdot \frac{3+\sqrt{-19}}{2},$$
$$\mathbf{Z} \cdot 7 + \mathbf{Z} \cdot \frac{-3+\sqrt{-19}}{2}, \quad \mathbf{Z} \cdot 11 + \mathbf{Z} \cdot \frac{5+\sqrt{-19}}{2}, \quad \mathbf{Z} \cdot 11 + \mathbf{Z} \cdot \frac{-5+\sqrt{-19}}{2}, \quad \mathbf{Z} \cdot 17 + \mathbf{Z} \cdot \frac{7+\sqrt{-19}}{2},$$
$$\mathbf{Z} \cdot 17 + \mathbf{Z} \cdot \frac{-7+\sqrt{-19}}{2}, \quad \mathbf{Z} \cdot 19 + \mathbf{Z} \cdot \frac{19+\sqrt{-19}}{2}.$$

Wir suchen für $|x|, |y| \leq 20$ nach primitiven Moduln $(x + y\omega)R_D$ mit Norm ≤ 20:

$$\omega R_D = \mathbf{Z} \cdot 5 + \mathbf{Z} \cdot \frac{1+\sqrt{-19}}{2}, \quad (1-2\omega)R_D = \mathbf{Z} \cdot 19 + \mathbf{Z} \cdot \frac{19+\sqrt{-19}}{2},$$
$$(1-\omega)R_D = \mathbf{Z} \cdot 5 + \mathbf{Z} \cdot \frac{-1+\sqrt{-19}}{2}, \quad (1)R_D = \mathbf{Z} + \mathbf{Z} \cdot \frac{1+\sqrt{-19}}{2},$$
$$(1+\omega)R_D = \mathbf{Z} \cdot 7 + \mathbf{Z} \cdot \frac{3+\sqrt{-19}}{2}, \quad (2-\omega)R_D = \mathbf{Z} \cdot 7 + \mathbf{Z} \cdot \frac{-3+\sqrt{-19}}{2},$$
$$(2+\omega)R_D = \mathbf{Z} \cdot 11 + \mathbf{Z} \cdot \frac{5+\sqrt{-19}}{2}, \quad (3-\omega)R_D = \mathbf{Z} \cdot 11 + \mathbf{Z} \cdot \frac{-5+\sqrt{-19}}{2},$$
$$(3+\omega)R_D = \mathbf{Z} \cdot 17 + \mathbf{Z} \cdot \frac{7+\sqrt{-19}}{2}, \quad (4-\omega)R_D = \mathbf{Z} \cdot 17 + \mathbf{Z} \cdot \frac{-7+\sqrt{-19}}{2}.$$

Man sieht, daß alle primitiven Moduln mit Norm ≤ 20 Hauptmoduln sind. Gilt daher schon $\mathrm{Mod}(-19) = \mathrm{HMod}(-19)$?

Beispiel: $D = -20$. Wir listen alle primitiven Moduln mit Norm ≤ 20 auf:

$$\mathbf{Z} + \mathbf{Z} \cdot \frac{0+\sqrt{-20}}{2}, \quad \mathbf{Z} \cdot 2 + \mathbf{Z} \cdot \frac{2+\sqrt{-20}}{2}, \quad \mathbf{Z} \cdot 3 + \mathbf{Z} \cdot \frac{2+\sqrt{-20}}{2}, \quad \mathbf{Z} \cdot 3 + \mathbf{Z} \cdot \frac{-2+\sqrt{-20}}{2},$$
$$\mathbf{Z} \cdot 5 + \mathbf{Z} \cdot \frac{0+\sqrt{-20}}{2}, \quad \mathbf{Z} \cdot 6 + \mathbf{Z} \cdot \frac{2+\sqrt{-20}}{2}, \quad \mathbf{Z} \cdot 6 + \mathbf{Z} \cdot \frac{-2+\sqrt{-20}}{2}, \quad \mathbf{Z} \cdot 7 + \mathbf{Z} \cdot \frac{6+\sqrt{-20}}{2},$$
$$\mathbf{Z} \cdot 7 + \mathbf{Z} \cdot \frac{-6+\sqrt{-20}}{2}, \quad \mathbf{Z} \cdot 9 + \mathbf{Z} \cdot \frac{4+\sqrt{-20}}{2}, \quad \mathbf{Z} \cdot 9 + \mathbf{Z} \cdot \frac{-4+\sqrt{-20}}{2}, \quad \mathbf{Z} \cdot 10 + \mathbf{Z} \cdot \frac{10+\sqrt{-20}}{2},$$
$$\mathbf{Z} \cdot 14 + \mathbf{Z} \cdot \frac{6+\sqrt{-20}}{2}, \quad \mathbf{Z} \cdot 14 + \mathbf{Z} \cdot \frac{-6+\sqrt{-20}}{2}, \quad \mathbf{Z} \cdot 15 + \mathbf{Z} \cdot \frac{10+\sqrt{-20}}{2},$$
$$\mathbf{Z} \cdot 15 + \mathbf{Z} \cdot \frac{-10+\sqrt{-20}}{2}, \quad \mathbf{Z} \cdot 18 + \mathbf{Z} \cdot \frac{14+\sqrt{-20}}{2}, \quad \mathbf{Z} \cdot 18 + \mathbf{Z} \cdot \frac{-14+\sqrt{-20}}{2}.$$

Nun listen wir alle primitiven Moduln $(x + y\omega)R_D$ mit $|x|, |y| \leq 20$ und Norm ≤ 20 auf:

$$\omega R_D = \mathbf{Z} \cdot 5 + \mathbf{Z} \cdot \frac{0+\sqrt{-20}}{2}, \quad (1-\omega)R_D = \mathbf{Z} \cdot 6 + \mathbf{Z} \cdot \frac{-2+\sqrt{-20}}{2},$$
$$(1)R_D = \mathbf{Z} + \mathbf{Z} \cdot \frac{0+\sqrt{-20}}{2}, \quad (1+\omega)R_D = \mathbf{Z} \cdot 6 + \mathbf{Z} \cdot \frac{2+\sqrt{-20}}{2},$$
$$(2-\omega)R_D = \mathbf{Z} \cdot 9 + \mathbf{Z} \cdot \frac{-4+\sqrt{-20}}{2}, \quad (2+\omega)R_D = \mathbf{Z} \cdot 9 + \mathbf{Z} \cdot \frac{4+\sqrt{-20}}{2},$$
$$(3-\omega)R_D = \mathbf{Z} \cdot 14 + \mathbf{Z} \cdot \frac{-6+\sqrt{-20}}{2}, \quad (3+\omega)R_D = \mathbf{Z} \cdot 14 + \mathbf{Z} \cdot \frac{6+\sqrt{-20}}{2}.$$

Es treten also hier keine primitiven Hauptmoduln mit Norm 2, 3, 7, 10, 15, 18 auf. Wegen $\mathrm{N}(x + y\omega) = x^2 + 5y^2$ sieht man auch sofort, daß es keine primitiven Hauptmoduln mit den angegebenen Normen geben kann. Wir haben also $\mathrm{HMod}(-20) \neq \mathrm{Mod}(20)$.

Beispiel: $D = 13$. Wir listen alle primitiven Moduln mit Norm ≤ 20 auf und geben gleich die dazu gefundenen Darstellungen $(x + y\omega)R_D$ mit $|x|, |y| \leq 4$ (bis aufs Vorzeichen) an.

$$\begin{aligned}
\mathbf{Z} + \mathbf{Z} \cdot \frac{1+\sqrt{13}}{2} &= (1)R_D = (1+\omega)R_D = (2-\omega)R_D = (4+3\omega)R_D \\
\mathbf{Z} \cdot 3 + \mathbf{Z} \cdot \frac{1+\sqrt{13}}{2} &= \omega R_D = (3-\omega)R_D = (3+2\omega)R_D \\
\mathbf{Z} \cdot 3 + \mathbf{Z} \cdot \frac{-1+\sqrt{13}}{2} &= (1-\omega)R_D = (2+\omega)R_D \\
\mathbf{Z} \cdot 9 + \mathbf{Z} \cdot \frac{7+\sqrt{13}}{2} &= (3-2\omega)R_D = (3+\omega)R_D \\
\mathbf{Z} \cdot 9 + \mathbf{Z} \cdot \frac{-7+\sqrt{13}}{2} &= (1+2\omega)R_D = (4-\omega)R_D \\
\mathbf{Z} \cdot 13 + \mathbf{Z} \cdot \frac{13+\sqrt{13}}{2} &= (1-2\omega)R_D \\
\mathbf{Z} \cdot 17 + \mathbf{Z} \cdot \frac{9+\sqrt{13}}{2} &= (4+\omega)R_D \\
\mathbf{Z} \cdot 17 + \mathbf{Z} \cdot \frac{-9+\sqrt{13}}{2} &= (2+3\omega)R_D.
\end{aligned}$$

Die angegebenen primitiven Moduln sind also alle Hauptmoduln. Die Darstellung ist nicht eindeutig, da es nichttriviale Einheiten gibt, z.B. $1 + \omega$.

12. Übungen

Aufgabe 6.1: Sei

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \quad \text{mit } A, B, C, D \in \mathbf{Z} \quad \text{und} \quad AD - BC = \pm 1.$$

Zeige, daß sich M als Produkt mit Faktoren aus $\{S, T, T_1\}$ schreiben läßt, wobei

$$S = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad T = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \quad T_1 = \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix}$$

ist.

Aufgabe 6.2: Sei K ein quadratischer Zahlkörper mit Fundamentaldiskriminante D. Bezeichnet R_f die Ordnung mit Diskriminante $f^2 D$, so gilt für das Produkt von Ordnungen (als Moduln):

$$R_{f_1} R_{f_2} = R_{\mathrm{ggT}(f_1, f_2)}.$$

Aufgabe 6.3: Sei D Fundamentaldiskriminante eines quadratischen Zahlkörpers und $M_1 \in \mathrm{Mod}(f_1^2 D)$, $M_2 \in \mathrm{Mod}(f_2^2 D)$. Dann gilt:

$$\mathrm{N}(M_1 M_2) = \mathrm{N}(M_1)\mathrm{N}(M_2) \quad \text{und} \quad M_1 M_2 \in \mathrm{Mod}(\mathrm{ggT}(f_1, f_2)^2 D).$$

Aufgabe 6.4: Sei D eine reellquadratische Diskriminante mit $D \equiv 1 \bmod 8$ und ε eine Grundeinheit der zugehörigen Ordnung. Dann ist ε auch Grundeinheit der Ordnung mit Diskriminante $4D$.

Aufgabe 6.5: Sei D eine reellquadratische Diskriminante mit $D \equiv 5 \bmod 8$ und ε Grundeinheit der Ordnung R_D.

(1) Ist $\varepsilon \in R_{4D}$, so ist ε Grundeinheit der Ordnung R_{4D}.
(2) Ist $\varepsilon \notin R_{4D}$, so ist ε^3 Grundeinheit von R_{4D}.

Aufgabe 6.6: Sei ε_D die Grundeinheit des Ringes R_D für eine reellquadratische Diskriminante D.

(1) Zeige: Gibt es eine Primzahl $p \equiv 3 \bmod 4$ mit $p|D$, so gilt $\mathrm{N}(\varepsilon_D) = 1$.
(2) Gib numerische Beispiele, die zeigen, daß die Umkehrung in 1. im allgemeinen nicht gilt.

Aufgabe 6.7: Sei $D = p$ eine Primzahl mit $p \equiv 1 \bmod 4$. Zeige, daß die Grundeinheit des Ringes R_D Norm -1 hat.

Aufgabe 6.8: Sei D eine reellquadratische Diskriminante und $\alpha \in \mathrm{Red}(D)$ mit der Kettenbruchentwicklung $\alpha = [\overline{u_0, u_1, \ldots, u_{k-1}}]$ und den Näherungsbrüchen $\frac{p_i}{q_i}$. Es wurde gezeigt, daß $\varepsilon = q_{k-2} + q_{k-1}\alpha$ eine Grundeinheit der Ordnung R_D ist. Hier soll die (einfache) Abschätzung

$$\varepsilon < e^D$$

in folgenden Schritten bewiesen werden.

(1) Definiere eine Folge β_i durch

$$\beta_{-2} = 1, \quad \beta_{-1} = 0, \quad \beta_i = \alpha_i \beta_{i-1} + \beta_{i-2} \text{ für } i \geq 0,$$

wobei $\alpha_i = \rho^i(\alpha)$ die Nachfolger von α in der Kettenbruchentwicklung sind, was insbesondere $u_i = \lfloor \alpha_i \rfloor$ und $\alpha_k = \alpha_0 = \alpha$ liefert.
(2) Zeige durch Induktion $q_i \leq \beta_i$ für alle $i \geq -2$.
(3) Zeige $\varepsilon \leq \beta_k$.
(4) Zeige durch Induktion $\beta_i < (\alpha_1 + 1)(\alpha_2 + 1) \ldots (\alpha_i + 1)$ für alle $i \geq 1$.
(5) Folgere $\beta_i < e^{\alpha_1 + \alpha_2 + \cdots + \alpha_i}$ und damit $\varepsilon < e^{\alpha_1 + \alpha_2 + \cdots + \alpha_k}$.
(6) Dann gilt

$$\varepsilon < e^{\sum_{\alpha \in \mathrm{Red}(D)} \alpha}$$

und mit der früher bewiesenen Ungleichung $\sum_{\alpha \in \mathrm{Red}(D)} \alpha < D$ folgt die behauptete Abschätzung für ε.

Aufgabe 6.9: Sei $D \equiv 0 \bmod 4$ eine quadratische Diskriminante. Charakterisiere die D's, für die ein primitiver Modul mit Norm 4 und Diskriminante D existiert.

KAPITEL 7

Imaginärquadratische Klassengruppen

1. Einführung

Sei D eine quadratische Diskriminante. Die Moduln mit Diskriminante D lassen sich in der Form

$$M = A(\mathbf{Z}\cdot a + \mathbf{Z}\cdot\frac{b+\sqrt{D}}{2}) \quad \text{mit} \quad D = b^2-4ac, \quad \mathrm{ggT}(a,b,c)=1, \quad \text{o.E. } -a < b \le b$$

schreiben. Durch

$$M_1 \sim M_2 \quad \Longleftrightarrow \quad M_1 = \alpha M_2 \text{ für ein } \alpha \in K$$

wird eine Äquivalenzrelation definiert, die mit der Modulmultiplikation verträglich ist. Die Menge der Äquivalenzklassen liefert die Klassengruppe $\mathrm{H}(D)$ zur Diskriminante D:

$$\mathrm{H}(D) = \{\overline{M} : M \in \mathrm{Mod}(D)\} = \mathrm{Mod}(D)/\sim .$$

Das neutrale Element ist die Klasse von

$$R_D = \mathbf{Z} + \mathbf{Z}\cdot\frac{(D \bmod 2)+\sqrt{D}}{2}.$$

Aus der Normrelation $MM' = \mathrm{N}(N)R_D \sim R_D$ folgt $\overline{M}^{-1} = \overline{M'}$.
Die Ordnung der Klassengruppe $\mathrm{H}(D)$ heißt die Klassenzahl $h(D)$ zur Diskriminante D:

$$h(D) = \#\mathrm{H}(D).$$

Um die Klassengruppe zu beschreiben, wollen wir versuchen, aus jeder Äquivalenzklasse möglichst einfache Repräsentanten auszuwählen.

2. Der Reduktionsalgorithmus

Sei $M \in \mathrm{Mod}(D)$ durch

$$M = A(\mathbf{Z}\cdot a + \mathbf{Z}\cdot\frac{b+\sqrt{D}}{2}), \quad D = b^2-4ac, \quad \mathrm{ggT}(a,b,c)=1, \quad \text{o.E. } -a < b \le a$$

gegeben. Da wir $D<0$ voraussetzen, ist

$$4ac = b^2 - D = b^2 + |D|.$$

Daher gilt wegen $a \ge 1$ auch $c \ge 1$. Wir haben

$$A(\mathbf{Z}\cdot a + \mathbf{Z}\cdot\frac{b+\sqrt{D}}{2}) \sim \mathbf{Z}\cdot a + \mathbf{Z}\cdot\frac{b+\sqrt{D}}{2},$$

also können wir uns sofort auf primitive Moduln beschränken.
Aus $4ac = b^2 - D$ folgt

$$\frac{b-\sqrt{D}}{2a}\left(\mathbf{Z}\cdot a + \mathbf{Z}\cdot\frac{b+\sqrt{D}}{2}\right) = \mathbf{Z}\cdot\frac{b-\sqrt{D}}{2} + \mathbf{Z}\cdot\frac{b^2-D}{4a} = \mathbf{Z}\cdot c + \mathbf{Z}\cdot\frac{-b+\sqrt{D}}{2}.$$

Damit erhalten wir

$$\begin{aligned}\mathbf{Z}\cdot a+\mathbf{Z}\cdot\frac{b+\sqrt{D}}{2} &\sim \mathbf{Z}\cdot c+\mathbf{Z}\cdot\frac{-b+\sqrt{D}}{2}=\\ &= \mathbf{Z}\cdot c+\mathbf{Z}\cdot\frac{(-b+2cm)+\sqrt{D}}{2}\\ &\quad \text{für } m\in\mathbf{Z}, \text{ also o.E. } -c<-b+2mc\le c\end{aligned}$$

Mit dieser Beziehung definieren wir einen Reduktionsalgorithmus:

Reduktionsalgorithmus: Gegeben sei $M\in \mathrm{Mod}(D)$ mit

$$M=\mathbf{Z}\cdot a+\mathbf{Z}\cdot\frac{b+\sqrt{D}}{2},\quad D=b^2-4ac,\quad \mathrm{ggT}(a,b,c)=1,\quad \text{o.E. } -a<b\le a.$$

Anfang: Ist die Ungleichung $-a<b\le a$ nicht gewährleistet, reduziere b modulo $2a$ so, daß $-a<b\le a$ gilt. Explizit: $b:=b \bmod 2a$, ist dann $b>a$, setze $b:=b-2a$.
Berechne $c:=\frac{b^2-D}{4a}$, wenn c noch nicht bekannt ist.

Reduktionsschritt: Führe folgendes aus, solange $a>c$ gilt:

$$a:=c,\quad b:=-b \bmod 2a \text{ mit } -a<b\le a,\quad c:=\frac{b^2-D}{4a}.$$

Abschluß: Ist $a=c$ und $b<0$, ersetze b durch $-b$. Gib den Modul $\mathbf{Z}\cdot a+\mathbf{Z}\cdot\frac{b+\sqrt{D}}{2}$ aus.

Nun gilt $-a<b\le a$, $a\le c$. Ist $a=c$, so gilt außerdem noch $b\ge 0$.

Bemerkungen:

(1) Da im Reduktionsschritt a kleiner wird, hört der Reduktionsalgorithmus nach endlich vielen Schritten auf.
(2) Die Reduktion $b:=-b \bmod 2a$ mit $-a<b\le a$, kann man explizit so erhalten: $b:=-b \bmod 2a$, ersetze im Fall $b>a$ dann $b:=b-2a$.
(3) Da nach obiger Relation im Fall $a=c$ die Beziehung $\mathbf{Z}\cdot a+\mathbf{Z}\cdot\frac{b+\sqrt{D}}{2}\sim\mathbf{Z}\cdot a+\mathbf{Z}\cdot\frac{-b+\sqrt{D}}{2}$ gilt, kann man o.E. $b\ge 0$ wählen.

Beispiel: Wir starten mit $M=\mathbf{Z}\cdot 1013+\mathbf{Z}\cdot\frac{497+\sqrt{-163}}{2}$ und erhalten bei Reduktion die folgenden Tripel (a,b,c):

a	b	c
1013	497	61
61	-9	1
1	1	41

Also ist M äquivalent zu $\mathbf{Z}+\mathbf{Z}\cdot\frac{1+\sqrt{-163}}{2}$.

Beispiel: Wir reduzieren den Modul

$$M=\mathbf{Z}\cdot 100019+\mathbf{Z}\cdot\frac{90505+\sqrt{-999}}{2}$$

und erhalten dabei folgende Tripel (a,b,c):

a	b	c
100019	90505	20474
20474	-8609	905
905	-441	54
54	9	5
5	1	50

Der Ausgangsmodul M ist also äquivalent zu $\mathbf{Z}\cdot 5+\mathbf{Z}\cdot\frac{1+\sqrt{-999}}{2}$.

Beispiel: Wir betrachten den Modul $M \in \mathrm{Mod}(-10007)$

$$M = \mathbf{Z} \cdot 100000000000000000129 + \mathbf{Z} \cdot \frac{17742755318685394947 + \sqrt{-10007}}{2}.$$

Beim Reduzieren ergeben sich folgende Tripel (a, b, c):

a	b	c
100000000000000000129	17742755318685394947	787013415746847176
787013415746847176	-428460172254757075	58314799829164008
58314799829164008	-38058226378554989	6209523994443004
6209523994443004	801082411896965	25836643486077
25836643486077	25690179657499	6386136527276
6386136527276	-145633548395	830280183
830280183	-495763813	74005668
74005668	51729805	9039756
9039756	2508731	174057
174057	-71933	7432
7432	-2387	192
192	83	22
22	5	114

Unser Ausgangsmodul M ist also äquivalent zu

$$\mathbf{Z} \cdot 22 + \mathbf{Z} \cdot \frac{5 + \sqrt{-10007}}{2}.$$

Wir nennen einen Modul $M \in \mathrm{Mod}(D)$ reduziert, wenn er durch das Reduktionsverfahren nicht mehr verändert wird, was zu folgender formalen Definition führt.

DEFINITION. Ein Modul $M \in \mathrm{Mod}(D)$ heißt reduziert, wenn gilt

$$\begin{aligned} M \quad = \quad & \mathbf{Z} \cdot a + \mathbf{Z} \cdot \frac{b + \sqrt{D}}{2}, \quad D = b^2 - 4ac, \quad \mathrm{ggT}(a, b, c) = 1, \\ & -a < b \leq a, \quad a \leq c \quad \text{und} \quad b \geq 0 \text{ im Fall } a = c. \end{aligned}$$

Die Menge der reduzierten Moduln wird

$$\begin{aligned} \mathrm{Red}(D) \quad = \quad & \{\mathbf{Z} \cdot a + \mathbf{Z} \cdot \frac{b + \sqrt{D}}{2} \in \mathrm{Mod}(D) : \quad D = b^2 - 4ac, \quad \mathrm{ggT}(a, b, c) = 1, \\ & -a < b \leq a, \quad a \leq c \quad \text{und} \quad b \geq 0 \text{ im Fall } a = c\}. \end{aligned}$$

Bemerkung: Genau genommen ist die Definition nicht ganz exakt, denn b ist durch den Modul M nicht eindeutig bestimmt, sondern nur modulo $2a$. Man sollte also genauer sagen, ein Modul M heißt reduziert, wenn er sich in obiger Form schreiben läßt.

Der Reduktionsalgorithmus liefert nun sofort folgendes Ergebnis:

SATZ. *Der Reduktionsalgorithmus wandelt jeden Modul $M \in \mathrm{Mod}(D)$ in einen reduzierten Modul um, d.h. jeder Modul $M \in \mathrm{Mod}(D)$ ist zu einem Modul aus $\mathrm{Red}(D)$ äquivalent.*

Bemerkung: In Maple stellen wir einen primitiven Modul $\mathbf{Z} \cdot a + \mathbf{Z} \cdot \frac{b+\sqrt{D}}{2}$ durch ein Tripel $[a, b, D]$ dar. Die Funktion HD_red liefert zu einem Modul M nach obigem Reduktionsalgorithmus den zugehörigen reduzierten Modul.

3. Die Menge $\mathrm{Red}(D)$ der reduzierten Moduln

Wir wollen jetzt die reduzierten Moduln untersuchen.

LEMMA. *Für* $\mathbf{Z}\cdot a+\mathbf{Z}\cdot\frac{b+\sqrt{D}}{2}\in\mathrm{Red}(D)$ *gilt*

$$|b|\le a\le\sqrt{\frac{|D|}{3}}=0.57735...\sqrt{|D|}.$$

Insbesondere ist $\mathrm{Red}(D)$ *eine endliche Menge.*

Beweis: Wir haben $|b|\le a\le c$ und $D=b^2-4ac$. Es folgt

$$|D|+a^2\ge|D|+b^2=b^2-D=4ac\ge 4a^2,$$

also $a^2\le\frac{1}{3}|D|$, was die Behauptung beweist. ■

Beispiel: Mit den obigen Abschätzungen erhält man sofort

$$\mathrm{Red}(-3)=\{\mathbf{Z}+\mathbf{Z}\cdot\frac{1+\sqrt{-3}}{2}\},\quad \mathrm{Red}(-4)=\{\mathbf{Z}+\mathbf{Z}\cdot\frac{0+\sqrt{-4}}{2}\}.$$

Es gibt auch eine Art von Umkehrung des letzten Lemmas:

LEMMA. *Gilt für* $M=\mathbf{Z}\cdot a+\mathbf{Z}\cdot\frac{b+\sqrt{D}}{2}\in\mathrm{Mod}(D)$

$$-a<b\le a\quad\text{ und }\quad a\le\sqrt{\frac{|D|}{4}},$$

so ist M *reduziert.*

Beweis: Mit $D=b^2-4ac$ und der Voraussetzung $4a^2\le|D|$ folgt

$$c=\frac{b^2-D}{4a}=\frac{b^2+|D|}{4a}\ge\frac{|D|}{4a}\ge\frac{4a^2}{4a}=a,$$

also ist $c\ge a$. Gilt $c>a$, so ist M reduziert. Gilt $c=a$, so folgt $b=0$, was mit $\mathrm{ggT}(a,b,c)=1$ sofort $a=c=1$, also $M=\mathbf{Z}+\mathbf{Z}\cdot\sqrt{-1}$ impliziert. Dieser Modul ist offensichtlich auch reduziert, was die Behauptung beweist. ■

LEMMA. *Sei* $M=\mathbf{Z}a+\mathbf{Z}\frac{b+\sqrt{D}}{2}$ *reduziert. Dann gilt*

$$\min\{\mathrm{N}(\alpha):\alpha\in M,\alpha\ne 0\}=a^2$$

und für $\alpha\in M$

$$\mathrm{N}(\alpha)=a^2\quad\Longleftrightarrow\quad\alpha\in\begin{cases}\{\pm a\} & \text{im Fall } a<c,\\ \{\pm a,\pm\frac{b+\sqrt{D}}{2}\} & \text{im Fall } a=c,a>1,\\ \{\pm1,\pm\frac{1+\sqrt{-3}}{2},\pm\frac{-1+\sqrt{-3}}{2}\} & \text{im Fall } a=b=c=1 \text{ und } D=-3.\end{cases}$$

Beweis: Für

$$\alpha=ax+\frac{b+\sqrt{D}}{2}y\quad\text{ gilt }\quad\mathrm{N}(\alpha)=\frac{|2ax+by|^2}{4}+\frac{|D|}{4}y^2.$$

Nach eventuellem Übergang zu $-\alpha$ können wir o.E. $y\ge 0$ voraussetzen. Außerdem sei $\alpha\ne 0$.
Fall $y=0$*:* Es ist $\mathrm{N}(\alpha)=a^2x^2\ge a^2$, interessant ist hier nur $\alpha=\pm a$.
Fall $y=1$*:* Wir haben $\mathrm{N}(\alpha)=\frac{1}{4}(|2ax+b|^2+|D|)$.
Ist $|2ax+b|>|b|$, so wird

$$\mathrm{N}(\alpha)>\frac{b^2+|D|}{4}=ac\ge a^2,$$

das Element kann also nicht minimal sein.
Es bleibt nur die Möglichkeit $|2ax+b| \leq |b|$. Wegen $-a < b \leq a$ bleiben nur die Fälle $x = 0$ oder $x = -1$, wenn $b = a$ gilt. Dann ist $|2ax + b| = |b|$ und damit

$$\mathrm{N}(\alpha) = \frac{b^2 + |D|}{4} = ac \geq a^2.$$

Der Fall ist also nur interessant, falls $c = a$ gilt. In diesem Fall gilt auch $\mathrm{N}(\alpha) = a^2$.
Für $x = 0$ hat man $\alpha = \frac{b+\sqrt{D}}{2}$ mit $c = a$.
Für $x = -1$ und $b = a$ und $a = c$ ist wegen $\mathrm{ggT}(a, b, c) = 1$ dann $a = b = c = 1$ und damit $D = -3$ und $\alpha = -1 + \frac{1+\sqrt{-3}}{2} = \frac{-1+\sqrt{-3}}{2}$.
Fall $y \geq 2$: Hier gilt

$$\mathrm{N}(\alpha) = (ax + \frac{b}{2}y)^2 + \frac{|D|}{4}y^2 \geq |D| \geq 3a^2 > a^2,$$

dieser Fall ist also nicht interessant.
Faßt man die obigen Fallunterscheidungen zusammen, erhält man die Behauptung. ■

SATZ. *Die Menge* Red(D) *ist ein Repräsentantensystem der Klassengruppe* H(D).

Beweis: Wir haben bereits gesehen, daß jeder Modul zu einem reduzierten äquivalent ist. Es bleibt noch zu zeigen, daß in jeder Klasse nur ein reduzierter Modul liegt. Seien also M_1, M_2 reduziert mit $M_2 = \lambda M_1$ und $a_1 = \mathrm{N}(M_1)$, $a_2 = \mathrm{N}(M_2)$. Dann ist

$$M_1' M_2 = \lambda M_1' M_1 = \lambda \mathrm{N}(M_1) R_D.$$

Sei $\alpha = \lambda \mathrm{N}(M_1)$, also $M_1' M_2 = \alpha R_D$. Wegen $M_1' \subseteq R_D$, $M_2 \subseteq R_D$ gilt $\alpha \in M_1'$ und $\alpha \in M_2$. Es folgt

$$\mathrm{N}(\alpha) = \mathrm{N}(\alpha R_D) = \mathrm{N}(M_1' M_2) = \mathrm{N}(M_1')\mathrm{N}(M_2) = a_1 a_2.$$

Wegen $\alpha \in M_1'$ folgt $\mathrm{N}(\alpha) \geq a_1^2$, wegen $\alpha \in M_2$ folgt $\mathrm{N}(\alpha) \geq a_2^2$, was sofort $a_1 = a_2$ impliziert. Außerdem ist nun α Element minimaler Norm in M_1' und M_2. Wir unterscheiden mit dem vorangegangnen Lemma zwei Fälle, je nachdem wie α als Element minimaler Norm in M_2 aussieht. Wir können $D < -3$ annehmen, da $\#\mathrm{Red}(-3) = 1$ gilt.

- $\alpha = \pm a_2 = \pm a_1$. Aus $M_1' M_2 = \alpha R_D = a_1 R_D = M_1' M_1$ folgt $M_1 = M_2$.
- $\alpha = \pm\frac{b_2+\sqrt{D}}{2}$ und $a_2 = c_2 > 1$. Da α' Element minimaler Norm in M_1 ist, folgt $\alpha' = \pm\frac{b_1+\sqrt{D}}{2}$ mit $a_1 = c_1$. Dies liefert $b_1 = -b_2$, wegen $a_1 = c_1 = a_2 = c_2$, also $b_1 = b_2 = 0$ und damit $M_1 = M_2$. (Aus $\mathrm{ggT}(a_1, b_1, c_1) = 1$ folgt auch $a_1 = c_1 = 1$ und damit $D = -4$.) ■

Bemerkungen:

(1) Durch Bestimmung von Red(D) erhalten wir also ein Repräsentantensystem vom H(D).
(2) Wie kann man testen, ob zwei Moduln $M_1, M_2 \in \mathrm{Mod}(D)$ äquivalent sind oder nicht? Man reduziert die Moduln zu $(M_1)_{\mathrm{red}}$ und $(M_2)_{\mathrm{red}}$ testet, ob diese Moduln gleich sind oder nicht.

FOLGERUNG. *Für die Klassenzahl* $h(D) = \#\mathrm{H}(D)$ *gilt*

$$h(D) = \#\mathrm{Red}(D).$$

Wie bestimmt man die Menge Red(D) der reduzierten Moduln mit Diskriminante D explizit? Der Modul

$$M = \mathbf{Z} \cdot a + \mathbf{Z} \cdot \frac{b + \sqrt{D}}{2}, \quad D = b^2 - 4ac, \quad \mathrm{ggT}(a, b, c) = 1$$

ist genau dann reduziert, wenn

$$-a < b \leq a, \quad 1 \leq a \leq c, \quad \text{im Fall } a = c \text{ ist } b \geq 0, \quad D = b^2 - 4ac, \quad \mathrm{ggT}(a, b, c) = 1$$

gilt. Wegen

$$4ac = b^2 + |D|, \quad b \equiv D \bmod 2, \quad |b| \leq a \leq \sqrt{ac} = \sqrt{\frac{b^2 + |D|}{4}}$$

sind die Bedingungen äquivalent zu folgenden:

$$|b| \le a \le \sqrt{\frac{b^2+|D|}{4}}, \quad b \equiv D \bmod 2, \quad 4ac = b^2 + |D|, \quad \text{im Fall } |b| = a \text{ oder } c = a \text{ ist } b \ge 0.$$

Wir erhalten damit folgendes Verfahren:

Algorithmus zur Bestimmung von $\mathrm{Red}(D)$**:** Für eine negative Diskriminante D werden alle reduzierten Moduln $\mathbf{Z}a + \mathbf{Z}\frac{b+\sqrt{D}}{2}$ bzw. die zugehörigen Tripel (a,b,c) aufgelistet:

(1) Setze $b := D \bmod 2$.
(2) Berechne $A := \frac{b^2+|D|}{4}$ und $w := \lfloor\sqrt{A}\rfloor$. Ist $b > w$, beende das Verfahren.
(3) Setze $a := b$. Ist $a = 0$, setze $a := 1$.
(4) Ist $A \bmod a = 0$, berechne $c := \frac{A}{a}$.
(5) Ist $\mathrm{ggT}(a,b,c) = 1$, gib (a,b,c) bzw. $\mathbf{Z}a + \mathbf{Z}\frac{b+\sqrt{D}}{2}$ aus, im Fall $b > 0$ und $b < a$ und $a < c$ dazu noch $(a,-b,c)$ bzw. $\mathbf{Z}a + \mathbf{Z}\frac{-b+\sqrt{D}}{2}$.
(6) Setze $a := a + 1$. Ist $a \le w$, gehe zu 4.
(7) Setze $b := b + 2$. Gehe zu 2.

Beispiel: $D = -103$. Wir wollen die reduzierten Tripel $(a,b,c) \simeq \mathbf{Z} \cdot a + \mathbf{Z} \cdot \frac{b+\sqrt{-103}}{2}$ nach obigem Verfahren bestimmen und testen daher die Möglichkeiten für b:

- $b = 1$ liefert $A = 26$, $\lfloor\sqrt{A}\rfloor = 5$. Wegen $b \le a \le \lfloor\sqrt{A}\rfloor$ und $a|A$ sind nur $a = 1, 2$ sind möglich. Dies liefert die Tripel $(1,1,26)$, $(2,1,13)$, $(2,-1,13)$.
- $b = 3$ liefert $A = 28$, $\lfloor\sqrt{A}\rfloor = 5$. Wegen $b \le a \le \lfloor\sqrt{A}\rfloor$ und $a|A$ ist nur $a = 4$ möglich und liefert $(4,3,7)$, $(4,-3,7)$.
- $b = 5$ liefert $A = 32$, $\lfloor\sqrt{A}\rfloor = 5$. Wegen $b \le a \le \lfloor\sqrt{A}\rfloor$, bleibt nur $a = 5$, was aber wegen der Bedingung $a|A$ nicht geht.
- $b = 7$ liefert $A = 38$, $\lfloor\sqrt{A}\rfloor = 6$. Wegen der Bedingung $b < \lfloor\sqrt{A}\rfloor$, sind wir nun fertig.

Wir haben also

$$\mathrm{Red}(-103) \simeq \{(1,1,26),(2,1,13),(2,-1,13),(4,3,7),(4,-3,7)\},$$

insbesondere ist $\mathrm{H}(-103)$ eine Gruppe mit 5 Elementen.

Bemerkung: Die Maple-Funktion $\mathrm{Red}(D)$ liefert zu einer imaginärquadratischen Diskriminante D die Menge der reduzierten Moduln mit Diskrimiante in der Form $[a,b,D]$, was $\mathbf{Z} \cdot a + \mathbf{Z} \cdot \frac{b+\sqrt{D}}{2}$ entspricht.

Beispiele: In der folgenden Tabelle finden sich zu den ersten negativen Diskriminanten die reduzierten Tripel (a, b, c) mit Diskriminante D.

D	(a, b, c)
-3	(1,1,1)
-4	(1,0,1)
-7	(1,1,2)
-8	(1,0,2)
-11	(1,1,3)
-12	(1,0,3)
-15	(1,1,4),(2,1,2)
-16	(1,0,4)
-19	(1,1,5)
-20	(1,0,5),(2,2,3)
-23	(1,1,6),(2,1,3),(2,-1,3)
-24	(1,0,6),(2,0,3)
-27	(1,1,7)
-28	(1,0,7)
-31	(1,1,8),(2,1,4),(2,-1,4)
-32	(1,0,8),(3,2,3)
-35	(1,1,9),(3,1,3)
-36	(1,0,9),(2,2,5)
-39	(1,1,10),(2,1,5),(2,-1,5),(3,3,4)
-40	(1,0,10),(2,0,5)
-43	(1,1,11)
-44	(1,0,11),(3,2,4),(3,-2,4)
-47	(1,1,12),(2,1,6),(2,-1,6),(3,1,4),(3,-1,4)
-48	(1,0,12),(3,0,4)
-51	(1,1,13),(3,3,5)
-52	(1,0,13),(2,2,7)
-55	(1,1,14),(2,1,7),(2,-1,7),(4,3,4)
-56	(1,0,14),(2,0,7),(3,2,5),(3,-2,5)
-59	(1,1,15),(3,1,5),(3,-1,5)
-60	(1,0,15),(3,0,5)
-63	(1,1,16),(2,1,8),(2,-1,8),(4,1,4)
-64	(1,0,16),(4,4,5)
-67	(1,1,17)
-68	(1,0,17),(2,2,9),(3,2,6),(3,-2,6)
-71	(1,1,18),(2,1,9),(2,-1,9),(3,1,6),(3,-1,6),(4,3,5),(4,-3,5)
-72	(1,0,18),(2,0,9)
-75	(1,1,19),(3,3,7)
-76	(1,0,19),(4,2,5),(4,-2,5)
-79	(1,1,20),(2,1,10),(2,-1,10),(4,1,5),(4,-1,5)
-80	(1,0,20),(4,0,5),(3,2,7),(3,-2,7)
-83	(1,1,21),(3,1,7),(3,-1,7)
-84	(1,0,21),(3,0,7),(2,2,11),(5,4,5)
-87	(1,1,22),(2,1,11),(2,-1,11),(3,3,8),(4,3,6),(4,-3,6)
-88	(1,0,22),(2,0,11)
-91	(1,1,23),(5,3,5)
-92	(1,0,23),(3,2,8),(3,-2,8)
-95	(1,1,24),(2,1,12),(2,-1,12),(3,1,8),(3,-1,8),(4,1,6),(4,-1,6),(5,5,6)
-96	(1,0,24),(3,0,8),(5,2,5),(4,4,7)
-99	(1,1,25),(5,1,5)

D	(a,b,c)
-100	(1,0,25),(2,2,13)
-103	(1,1,26),(2,1,13),(2,-1,13),(4,3,7),(4,-3,7)
-104	(1,0,26),(2,0,13),(3,2,9),(3,-2,9),(5,4,6),(5,-4,6)
-107	(1,1,27),(3,1,9),(3,-1,9)
-108	(1,0,27),(4,2,7),(4,-2,7)
-111	(1,1,28),(2,1,14),(2,-1,14),(4,1,7),(4,-1,7),(3,3,10),(5,3,6),(5,-3,6)
-112	(1,0,28),(4,0,7)
-115	(1,1,29),(5,5,7)
-116	(1,0,29),(2,2,15),(3,2,10),(3,-2,10),(5,2,6),(5,-2,6)
-119	(1,1,30),(2,1,15),(2,-1,15),(3,1,10),(3,-1,10),(5,1,6),(5,-1,6),(4,3,8),(4,-3,8),(6,5,6)
-120	(1,0,30),(2,0,15),(3,0,10),(5,0,6)
-123	(1,1,31),(3,3,11)
-124	(1,0,31),(5,4,7),(5,-4,7)
-127	(1,1,32),(2,1,16),(2,-1,16),(4,1,8),(4,-1,8)
-128	(1,0,32),(3,2,11),(3,-2,11),(4,4,9)
-131	(1,1,33),(3,1,11),(3,-1,11),(5,3,7),(5,-3,7)
-132	(1,0,33),(3,0,11),(2,2,17),(6,6,7)
-135	(1,1,34),(2,1,17),(2,-1,17),(4,3,9),(4,-3,9),(5,5,8)
-136	(1,0,34),(2,0,17),(5,2,7),(5,-2,7)
-139	(1,1,35),(5,1,7),(5,-1,7)
-140	(1,0,35),(5,0,7),(3,2,12),(3,-2,12),(4,2,9),(4,-2,9)
-143	(1,1,36),(2,1,18),(2,-1,18),(3,1,12),(3,-1,12),(4,1,9),(4,-1,9),(6,1,6),(6,5,7),(6,-5,7)
-144	(1,0,36),(4,0,9),(5,4,8),(5,-4,8)
-147	(1,1,37),(3,3,13)
-148	(1,0,37),(2,2,19)
-151	(1,1,38),(2,1,19),(2,-1,19),(4,3,10),(4,-3,10),(5,3,8),(5,-3,8)
-152	(1,0,38),(2,0,19),(3,2,13),(3,-2,13),(6,4,7),(6,-4,7)
-155	(1,1,39),(3,1,13),(3,-1,13),(5,5,9)
-156	(1,0,39),(3,0,13),(5,2,8),(5,-2,8)
-159	(1,1,40),(2,1,20),(2,-1,20),(4,1,10),(4,-1,10),(5,1,8),(5,-1,8),(3,3,14),(6,3,7),(6,-3,7)
-160	(1,0,40),(5,0,8),(4,4,11),(7,6,7)
-163	(1,1,41)
-164	(1,0,41),(2,2,21),(3,2,14),(3,-2,14),(6,2,7),(6,-2,7),(5,4,9),(5,-4,9)
-167	(1,1,42),(2,1,21),(2,-1,21),(3,1,14),(3,-1,14),(6,1,7),(6,-1,7),(4,3,11),(4,-3,11),(6,5,8),(6,-5,8)
-168	(1,0,42),(2,0,21),(3,0,14),(6,0,7)
-171	(1,1,43),(5,3,9),(5,-3,9),(7,5,7)
-172	(1,0,43),(4,2,11),(4,-2,11)
-175	(1,1,44),(2,1,22),(2,-1,22),(4,1,11),(4,-1,11),(7,7,8)
-176	(1,0,44),(4,0,11),(3,2,15),(3,-2,15),(5,2,9),(5,-2,9)
-179	(1,1,45),(3,1,15),(3,-1,15),(5,1,9),(5,-1,9)
-180	(1,0,45),(5,0,9),(2,2,23),(7,4,7)
-183	(1,1,46),(2,1,23),(2,-1,23),(3,3,16),(4,3,12),(4,-3,12),(6,3,8),(6,-3,8)
-184	(1,0,46),(2,0,23),(5,4,10),(5,-4,10)
-187	(1,1,47),(7,3,7)
-188	(1,0,47),(3,2,16),(3,-2,16),(7,6,8),(7,-6,8)
-191	(1,1,48),(2,1,24),(2,-1,24),(3,1,16),(3,-1,16),(4,1,12), (4,-1,12),(6,1,8),(6,-1,8),(5,3,10),(5,-3,10) (6,5,9),(6,-5,9)
-192	(1,0,48),(3,0,16),(7,2,7),(4,4,13)
-195	(1,1,49),(7,1,7),(3,3,17),(5,5,11)
-196	(1,0,49),(2,2,25),(5,2,10),(5,-2,10)
-199	(1,1,50),(2,1,25),(2,-1,25),(5,1,10),(5,-1,10),(4,3,13),(4,-3,13),(7,5,8),(7,-5,8)

4. Die Klassenzahl $h(D)$

Eine wichtige Größe ist die Klassenzahl

$$h(d) = \#\mathrm{H}(D) = \#\mathrm{Red}(D).$$

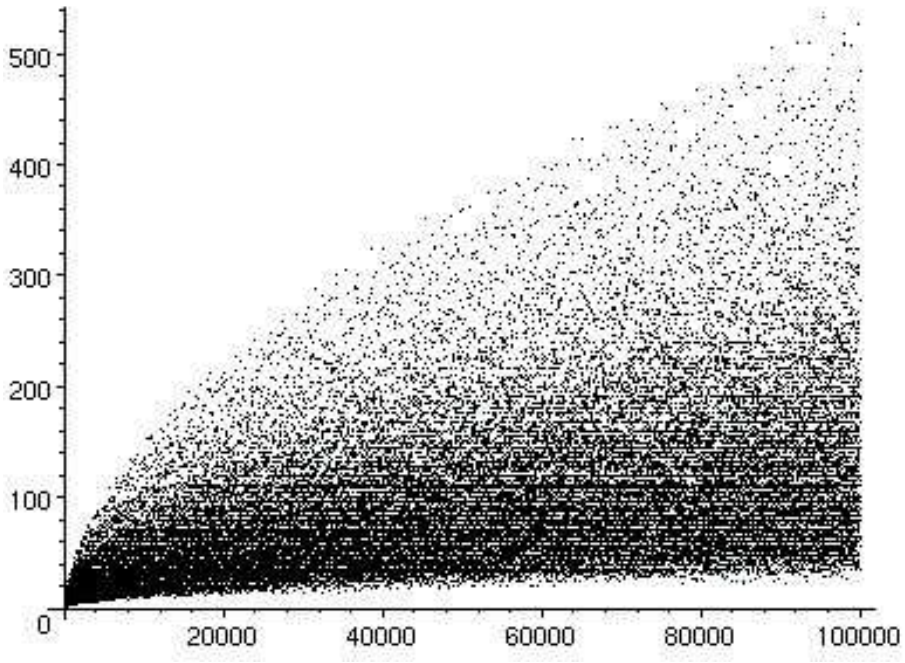
Eine Möglichkeit die Klassenzahl $h(D)$ zu bestimmen, besteht darin, alle reduzierten Moduln mit Diskriminante D zu zählen. Dann erhält man folgendes Verfahren:

Algorithmus zur Bestimmung von $h(D)$ durch Zählen der reduzierten Moduln: Für eine negative Diskriminante D wird die Klassenzahl $h(D)$ bestimmt.

(1) Setze $b := D \bmod 2$, $h := 0$.
(2) Berechne $A := \frac{b^2+|D|}{4}$ und $w := \lfloor\sqrt{A}\rfloor$. Ist $b > w$, gib h als Klassenzahl aus und beende das Verfahren.
(3) Setze $a := b$. Ist $a = 0$, setze $a := 1$.
(4) Ist $A \bmod a = 0$, berechne $c := \frac{A}{a}$.
(5) Ist $\mathrm{ggT}(a, b, c) = 1$, setze $h := h + 2$, ist $b = 0$ oder $a = b$ oder $a = c$, setze weiter $h := h - 1$.
(6) Setze $a := a + 1$. Ist $a \leq w$, gehe zu 4.
(7) Setze $b := b + 2$. Gehe zu 2.

Bemerkung: Zu dem Verfahren gibt es eine gmp-Funktion 'hD' und eine Maple-Funktion 'hD(D)', mit der die nachfolgenden Beispiele gerechnet wurden.

Beispiel: Im folgenden Bild sind alle Paare $(|D|, h(D))$ für $|D| \leq 10^5$ eingezeichnet:



Ein wesentliches (nichttriviales) Resultat ist der folgende Satz:

Satz (Siegel). *Für negative Fundamentaldiskriminanten gilt*

$$\lim_{D\to-\infty} \frac{\ln h(D)}{\ln\sqrt{|D|}} = 1.$$

FOLGERUNG. *Für jedes $\varepsilon > 0$ gibt es ein D_ε, so daß für alle Fundamentaldiskriminanten D gilt:*

$$D < D_\varepsilon \quad \Longrightarrow \quad \sqrt{|D|}^{1-\varepsilon} < h(D) < \sqrt{|D|}^{1+\varepsilon}.$$

Beweis der Folgerung: Nach dem Satz von Siegel gibt es D_ε mit

$$D < D_\varepsilon \quad \Longrightarrow \quad 1 - \varepsilon < \frac{\ln h(D)}{\ln \sqrt{|D|}} < 1 + \varepsilon.$$

Dann folgt

$$(1-\varepsilon)\ln\sqrt{|D|} < \ln h(D) < (1+\varepsilon)\ln\sqrt{|D|} \quad \text{und} \quad \sqrt{|D|}^{1-\varepsilon} < h(D) < \sqrt{|D|}^{1+\varepsilon},$$

was zu zeigen war. ■

Mit analytischen Methoden erhält man folgende Abschätzung der Klassenzahl nach oben:

SATZ. *Für $D < -4$ gilt*

$$h(D) < \frac{1}{\pi}\sqrt{|D|}\ln|D|.$$

Gute Abschätzung in der anderen Richtung sind wesentlich schwieriger. Wir geben ein nichttriviales Beispiel (nach Cohen):

SATZ (Gross-Zagier). *Für $D < 0$ und $\mathrm{ggT}(D, 5077) = 1$ gilt*

$$h(D) > \frac{1}{55}\ln|D| \cdot \prod_{p|D}\left(1 - \frac{2\sqrt{p}}{p+1}\right).$$

Es ist ein schwieriges Problem, zu einem gegebenen h alle negativen Diskriminanten D mit $h(D) = h$ aufzulisten.
Experimentell kommt man schnell auf Vermutungen. In der nachfolgenden Tabelle sind alle D's mit $h(D) = 1, 2, 3, 4, 5$ und $|D| < 10^5$ aufgelistet:

h	Diskriminanten D mit $h(D) = h$ und $-10^5 < D < 0$
1	$-3, -4, -7, -8, -11, -12, -16, -19, -27, -28, -43, -67, -163$
2	$-15, -20, -24, -32, -35, -36, -40, -48, -51, -52, -60, -64, -72, -75, -88, -91, -99,$ $-100, -112, -115, -123, -147, -148, -187, -232, -235, -267, -403, -427$
3	$-23, -31, -44, -59, -76, -83, -92, -107, -108, -124, -139, -172, -211, -243, -268, -283,$ $-307, -331, -379, -499, -547, -643, -652, -883, -907$
4	$-39, -55, -56, -63, -68, -80, -84, -96, -120, -128, -132, -136, -144, -155, -156, -160,$ $-168, -171, -180, -184, -192, -195, -196, -203, -208, -219, -220, -228, -240, -252, -256,$ $-259, -275, -280, -288, -291, -292, -312, -315, -323, -328, -340, -352, -355, -363, -372,$ $-387, -388, -400, -408, -435, -448, -475, -483, -507, -520, -532, -555, -568, -592, -595,$ $-603, -627, -667, -708, -715, -723, -760, -763, -772, -795, -928, -955, -1003, -1012,$ $-1027, -1227, -1243, -1387, -1411, -1435, -1467, -1507, -1555$
5	$-47, -79, -103, -127, -131, -179, -188, -227, -316, -347, -412, -443, -508, -523, -571,$ $-619, -683, -691, -739, -787, -947, -1051, -1123, -1723, -1747, -1867, -2203, -2347,$ -2683

Bewiesen wurde für $h = 1, 2, 3, 4$, daß obige Diskriminanten alle sind.

Wir wollen jetzt sehen, wie schnell man die Klassenzahl $h(D)$ mit obigem Verfahren berechnen kann.

Beispiel: Mit der gmp-Funktion 'hD' erhalten wir folgende Ergebnisse:

D	$h(D)$	Zeit
-38227	24	0 sec
-38228	52	0 sec
-152915	128	0 sec
-152916	128	0 sec
-611667	128	0 sec
-611668	120	0 sec
-2446675	240	1 sec
-2446676	824	0 sec
-9786707	656	0 sec
-9786708	820	1 sec
-39146835	1632	1 sec
-39146836	1696	2 sec

D	$h(D)$	Zeit
-156587347	1048	8 sec
-156587348	3550	7 sec
-626349395	8480	29 sec
-626349396	10600	29 sec
-2505397587	9728	117 sec
-2505397588	7486	116 sec
-10021590355	20800	465 sec
-10021590356	54264	466 sec
-40086361427	53380	1854 sec
-40086361428	49000	1855 sec
-160345445715	81920	7417 sec
-160345445716	104520	7418 sec

Überlegung: Wir wollen sehen, wieviele Schritte man zur Berechnung von $h(D)$ braucht, wenn man nach obigem Verfahren die reduzierten Moduln mit Diskriminante D zählt. Wir lassen b und a alles ganzen Zahlen durchlaufen mit den Bedingungen

$$0 \le b \le \sqrt{\frac{|D|}{3}}, \quad b \equiv D \bmod 2, \quad b \le a \le \sqrt{\frac{b^2+|D|}{4}}.$$

Also ist die Schrittzahl ungefähr

$$\frac{1}{2}\int_{b=0}^{\sqrt{\frac{|D|}{3}}} \int_{a=b}^{\sqrt{\frac{b^2+|D|}{4}}} dadb = \frac{\ln 3}{16}|D|.$$

Beispiel: Mit Maple erhalten wir folgende Ergebnisse:

D	$h(D)$	Maple-Rechenzeit	Zeit$/\lvert D\rvert$ in sec
-1903	22	.011	5.780347e-06
-3160	16	0.000	0.000000e-01
-5531	23	.019	3.435184e-06
-11343	72	.021	1.851362e-06
-22131	38	.039	1.762234e-06
-65520	80	.120	1.831502e-06
-76780	48	.110	1.432665e-06
-220651	76	.351	1.590747e-06
-507536	504	.800	1.576243e-06
-903000	240	1.379	1.527132e-06
-1403792	544	2.160	1.538689e-06
-2628487	645	4.000	1.521788e-06
-7053523	306	10.631	1.507190e-06
-9296039	2986	14.059	1.512365e-06
-27537971	2037	42.030	1.526256e-06
-36669503	5212	56.100	1.529882e-06
-72499523	2424	111.810	1.542217e-06
-190577471	16184	298.610	1.566869e-06
-394785879	19140	632.229	1.601448e-06
-615874840	6400	992.781	1.611985e-06
-1144599163	3008	1872.409	1.635864e-06

Es ist klar, daß man $h(D)$ für große D auf diese Weise nicht berechnen kann. Es gibt allerdings auch noch andere Ansätze:

Analytische Klassenzahlformel: Es gilt

$$\frac{2\pi h(D)}{\#R_D^*\sqrt{|D|}} = \sum_{n\geq 1}\frac{\left(\frac{D}{n}\right)}{n} = \prod_p \frac{1}{1-\frac{\left(\frac{D}{p}\right)}{p}}$$

mit $\#R_D^* = 2$ für $D < -4$. Die Konvergenz der Reihe bzw. des Produkts ist schlecht. Man kann aber dennoch damit versuchen $h(D)$ näherungsweise zu bestimmen, was zuerst Shanks vorgeschlagen hat. Wir geben hier nur zwei Beispiele.

Beispiel: Wir wählen $D = -163$ und berechnen mit Maple (100-stellige Genauigkeit) die Produkte

$$\frac{\sqrt{|D|}}{\pi}\prod_{i=1}^{j}\frac{1}{1-\frac{\left(\frac{D}{p_i}\right)}{p_i}}.$$

j	$\frac{1}{\pi}\sqrt{\lvert D\rvert}\prod_{1<i<j}\cdots$
1	2.709272
2	2.031954
3	1.693295
4	1.481633
5	1.358164
6	1.261152
7	1.191088
8	1.131534
9	1.084387

j	$\frac{1}{\pi}\sqrt{\lvert D\rvert}\prod_{1<i<j}\cdots$
10	1.048240
20	1.079573
30	1.033170
40	1.023696
50	1.023341
60	1.006984
70	0.994190
80	1.004726
90	1.004876
100	1.000844

Bekanntlich ist $h(-163) = 1$.

Beispiel: Wir wählen $D = -160345445716$ und berechnen mit Maple (100-stellige Genauigkeit) die Produkte

$$\frac{\sqrt{|D|}}{\pi}\prod_{i=1}^{j}\frac{1}{1-\frac{\left(\frac{D}{p_i}\right)}{p_i}}.$$

j	$\frac{1}{\pi}\sqrt{\lvert D\rvert}\prod_{1<i<j}\cdots$
1	127461.328848
2	95595.996636
3	119494.995795
4	104558.121320
5	95844.944544
6	103832.023256
7	110321.524709
8	104805.448474
9	100438.554787
10	104025.646030
100	106290.569384

j	$\frac{1}{\pi}\sqrt{\lvert D\rvert}\prod_{1<i<j}\cdots$
10000	104492.217119
20000	104546.825881
30000	104557.745672
40000	104556.844573
50000	104551.693642
60000	104532.883019
70000	104525.060476
80000	104534.967509
90000	104534.646773
100000	104534.161666
110000	104532.757458
120000	104539.577927
130000	104542.622131
132000	104542.135151

(Mir ist nicht klar, wie weit man sich auf die angegebenen Näherungswerte verlassen kann.) Hier hatten wir schon früher berechnet: $h(D) = 104520$.

Shanks baby-step-giant-step-Methode: Wir skizzieren nur die Grundidee: Sei G eine endliche Gruppe, wobei für die Gruppenordnung $\#G$ eine Abschätzung $\#G \leq B$ bekannt ist. Sei $q = \lceil\sqrt{B}\rceil$.

- Wähle $g \in G$, setze $g_1 = g^{-q}$ und erzeuge zwei Mengen/Listen

$$\{g_1^a : 1 \le a \le q\} \quad \text{und} \quad \{g^b : 0 \le b \le q-1\}.$$

  Die Listen werden sortiert, was mit 'quicksort' oder 'heapsort' schnell geht, und auf gemeinsame Elemente hin untersucht. Dies liefert Beziehungen der Gestalt

$$g_1^a = g^b, \quad \text{also} \quad g^{aq+b} = 1,$$

  und damit $\operatorname{ord}(g) | aq+b$. (Man sieht auch sofort, daß die Listen/Mengen gemeinsame Elemente haben falls für $\operatorname{ord}(g)$ gilt $\lceil \sqrt{B} \rceil \le \operatorname{ord}(g) \le B$.)
- Kann man die Zahl $aq+b$ faktorisieren, so kann man auch $\operatorname{ord}(g)$ bestimmen, man muß im Prinzip nur alle Teiler von $aq+b$ durchprobieren.
- Wegen $\operatorname{ord}(g) | \#G$ erhält man damit auch Informationen über die Gruppenordnung.
- Wendet man diese Ideen auf die Berechnung von $h(D)$ an, stellt man fest, daß die Schrittzahl wie $|D|^{\frac{1}{4}+\varepsilon}$ wächst, allerdings ist auch der Speicherbedarf in dieser Größenordnung.

Beispiel: $D = -13579$. Wir wählen versuchsweise $B = \lfloor \sqrt{|D|} \rfloor = 116$ und $q = \lceil \sqrt{B} \rceil = 11$. Als Element von $\mathrm{H}(D)$ wählen wir $G = \mathbf{Z} \cdot 5 + \mathbf{Z} \cdot \frac{1+\sqrt{-13579}}{2}$. Dann ist $G_1 = G^{-q} = \mathbf{Z} \cdot 35 + \mathbf{Z} \cdot \frac{1+\sqrt{-13579}}{2}$. Wir bilden nun die zwei Listen, wobei $[a, b, D]$ abkürzend für $\mathbf{Z} \cdot a + \mathbf{Z} \cdot \frac{b+\sqrt{D}}{2}$ steht:

$G_1^1 = [35, 1, -13579]$
$G_1^2 = [53, 25, -13579]$
$G_1^3 = [19, -5, -13579]$
$G_1^4 = [7, -1, -13579]$
$G_1^5 = [5, 1, -13579]$
$G_1^6 = [47, 45, -13579]$
$G_1^7 = [29, -15, -13579]$
$G_1^8 = [49, 27, -13579]$
$G_1^9 = [35, -29, -13579]$
$G_1^{10} = [25, 11, -13579]$
$G_1^{11} = [17, 15, -13579]$

$G^0 = [1, 1, -13579]$
$G^1 = [5, 1, -13579]$
$G^2 = [25, 11, -13579]$
$G^3 = [43, 3, -13579]$
$G^4 = [49, -27, -13579]$
$G^5 = [19, 5, -13579]$
$G^6 = [53, 25, -13579]$
$G^7 = [29, -15, -13579]$
$G^8 = [41, -19, -13579]$
$G^9 = [17, -15, -13579]$
$G^{10} = [47, -45, -13579]$

Man sieht: $G_1^5 = G^1$, also $1 = G^{5 \cdot 11+1} = G^{56}$ und damit $\operatorname{ord}(G) | 56$. Man sieht auch noch die Relation $G_1^{10} = G^2$, die aber aus der ersten folgt. $G_1^2 = G^6$ liefert $1 = G^{2 \cdot 11+6} = G^{28}$, also $\operatorname{ord}(G) | 28$. Dann $G_1^7 = G^7$, also $1 = G^{7 \cdot 11+7} = G^{84}$. Es folgt $\operatorname{ord}(G) | 28$. Man testet nun leicht, daß tatsächlich $\operatorname{ord}(G) = 28$ gilt. (Durch Zählen der reduzierten Moduln sieht man $h(D) = 28$.)

Bemerkung: Es gibt noch andere Methoden, die Klassenzahl $h(D)$ zu bestimmen. Keine davon ist aber im Allgemeinfall schnell und für große D praktikabel, was auch weiter unten nochmals klar werden wird.

Der folgende Satz zeigt eine Verbindung von Diskriminante zu Fundamentaldiskriminante.

SATZ. *Ist D imaginärquadratische Diskriminante, D_1 die zugehörige Fundamentaldiskriminante und $D = f^2 D_1$, so gilt*

$$\frac{h(D)}{\#R_D^*} = \frac{h(D_1)}{\#R_{D_1}^*} f \prod_{p|f} \left(1 - \frac{\left(\frac{D_1}{p}\right)}{p}\right).$$

(Dabei ist $\#R_{-3}^* = 6$, $\#R_{-4}^* = 4$, $\#R_D^* = 2$ für $D < -4$.)

Bemerkungen:

(1) Der Satz zeigt, daß man sich für die Berechnung von $h(D)$ auf Fundamentaldiskriminanten zurückziehen kann, wenn man D faktorisieren bzw. die zu D gehörige Fundamentaldiskriminante bestimmen kann. Die nachfolgende Bemerkung macht den Vorteil dieser Methode deutlich.

(2) Ist D eine Fundamentaldiskriminante und $D = b^2 - 4ac$, so gilt bereits $\mathrm{ggT}(a,b,c) = 1$. Denn ist $a = d\widetilde{a}$, $b = d\widetilde{b}$, $c = d\widetilde{c}$, so folgt

$$D = b^2 - 4ac = d^2(\widetilde{b}^2 - 4\widetilde{a}\widetilde{c}).$$

Da auch $\widetilde{b}^2 - 4\widetilde{a}\widetilde{c}$ eine Diskriminante ist, muß $d = 1$ gelten. Insbesondere impliziert dies, daß man nicht testen muß, ob $\mathrm{ggT}(a,b,c) = 1$ gilt, wenn man reduzierte Moduln mit einer Fundamentaldiskriminante zählt.

5. Rechnen in der Klassengruppe $\mathrm{H}(D)$

Die Elemente der Klassengruppe $\mathrm{H}(D)$ werden eindeutig durch die Elemente von $\mathrm{Red}(D)$ repräsentiert. Durch den Reduktionsalgorithmus können wir für einen Modul $M \in \mathrm{Mod}(D)$ schnell den reduzierten Repräsentanten in der Klasse $\overline{M}$ bestimmen. Wir stellen im folgenden die Algorithmen für Multiplikation, Quadrieren, Potenzieren nochmals zusammen.

Multiplikation: Gegeben sind $M_1, M_2 \in \mathrm{Mod}(D)$ durch

$$M_1 = \mathbf{Z}\cdot a_1 + \mathbf{Z}\cdot\frac{b_1+\sqrt{D}}{2}, \quad M_2 = \mathbf{Z}\cdot a_2 + \mathbf{Z}\cdot\frac{b_2+\sqrt{D}}{2}.$$

Berechnet wird ein Repräsentant von $\overline{M_1}\cdot\overline{M_2}$.

(1) Berechne mit dem erweiterten euklidischen Algorithmus $a_{12} \in \mathbf{N}$, $x, y \in \mathbf{Z}$ mit

$$a_{12} = \mathrm{ggT}(a_1, a_2) = a_1 x + a_2 y.$$

(2) Berechne

$$a := a_1 a_2, \quad b := a_1 b_2 x + a_2 b_1 y.$$

(3) Im Fall $a_{12} > 1$:

(a) Berechne mit dem erweiterten euklidischen Algorithmus $A \in \mathbf{N}$, $u, v \in \mathbf{Z}$ mit

$$A = \mathrm{ggT}(a_{12}, \frac{b_1+b_2}{2}) = a_{12}u + \frac{b_1+b_2}{2}v.$$

(b) Berechne

$$b := bu + \frac{b_1 b_2 + D}{2}v.$$

(c) Ist $A > 1$, berechne

$$a := \frac{a}{A^2}, \quad b := \frac{b}{A}.$$

(4) Berechne

$$b := b \bmod 2a, \text{ ist } b > a, \text{ setze } b := b - 2a.$$

(5) Reduziere $\mathbf{Z}\cdot a + \mathbf{Z}\cdot\frac{b+\sqrt{D}}{2}$ und gib das Ergebnis aus.

Beweis: Wir hatten folgende Formel für die Multiplikation von Moduln bewiesen: Bestimmt man $\widetilde{x}, \widetilde{y}, z, A, a, b \in \mathbf{Z}$ mit

$$\begin{aligned} A &= \mathrm{ggT}(a_1, a_2, \frac{b_1+b_2}{2}) = a_1\widetilde{x} + a_2\widetilde{x} + \frac{b_1+b_2}{2}z, \\ a &= \frac{a_1 a_2}{A^2}, \\ b &= \frac{1}{A}(a_1 b_2 \widetilde{x} + a_2 b_1 \widetilde{y} + \frac{b_1 b_2 + D}{2}z), \end{aligned}$$

so gilt:

$$M_1 M_2 = A(\mathbf{Z}\cdot a + \mathbf{Z}\cdot\frac{b+\sqrt{D}}{2}).$$

In unserem Fall: Ist $a_{12} = 1$, so folgt die Behauptung mit $A = a_{12} = 1$, $x = \widetilde{x}$, $y = \widetilde{y}$, $z = 0$. Im allgemeinen Fall ist $\widetilde{x} = xu$, $\widetilde{y} = yu$, $z = v$. ■

Als Spezialfall der Multiplikation erhält man das Quadrieren:

Quadrieren: Gegeben sei $M = \mathbf{Z}\cdot a + \mathbf{Z}\cdot\frac{b+\sqrt{D}}{2}$. Berechnet wird $\overline{M}^2$ in $\mathrm{H}(D)$.

(1) Berechne mit dem erweiterten euklidischen Algorithmus $\widetilde{A} \in \mathbf{N}$, $x, y \in \mathbf{Z}$ mit
$$\widetilde{A} = \mathrm{ggT}(a, b) = ax + by.$$
(2) Berechne
$$\widetilde{a} = a^2, \quad \widetilde{b} = abx + \frac{b^2 + D}{2} y.$$
(3) Ist $\widetilde{A} > 1$, setze
$$\widetilde{a} = \frac{1}{A^2}\widetilde{a}, \quad \widetilde{b} := \frac{1}{A}\widetilde{b}.$$
(4) $\widetilde{b} := \widetilde{b} \bmod 2\widetilde{a}$ mit $-\widetilde{a} < \widetilde{b} \leq \widetilde{a}$.
(5) Reduziere $\mathbf{Z} \cdot \widetilde{a} + \mathbf{Z} \cdot \frac{\widetilde{b}+\sqrt{D}}{2}$ und gib das Ergebnis aus.

Multiplizieren und Quadrieren verwendet man mit der square-and-multiply-Methode beim Potenzieren:

Potenzieren: Zu $C \in \mathrm{H}(D)$ und $k \in \mathbf{N}_0$ wird $C^k \in \mathrm{H}(D)$ berechnet.

(1) Setze
$$X = C, \quad Y = \begin{cases} \mathbf{Z} + Z \cdot \frac{(D \bmod 2)+\sqrt{D}}{2} & \text{für } k \equiv 0 \bmod 2, \\ C & \text{für } k \equiv 1 \bmod 2 \end{cases}.$$
(2) Solange $k > 1$ ist, führe folgendes aus:
$$k := \lfloor \frac{k}{2} \rfloor, X := X \cdot X,$$
gilt $k \equiv 1 \bmod 2$, so $Y := X \cdot Y$.
(3) Gib Y als Ergebnis aus.

Der Vollständigkeit halber erinnern wir noch an die Inversenbildung: $\overline{M}^{-1} = \overline{M'}$.

Bemerkung: Zu den obigen Operationen haben wir Maple- und gmp-Funktionen geschrieben, die in den nachfolgenden Beispielen verwendet werden.

6. Die 2-Torsion der Klassengruppe H(D)

Für eine (multiplikativ geschriebene) abelsche Gruppe A nennt man
$$A_n = \{a \in A : a^n = 1\}$$
die Untergruppe der n-Torsionselemente von A.

Wir wollen die 2-Torsion der Klassengruppe H(D) anschauen.

LEMMA. *Für* $M = \mathbf{Z} \cdot a + \mathbf{Z} \cdot \frac{b+\sqrt{D}}{2} \in \mathrm{Red}(D)$ *mit* $D = b^2 - 4ac$, $\mathrm{ggT}(a, b, c) = 1$ *gilt:*
$$\overline{M} \in \mathrm{H}(D)_2 \iff \overline{M}^2 = \overline{R_D} \text{ in } \mathrm{H}(D) \iff b = 0 \text{ oder } a = b \text{ oder } a = c.$$

Beweis: Wir haben
$$\overline{M} \in \mathrm{H}(D)_2 \iff \overline{M}^2 = 1 \iff \overline{M} = \overline{M}^{-1} \iff \overline{M} = \overline{M'} \iff M \sim M'.$$
Mit $M' = \mathbf{Z} \cdot a + \mathbf{Z} \cdot \frac{-b+\sqrt{D}}{2}$ unterscheiden wir ein paar Fälle:

- Ist $-a < b < a$, $b \neq 0$ und $a < c$, so ist auch M' reduziert und $M' \neq M$, insbesondere folgt $M' \not\sim M$ und damit $\overline{M}^2 \neq 1$.
- Ist $b = 0$, so ist $M' = M$, also $\overline{M}^2 = 1$.
- Ist $b = a$, so ist $M' = M$, also $\overline{M}^2 = 1$.
- Ist $a = c$, so ist $M' \sim M$, also $\overline{M}^2 = 1$,

woraus sofort die Behauptung folgt. ■

Bemerkung: Ist $M = \mathbf{Z} \cdot a + \mathbf{Z} \cdot \frac{b+\sqrt{D}}{2} \in \mathrm{Red}(D)$ mit $D = b^2 - 4ac$, $\mathrm{ggT}(a,b,c) = 1$ und $\overline{M}^2 = 1$, so folgt mit dem vorangegangenen Lemma $b = 0$ oder $a = b$ oder $a = c$, und somit

$$D = \begin{cases} -4ac = -4 \cdot a \cdot c & \text{im Fall } b = 0, \\ a(a-4c) = -a \cdot (4c-a) & \text{im Fall } a = b, \\ (b-2a)(b+2a) = -(2a-b) \cdot (2a+b) & \text{im Fall } a = c, \end{cases}$$

also erhält man eine (eventuell triviale) Faktorzerlegung der Diskriminante.

Beispiel: $D = -420 = -4 \cdot 3 \cdot 5 \cdot 7$. Wir listen die zu den Elementen aus $\mathrm{H}(D)_2$ gehörigen Tripel (a,b,c) auf zusammen dem den Faktorzerlegungen der Diskriminante.

(a,b,c)	Faktorzerlegung der Diskriminante
$(1,0,105)$	$D = -4 \cdot a \cdot c = -4 \cdot 1 \cdot 105$
$(5,0,21)$	$D = -4 \cdot a \cdot c = -4 \cdot 5 \cdot 21$
$(7,0,15)$	$D = -4 \cdot a \cdot c = -4 \cdot 7 \cdot 15$
$(3,0,35)$	$D = -4 \cdot a \cdot c = -4 \cdot 3 \cdot 35$
$(10,10,13)$	$D = -a \cdot (4c-a) = -10 \cdot 42$
$(6,6,19)$	$D = -a \cdot (4c-a) = -6 \cdot 70$
$(2,2,53)$	$D = -a \cdot (4c-a) = -2 \cdot 210$
$(11,8,11)$	$D = -(2a-b) \cdot (2a+b) = -14 \cdot 30$

Bemerkung: Die vorangegangenen Bemerkungen führen zu folgender Faktorisierungsidee: Sei N eine (große) zusammengesetzte Zahl (ohne kleine Teiler). Wähle eine kleine natürliche Zahl m, so daß $D = -mN$ eine imaginärquadratische Diskriminante ist, z.B. $m = 4$ für $N \equiv 1 \bmod 4$ und $m = 1$ für $N \equiv 3 \bmod 4$. Konstruiere Elemente aus $\mathrm{H}(D)_2$ und teste, ob die zugehörige Faktorisierung der Diskriminante eine (nichttriviale) Faktorisierung von N liefert. Damit dies sinnvoll ist, sollte man sicherstellen, daß es Elemente in $\mathrm{H}(D)_2$ gibt, die zu einer nichttrivialen Faktorisierung führen. Dies geschieht im folgenden Lemma.

LEMMA. *Sei N eine natürliche Zahl, $D = -mN$, p eine Primzahl mit*

$$p \mid N, \quad p^2 \nmid N, \quad p \nmid m, \quad p < \sqrt{N}.$$

Sei

$$(a,b,c) = \begin{cases} (p, 0, \frac{m}{4}\frac{N}{p}) & \text{im Fall } m \equiv 0 \bmod 2, \\ (p, p, \frac{1}{4}(p + \frac{mN}{p}) & \text{im Fall } m \equiv 1 \bmod 2, \quad p < \sqrt{\frac{mN}{3}}, \\ (\frac{1}{4}(p+\frac{mN}{p}), \frac{1}{2}(\frac{mN}{p} - p), \frac{1}{4}(p+\frac{mN}{p})) & \text{im Fall } m \equiv 1 \bmod 2, \quad \sqrt{\frac{mN}{3}} < p < \sqrt{N}. \end{cases}$$

Dann ist $D = b^2 - 4ac$, $\mathrm{ggT}(a,b,c) = 1$ und $M = \mathbf{Z} \cdot a + \mathbf{Z} \cdot \frac{b+\sqrt{D}}{2} \in \mathrm{Red}(D)$ mit $\overline{M}^2 = 1$ und

$$a = p \quad bzw. \quad a = p \quad bzw. \quad 2a - b = p.$$

Beweis:

(1) Wir wollen einen reduzierten Modul mit $(a,b,c) = (p,0,c)$ und $D = -4pc$ konstruieren. Dazu muß man $c = \frac{m}{4}\frac{N}{p}$ wählen. Nach Voraussetzung gilt dann $\mathrm{ggT}(p,c) = 1$. Die Reduktionsbedingung $a \le c$ ist äquivalent zu $p^2 \le \frac{mN}{4}$, hier also wegen $4|m$ und der Voraussetzung $p < \sqrt{N}$ erfüllt.

(2) Wir wollen einen reduzierten Modul mit $(a,b,c) = (a,a,c) = (p,p,c)$ konstruieren und haben daher die Bedingung

$$-mN = D = p(p-4c),$$

also

$$c = \frac{1}{4}(p + \frac{mN}{p}).$$

Wir betrachten die Bedingung $p \le c$:

$$p \le c \iff p \le \frac{1}{4}(p + \frac{mN}{p}) \iff 4p \le p + \frac{mN}{p} \iff 3p^2 \le mN$$
$$\iff p \le \sqrt{\frac{mN}{3}},$$

was die Behauptung zeigt.

(3) Wir setzen an $a = c$ und

$$D = -(2a - b)(2a + b) = -p \cdot m\frac{N}{p}$$
$$2a - b = p, \quad 2a + b = \frac{mN}{p}.$$

Dies führt zu

$$a = \frac{mN + p^2}{4p}, \quad b = \frac{mN - p^2}{2p}.$$

Die Bedingung $b \ge 0$ ist wegen $p \le \sqrt{N}$ erfüllt. Für die Bedingung $b \le a$ haben wir die Äquivalenzen

$$b \le a \iff \frac{mN - p^2}{2p} \le \frac{mN + p^2}{4p} \iff 2mN - 2p^2 \le mN + p^2 \iff$$
$$\iff mN \le 3p^2 \iff \sqrt{\frac{mN}{3}} \le p,$$

was schließlich die Behauptung beweist. ■

Bemerkung: Das Lemma zeigt, daß es zu jedem Primteiler p von N wie im Lemma ein Element aus $\mathrm{N}(D)_2$ gibt, mit dem man p erhält.

Beispiele:

(1) Wir betrachten $N = 19805322830671970989$ und wählen $D = -4 \cdot N = -79221291322687883956$. Die zu $\mathrm{H}(D)_2$ gehörigen Tripel (a, b, c) sind

$$(1, 0, 19805322830671970989), (3146849437, 0, 6293698897),$$
$$(2, 2, 9902661415335985495), (4720274167, 3146849460, 4720274167),$$

was zur Primfaktorzerlegung

$$N = 3146849437 \cdot 6293698897$$

führt.

(2) Für $N = 29707984301077821631$ wählen wir $D = -N = -29707984301077821631$ und erhalten die Tripel

$$(3146849437, 3146849437, 3146849450), (1, 1, 7426996075269455408)$$

und die Primfaktorzerlegung

$$N = 3146849437 \cdot 9440548363.$$

(3) Für $N = 29707984036742468923$ sei $D = -N = -29707984036742468923$. Als Tripel finden wir

$$(1, 1, 7426996009185617231), (3146849429, 3146849421, 3146849429),$$

was die Primfaktorzerlegung

$$N = 3146849437 \cdot 9440548279$$

liefert.

Das folgende Lemma beschreibt in einem Spezialfall die ganze Gruppe $\mathrm{H}(D)_2$.

LEMMA. *Sei $D = -pq$ mit verschiedenen ungeraden Primzahlen p und q, o.E. $p < q$. Dann gibt es genau einen reduzierten Modul $M = \mathbf{Z} \cdot a + \mathbf{Z} \cdot \frac{b+\sqrt{D}}{2} \in \mathrm{Red}(D)$, $D = b^2 - 4ac$ mit $\overline{M}^2 = \overline{R_D}$ und $M \neq R_D$, und zwar ist*

$$(a,b,c) = \begin{cases} (p, p, \frac{p+q}{4}) & \text{im Fall } p < 3p < q, \\ (\frac{p+q}{4}, \frac{q-p}{2}, \frac{p+q}{4}) & \text{im Fall } p < q < 3p. \end{cases}$$

Der Beweis verläuft genau wie beim vorangegangenen Lemma.
Wir geben den folgenden Satz ohne Beweis an:

SATZ. *Sei $D = -p_1^{e_1} p_2^{e_2} \dots p_r^{e_r}$ eine Fundamentaldiskriminante. (Dies impliziert: $p_i \neq 2 \Rightarrow e_i = 1$ und $p_i = 2 \Rightarrow e_i \in \{0,2,3\}$.) Dann gilt $\#\mathrm{H}(D)_2 = 2^{r-1}$.*

7. Faktorisierung bei Kenntnis von $h(D)$

Wir wollen sehen, was man machen kann, wenn man Klassenzahlen $h(D)$ schnell berechnen kann.

Überlegungen:

(1) Sei N eine zusammengesetzte natürliche Zahl ohne kleine Teiler. Dann ist

$$D = \begin{cases} -N & \text{für } N \equiv 3 \bmod 4, \\ -4N \text{ für } N \equiv 1 \bmod 4 \end{cases}$$

eine imaginärquadratische Diskriminante.

(2) Wir nehmen an, wir können die Klassenzahl $h(D)$ schnell berechnen. Wir zerlegen dann $h(D) = 2^m u$ mit $u \equiv 1 \bmod 2$. Wir wissen, daß die 2-Torsionsgruppe $\mathrm{H}(D)_2$ nichttrivial ist und daß es Elemente darin gibt, die zu einer nichttrivialen Faktorisierung von N führen.

(3) Wähle $C \in \mathrm{H}(D)$ zufällig und berechne $C_1 = C^u$. Dann ist

$$C_1^{2^m} = (C^u)^{2^m} = 1,$$

also gibt es einen Index i mit $0 \leq i \leq m-1$, so daß gilt

$$C_1^{2^i} \neq 1, \quad C_1^{2^{i+1}} = 1, \quad \text{d.h.} \quad C_1^{2^i} \in \mathrm{H}(D)_2 \setminus \{1\}.$$

Mit diesem Element $C_1^{2^i}$ erhält man eine (eventuell triviale) Faktorisierung von N.

(4) Ist die erhaltene Faktorisierung trivial, wählt man ein anderes $C \in \mathrm{H}(D)$ zufällig. Da $\mathrm{H}(D)_2$ nicht zu groß ist, braucht man nur einige C zufällig zu wählen, bis man eine nichttriviale Faktorisierung erhält.

Eine algorithmische Version dieser Überlegungen folgt:

Algorithmus zur Faktorisierung von N bei Kenntnis von $h(-N)$ bzw. $h(-4N)$: Eingegeben wird eine zusammengesetzte natürliche Zahl N ohne kleine Teiler.

(1) Setze $D := -N$, falls $N \equiv 3 \bmod 4$ und $D := -4N$, falls $N \equiv 1 \bmod 4$.
(2) Berechne $h(D)$.
(3) Zerlege: $h(D) = 2^m u$ mit $2 \nmid u$.
(4) Wähle $C \in \mathrm{H}(D)$ zufällig.
(5) Berechne $C_1 := C^u$. (Die Ordnung von C_1 in nun eine 2-Potenz und teilt 2^m.) Ist $C_1 = 1$ gehe zu 4.
(6) Berechne $C_2 := C_1^2$. Ist $C_2 \neq 1$, setze $C_1 := C_2$ und gehe zurück zu 6. Andernfalls hat C_1 Ordnung 2, liefert also eine Faktorisierung von D. Liefert dies eine nichttriviale Faktorisierung von N, beende das Verfahren.

Bemerkung: Shanks hat wohl als erster obiges Faktorisierungsverfahren vorgeschlagen als Anwendung seiner Methoden zur Berechnung von Klassenzahlen $h(D)$.

Beispiel: Wir wollen $N = 12193264431633919465 63$ faktorisieren und setzen $D = -N$. Die Klassenzahl ist (wahrscheinlich) $h(D) = 5609655886 = 2u$ mit $u = 2804827943$. Wir wählen

$$C_1 = \mathbf{Z} \cdot 7 + \mathbf{Z} \cdot \frac{3 + \sqrt{D}}{2}, \quad C_2 = \mathbf{Z} \cdot 17 + \mathbf{Z} \cdot \frac{15 + \sqrt{D}}{2}$$

und berechnen

$$C_1^u = \mathbf{Z} \cdot 12345679937 + \mathbf{Z} \cdot \frac{12345679937 + \sqrt{D}}{2}, \quad C_2^u = \mathbf{Z} \cdot 1 + \mathbf{Z} \cdot \frac{1 + \sqrt{D}}{2}.$$

C_1^u hat Ordnung 2 und liefert die Primfaktorzerlegung

$$N = 12345679937 \cdot 98765434499.$$

Bemerkung: Das obige Verfahren zeigt, daß man N schnell faktorisieren kann, wenn man $h(-N)$ bzw. $h(-4N)$ schnell berechnen kann. Da bis jetzt kein schnelles Faktorisierungsverfahren bekannt ist, daß man für große negative D die Klassenzahl $h(D)$ praktisch nicht bestimmen kann.

8. Faktorisierung mit Klassengruppen H($-mN$)

Überlegungen:

(1) Wir wollen einen nichttrivialen Teiler einer zusammengesetzten natürlichen Zahl N (ohne kleine Teiler) finden. Das obige Verfahren funktioniert nur, wenn man $h(-N)$ bzw. $h(-4N)$ berechnen kann.

(2) Findet man eine (kleine) Zahl m, so daß man $h(-mN)$ berechnen kann, so kann man die Faktorisierung auch mit H($-mN$) durchführen, indem man sich wie oben nichttriviale Elemente in H$(-mN)_2$ konstruiert.

(3) Sei
$$h(-mN) = p_1^{e_1} p_2^{e_2} \dots p_r^{e_r}$$
die Primfaktorzerlegung von $h(-mN)$. Gilt
$$p_i^{e_i} \leq K$$
für eine natürliche Zahl K, so folgt
$$h(-mN) | \mathrm{kgV}(1..K),$$
insbesondere gilt also
$$C \in \mathrm{H}(-mN) \quad \Longrightarrow \quad C^{\mathrm{kgV}(1..K)} = 1 \text{ in } \mathrm{H}(-mN).$$
Die Klassenzahlen $h(-mN)$ sind in der Größenordnung von $\sqrt{N}$. Variiert man m, so ist zu erwarten, daß unter den Zahlen $h(-mN)$ auch glatte Zahlen auftreten, d.h. Zahlen, für die ein relativ kleines K mit $h(-mN) | \mathrm{kgV}(1..K)$ existiert.

(4) Daher erhält man folgende Idee: Man wählt K so, daß man $C^{\mathrm{kgV}(1..K)}$ noch gut berechnen kann, dann variiert man m so lange, bis man ein $C \in \mathrm{H}(-mN)$ (zufällig) gefunden hat mit $C^{\mathrm{kgV}(1..K)} = 1$.

(5) Hat man ein $C \in \mathrm{H}(D)$ mit $C^{\mathrm{kgV}(1..K)} = 1$, erhält man aber damit nur die triviale Faktorisierung, so sollte man m beibehalten und sich ein anderes $C \in \mathrm{H}(D)$ suchen, denn es ist dann wahrscheinlich, daß $h(-mN) | \mathrm{kgV}(1..K)$ gilt (oder fast gilt).

(6) Wir bemerken noch, daß gilt
$$\mathrm{kgV}(1..K) = \prod_{p \leq K} p^{\lfloor \frac{\ln K}{\ln p} \rfloor}.$$

Die vorangegangenen Überlegungen führen zu folgendem Algorithmus:

Algorithmus: Gegeben sei eine zusammengesetzte natürliche Zahl (ohne kleine Teiler).

(1) Wähle eine natürliche Zahl K, die nicht zu groß sein sollte.

(2) Erstelle eine Liste aller Primzahlen $\leq K$: $p_1 = 2$, $p_2 = 3$, $\dots$, p_r. Bestimme e_i maximal mit $p_i^{e_i} \leq K$, also $e_i = \lfloor \frac{\ln K}{\ln p_i} \rfloor$.

(3) Setze $m := 0$.

(4) Setze $m := m + 1$ und $D := -mN$. Ist D keine imaginärquadratische Diskriminante, d.h. $D \not\equiv 0 \bmod 4$ und $D \not\equiv 1 \bmod 4$, gehe zu 4.
(5) Setze $a := m$.
(6) Sei a die nächste Primzahl $> a$. Versuche einen Modul $M = \mathbf{Z} \cdot a + \mathbf{Z} \cdot \frac{b+\sqrt{D}}{2}$ mit Norm a zu konstruieren, d.h. die Gleichung $D = b^2 - 4ac$, $\mathrm{ggT}(a, b, c) = 1$, $0 \le b \le a$ zu lösen. Geht das nicht, gehe zu 6.
(7) Berechne in der Klassengruppe
$$\widetilde{M} = M^{\mathrm{kgV}(3,5,7,\ldots,2\lfloor \frac{K+1}{2} \rfloor - 1)} = M^{\prod_{i=2}^{r} p_i^{e_i}}.$$
(8) Setze $\widetilde{M}_0 := \widetilde{M}$ und $i := 0$.
(9) Setze $i := i + 1$ und $\widetilde{M}_i := \widetilde{M}_{i-1}^2$ in der Klassengruppe.
(10) Ist $\widetilde{M}_i = R_D$, so ist $\widetilde{M}_{i-1}$ ein 2-Torsionselement, mit dem man einen Faktorisierungsversuch unternehmen kann. Ist dieser erfolgreich, ist man fertig.
(11) Ist $i = e_1$, gehe zu 4, andernfalls zu 9.

Bemerkungen:

(1) Das obige Faktorisierungsverfahren wurde von Schnorr/Lenstra und unabhängig davon von Shanks/Pollard/Atkin/Rickert (SPAR) vorgeschlagen.
(2) Wir haben zu obigem Faktorisierungsverfahren eine gmp-Funktion 'hd_fac' geschrieben, mit der die nachfolgenden Beispiele gerechnet wurden.

Beispiel: Wir wollen $N = 12193263122374638001$ faktorisieren und wählen $K = 1000$. Wir erhalten folgende Versuche:

```
1. Versuch: Multiplikator=3, M=[13,5,-36579789367123914003]
kgV(3,5,7,9,...,999)*M=[1319140237,863392889,-36579789367123914003]
Quadrat: [108882127,86105103,-36579789367123914003]
Quadrat: [584267611,551969533,-36579789367123914003]
Quadrat: [375805183,-271277417,-36579789367123914003]
Quadrat: [2403490921,308533181,-36579789367123914003]
Quadrat: [2767935091,-623684455,-36579789367123914003]
Quadrat: [527783911,472692753,-36579789367123914003]
Quadrat: [2113845537,845339565,-36579789367123914003]
Quadrat: [2641323007,-1603588355,-36579789367123914003]
Quadrat: [1508730493,-992472719,-36579789367123914003]
Dies ist nicht 1, liefert also nichts.

33. Versuch: Multiplikator=67, M=[73,45,-816948629199100746067]
kgV(3,5,7,9,...,999)*M=[1,1,-816948629199100746067]
Quadrat: [1,1,-816948629199100746067]
Dies ist 1, liefert nur die triviale Faktorisierung, daher werden bei
gleichem Multiplikator andere M's versucht.

M=[89,5,-816948629199100746067]
kgV(3,5,7,9,...)*M=[67,67,-816948629199100746067]
Quadrat: [1,1,-816948629199100746067] (trivial)

M=[97,67,-816948629199100746067]
kgV(3,5,7,9,...)*M=[1101986263,814640163,-816948629199100746067]
Quadrat: [1234567891,1234567891,-816948629199100746067]
Quadrat: [1,1,-816948629199100746067]
Dies liefert einen nichttrivialen Teiler: 1234567891
```

Also erhält man die Primfaktorisierung

$$12193263122374638001 = 1234567891 \cdot 9876543211.$$

Beispiel: Wir wollen $N = 1057074580493026582727698843 81$ mit $K = 10000$ faktorisieren.

```
Multiplikator=52
M=[53,44,-549678781856373823018403398 7812]
kgV(3,5,7,9,...)*M=[948385547116357,-356006368833216,-549678781856373823018403398 7812]
Quadrat: [13,0,-549678781856373823018403398 7812]
Quadrat: [1,0,-549678781856373823018403398 7812]
M=[73,18,-549678781856373823018403398 7812]
kgV(3,5,7,9,...)*M=[582620772428377,9400734870426,-549678781856373823018403398 7812]
Quadrat: [13,0,-549678781856373823018403398 7812]
Quadrat: [1,0,-549678781856373823018403398 7812]
M=[101,16,-549678781856373823018403398 7812]
kgV(3,5,7,9,...)*M=[1,0,-549678781856373823018403398 7812]
Quadrat: [1,0,-549678781856373823018403398 7812]
M=[107,102,-549678781856373823018403398 7812]
kgV(3,5,7,9,...)*M=[694639324751134,680128474002534,-549678781856373823018403398 7812]
Quadrat: [217835178263837,0,-549678781856373823018403398 7812]
Quadrat: [1,0,-549678781856373823018403398 7812]
ggT=217835178263837
```

Damit ergibt sich die Primfaktorzerlegung

$$1057074580493026582727698843 81 = 217835178263837 \cdot 485263486328513.$$

Wir wollen nun noch an Beispielen sehen, welchen Einfluss der Parameter K hat.

Beispiel: Wir wollen

$$N = 1057074580493026582727698843 81 = 217835178263837 \cdot 485263486328513$$

faktorisieren.

K	Multiplikator	Teiler	Zeit
1000	1092	217835178263837	107 sec
2000	527	217835178263837	103 sec
3000	527	217835178263837	154 sec
4000	527	217835178263837	205 sec
5000	52	217835178263837	27 sec
6000	52	217835178263837	33 sec
7000	52	217835178263837	38 sec
8000	52	217835178263837	42 sec
9000	52	217835178263837	47 sec
10000	52	217835178263837	52 sec

Beispiele: Wir faktorisieren die nachfolgenden Zahlen N mit $K = 5000$. Die Zahlen sind 'zufällig' konstruiert mit einer Faktorisierung $N = pq$.

N	Multiplikator	Teiler	Zeit
664611256030932029991149705201	2980	922537519955053	1522 sec
256112604031810931418893914253	303	413651631798463	148 sec
136470722644397386869660777467	173	649586894087171	84 sec
142139171560267204905441854563	273	267005221791157	143 sec
381286984511903087964398016979999	152	9720416505188111	79 sec
340694217683190746303980593255553	483	3619150474321313	251 sec
389314483475500520616903431490 49	295	7143206002489871	156 sec
322054376529337510978447325827 49	1240	4210088479134359	661 sec
560909305991587066515727105393 5409	611	59720416505188133	340 sec
216540804548063550355563992705 6149	2280	73619150474321219	1278 sec
573739612347193181217333764181 9289	2395	67143206002489849	1358 sec
163780248428273969204397852924 4817	1239	67649586894087157	690 sec
510607331328473892464505087325180187	10968	593922537519954971	6755 sec
280775765495881953922556455524157447	1740	314686912080777253	1037 sec
39367539962664600442278315974145535 1	5	934721542472337679	3 sec
180574207116129518458884684839007667	881	343503006329051743	516 sec
43966667176494099684425149700499172741	?	?	?
26523531057020038844229821388725490307	2797	3073619150474321221	1774 sec
83287622493042868513153560055731229963	5997	9167143206002489887	3917 sec
27989879308657879254126896075149895569	13251	3067649586894087191	8891 sec
13500863217070568252507656089738985644 67	8029	75593922537519955067	5376 sec
15194665966480963893472879960482700325 47	11001	53073619150474321279	7508 sec
17075135803390629233212811711852486923 43	2392	19167143206002489869	1583 sec
39565321207567248175176995919536975670 11	75836	48022267005221791141	56479 sec

Die offenstehende Faktorisierung wurde mit $K = 10000$ erzielt:

Faktorisierung von N mit $K = 10000$	Multiplikator	Teiler	Zeit
43966667176494099684425149700499172741	3415	7859720416505188087	4646 sec

Für die letzte Zahl

$$N = 39565321207567248175176995919536975670 11 = 48022267005221791141 * 82389532346036556071$$

haben wir noch verschiedene Parameter K ausprobiert:

K	Multiplikator	Teiler	Zeit
10000	14492	48022267005221791141	20485 sec
20000	889	82389532346036556071	2523 sec
30000	889	82389532346036556071	3657 sec
40000	889	82389532346036556071	4872 sec
50000	889	82389532346036556071	6097 sec
100000	889	82389532346036556071	12204 sec

Bemerkung: Das Faktorisierungsverfahren funktioniert mit Multiplikator m, wenn $h(-mN)$ nur kleine Primteiler hat, d.h. wenn $h(-mN)$ glatt ist. Indem man sich anschaut, wieviele glatte Zahlen es in der Größenordnung $\sqrt{N} \approx h(-mN)$ gibt, wurde für die Wahl von K der Wert

$$K(N) = \lfloor e^{\frac{1}{2}\sqrt{\ln N \ln\ln N}} \rfloor$$

vorgeschlagen. Hier ist eine kleine Tabelle:

N	$K(N)$
10^{10}	70
10^{20}	765
10^{30}	5178
10^{40}	27040
10^{50}	119098
10^{60}	463630
10^{70}	1641126
10^{80}	5382469
10^{90}	16574471
10^{100}	48389443

Wir wollen nun noch abschließend an eine sogenannte $(p-1)$-Methode zum Faktorisieren erinnern, die in ähnlicher Weise wie obiges Faktorisierungsverfahren arbeitet.

LEMMA. *Sei N eine ungerade natürliche Zahl, p ein Primteiler von N mit*

$$p-1 = q_1^{e_1} \dots q_r^{e_r} \text{ (Primfaktorzerlegung).}$$

Ist K eine natürliche Zahl mit

$$q_1^{e_1} \le K, \quad \dots, \quad q_r^{e_r} \le K,$$

so folgt

$$p | \mathrm{ggT}(2^{\mathrm{kgV}(1..K)} - 1, N).$$

Beweis: Aus $q_i^{e_i} \le K$ folgt $p-1 | \mathrm{kgV}(1..K)$. Nach dem kleinen Satz von Fermat gilt $2^{p-1} \equiv 1 \bmod p$ und damit auch $2^{\mathrm{kgV}(1..K)} \equiv 1 \bmod p$. Dies liefert sofort

$$p | \mathrm{ggT}(2^{\mathrm{kgV}(1..K)} - 1, N),$$

wie behauptet. ■

Bemerkung: Man kann das Lemma benutzen um einen schnellen Faktorisierungsversuch zu machen, indem man testet, ob

$$1 < \mathrm{ggT}(2^{\mathrm{kgV}(1..K)} - 1, N) < N$$

gilt. Allerdings hat man im allgemeinen nur Aussicht auf Erfolg, falls N einen Primteiler p besitzt, für den $p-1$ glatt ist. Dies ist im allgemeinen sehr unwahrscheinlich.

Beispiele: Wir wollen die folgenden 40-stelligen Zahlen N zu faktorisieren versuchen mit $K = 10^6$, d.h. wir schauen, ob

$$\mathrm{ggT}(2^{\mathrm{kgV}(1..10^6)} - 1, N)$$

ein nichttrivialer Teiler von N ist.

(1) $N = 1350086321707056825250765608973898564467$. Nach 7 sec finden wir

$$1350086321707056825250765608973898564467 = 75593922537519955067 * 17859720416505188201.$$

$$\begin{aligned} 75593922537519955067 - 1 &= 2 \cdot 41 \cdot 127 \cdot 443 \cdot 907 \cdot 362717 \cdot 49807, \\ 17859720416505188201 - 1 &= 2^3 \cdot 5^2 \cdot 7 \cdot 39267307 \cdot 324874409. \end{aligned}$$

(2) $N = 151946659664809638934728799604827003254 7$. Der Faktorisierungsversuch hat (nach 7 sec) keinen Erfolg. Die Faktorisierung ist

$$N = 28629413651631798493 \cdot 53073619150474321279$$

und

$$\begin{aligned} 28629413651631798493 - 1 &= 2^2 \cdot 3 \cdot 1279 \cdot 1865351423744579, \\ 53073619150474321279 - 1 &= 2 \cdot 3 \cdot 11 \cdot 229 \cdot 1868183 \cdot 67369 \cdot 27901 \end{aligned}$$

9. IQ-Kryptographie

Wesentlich bei vielen 'klassischen' Public-Key-Kryptoverfahren ist, daß man schnell modulo einer natürlichen Zahl potenzieren kann, daß man aber z.B. diskrete Logarithmen oder Wurzeln nicht schnell bzw. praktisch überhaupt nicht berechnen kann.

Beispiele:

(1) Beim RSA-Verfahren verschlüsselt man mit Funktionen

$$f_{(N,e)} : \mathbf{Z}/N\mathbf{Z} \to \mathbf{Z}/N\mathbf{Z}, \quad x \mapsto x^e \bmod N,$$

wobei (N, e) öffentlich bekannt ist. Wichig ist, daß man $x^e \bmod N$ schnell berechnen kann, daß man aber praktisch keine e-ten Wurzeln ziehen kann, d.h. im allgemeinen kann man bei gegebenem y kein x mit $x^e \equiv y \bmod N$ in vernünftiger Zeit berechnen.

(2) Beim Diffie-Hellman-Schlüsselaustausch hat man Größen p, g, $g_a \equiv g^a \bmod p$, $g_b \equiv g^b \bmod p$. Dabei sind p, g_a, g_b (eventuell) öffentlich bekannt. Der geheime Schlüssel ist

$$g^{ab} \equiv g_a^b \equiv g_b^a \bmod p.$$

Damit der Schlüssel wirklich geheim bleibt, ist wichtig, daß man die Gleichung $g^x \equiv g_a \bmod p$ nicht lösen kann, d.h. daß man den diskreten Logarithmus von g_a zur Basis g (modulo p) nicht bestimmen kann.

Bei den angegebenen Beispielen steckt jeweils eine Gruppenstruktur im Hintergrund, nämlich $(\mathbf{Z}/N\mathbf{Z})^*$ oder $(\mathbf{Z}/p\mathbf{Z})^*$. Es ist nun naheliegend zu untersuchen, ob man diese kryptographischen Verfahren nicht auch mit anderen endlichen Gruppen G statt $(\mathbf{Z}/N\mathbf{Z})^*$ oder $(\mathbf{Z}/p\mathbf{Z})^*$ durchführen kann. Wichtig dabei ist, daß man in G schnell potenzieren kann, daß aber Wurzelziehen oder die Berechnung diskreter Logarithmen sehr schwierig ist. Für sogennante elliptische Kurven hat man auf diesem Weg kryptographische Verfahren entwickelt. Hier wollen wir an einigen Beispielen imaginärquadratische Klassengruppen kryptographisch nutzen.

Für die Kryptographie sind folgende Eigenschaften imaginärquadratischer Klassengruppen $\mathrm{H}(D)$ wichtig:

- Hat man $M_1, M_2 \in \mathrm{Mod}(D)$ gegeben, kann man schnell testen, ob $\overline{M_1} = \overline{M_2}$ in $\mathrm{H}(D)$ gilt.
- Man kann in $\mathrm{H}(D)$ schnell multiplizieren und potenzieren.
- Für große D kennt man kein effektives Verfahren um die Klassenzahl $h(D) = \#\mathrm{H}(D)$ auszurechnen.
- Wurzelziehen ist schwierig. (Kennt man die Gruppenordnung $h(D)$ und gilt $\mathrm{ggT}(e, h(D)) = 1$, wählt man d mit $ed \equiv 1 \bmod h(D)$, so ist e-tes Wurzelziehen das gleiche wie Potenzieren mit d. Es ist unklar, wie man Wurzeln ziehen kann, wenn man die Gruppenordnung nicht kennt.)
- Diskrete Logarithmen lassen sich nicht vernünftig berechnen. (Heuristisches Argument: Könnte man diskrete Logarithmen berechnen, so könnte man so vorgehen: Man wählt $G \in \mathrm{H}(D)$ zufällig und $n \in \mathbf{N}$ groß. Dann berechnet man den diskreten Logarithmus von G^n zur Basis G, d.h. ein x mit $0 \le x \le h(D)$ und $G^x = G^n$. Es folgt $\mathrm{ord}\,(G) | n - x$. Damit kommt man an $\mathrm{ord}\,(G)$ und somit auch an $h(D)$.)

Wir wollen das ElGamal-Kryptosystem auf imaginärquadratische Klassengruppen übertragen und erinneren dazu zunächst an das klassische Verschlüsselungsverfahren, das im 2. Kapitel vorgestellt wurde:

Das ElGamal-Kryptosystem zur Verschlüsselung von Daten:

(1) Man einigt sich darauf, wie man bei gegebener Primzahl p Text in eine Folge von Zahlen aus $\{0, 1, \ldots, p-1\} \simeq \mathbf{Z}/p\mathbf{Z}$ umwandelt und umgekehrt.

(2) Jeder Teilnehmer A
- wählt eine (große) Primzahl p_A,
- eine Zahl g_A mit $2 \le g_A \le p_A - 2$,
- eine (zufällige) Zahl e_A mit $2 \le e_A \le p_A - 2$
- und berechnet $f_A := g_A^{e_A} \bmod p_A$.

A's öffentlicher Schlüssel (p_A, g_A, f_A), sein privater Schlüssel (p_A, g_A, e_A).

(3) Wie kann ein Teilnehmer B eine Nachricht T verschlüsselt an A schicken?

(a) B besorgt sich den öffentlichen Schlüssel (p_A, g_A, f_A) von A.
(b) B wandelt die Nachricht T (nach dem vereinbarten Schema) in eine Folge von Zahlen a_i aus $\mathbf{Z}/p_A\mathbf{Z}$ um.
(c) Zu jeder solchen Zahl $a_i \in \mathbf{Z}/p_A\mathbf{Z}$ wählt sich B eine zufällige Zahl $z_i \in (\mathbf{Z}/p_A\mathbf{Z})^*$.
(d) B berechnet nun

$$b_i \equiv g_A^{z_i} \bmod p_A \quad \text{und} \quad c_i \equiv a_i \cdot f_A^{z_i} \bmod p_A$$

und schickt die Folge (b_i, c_i) an A.

(4) Wie erhält A aus dem verschlüsselten Paar (b_i, c_i) die ursprüngliche Zahl a_i zurück? Er berechnet einfach $b_i^{-e_A} \cdot c_i \bmod p_A$, denn es gilt

$$\frac{c_i}{b_i^{e_i}} \equiv \frac{a_i f_A^{z_i}}{g_A^{z_i e_i}} \equiv \frac{a_i g_A^{e_i z_i}}{g_A^{z_i e_i}} \equiv a_i \bmod p_A.$$

Wir hatten überlegt, daß das Verfahren sicher ist, solange man diskrete Logarithmen in $(\mathbf{Z}/p_A\mathbf{Z})^*$ zur Basis g_A nicht berechnen kann und solange der verwendete Zufallszahlengenerator gut arbeitet.

Überlegungen:

(1) Das 'klassische' ElGamal-Verschlüsselungsverfahren nutzt die Gruppenstruktur von $(\mathbf{Z}/p\mathbf{Z})^*$. Außerdem wird der zu verschlüsselnde Text in eine Zahlenfolge a_i umgewandelt, die man i.a. als Gruppenelemente $a_i \in (\mathbf{Z}/p_A\mathbf{Z})^*$ auffassen kann, um sie anschließend mit $f_A^{z_i}$ modulo p_A zu multiplizieren.
(2) Statt $(\mathbf{Z}/p_A)^*$ wollen wir jetzt imaginärquadratische Klassengruppen $\mathrm{H}(D)$ benutzen.
(3) Will man das ElGamal-Verfahren direkt übersetzen, müßte man Text in eine Folge von Elementen aus $\mathrm{H}(D)$ umwandeln. Es ist nicht klar, wie man das auf natürlichem Weg machen kann, denn die Elemente aus $\mathrm{H}(D)$ werden repräsentiert durch Moduln

$$\mathbf{Z} \cdot a + \mathbf{Z} \cdot \frac{b + \sqrt{D}}{2} \in \mathrm{Mod}(D) \text{ mit } D = b^2 - 4ac, \quad \mathrm{ggT}(a, b, c) = 1,$$

also durch ein Zahlenpaar (a, b), wobei aber nicht alle a's möglich sind.
(4) Eine andere Möglichkeit wird weiter unten behandelt, wobei Hashfunktionen benutzt werden. Dabei wird der Text nicht als Folge von Gruppenelementen geschrieben.

Erinnerung an Hashfunktionen: Eine n-Bit-Hashfunktion h berechnet zu einer beliebig langen Zeichenkette S eine Zahl mit $\leq n$ Bits, den sogenannten Hash-Wert $h(S)$. Für die nachfolgenden kryptographischen Anwendungen sollten folgende Bedingungen erfüllt sein, wir sprechen dann von einer kryptographischen Hashfunktion:

- Zu einer vorgegebenen Zeichenkette S läßt sich $h(S)$ schnell berechnen.
- Es ist praktisch unmöglich bzw. sehr schwer, zu w ein S mit $h(S) = w$ zu finden. (Einweg-Eigenschaft)
- Zu vorgegebenem S ist es praktisch unmöglich ein $S' \neq S$ mit $h(S) = h(S')$ zu finden. (Schwache Kollisionsresistenz)
- Es ist praktisch unmöglich zwei Zeichenketten $S_1 \neq S_2$ zu finden mit $h(S_1) = h(S_2)$. (Kollisionsresistenz)

Ein Beispiel ist die 128-Bit-Hashfunktion MD5 (message digest). So ist z.B.

```
MD5 ("Zahlentheorie") = 878e90dfbbc16953e32f4fe5a4e97382
```

IQ-ElGamal — ElGamal-Verschlüsselung mit imaginärquadratischen Klassengruppen:

(1) Man vereinbart eine kryptographische n-Bit-Hashfunktion f und wie sie auf Moduln $\mathbf{Z} \cdot a + \mathbf{Z} \cdot \frac{b+\sqrt{D}}{2}$ operieren soll, z.B. hänge die Dezimaldarstellungen von a und b aneinander und bilde den Hash-Wert dieser Zeichenkette.
Man vereinbart, wie man einen Text in eine Folge von n-Bit-Zeichenketten umwandelt.
(2) Jeder Teilnehmer A
- wählt eine imaginärquadratische Diskriminante D_A,
- einen reduzierten Modul $G_A \in \mathrm{Mod}(D_A)$,
- eine (zufällige) Zahl e_A mit $2 \leq e_A \leq \sqrt{|D_A|}$

- und berechnet $F_A = G_A^{e_A}$ in $\mathrm{H}(D)$.

A's öffentlicher Schlüssel ist (D_A, G_A, F_A), sein privater Schlüssel ist (D_A, G_A, e_A).

(3) Wie kann ein Teilnehmer B eine Nachricht T verschlüsselt an A schicken?
   (a) B besorgt sich den öffentlichen Schlüssel (D_A, G_A, F_A) von A.
   (b) B wandelt die Nachricht T (nach dem vereinbarten Schema) in eine Folge von n-Bit-Zeichenketten a_i um.
   (c) Zu jedem a_i wählt sich B eine Zufallszahl z_i mit $1 \le z_i \le \sqrt{|D_A|}$.
   (d) B berechnet in $\mathrm{H}(D)$ (repräsentiert durch reduzierte Moduln)

$$B_i = G_A^{z_i} \quad \text{und} \quad \widetilde{C}_i = F_A^{z_i}.$$

$$c_i = a_i \oplus f(\widetilde{C}_i),$$

   wobei $\oplus$ für die komponentenweise Addition modulo 2 steht (XOR). Die aus (B_i, c_i) bestehende Folge wird an A geschickt.

(4) Wie erhält A aus dem verschlüsselten Paar (B_i, c_i) die ursprüngliche Zahl a_i zurück? Wegen

$$B_i^{e_A} = (G_A^{z_i})^{e_A} = (G_A^{e_A})^{z_i} = F_A^{z_i} = \widetilde{C}_i$$

kann sich A den Modul $\widetilde{C}_i$ und wegen

$$a_i = c_i \oplus f(\widetilde{C}_i)$$

dann a_i berechnen.

Beispiel:

(1) Zur Konstruktion eines Schlüssels wählen wir zunächst eine Primzahl

$$p = 78461289346189273648123648126335523452 59$$

mit $p \equiv 3 \bmod 4$ und setzen dann $D = -p$. Als Basis-Modul wählen wir

$$G = [3, 1, -78461289346189273648123648126335523452 59]$$

und dann ($\lfloor\sqrt{|D|}\rfloor = 88578377353725143551$)

$$e = 24482733571884976033.$$

Mit

$$F = G^e = [5487200746003049467, 1899564277622157397, D]$$

ist dann (D, G, F) der öffentliche Schlüssel und (D, G, e) der private Schlüssel.

(2) Als Hashfunktion verwenden wir MD5, daher teilen wir den Text in Blöcke mit 128 Bits bzw. 16 Bytes auf. Wir wollen den Text

Algorithmische Zahlentheorie

verschlüsseln. Dies sind 28 Zeichen, die ascii-Werte in Hexadezimaldarstellung sind

$$416C676F726974686D6973636865205A61686C656E7468656F726965.$$

Wir bilden daraus zwei Blöcke mit 16 Bytes, wobei wir mit 0 die fehlenden Zeichen auffüllen:

$$\begin{aligned} a_1 &= 416C676F726974686D6973636865205A, \\ a_2 &= 61686C656E7468656F72696500000000. \end{aligned}$$

(3) Wir wählen nun zwei Zufallszahlen z_i und berechnen damit $B_i = G^{z_i}$ und $C_i = F^{z_i}$:

$$\begin{aligned} z_1 &= 20405182708966081046, \\ z_2 &= 78765650665739485434, \\ B_1 &= [14341632944477543883, 5399240213427182597, D], \\ C_1 &= [18001873890848523145, -5893305375058672149, D], \\ B_2 &= [39131824122806366515, 7563568923409676051, D], \\ C_2 &= [46464428094034109885, -36216068594728755239, D], \end{aligned}$$

Nun bilden wir die Hashwerte (in Hexadezimaldarstellung)

$$\begin{aligned} f(C_1) &= \text{MD5 ('1800187389084852314 5-5893305375058672149') =} \\ &= \text{ec9b2f89f8ed685ff220348ec43af99f,} \\ f(C_2) &= \text{MD5 ('4646442809403410988 5-36216068594728755239') =} \\ &= \text{efb8bf73e2a1586d38ba3a13ae5fe35a.} \end{aligned}$$

Es ergibt sich (in Hexadezimaldarstellung)

$$\begin{aligned} c_1 = a_1 \oplus f(C_1) &= \text{adf748e68a841c379f4947edac5fd9c5,} \\ c_2 = a_2 \oplus f(C_2) &= \text{8ed0d3168cd5300857c85376ae5fe35a,} \end{aligned}$$

so daß die verschlüsselte Nachricht nun aus

$$(B_1, c_1), \quad (B_2, c_2)$$

besteht.

Das ElGamal-Verfahren beruht darauf, daß die Berechnung diskreter Logarithmen schwierig ist. Das nächste Beispiel nutzt die Schwierigkeit des Wurzelziehens.

Guillou-Quisquater-Signatur-Schema (GQ-Signatur):

(1) Schlüsselerzeugung: Jeder Teilnehmer i konstruiert sich folgende Zahlen:
   (a) $N_i = pq$ (mit verschiedenen Primzahlen p, q, so daß N_i praktisch nicht faktorisiert werden kann.)
   (b) e_i mit $\text{ggT}(e_i, (p-1)(q-1)) = 1$.
   (c) J_i mit $\text{ggT}(J_i, N_i) = 1$. (J_i kann dazu benutzt werden, persönliche Informationen, z.B. den Namen, unterzubringen.)
   (d) Man berechnet a_i mit $J_i a_i^{e_i} \equiv 1 \bmod N_i$. (Berechne d_i mit $d_i e_i \equiv 1 \bmod (p-1)(q-1)$. Dann ist $a_i \equiv J_i^{-d_i} \bmod N_i$.)
   (e) Der öffentliche Schlüssel ist (N_i, e_i, J_i), der private a_i.

(2) Wie unterschreibt i eine Nachricht T?
   (a) i wählt eine Zufallszahl z.
   (b) $r \equiv z^{e_i} \bmod N_i$.
   (c) $t = h(T||r)$, d.h. an die Nachricht T hängt man r (z.B. in Dezimaldarstellung) an und bildet dann den Hashwert t, wo h eine vereinbarte Hashfunktion ist.
   (d) $s \equiv z a_i^t \bmod N_i$.

   Die GQ-Signatur der Nachricht T ist (s, t).

(3) Wie kann man testen, ob die Signatur/Unterschrift (s, t) der Nachricht T wirklich von i stammt? Man kennt die Nachricht T, den öffentlichen Schlüssel (N_i, e_i, J_i) und die Signatur (s, t) und berechnet dann

$$\widetilde{r} \equiv s^{e_i} J_i^t \bmod N_i \quad \text{ und damit } \quad \widetilde{t} = h(T||\widetilde{r}).$$

   Die Signatur/Unterschrift wird akzeptiert, falls $t = \widetilde{t}$ gilt, denn wenn i die Signatur erstellt hat, gilt

$$\widetilde{r} \equiv s^{e_i} J_i^t \equiv (z a_i^t)^{e_i} J_i^t \equiv z^{e_i} (a_i^{e_i} J_i)^t \equiv z^{e_i} \equiv r \bmod N_i$$

   und damit auch $t = \widetilde{t}$.

Beispiel: Wir wählen $N = pq$, e, J, $d \equiv \frac{1}{e} \bmod (p-1)(q-1)$, $a \equiv (\frac{1}{J})^d \bmod N$ mit

$$\begin{aligned} p &= \text{8343685320286294136516317985 23,} \\ q &= \text{9611577429530736191504743212 23,} \\ N &= \text{8019597750357067631059806794694132847620362494060711189536 29,} \\ e &= 5, \\ J &= \text{1234567890123456789012345678901234567890123456789012345678 90,} \\ d &= \text{1603919550071413526211961358935235516974109092746538025667 77,} \\ a &= \text{5818724478248228755526762261274054803210073558099797751461 98.} \end{aligned}$$

Dann ist (N, e, J) ein öffentlicher GQ-Schlüssel, a der zugehörige private Schlüssel.
Als Hashfunktion nehmen wir MD5. Die Bezeichnung $T||r$ soll heißen, daß wir an die Zeichenkette T die Dezimaldarstellung der Zahl r anhängen.
Wir wollen 'Heute ist Mittwoch' signieren. Als Zufallszahl wählen wir

$$z = 75738497568934756983278573264785943785967485967430985763049 8.$$

Mit den Formeln

$$r = z^e \bmod N, \quad t = h(T||r), \quad s = za^t \bmod N$$

erhalten wir

$$\begin{aligned} r &= 61019502076511940080119301332259006833001914903679139793375 2, \\ s &= 75446646255924741213862951127339972805693850356631133904651 7, \\ t &= 10240070785972941918892841862820253871 5. \end{aligned}$$

Dann ist also (s, t) die GQ-Signatur von T mit dem Schlüssel (N, e, J, a).
Als Test bilden wir

$$\widetilde{r} = s^e J^t \bmod N$$

und erhalten $\widetilde{r} = r$, wie es bei einer gültigen Signatur sein sollte.

Überlegungen: Gegeben sei eine Zeichenkette T und ein öffentlicher GQ-Schlüssel (N_i, e_i, J_i). Wir wollen (s, t) finden, so daß (s, t) eine gültige GQ-Signatur für T bzgl. des Schlüssels (N_i, e_i, J_i) ist, d.h. wir wollen die Signatur fälschen. Wie kann man das machen?

(1) (s, t) sollte mit $r \equiv s^{e_i} J_i^t \bmod N$ die Gleichung $t = h(T||r)$ erfüllen.

(2) *1. Versuch:* Da die Hashfunktion h nicht durchschaubar sein sollte, geben wir uns r vor, berechnen dann $t = h(T||\widetilde{r})$ und erhalten als Bedingung für s

$$s^{e_i} \equiv r J_i^{-t} \bmod N.$$

Man muß also die e_i-te Wurzel ziehen können. Dafür ist aber kein vernünftiger Algorithmus bekannt, wenn man nicht die Faktorisierung von N_i kennt.

(3) *2. Versuch:* Wir setzen jetzt r etwas spezieller an, nämlich als $r_x \equiv J_i^x \bmod N$ mit einem Parameter x. Dann wird das zugehörige t zu $t_x = h(T||r_x)$ und wir erhalten als Bedingung

$$s^{e_i} \equiv r_x J_i^{-t_x} \equiv J_i^{x - t_x} \bmod N.$$

Wir setzen jetzt an $s = J_i^{-y} \bmod N$ mit einem weiteren Parameter y. Dann wird die Bedingung für eine gültige Signatur

$$J_i^{-ye_i} \equiv J_i^{x - t_x} \bmod N \quad \text{, also} \quad J_i^{t_x - x - ye_i} \equiv 1 \bmod N.$$

Diese Gleichung ist äquivalent mit

$$t_x - x - ye_i \equiv 0 \bmod \operatorname{ord}(J_i),$$

wenn $\operatorname{ord}(J_i)$ die Ordnung von J_i in der multiplikativen Gruppe $(\mathbf{Z}/N\mathbf{Z})^*$ bezeichnet. Da aber die Faktorzerlegung von N unzugänglich ist bzw. sein sollte, kann man die Gruppenordnung $\#(\mathbf{Z}/N\mathbf{Z})^*$ nicht berechnen und somit im allgemeinen auch nicht die Elementordnung $\operatorname{ord}(J_i)$. Somit bleibt als einzige Möglichkeit, obige Gleichung zu erfüllen, der Ansatz

$$t_x - x - ye_i = 0.$$

Dies ist dann gleichwertig mit $t_x \equiv x \bmod e_i$ und $y = \frac{t_x - x}{e_i}$. Dies liefert dann folgenden Ansatz:

(4) *Ansatz:* Zu $x = 0, 1, 2, \ldots$ berechnet man

$$r_x \equiv J_i^x \bmod N, \quad t_x = h(T||r_x)$$

und testet, ob $t_x \equiv x \bmod e_i$ gilt. Wenn ja, dann ist mit $y = \frac{t_x - x}{e_i}$

$$(s, t) = ((\frac{1}{J_i})^y \bmod N, t_x)$$

eine gültige GQ-Signatur.

(5) Ist e_i klein, so hat der beschriebene Ansatz große Aussicht auf Erfolg. Um sicherzustellen, daß eine Signatur nicht auf diesem Weg gefälscht wird, sollte e_i hinreichend groß gewählt werden.

Beispiel: Wir wollen für den obigen GQ-Schlüssel

$$\begin{aligned} N &= 80195977503570676310598067946941328476203624940607111895362 9, \\ e &= 5, \\ J &= 12345678901234567890123456789012345678901234567890123456789012345678901234567890, \end{aligned}$$

eine Unterschrift nach obigem Ansatz zu fälschen versuchen, da $e = 5$ sehr klein ist. Als Text T wählen wir 'Heute ist Donnerstag'. Wir bilden für $x = 0, \ldots, 9$ jeweils

$$r_x = J^x \bmod N, \quad t_x = h(T||r_x)$$

und erhalten

$$\begin{aligned} r_0 &= 1, \\ t_0 &= 187021716417897918939081393701890125706, \\ t_0 - 0 &= 1 \bmod e, \\ r_1 &= 12345678901234567890123456789012345678901234567890123456789012345678901234567890, \\ t_1 &= 37304281428875460480720698791824103419, \\ t_1 - 1 &= 3 \bmod e, \\ r_2 &= 781214447880370836884448627694047713330715287645400737047721, \\ t_2 &= 164484284377829216808608781421987753656, \\ t_2 - 2 &= 4 \bmod e, \\ r_3 &= 640794249104102361268553640000341898524168384644179347014979, \\ t_3 &= 264615935347913259158060516897580300840, \\ t_3 - 3 &= 2 \bmod e, \\ r_4 &= 303389178110818526859340341863956867793179792184275515415625, \\ t_4 &= 102286417163487719369022166884419733422, \\ t_4 - 4 &= 3 \bmod e, \\ r_5 &= 474185587288454249844683803639120559412812852676257244290599, \\ t_5 &= 268772620283729069052289938076922549420, \\ t_5 - 5 &= 0 \bmod e, \\ r_6 &= 560324332903434294049622246619626654128104156630163558576917, \\ t_6 &= 29923114473883795800229817866833791256 9, \\ t_6 - 6 &= 3 \bmod e, \\ r_7 &= 402801520332056775006757284232173137244807905909863450615631, \\ t_7 &= 90546601547327142154378978808189933615, \\ t_7 - 7 &= 3 \bmod e, \\ r_8 &= 631716295813960009264107822008938487832927601700145849404460, \\ t_8 &= 72093047944411693935173449838091619957, \\ t_8 - 8 &= 4 \bmod e, \\ r_9 &= 329541224308114777089953273851766978120406197879553117672228, \\ t_9 &= 29502748007972036202031447066060994 2286, \\ t_9 - 9 &= 2 \bmod e. \end{aligned}$$

Man sieht $t_5 \equiv 5 \bmod 5$, also wählen wir $x = 5$ und erhalten aus $t_x - x = ye$ sofort

$$y = 53754524056745813810457987615384509883$$

und mit $s = (\frac{1}{J})^e \bmod N$ und $t = t_5$ dann

$$\begin{aligned} s &= 591184573847810799536030538158547044549874285553030000530538, \\ t &= 10240070785972941918892841862820253871 5. \end{aligned}$$

Durch einen Test stellt man dann fest, daß tatsächlich (s,t) eine gültige GQ-Signatur ist.

Wir wollen jetzt das GQ-Signatur-Schema auf imaginärquadratische Klassengruppen übertragen.

IQ-GQ-Signatur:

(1) Schlüsselerzeugung: i wählt bzw. berechnet folgende Größen:
   (a) Imaginärquadratische Diskriminante D_i.
   (b) $A_i \in \mathrm{H}(D_i)$.
   (c) $e_i \in \mathbf{N}$.
   (d) $J_i = A_i^{-e_i} \in \mathrm{H}(D_i)$.

   Der öffentliche Schlüssel ist (D_i, e_i, J_i), der private A_i.

(2) Wie signiert i eine Nachricht T?
   (a) Wahl eines Zufallselements $Z \in \mathrm{H}(D_i)$.
   (b) $R = Z^{e_i} \in \mathrm{H}(D_i)$.
   (c) $t = h(T||a_R||b_R)$, wenn $R = [a_R, b_R, D_i]$.
   (d) $S = ZA^t$.

   Die Signatur ist (S,t).

(3) Wie überprüft man die Signatur?
   Man berechnet
$$\widetilde{R} = S^{e_i} J_i^t \quad \text{und damit} \quad \widetilde{t} = h(T||a_{\widetilde{R}}||b_{\widetilde{R}}).$$
   Die Signatur wird akzeptiert, falls $t = \widetilde{t}$ gilt, denn wenn die Signatur tatsächlich von i konstruiert wurde, gilt
$$\widetilde{R} = S^{e_i} J_i^t = (ZA^t)^{e_i} J_i^t = Z^{e_i} (A^{e_i} J_i)^t Z^{e_i} = R$$
   und damit auch $t = \widetilde{t}$.

Bemerkung: Im Unterschied zur klassischen GQ-Signatur kann man hier nicht mit J_i starten und dann A_i aus $J_i A_i^{e_i} = 1$ berechnen, da man in $\mathrm{H}(D_i)$ keine Wurzeln ziehen kann. Also fängt man mit A_i an und berechnet dann J_i.

Beispiel: Wir haben

$$\begin{aligned} D &= -786327846523874652837645827364785236 5107, \\ e &= 10007, \\ J &= [23641540067042968539, 21608794274197195885, D], \\ A &= [27346955561337885993, -20527805173231553701, D] \end{aligned}$$

so gewählt, daß $K = (D, e, J, A)$ ein IQ-GQ-Signatur-Schlüssel ist. Wir wollen den Text

Heute ist Donnerstag!

mit obigem Schlüssel unterschreiben.
Um ein zufälliges Element $Z \in \mathrm{H}(D)$ zu erhalten, wählen wir eine Zufallszahl

$$z = 78435763498576376487562387452,$$

suchen dann den ersten primitiven Modul Z_p mit Primzahlnorm $p \geq z$, was

$$Z_p := [78435763498576376487562387547, 20951158651502139237564903275, D]$$

ergibt, abschließend reduzieren wir Z_p um $Z \in \mathrm{Red}(D)$ zu erhalten:

$$Z = [6369277163403290147, 1586650442182558353, D].$$

Wir bilden in $\mathrm{H}(D)$

$$R = Z^e = [18280249721434612429, -3213412177941356677, D] = [a_R, b_R, D]$$

und bilden den MD5-Hashwert t der Zeichenkette

$$T||a_R||b_R = \text{'Heute ist Donnerstag!18280249721434612429-3213412177941356677'}$$

und erhalten

$$t = 149956964410269580219320886171795558621.$$

Dann ist

$$S = ZA^t = [19854198644356760739, 1855303548112325593, D].$$

Die IQ-GQ-Signatur ist nun (S, t)

$$\begin{aligned} S &= [19854198644356760739, 1855303548112325593, -7863278465238746528376458273647852365107], \\ t &= 14995696441026958021932088617179555862 1. \end{aligned}$$

Fälschen der Unterschrift: Hier kann man genauso wie bei der 'klassischen' GQ-Signatur vorgehen. Will man für eine Zeichenkette T und einen IQ-GQ-Schlüssel (D, e, J) eine gültige Unterschrift konstruieren, setzt man für $x = 0, 1, 2, \ldots$ an und berechnet (in der Klassengruppe)

$$R = J^e = [a_R, b_R, D], \quad t_x = h(T||a_R||b_R).$$

Ist $t_x \equiv x \bmod e$, so erhält man mit

$$y = \frac{t_x - x}{e} \quad \text{und} \quad S = J^{-y}$$

die gültige Signatur (S, t_x). Aussicht auf Erfolg hat dieser Ansatz, wenn e nicht zu groß ist.

Beispiel: Wir wollen für den IQ-GQ-Schlüssel (D, e, J) des letzten Beispiels und den Text 'Heute ist Donnerstag!' eine Signatur nach dem beschriebenen Verfahren erzeugen, was wegen $e = 10007$ sinnvoll erscheint. Für $x = 4836$ erhält man tatsächlich die Signatur (S, t) mit

$$\begin{aligned} S &= [12249110803920915507, -6745291849523563439, D], \\ t &= 85913772141550046708233214688734262732. \end{aligned}$$

Bemerkung: Zur GQ-Signatur und IQ-GQ-Signatur haben wir Maple-Funktionen geschrieben, mit denen die vorangegangenen Beispiele gerechnet wurden.

10. Geometrische Deutung

Erinnerung: Wir hatten früher folgende Aussage bewiesen: Sind $M_1, M_2 \in \mathrm{Mod}(D)$ mit

$$M_1 = A_1(\mathbf{Z} \cdot a_1 + \mathbf{Z} \cdot \frac{b_1 + \sqrt{D}}{2}), \quad M_2 = A_2(\mathbf{Z} \cdot a_2 + \mathbf{Z} \cdot \frac{b_2 + \sqrt{D}}{2}),$$

so gilt mit

$$\alpha_1 = \frac{b_1 + \sqrt{D}}{2a_1}, \quad \alpha_2 = \frac{b_2 + \sqrt{D}}{2a_2}$$

die Äquivalenz

$$M_1 \sim M_2 \quad \Longleftrightarrow \quad \text{es gibt } A, B, C, D \in \mathbf{Z} \text{ mit } AD - BC = \pm 1 \text{ und } \alpha_2 = \frac{A\alpha_1 + B}{C\alpha_1 + D}.$$

Außerdem sind α_1 und α_2 Zahlen mit Diskriminante D. O.E. können wir $a_i \geq 1$ annehmen.

Um die Modulklassen zu beschreiben, können wir als Zahlen $\alpha = \frac{b+\sqrt{D}}{2a}$ mit Diskriminante D und $a \geq 1$ betrachten. Durch

$$\alpha = \frac{b + \sqrt{D}}{2a} \mapsto \frac{b}{2a} + \frac{\sqrt{|D|}}{2a} i$$

erhalten wir eine Einbettung in die obere Halbebene der komplexen Zahlenebene. Dabei gilt für den komplexen Absolutbetrag

$$|\alpha|^2 = \alpha\alpha' = \frac{(b + \sqrt{D})(b - \sqrt{D})}{4a^2} = \frac{b^2 - D}{4a^2} = \frac{4ac}{4a^2} = \frac{c}{a},$$

also

$$|\alpha| = \sqrt{\frac{c}{a}}.$$

Wir wollen den Reduktionsprozess geometrisch deuten: Dabei sei jeweils

$$M = \mathbf{Z} \cdot a + \mathbf{Z} \cdot \frac{b + \sqrt{D}}{2}, \quad D = b^2 - 4ac, \quad \mathrm{ggT}(a, b, c) = 1, \quad \alpha = \frac{b + \sqrt{D}}{2a}$$

gegeben.

(1) Wir können b durch $b+2am$ mit $m \in \mathbf{Z}$ ersetzen, da sich dadurch der Modul nicht ändert. Dies entspricht dem Übergang von α zu $\alpha + m$. Wir können dann o.E. $-a < b \leq a$ erreichen, was zu der Realteilbedingung $-\frac{1}{2} < \mathrm{Re}\,(\alpha) \leq \frac{1}{2}$ führt.

(2) Der zweite wesentliche Reduktionsschritt ergab sich aus

$$\frac{b-\sqrt{D}}{2a}(\mathbf{Z}\cdot a + \mathbf{Z}\cdot\frac{b+\sqrt{D}}{2}) = \mathbf{Z}\cdot c + \mathbf{Z}\cdot\frac{-b+\sqrt{D}}{2}.$$

Der letzte Modul entspricht der quadratischen Zahl

$$\frac{-b+\sqrt{D}}{2c} = \frac{-(b-\sqrt{D})(b+\sqrt{D}+b)}{2c(b+\sqrt{D})} = \frac{-4ac}{2c(b+\sqrt{D})} = -\frac{2a}{b+\sqrt{D}} = -\frac{1}{\alpha}$$

Dieser Reduktionsschritt wurde nur ausgeführt im Fall $a > c$, was der Bedingung $|\alpha| < 1$ entspricht.

Damit erhalten wir folgende Übersetzung:

Reduktionsprozess für imaginärquadratische Zahlen mit Diskriminante D: Gegeben sei

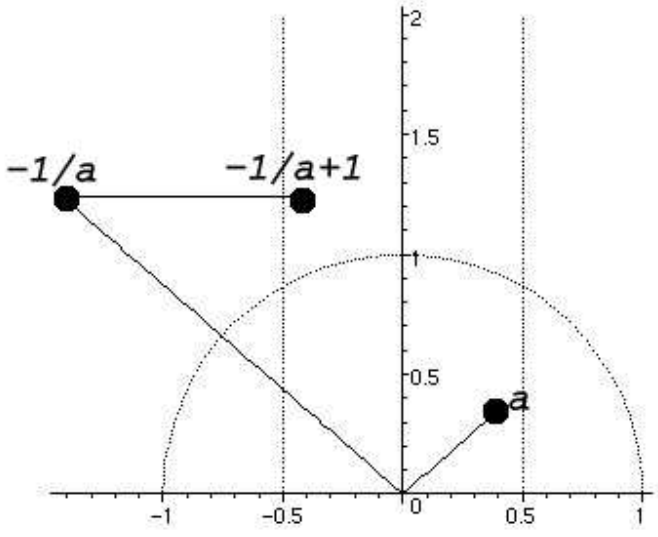
$$\alpha = \frac{b+\sqrt{D}}{2a}, \quad D = b^2 - 4ac, \quad \mathrm{ggT}(a,b,c) = 1.$$

Anfang: Ersetze α durch $\alpha + m$, $m \in \mathbf{Z}$, so daß $-\frac{1}{2} < \mathrm{Re}\,(\alpha) \leq \frac{1}{2}$ gilt.
Reduktionsschritt: Durchlaufe folgende Schritte, solange $|\alpha| < 1$ gilt:
- Ersetze α durch $-\frac{1}{\alpha}$.
- Ersetze α durch $\alpha + m$, $m \in \mathbf{Z}$, so daß $-\frac{1}{2} < \mathrm{Re}\,(\alpha) \leq \frac{1}{2}$ gilt.

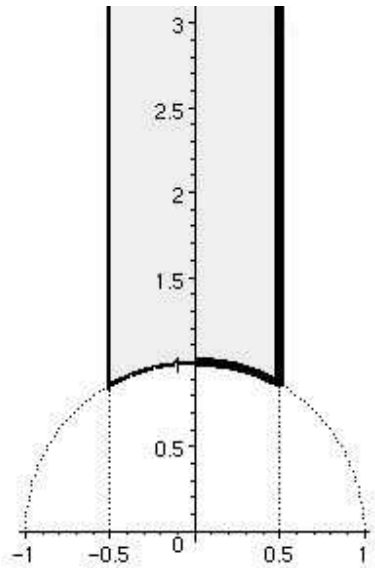
Abschluß: Ist $|\alpha| = 1$ und $\mathrm{Re}\,(\alpha) < 0$, ersetze α durch $-\alpha'$.

Durch den Reduktionsprozess erhält man imaginärquadratische Zahlen, die wir als reduziert bezeichnen:

$$-\frac{1}{2} < \mathrm{Re}\,(\alpha) \leq \frac{1}{2}, \quad |\alpha| > 1, \qquad \text{im Fall } |\alpha| = 1 \text{ gilt außerdem } \mathrm{Re}\,(\alpha) \geq 0.$$



Die folgende Skizze zeigt das Gebiet mit den reduzierten Zahlen.



11. Übungen

Aufgabe 7.1: Sei D eine imaginärquadratische Diskriminante und $M \in \mathrm{Mod}(D)$. Sei $M_0, M_1, \ldots, M_n \in \mathrm{Mod}(D)$ die Folge der Moduln, die sich beim Reduktionsprozess von $M = M_0$ ergeben, also

$$M_i = \mathbf{Z} \cdot a_i + \mathbf{Z} \cdot \frac{b_i + \sqrt{D}}{2}, \quad D = b_i^2 - 4a_ic_i, \quad \mathrm{ggT}(a_i, b_i, c_i) = 1, \quad -a_i < b_i \leq a_i,$$

wobei für $0 \leq i \leq n-1$ bzw. $0 \leq i \leq n-2$ gilt $a_i > c_i$ und $a_{i+1} = c_i$.

(1) Ist $a_i \geq \sqrt{|D|}$, so ist $a_{i+1} \leq \frac{1}{2}a_i$.
(2) Ist $a_i < \sqrt{|D|}$, so ist M_i reduziert oder M_{i+1}.
(3) Schätze die Anzahl der Reduktionsschritte in Abhängigkeit von $a = a_0$ und $|D|$ ab.

Aufgabe 7.2: Sei $M \in \mathrm{Mod}(D)$ und M_{red} der zu M äquivalente reduzierte Modul. Dann gibt es ein $\alpha \in \mathbf{Q}(\sqrt{D})$ mit $M = \alpha M_{\mathrm{red}}$. Gib einen Algorithmus für die Bestimmung von α an.

Aufgabe 7.3: Es gibt genau einen imaginärquadratischen reduzierten Modul M, der die Gleichung $\mathrm{N}(M) = \sqrt{\frac{|\mathrm{disc}\,(M)|}{3}}$ erfüllt. Welchen?

Aufgabe 7.4: Wie sehen die primitiven imaginärquadratischen Moduln M aus, wo Diskriminante und Norm die Gleichung $\mathrm{N}(M) = \sqrt{\frac{|\mathrm{disc}\,(M)|}{4}}$ erfüllen.

Aufgabe 7.5: Gilt für die Norm a eines primitiven Moduls $M = \mathbf{Z} \cdot a + \mathbf{Z} \cdot \frac{b+\sqrt{D}}{2} \in \mathrm{Mod}(D)$ die Ungleichung $a \leq \frac{1}{2}\sqrt{|D|}$, so ist M reduziert. Zeige, daß die Ungleichung nicht durch eine Ungleichung $a \leq (\frac{1}{2}+\varepsilon)\sqrt{|D|}$ ersetzt werden kann. (Hinweis: Betrachte für $n \geq 2$ die Diskriminante $D = -4n^2+4n+1$ und den Modul $M = \mathbf{Z} \cdot n + \mathbf{Z} \cdot \frac{1+\sqrt{D}}{2}$.)

Aufgabe 7.6: Ist M ein reduzierter Modul mit Diskriminante D, so gilt $\mathrm{N}(M) \leq \frac{1}{\sqrt{3}}\sqrt{|D|}$. Zeige, daß diese Abschätzung nicht verbessert werden kann, d.h. man kann die Ungleichung nicht ersetzen durch $\mathrm{N}(M) \leq (\frac{1}{\sqrt{3}}-\varepsilon)\sqrt{|D|}$. (Hinweis: Betrachte für $n \in \mathbf{N}$ die Diskriminante $D = -3n^2 - 4n$ und den Modul $M = \mathbf{Z} \cdot n + \mathbf{Z} \cdot \frac{n+\sqrt{D}}{2}$.)

Aufgabe 7.7: Für eine imaginärquadratische Diskriminante D sei

$$\begin{aligned} S(D) &= \{M \in \mathrm{Mod}(D) : M \text{ primitiv}, \frac{\sqrt{|D|}}{4} \le \mathrm{N}(M) \le \frac{\sqrt{|D|}}{3}\}, \\ S_r(D) &= \{M \in \mathrm{Red}(D) : \frac{\sqrt{|D|}}{4} \le \mathrm{N}(M) \le \frac{\sqrt{|D|}}{3}\}. \end{aligned}$$

Untersuche experimentell das Verhältnis

$$\frac{\#S_r(D)}{\#S(D)},$$

d.h. die Frage, welchen Anteil die reduzierten Moduln an allen primitiven Moduln mit Norm zwischen $\sqrt{\frac{|D|}{4}}$ und $\sqrt{\frac{|D|}{3}}$ haben.

Aufgabe 7.8: Bestimme (vermutungsweise) alle imaginärquadratischen Diskriminanten D mit $h(D) = 6$.

Aufgabe 7.9: Es gibt 3 abelsche Gruppen der Ordnung 8. Gib zu jedem Fall eine imaginärquadratische Diskriminante D an, so daß die Klassengruppe $\mathrm{H}(D)$ die zugehörige Gruppenstruktur hat.

Aufgabe 7.10: Sei D eine imaginärquadratische Diskriminante mit $D \equiv 1 \bmod 4$ und $D < -3$.

(1) Sei $D = -uv$ eine Faktorisierung mit $u < v$ und $\mathrm{ggT}(u, v) = 1$. Setzt man

$$(a, b, c) = \begin{cases} (u, u, \frac{u+v}{4}) & \text{im Fall } u < 3u < v, \\ (\frac{u+v}{4}, \frac{v-u}{2}, \frac{u+v}{4}) & \text{im Fall } u < v < 3u, \end{cases}$$

so ist $M = \mathbf{Z} \cdot a + \mathbf{Z} \cdot \frac{b+\sqrt{D}}{2}$ ein reduzierter Modul mit $\overline{M} \in \mathrm{H}(D)_2$.

(2) Das Tripel (a, a, c) gehöre zu einem reduzierten Modul mit Diskriminante D. Dann ist

$$D = -a(4c - a), \quad \mathrm{ggT}(a, 4c - a) = 1, \quad a < 3a < 4c - a.$$

(3) Das Tripel (a, b, a) gehöre zu einem reduzierten Modul mit Diskriminante D. Dann ist

$$D = -(2a - b)(2a + b), \quad \mathrm{ggT}(2a - b, 2a + b) = 1, \quad 2a - b < 2a + b < 3(2a - b).$$

(4) Zeige, daß sich aus den vorangegangenen Aussagen eine Bijektion zwischen den Elementen aus $\mathrm{H}(D)_2$ und den Faktorisierungen der Diskriminante $D = -uv$ mit $\mathrm{ggT}(u, v) = 1$ ergibt.

Aufgabe 7.11: Beweise oder widerlege:

$$h(-1219326443163391946563) = 5609655886.$$

Aufgabe 7.12: Für $D = -5919282544409406213191631366079$ und

$$C = \mathbf{Z} \cdot 2 + \mathbf{Z} \cdot \frac{1 + \sqrt{D}}{2} \quad \text{gilt in der Klassengruppe} \quad \mathrm{ord}\,(C) = 534511162197935.$$

Bestimme $h(D)$.

Aufgabe 7.13: Hans verwendet das GQ-Signatur-Verfahren mit dem öffentlichen Schlüssel (N, e, J), wobei

$$\begin{aligned} N &= 136734394546445949436568444600131386808770255425612173611677, \\ e &= 1234567, \\ J &= 111111111111111111111111111111111111111111111111111111111111111 \end{aligned}$$

gilt. Schicke an Peter die Nachricht

Lieber Peter, das heutige Kennwort ist NEPTUN. Gruss, Hans

Bestimme eine mögliche (dann natürlich gefälschte) GQ-Unterschrift.

Aufgabe 7.14: Erich verwendet das IQ-GQ-Signatur-Verfahren mit dem öffentlichen Schlüssel (D, e, J), wobei

$$\begin{aligned} D &= -9857643985763049875608934756098347059363, \\ e &= 1234567, \\ J &= [45820136558036439781, -18699009596096745785, D] \end{aligned}$$

gilt. Sende an Peter die Nachricht

Lieber Peter, das heutige Kennwort ist URANUS. Gruss, Erich

und fälsche Erichs IQ-GQ-Unterschrift.

Aufgabe 7.15: Faktorisiere die 100-stellige Zahl

$$\begin{aligned} N = {} & 11280781795606863639225618752326448610116891909234 \\ & 03732043344289789116696232140439422406743841719099. \end{aligned}$$

KAPITEL 8

Reellquadratische Klassengruppen

Das Kapitel enthält bisher nur die Beschreibung der Klassengruppe durch reduzierte Zahlen.

1. Reduzierte reellquadratische Zahlen

Wir stellen hier einige Aussagen zusammen, die bereits früher bewiesen wurden.

(1) Die Abbildung

$$\rho : \mathbf{R} \setminus \mathbf{Z} \to \mathbf{R}, \quad \alpha \mapsto \frac{1}{\alpha - \lfloor \alpha \rfloor}$$

liefert den Nachfolger in der Kettenbruchentwicklung, d.h. $\alpha = \lfloor \alpha \rfloor + \frac{1}{\rho(\alpha)}$ und damit

$$\alpha = [a_0, a_1, a_2, a_3, a_4, \dots] \quad \Longrightarrow \quad \rho(\alpha) = [a_1, a_2, a_3, a_4, \dots].$$

(2) Eine reellquadratische Diskriminante ist eine ganze Zahl D mit $D > 0$, D kein Quadrat, $D \equiv 0$ oder $1 \bmod 4$. Eine Zahl $\alpha \in \mathbf{R} \setminus \mathbf{Q}$ ist reellquadratisch mit Diskriminante D, wenn sich α als

$$\alpha = \frac{b + \sqrt{D}}{2a} \quad \text{mit} \quad D = b^2 - 4ac, \quad \mathrm{ggT}(a,b,c) = 1$$

schreiben läßt. Mit α ist dann auch $\rho(\alpha)$ reellquadratisch mit Diskriminante D. Explizit: Mit

$$u = \lfloor \alpha \rfloor = \begin{cases} \lfloor \frac{b + \lfloor \sqrt{D} \rfloor}{2a} \rfloor & \text{für } a > 0, \\ \lfloor \frac{b + \lceil \sqrt{D} \rceil}{2a} \rfloor & \text{für } a < 0, \end{cases} \quad \widetilde{b} = 2au - b, \quad \widetilde{a} = \frac{D - \widetilde{b}^2}{4a} \quad \text{gilt} \quad \rho(\alpha) = \frac{\widetilde{b} + \sqrt{D}}{2\widetilde{a}}.$$

(3) Ist α reellquadratisch mit Diskriminante D und n hinreichend groß, so ist $\rho^n(\alpha)$ reduziert, d.h. Element der Menge

$$\begin{aligned} \mathrm{Red}(D) &= \{ \frac{b + \sqrt{D}}{2a} : \quad D = b^2 - 4ac, \quad \mathrm{ggT}(a,b,c) = 1 \\ &\qquad \text{mit } 1 \le b \le \lfloor \sqrt{D} \rfloor, \quad \lfloor \frac{\lfloor \sqrt{D} \rfloor - b}{2} \rfloor + 1 \le a \le \lfloor \frac{\lfloor \sqrt{D} \rfloor + b}{2} \rfloor \} = \\ &= \{ \alpha \text{ reellquadratisch mit Diskriminante } D \text{ und } \alpha > 1 \text{ und } -1 < \alpha' < 0 \}. \end{aligned}$$

(4) Die Abbildung ρ erfüllt $\rho(\mathrm{Red}(D)) \subseteq \mathrm{Red}(D)$ und

$$\rho|_{\mathrm{Red}(D)} : \mathrm{Red}(D) \to \mathrm{Red}(D)$$

ist bijektiv.

(5) Genau die reduzierten reellquadratischen Zahlen haben eine (rein) periodische Kettenbruchentwicklung. Ist $\alpha \in \mathrm{Red}(D)$ mit

$$\alpha = [\overline{a_0, a_1, a_2, \dots, a_{k-1}}],$$

so ist für alle $m \in \mathbf{N}$

$$\rho^m(\alpha) = [\overline{a_{(0+m) \bmod k}, a_{(1+m) \bmod k}, a_{(2+m) \bmod k}, \dots, a_{(k+1-m) \bmod k}}].$$

Insbesondere gilt $\rho^k(\alpha) = \alpha$. Die ρ-Bahn von α in $\mathrm{Red}(D)$ ist

$$Z(\alpha) = \{\alpha, \rho(\alpha), \rho^2(\alpha), \dots, \rho^{k-1}(\alpha)\}$$

und wird der von α erzeugte Zyklus genannt. In Kettenbruchschreibweise:

$$Z(\alpha) = \{[\overline{a_0, a_1, \dots, a_{k-1}}], [\overline{a_1, a_2, \dots, a_{k-1}, a_0}], \dots, [\overline{a_{k-1}, a_0, \dots, a_{k-3}, a_{k-2}}]\}.$$

Dem von α erzeugten Zyklus $Z(\alpha)$ entspricht dann also ein periodischer Kettenbruch modulo periodischer Shifts.

(6) Die Menge der reduzierten Zahlen $\mathrm{Red}(D)$ zerfällt in die disjunkte Vereinigung solcher Zyklen:

$$\mathrm{Red}(D) = Z(\alpha_1) \cup Z(\alpha_2) \cup \cdots \cup Z(\alpha_h) \quad \text{mit} \quad Z(\alpha_i) \cap Z(\alpha_j) = \emptyset \text{ für } i \neq j,$$

wobei wegen $Z(\alpha_i) = Z(\rho(\alpha_i))$ die repräsentierenden Elemente α_i im allgemeinen nicht eindeutig bestimmt sind.

Beispiel: Wir betrachten die reduzierten reellquadratischen Zahlen mit Diskriminante $D = 1297$. Die Zyklen sind dabei so notiert, daß $Z(\alpha_i) = \{\alpha_i, \rho(\alpha_i), \rho^2(\alpha_i), \dots\}$ gilt. Die angegebene Kettenbruchentwicklung ist die des ersten angegeben Elements α_i in $Z(\alpha_i)$.

i	$Z(\alpha_i)$	Kettenbruchentwicklung
1	$\{\frac{31+\sqrt{1297}}{12}, \frac{29+\sqrt{1297}}{38}, \frac{9+\sqrt{1297}}{32}, \frac{23+\sqrt{1297}}{24}, \frac{25+\sqrt{1297}}{28}\}$	$[\overline{5,1,1,2,2}]$
2	$\{\frac{31+\sqrt{1297}}{14}, \frac{25+\sqrt{1297}}{48}, \frac{23+\sqrt{1297}}{16}, \frac{25+\sqrt{1297}}{42}, \frac{17+\sqrt{1297}}{24}\}$	$[\overline{4,1,3,1,2}]$
3	$\{\frac{31+\sqrt{1297}}{24}, \frac{17+\sqrt{1297}}{42}, \frac{25+\sqrt{1297}}{16}, \frac{23+\sqrt{1297}}{48}, \frac{25+\sqrt{1297}}{14}\}$	$[\overline{2,1,3,1,4}]$
4	$\{\frac{31+\sqrt{1297}}{28}, \frac{25+\sqrt{1297}}{24}, \frac{23+\sqrt{1297}}{32}, \frac{9+\sqrt{1297}}{38}, \frac{29+\sqrt{1297}}{12}\}$	$[\overline{2,2,1,1,5}]$
5	$\{\frac{31+\sqrt{1297}}{42}, \frac{11+\sqrt{1297}}{28}, \frac{17+\sqrt{1297}}{36}, \frac{19+\sqrt{1297}}{26}, \frac{33+\sqrt{1297}}{8}\}$	$[\overline{1,1,1,2,8}]$
6	$\{\frac{31+\sqrt{1297}}{56}, \frac{25+\sqrt{1297}}{12}, \frac{35+\sqrt{1297}}{6}\}$	$[\overline{1,5,11}]$
7	$\{\frac{33+\sqrt{1297}}{4}, \frac{35+\sqrt{1297}}{18}, \frac{19+\sqrt{1297}}{52}\}$	$[\overline{17,3,1}]$
8	$\{\frac{33+\sqrt{1297}}{26}, \frac{19+\sqrt{1297}}{36}, \frac{17+\sqrt{1297}}{28}, \frac{11+\sqrt{1297}}{42}, \frac{31+\sqrt{1297}}{8}\}$	$[\overline{2,1,1,1,8}]$
9	$\{\frac{35+\sqrt{1297}}{2}, \frac{35+\sqrt{1297}}{36}, \frac{1+\sqrt{1297}}{36}\}$	$[\overline{35,1,1}]$
10	$\{\frac{35+\sqrt{1297}}{4}, \frac{33+\sqrt{1297}}{52}, \frac{19+\sqrt{1297}}{18}\}$	$[\overline{17,1,3}]$
11	$\{\frac{35+\sqrt{1297}}{12}, \frac{25+\sqrt{1297}}{56}, \frac{31+\sqrt{1297}}{6}\}$	$[\overline{5,1,11}]$

2. Beschreibung der Klassengruppe durch Zyklen reduzierter Zahlen

LEMMA. *Sei D eine reellquadratische Diskriminante.*

(1) *Ist $M \in \mathrm{Mod}(D)$ mit*

$$M = A \cdot (\mathbf{Z} \cdot a + \mathbf{Z} \cdot \frac{b+\sqrt{D}}{2}), \quad D = b^2 - 4ac, \quad \mathrm{ggT}(a,b,c) = 1$$

und

$$\alpha = \frac{b+\sqrt{D}}{2a},$$

so gilt

$$M \sim \mathbf{Z} + \mathbf{Z}\alpha \sim \mathbf{Z} + \mathbf{Z}\rho(\alpha).$$

(2) *In jeder Klasse von $\mathrm{H}(D)$ gibt es einen Modul $\mathbf{Z} + \mathbf{Z}\alpha$ mit $\alpha \in \mathrm{Red}(D)$.*

(3) *Für $\alpha_1, \alpha_2 \in \mathrm{Red}(D)$ gilt*

$$\mathbf{Z} + \mathbf{Z}\alpha_1 \sim \mathbf{Z} + \mathbf{Z}\alpha_2 \iff \alpha_2 = \rho^m(\alpha_1) \text{ für ein } m \geq 0 \iff Z(\alpha_1) = Z(\alpha_2).$$

Beweis:

(1) Es gilt

$$\begin{aligned} M &= A(\mathbf{Z}a + \mathbf{Z}\frac{b+\sqrt{D}}{2}) = Aa(\mathbf{Z} + \mathbf{Z}\frac{b+\sqrt{D}}{2a}) = Aa(\mathbf{Z} + \mathbf{Z}\alpha) \sim \\ &\sim \mathbf{Z} + \mathbf{Z}\alpha = \mathbf{Z} + \mathbf{Z}(\lfloor\alpha\rfloor + \frac{1}{\rho(\alpha)}) = \mathbf{Z} + \mathbf{Z}\frac{1}{\rho(\alpha)} = \frac{1}{\rho(\alpha)}(\mathbf{Z} + \mathbf{Z}\rho(\alpha)) \sim \\ &\sim \mathbf{Z} + \mathbf{Z}\rho(\alpha). \end{aligned}$$

(2) Ist $M \in \mathrm{Mod}(D)$, so gilt mit 1. und $m \in \mathbf{N}$

$$M \sim \mathbf{Z} + \mathbf{Z}\alpha \sim \mathbf{Z} + \mathbf{Z}\rho(\alpha) \sim \mathbf{Z} + \mathbf{Z}\rho^2(\alpha) \sim \mathbf{Z} + \mathbf{Z}\rho^3(\alpha) \sim \cdots \sim \mathbf{Z} + \mathbf{Z}\rho^m(\alpha).$$

Für hinreichend großes m ist $\rho^m(\alpha)$ reduziert, woraus die Behauptung folgt.

(3) Wir haben bereits gesehen

$$\mathbf{Z} + \mathbf{Z}\alpha_1 \sim \mathbf{Z} + \mathbf{Z}\alpha_2 \quad \Longleftrightarrow \quad \alpha_1 = \frac{A\alpha_2 + B}{C\alpha_2 + D} \text{ mit } A, B, C, D \in \mathbf{Z}, AD - BC = \pm 1.$$

Nach dem Satz von Serret ist die Bedingung der rechten Seite äquivalent mit der Bedingung $\rho^{m_1}(\alpha_1) = \rho^{m_2}(\alpha_2)$ mit $m_1, m_2 \in \mathbf{N}$. Somit erhalten wir unter Beachtung der Bijektivität von ρ auf $\mathrm{Red}(D)$

$$\begin{aligned} \mathbf{Z} + \mathbf{Z}\alpha_1 \sim \mathbf{Z} + \mathbf{Z}\alpha_2 \quad &\Longleftrightarrow \quad \rho^{m_1}(\alpha_1) = \rho^{m_2}(\alpha_2) \text{ mit } m_1, m_2 \geq 0 \\ &\Longleftrightarrow \quad \alpha_2 = \rho^m(\alpha_1) \text{ mit } m \geq 0 \\ &\Longleftrightarrow \quad Z(\alpha_2) = Z(\alpha_1), \end{aligned}$$

was die Behauptung beweist. ■

Damit ergibt sich sofort der folgende Satz:

SATZ. *Sei D eine reellquadratische Diskriminante und*

$$\mathrm{Red}(D) = Z(\alpha_1) \cup Z(\alpha_2) \cup \cdots \cup Z(\alpha_h) \quad \textit{mit} \quad Z(\alpha_i) \cap Z(\alpha_j) = \emptyset \textit{ für } i \neq j.$$

Dann bilden die Moduln $\mathbf{Z} + \mathbf{Z}\alpha_i$ ein Repräsentantensystem von $\mathrm{H}(D)$. Insbesondere folgt

$$h(D) = \#\mathrm{H}(D) = h,$$

d.h. $h(D)$ ist die Anzahl der Zyklen, in die $\mathrm{Red}(D)$ unter der ρ-Operation zerfällt.

Bemerkungen:

(1) Im Unterschied zu den imaginärquadratischen Klassengruppen ist das im Satz angegebene Repräsentantensystem $\mathbf{Z}+\mathbf{Z}\alpha_i$, $i = 1, \ldots, h$ für die Klassengruppe $\mathrm{H}(D)$ nicht eindeutig bestimmt. Ist $\widetilde{\alpha}_i \in Z(\alpha_i)$, so ist auch $\mathbf{Z}+\mathbf{Z}\widetilde{\alpha}_i$, $i = 1, \ldots, h$ ein Repräsentantensystem. Das macht das Rechnen deutlich unangenehmer.

(2) Ist $M \in \mathrm{Mod}(D)$, schreibt man $M = \beta(\mathbf{Z} + \mathbf{Z}\alpha)$, entwickelt man α in einen Kettenbruch, bis $\rho^m(\alpha)$ reduziert ist, so ist

$$M \sim \mathbf{Z} + \mathbf{Z}\rho^m(\alpha),$$

also hat man für M einen Repräsentanten in der Klassengruppe (wie im Satz angegeben) gefunden.

(3) Sind $\alpha, \beta \in \mathrm{Red}(D)$, wie testet man, ob $\mathbf{Z} + \mathbf{Z}\alpha$ und $\mathbf{Z} + \mathbf{Z}\beta$ in der gleichen Klasse liegen? Geht man wie im Lemma vor, muß man testen, ob ein $m \geq 0$ existiert mit $\beta = \rho^m(\alpha)$. Ist k die Periodenlänge der Kettenbruchentwicklung von α, so kann jedes m mit $0 \leq m \leq k - 1$ vorkommen. Dies macht das Verfahren unpraktikabel, wenn die Periodenlänge k groß ist.

Das folgende Verfahren funktioniert für nicht zu große D:

Algorithmus zur Beschreibung von $\mathrm{H}(D)$ durch Zerlegung von $\mathrm{Red}(D)$ in Zyklen: Sei D eine reellquadratische Diskriminante. Es wird $H = \{Z_1, Z_2, \ldots, Z_h\}$ bestimmt, so daß $Z_1 \cup \cdots \cup Z_h$ eine Zyklenzerlegung von $\mathrm{Red}(D)$ wie im Satz ist. Insbesondere ist $h = h(D)$.

(1) Bestimme $\mathrm{Red}(D)$ und setze dann $R := \mathrm{Red}(D)$, $H := \emptyset$, $h := 0$.
(2) Setze $h := h + 1$. Wähle ein Element $\alpha_h \in \mathrm{Red}(D)$. Setze $\alpha := \alpha_h$ und $Z := \{\alpha\}$, $R := R \setminus \{\alpha\}$.
(3) Berechne und setze $\alpha := \rho(\alpha)$.
(4) Ist $\alpha \neq \alpha_h$, setze $Z := Z \cup \{\alpha\}$, $R := R \setminus \{\alpha\}$. Gehe zu 3.
(5) Ist $\alpha = \alpha_h$, setze $H := H \cup \{Z\}$.
(6) Ist $R \neq \emptyset$, gehe zu 2. Andernfalls ist $h = h(D)$ und H die Zykelzerlegung von $\mathrm{Red}(D)$.

Bemerkung: Die Maple-Funktion 'HD' (in algo8_ma) funktioniert nach dem beschriebenen Verfahren.

3. Beispiele - Experimente

Beispiel: In der folgenden Tabelle haben wir für die kleinsten Diskriminanten die Zykelzerlegung von $\mathrm{Red}(D)$ sowie eine zugehörige Kettenbruchentwicklung angegeben:

i	$Z(\alpha_i)$ $(D=5,\ h(D)=1)$	Kettenbruchentwicklung von α_i
1	$\{\frac{1+\sqrt{5}}{2}\}$	$\overline{[1]}$

i	$Z(\alpha_i)$ $(D=8,\ h(D)=1)$	Kettenbruchentwicklung von α_i
1	$\{\frac{2+\sqrt{8}}{2}\}$	$\overline{[2]}$

i	$Z(\alpha_i)$ $(D=12,\ h(D)=1)$	Kettenbruchentwicklung von α_i
1	$\{\frac{2+\sqrt{12}}{2}, \frac{2+\sqrt{12}}{4}\}$	$\overline{[2,1]}$

i	$Z(\alpha_i)$ $(D=13,\ h(D)=1)$	Kettenbruchentwicklung von α_i
1	$\{\frac{3+\sqrt{13}}{2}\}$	$\overline{[3]}$

i	$Z(\alpha_i)$ $(D=17,\ h(D)=1)$	Kettenbruchentwicklung von α_i
1	$\{\frac{1+\sqrt{17}}{4}, \frac{3+\sqrt{17}}{2}, \frac{3+\sqrt{17}}{4}\}$	$\overline{[1,3,1]}$

i	$Z(\alpha_i)$ $(D=20,\ h(D)=1)$	Kettenbruchentwicklung von α_i
1	$\{\frac{4+\sqrt{20}}{2}\}$	$\overline{[4]}$

i	$Z(\alpha_i)$ $(D=21,\ h(D)=1)$	Kettenbruchentwicklung von α_i
1	$\{\frac{3+\sqrt{21}}{2}, \frac{3+\sqrt{21}}{6}\}$	$\overline{[3,1]}$

i	$Z(\alpha_i)$ $(D=24,\ h(D)=1)$	Kettenbruchentwicklung von α_i
1	$\{\frac{4+\sqrt{24}}{2}, \frac{4+\sqrt{24}}{4}\}$	$\overline{[4,2]}$

i	$Z(\alpha_i)$ $(D=28,\ h(D)=1)$	Kettenbruchentwicklung von α_i
1	$\{\frac{2+\sqrt{28}}{4}, \frac{2+\sqrt{28}}{6}, \frac{4+\sqrt{28}}{2}, \frac{4+\sqrt{28}}{6}\}$	$\overline{[1,1,4,1]}$

i	$Z(\alpha_i)$ $(D=29,\ h(D)=1)$	Kettenbruchentwicklung von α_i
1	$\{\frac{5+\sqrt{29}}{2}\}$	$\overline{[5]}$

i	$Z(\alpha_i)$ $(D=32,\ h(D)=1)$	Kettenbruchentwicklung von α_i
1	$\{\frac{4+\sqrt{32}}{2}, \frac{4+\sqrt{32}}{8}\}$	$\overline{[4,1]}$

i	$Z(\alpha_i)$ $(D=33,\ h(D)=1)$	Kettenbruchentwicklung von α_i
1	$\{\frac{3+\sqrt{33}}{4}, \frac{5+\sqrt{33}}{2}, \frac{5+\sqrt{33}}{4}, \frac{3+\sqrt{33}}{6}\}$	$\overline{[2,5,2,1]}$

i	$Z(\alpha_i)$ $(D=37,\ h(D)=1)$	Kettenbruchentwicklung von α_i
1	$\{\frac{1+\sqrt{37}}{6}, \frac{5+\sqrt{37}}{2}, \frac{5+\sqrt{37}}{6}\}$	$\overline{[1,5,1]}$

i	$Z(\alpha_i)$ $(D=40,\ h(D)=2)$	Kettenbruchentwicklung von α_i
1	$\{\frac{2+\sqrt{40}}{6}, \frac{4+\sqrt{40}}{4}, \frac{4+\sqrt{40}}{6}\}$	$\overline{[1,2,1]}$
2	$\{\frac{6+\sqrt{40}}{2}\}$	$\overline{[6]}$

i	$Z(\alpha_i)$ $(D=41,\ h(D)=1)$	Kettenbruchentwicklung von α_i
1	$\{\frac{3+\sqrt{41}}{4}, \frac{5+\sqrt{41}}{4}, \frac{3+\sqrt{41}}{8}, \frac{5+\sqrt{41}}{2}, \frac{5+\sqrt{41}}{8}\}$	$\overline{[2,2,1,5,1]}$

i	$Z(\alpha_i)$ $(D=44,\ h(D)=1)$	Kettenbruchentwicklung von α_i
1	$\{\frac{6+\sqrt{44}}{2}, \frac{6+\sqrt{44}}{4}\}$	$\overline{[6,3]}$

i	$Z(\alpha_i)$ $(D=45,\ h(D)=1)$	Kettenbruchentwicklung von α_i
1	$\{\frac{5+\sqrt{45}}{2}, \frac{5+\sqrt{45}}{10}\}$	$\overline{[5,1]}$

i	$Z(\alpha_i)$ $(D=48,\ h(D)=1)$	Kettenbruchentwicklung von α_i
1	$\{\frac{6+\sqrt{48}}{2}, \frac{6+\sqrt{48}}{6}\}$	$\overline{[6,2]}$

i	$Z(\alpha_i)$ $(D=52,\ h(D)=1)$	Kettenbruchentwicklung von α_i
1	$\{\frac{2+\sqrt{52}}{6}, \frac{4+\sqrt{52}}{6}, \frac{2+\sqrt{52}}{8}, \frac{6+\sqrt{52}}{2}, \frac{6+\sqrt{52}}{8}\}$	$\overline{[1,1,1,6,1]}$

i	$Z(\alpha_i)$ $(D=53,\ h(D)=1)$	Kettenbruchentwicklung von α_i
1	$\{\frac{7+\sqrt{53}}{2}\}$	$\overline{[7]}$

i	$Z(\alpha_i)$ $(D=56,\ h(D)=1)$	Kettenbruchentwicklung von α_i
1	$\{\frac{4+\sqrt{56}}{4}, \frac{4+\sqrt{56}}{10}, \frac{6+\sqrt{56}}{2}, \frac{6+\sqrt{56}}{10}\}$	$\overline{[2,1,6,1]}$

i	$Z(\alpha_i)$ $(D=57,\ h(D)=1)$	Kettenbruchentwicklung von α_i
1	$\{\frac{3+\sqrt{57}}{6}, \frac{3+\sqrt{57}}{8}, \frac{5+\sqrt{57}}{4}, \frac{7+\sqrt{57}}{2}, \frac{7+\sqrt{57}}{4}, \frac{5+\sqrt{57}}{8}\}$	$\overline{[1,1,3,7,3,1]}$

Beispiel: Wir betrachten jetzt alle reellquadratischen Diskriminanten $D \leq 1000$ und bestimmen die zugehörige Klassenzahl $h(D)$. Die folgende Tabelle gibt zu h die D's mit $h(D) = h$ an:

h	D mit $h(D) = h$ und $1 \leq D \leq 1000$
1	5, 8, 12, 13, 17, 20, 21, 24, 28, 29, 32, 33, 37, 41, 44, 45, 48, 52, 53, 56, 57, 61, 68, 69, 72, 73, 76, 77, 80, 84, 88, 89, 92, 93, 97, 101, 108, 109, 112, 113, 116, 117, 124, 125, 128, 129, 132, 133, 137, 141, 149, 152, 153, 157, 161, 164, 172, 173, 176, 177, 180, 181, 184, 188, 189, 193, 197, 201, 208, 209, 212, 213, 216, 217, 228, 233, 236, 237, 241, 244, 245, 248, 249, 253, 261, 268, 269, 272, 276, 277, 281, 284, 292, 293, 297, 301, 304, 308, 309, 313, 317, 329, 332, 333, 337, 341, 344, 349, 353, 356, 368, 369, 372, 373, 376, 381, 388, 389, 392, 393, 397, 405, 409, 412, 413, 417, 421, 428, 432, 433, 436, 437, 449, 452, 453, 457, 461, 464, 468, 472, 477, 489, 496, 497, 500, 501, 508, 509, 512, 513, 516, 517, 521, 524, 532, 536, 537, 541, 548, 549, 553, 556, 557, 569, 573, 581, 589, 593, 596, 597, 601, 604, 605, 612, 613, 617, 628, 632, 633, 637, 641, 644, 648, 649, 652, 653, 656, 657, 661, 664, 668, 669, 673, 677, 681, 688, 692, 701, 708, 709, 713, 716, 717, 721, 724, 737, 749, 752, 753, 757, 764, 769, 772, 773, 781, 789, 796, 797, 801, 804, 809, 813, 821, 824, 829, 833, 836, 844, 848, 849, 852, 853, 856, 857, 868, 869, 873, 877, 881, 889, 893, 908, 909, 913, 917, 921, 929, 932, 933, 937, 941, 944, 948, 953, 956, 964, 968, 972, 973, 976, 977, 980, 981, 989, 996, 997
2	40, 60, 65, 85, 96, 104, 105, 120, 136, 140, 156, 160, 165, 168, 185, 192, 200, 204, 205, 220, 221, 224, 232, 240, 252, 260, 264, 265, 273, 280, 285, 288, 296, 300, 305, 312, 320, 325, 336, 340, 345, 348, 352, 357, 360, 364, 365, 377, 380, 384, 385, 408, 416, 420, 424, 425, 429, 440, 444, 448, 456, 460, 465, 476, 481, 485, 488, 492, 493, 504, 525, 528, 533, 540, 545, 552, 560, 561, 565, 572, 584, 585, 588, 600, 608, 609, 616, 620, 629, 636, 640, 645, 660, 665, 684, 685, 693, 696, 700, 704, 705, 712, 720, 725, 728, 732, 736, 740, 741, 744, 745, 748, 760, 765, 768, 776, 792, 800, 805, 808, 812, 816, 820, 825, 828, 832, 845, 860, 861, 864, 865, 872, 884, 885, 888, 912, 920, 925, 928, 936, 945, 949, 952, 957, 965, 969, 984, 988, 992, 1000
3	148, 229, 257, 316, 321, 404, 469, 473, 564, 568, 592, 621, 733, 756, 761, 788, 837, 892, 916, 993
4	145, 328, 396, 445, 480, 505, 520, 544, 580, 624, 672, 680, 689, 777, 780, 793, 840, 876, 880, 896, 897, 901, 905, 924, 960
5	401, 817
6	697, 785, 940, 985
7	577
8	904

Die folgende Tabelle enthält die Anzahl der Diskriminanten $D \leq 1000$ mit einer bestimmten Klassenzahl.

h	1	2	3	4	5	6	7	8
Anzahl der D's mit $h(D) = h$ und $D \leq 1000$	266	150	20	25	2	4	1	1

Beispiel: Wir betrachten nun alle Diskriminanten $D \leq 10000$. In der folgenden Tabelle bezeichnet $\#D$ die Anzahl der D's ≤ 10000 mit $h(D) = h$.

h	1	2	3	4	5	6	7	8	9	10	11	12	13	14	16	18	20	21	27
$\#D$	1864	1597	266	697	61	166	30	129	13	24	4	31	2	6	3	3	2	1	1

Nachfolgend sind alle bzw. die ersten D's ≤ 10000 mit $h(D) = h$ angegeben.

h	D mit $h(D) = h$ und $1 \leq D \leq 10000$
1	5, 8, 12, 13, 17, 20, 21, 24, 28, 29, 32, 33, 37, 41, 44, 45, 48, 52, 53, 56, 57, 61, 68, 69, 72, ...
2	40, 60, 65, 85, 96, 104, 105, 120, 136, 140, 156, 160, 165, 168, 185, 192, 200, 204, 205, 220, ...
3	148, 229, 257, 316, 321, 404, 469, 473, 564, 568, 592, 621, 733, 756, 761, 788, 837, 892, 916, ...
4	145, 328, 396, 445, 480, 505, 520, 544, 580, 624, 672, 680, 689, 777, 780, 793, 840, 876, 880, ...
5	401, 817, 1093, 1393, 1429, 1604, 1641, 1756, 1897, 1996, 2081, 2153, 2908, 3121, 3181, ...
6	697, 785, 940, 985, 1300, 1345, 1384, 1425, 1708, 1765, 1825, 1937, 1940, 2024, 2233, 2256, ...
7	577, 1009, 1601, 1761, 2029, 2308, 2913, 4036, 4229, 4348, 5176, 5193, 5273, 5417, 6404, ...
8	904, 1596, 1705, 1768, 1785, 2496, 2584, 2605, 2705, 3081, 3196, 3201, 3360, 3393, 3480, ...
9	1129, 2889, 3137, 4409, 4516, 5521, 6616, 6809, 7573, 7604, 7873, 8404, 8937
10	3129, 3585, 4097, 4321, 4444, 4625, 4865, 4904, 5777, 6085, 6945, 7000, 7049, 7221, 7353, ...
11	1297, 4009, 5188, 6081
12	2920, 3604, 4345, 4360, 4641, 5629, 5980, 6088, 6396, 6401, 6856, 7084, 7224, 7540, 7665, ...
13	8101, 8441
14	6097, 6136, 8104, 8185, 8465, 9505
16	2305, 8145, 9220
18	7060, 7465, 9217
20	3601, 5185
21	7057
27	8761

Die Beispiele machen folgende Vermutung plausibel, die bereits Gauß vermutet hat, die aber bis heute nicht bewiesen ist:

Vermutung: Es gibt unendlich viele reellquadratische Diskriminanten mit Klassenzahl 1.

Das folgende Beispiel dient nochmals der Illustration dieser Vermutung.

Beispiel: Wir betrachten die Diskriminanten zwischen 10^6 und 10^6+100 und berechnen die zugehörigen Klassenzahlen.

D	$h(D)$	D	$h(D)$	D	$h(D)$	D	$h(D)$
10^6+1	94	10^6+28	1	10^6+53	1	10^6+80	60
10^6+4	52	10^6+29	2	10^6+56	2	10^6+81	1
10^6+5	16	10^6+32	32	10^6+57	4	10^6+84	1
10^6+8	48	10^6+33	1	10^6+60	8	10^6+85	2
10^6+9	2	10^6+36	4	10^6+61	3	10^6+88	2
10^6+12	2	10^6+37	1	10^6+64	24	10^6+89	3
10^6+13	8	10^6+40	28	10^6+65	8	10^6+92	6
10^6+16	27	10^6+41	2	10^6+68	1	10^6+93	12
10^6+17	4	10^6+44	4	10^6+69	37	10^6+96	2
10^6+20	36	10^6+45	4	10^6+72	12	10^6+97	1
10^6+21	7	10^6+48	4	10^6+73	2	10^6+100	32
10^6+24	1	10^6+49	2	10^6+76	8		
10^6+25	64	10^6+52	1	10^6+77	2		

Bemerkung: Wir hatten früher bewiesen: Ist $\alpha \in \mathrm{Red}(D)$, hat die Kettenbruchentwicklung von α Periodenlänge k und sind $\frac{p_i}{q_i}$ die Näherungsbrüche, so ist

$$\varepsilon = q_{k-2} + q_{k-1}\alpha$$

eine Grundeinheit der Ordnung R_D^*. Ist der Zyklus $Z(\alpha)$ kurz, so wird ε im allgemeinen klein sein, ist der Zyklus $Z(\alpha)$ lang, so wird ε im allgemeinen groß sein. Dies wird auch durch das folgende Beispiel illustriert.

Beispiel: Für $D = 10001$ ist $h(D) = 16$, die zu den Zykeln gehörigen Kettenbruchentwicklungen sind

$$(1,1,1,2,1,4,3),\quad (1,1,1,24,2),\quad (1,1,99),\quad (1,19,9),\quad (1,1,4,2,9),\quad (1,1,1,2,24),$$
$$(1,1,9,2,4),\quad (1,1,1,3,4,1,2),\quad (1,2,4,1,11),\quad (1,4,2,1,11),\quad (1,3,49),\quad (1,49,3),$$
$$(1,2,2,7,3),\quad (1,3,7,2,2),\quad (1,5,2,2,5),\quad (1,9,19).$$

Für $D = 10009$ ist $h(D) = 1$, es gibt nur eine zugehörige Kettenbruchentwicklung:

$$(1,1,1,1,2,2,1,19,3,3,1,1,11,1,15,1,3,16,2,2,1,1,1,3,1,2,1,1,4,2,2,1,7,1,1,1,2,7,1,24,7,$$
$$1,1,1,9,2,1,5,4,1,4,1,3,49,1,3,5,3,3,1,5,1,9,6,1,1,3,5,1,32,1,1,32,1,5,3,1,1,6,9,1,5,$$
$$1,3,3,5,3,1,49,3,1,4,1,4,5,1,2,9,1,1,1,7,24,1,7,2,1,1,1,7,1,2,2,4,1,1,2,1,3,1,1,1,2,2,$$
$$16,3,1,15,1,11,1,1,3,3,19,1,2,2,1,1,1,1,10,1,1,99,1,1,10).$$

Zugehörig finden wir Grundeinheiten der jeweiligen Ordnungen:

$$\varepsilon_{10001} = 99 + 2\frac{1+\sqrt{10001}}{2} = 100 + \sqrt{10001}.$$

$$\begin{aligned}
\varepsilon_{10009} &= e_1 + e_2\frac{1+\sqrt{10009}}{2} \quad \text{mit}\\
e_1 &= 564930594544367831338736883495413861310897432399313531361300910399 4267,\\
e_2 &= 114075551975191638356315362974662605483538277625210425968307215789530.
\end{aligned}$$

4. Übungen

Aufgabe 8.1: Definiere

$$f(x) = \sum_{\substack{p\text{prim} \\ p\equiv 1 \bmod 4 \\ p\leq x}} h(p).$$

Untersuche $f(x)$ experimentell und stelle eine Vermutung auf. (Es gibt dazu eine Vermutung von Cohen-Lenstra.)

Aufgabe 8.2: In jedem Zyklus $Z(\alpha)$ reduzierter reellquadratischer Zahlen gibt es ein $\alpha = \frac{b+\sqrt{D}}{2a}$ mit $a \leq \sqrt{\frac{D}{5}}$.

Aufgabe 8.3: Sei D eine reellquadratische Diskriminante.

(1) Ist $\alpha \in \mathrm{Red}(D)$, so auch gilt auch $-\frac{1}{\alpha'} \in \mathrm{Red}(D)$. Stelle nochmals die Aussagen über die Beziehungen der Kettenbruchentwicklungen und Normalformen von α und $-\frac{1}{\alpha'}$ zusammen.

(2) In der Klassengruppe $\mathrm{H}(D)$ gilt

$$(\mathbf{Z} + \mathbf{Z}\alpha)(\mathbf{Z} + \mathbf{Z}(-\frac{1}{\alpha'})) \sim R_D$$

und damit

$$Z(\alpha)^{-1} = Z(-\frac{1}{\alpha'}).$$

(3) Charakterisiere einen Zyklus $Z \subseteq \mathrm{Red}(D)$ mit $Z = Z^{-1}$ durch Kettenbrüche.

ANHANG A

Programme

1. algo1_ma

```
# algo1_ma
# Kapitel: Der euklidische Algorithmus
# Version: 8.2.2002
# Funktionen: EA_tex(a_0,a_1)

# EA_tex(a_0,a_1) fuehrt den euklidischen und den erweiterten
# euklidischen Algorithmus aus und gibt die Ergebnisse in
# TeX-Format aus.
# Eingabe: a_0, a_1
# Ausgabe: a_(i-1)=q_i*a_i+a_(i+1) fuer 1<=i<=n
#          a_i=x_i*a_0+y_i*a_1
EA_tex:=proc()
  local a, q, i, n, x, y;
  a:=array(0..1000); q:=array(0..1000);
  x:=array(0..1000); y:=array(0..1000);

  a[0]:=args[1]; a[1]:=args[2]; x[0]:=1; x[1]:=0; y[0]:=0; y[1]:=1;

  for n from 1 to 1000 do
    q[n-1]:=iquo(a[n-1],a[n]);
    a[n+1]:=irem(a[n-1],a[n]);
    if a[n+1]=0 then break; fi;
  od;

  for i from 2 to n do
    x[i]:=x[i-2]-q[i-2]*x[i-1];
    y[i]:=y[i-2]-q[i-2]*y[i-1];
  od;

  printf("Der euklidische Algorithmus zur Berechnung von\n");
  printf("$\\ggT(%d,%d)$ ",a[0],a[1]);
  printf(" ben\"otigt %d Divisionen:\n",n);
  printf("\\begin{eqnarray*}\n");
  for i from 1 to n do
    printf("%d &=& %d \\cdot %d + %d\\\\\n",a[i-1],q[i-1],a[i],a[i+1]);
  od;
  printf("\\end{eqnarray*}\n");
  printf("Es ist $\\ggT(%d,%d)=%d$.\n\n",a[0],a[1],a[n]);

  printf("Der erweiterte euklidische Algorithmus zur Berechnung von ");
  printf("$\\ggT(a_0,a_1)$ mit \n$a_0=%d$, $a_1=%d$\n",a[0],a[1]);
  printf("liefert Relationen $a_i=x_ia_0+y_ia_1$, ");
```

Version vom 12.2.2002

```
  printf("wobei $a_{%d}=\\ggT(a_0,a_1)$ ist:\n",n);
  printf("$$\\begin{tabular}{|c|c|c|c|}\\hline\n");
  printf("$i$ & $a_i$ & $x_i$ & $y_i$\\\\\\hline\n");
  for i from 0 to n do
    printf("%d & %d & %d & %d\\\\\\hline\n",i,a[i],x[i],y[i]);
  od;
  printf("\\end{tabular}$$\n");
end;
```

2. algo2_ma

```
# algo2_ma
# Kapitel: Schnelle Exponentiation - Die square-and-multiply-Methode
# Version: 8.2.2002
# Funktionen:
# smm_tex(a,d,n)
# ft2(e,n)

# Die sqare-and-multiply-Methode zur Berechnung von a^d mod n
# Eingabe: a, d, n
# Ausgabe: Zwischenergebnisse nach Vorlesungsskript im TeX-Format
smm_tex:=proc()
  local a, dd, d, n, r, x, y, c, i;
  a:=args[1]; dd:=args[2]; n:=args[3];
  r:=0; while 2^(r+1)<=dd do r:=r+1; od; # r=log_2(dd)
  x:=array(0..r); y:=array(0..r); c:=array(0..r); d:=array(0..r);

  c[0]:=dd; d[0]:=c[0] mod 2; x[0]:=a;
  if d[0]=0 then y[0]:=1; else y[0]:=a; fi;

  for i from 1 to r do
    c[i]:=iquo(c[i-1],2);
    d[i]:=c[i] mod 2;
    x[i]:=x[i-1]*x[i-1] mod n;
    if d[i]=0 then
      y[i]:=y[i-1];
    else
      y[i]:=y[i-1]*x[i] mod n;
    fi;
  od;

  printf("Berechnung von $%d^{%d}\\bmod %d$:\n",a,dd,n);
  printf("$$\\begin{tabular}{c|c|c|c|c}\n");
  printf("$i$ & $c_i$ & $d_i$ & $x_i$ & $y_i$\\\\\\ \\hline\n");
  for i from 0 to r do
    printf("%d & %d & %d & %d & %d\\\\\\ \n",i,c[i],d[i],x[i],y[i]);
  od;
  printf("\\end{tabular}$$\n");
  printf("Ergebnis: $%d^{%d}\\equiv %d\\bmod %d$.\n",a,dd,y[r],n);

  y[r];
end;

# ft2(e,n) - Version vom 27.10.2001
```

```
# Eingabe: e, n
# Ausgabe: die n groessten wahrscheinlichen Primzahlen mit n
#          Dezimalstellen, wobei ein Fermattest zur Basis 2 gemacht
#          wird.
ft2:=proc()
  local e, n, n0, zeit1, p, r;
  if nargs<>2 then error "Aufruf: ft2(e,n)"; return; fi;
  e:=args[1]; n:=args[2]; n0:=0;
  printf("Die %d groessten wahrscheinlichen Primzahlen mit %d ",n,e);
  printf("Dezimalstellen werden \nbestimmt. ");
  printf("Dabei wird jeweils ein Fermattest zur Basis 2 gemacht.\n");

  zeit1:=time();
  r:=1; p:=10^e-r;

  while n0<n do
    if Power(2,p-1) mod p=1 then
      printf("p=10^%d-%d\n",e,r);
      n0:=n0+1;
    fi;
    r:=r+2; p:=p-2;
  od;

  printf("Zeit: %f\n",time()-zeit1);
end;
```

3. ft2_gmp.c

```
/* ft2_gmp.c - Version vom 27.10.2001
   Die n groessten wahrscheinlichen Primzahlen mit e Dezimalstellen
   werden bestimmt. Dabei wird ein Fermattest zur Basis 2 gemacht.
   Das Potenzieren erfolgt mit der Funktion mpz_powm.
   Uebersetzung mit gcc ft2_gmp.c -lgmp
*/

#include <stdio.h>
#include <time.h>
#include <gmp.h>

main()
{
  int e, n, n0=0, r;
  mpz_t p, p1, b, z;
  time_t zeit1, zeit2;

  mpz_init(p); mpz_init(p1); mpz_init_set_ui(b,2); mpz_init(z);

  printf("Die n groessten wahrscheinlichen Primzahlen mit e ");
  printf("Dezimalstellen werden \nbestimmt. ");
  printf("Dabei wird jeweils ein Fermattest zur Basis 2 gemacht.\n");
  printf("e="); scanf("%d",&e);
  printf("n="); scanf("%d",&n); n0=0;

  time(&zeit1);
  mpz_ui_pow_ui(p,10,e); r=1; mpz_sub_ui(p,p,r);
```

```
  while (n0<n)
  {
    mpz_sub_ui(p1,p,1); mpz_powm(z,b,p1,p); /* z=2^(p-1) mod p */
    if (mpz_cmp_ui(z,1)==0)
    {
      printf("p=10^%d-%d\n",e,r);
      n0++;
    }
    r=r+2; mpz_sub_ui(p,p,2);
  }

  time(&zeit2); printf("Zeit: %.0f sec\n",difftime(zeit2,zeit1));
}
```

4. ft2_ntl.c

```
/* ft2_ntl.c - Version vom 27.10.2001
   Die n groessten wahrscheinlichen Primzahlen mit e Dezimalstellen
   werden bestimmt. Dabei wird ein Fermattest zur Basis 2 gemacht.
   Uebersetzung mit gcc ft2_ntl.c -lntl -lgmp
*/

#include <NTL/ZZ.h>
#include <time.h>

int main()
{
  long e, n, n0=0, r;
  ZZ p, b;
  time_t zeit1, zeit2;
  b=2;

  cout<<"Die n groessten wahrscheinlichen Primzahlen mit e ";
  cout<<"Dezimalstellen werden \nbestimmt. ";
  cout<<"Dabei wird jeweils ein Fermattest zur Basis 2 gemacht.\n";
  cout<<"e="; cin>>e;
  cout<<"n="; cin>>n; n0=0;

  time(&zeit1);

  r=1; p=power_ZZ(10,e)-r;

  while (n0<n)
  {
    if (PowerMod(b,p-1,p)==1)
    {
      cout<<"p=10^"<<e<<"-"<<r<<"\n";
      n0++;
    }
    r=r+2; p=p-2;
  }

  time(&zeit2); cout<<"Zeit: "<<difftime(zeit2,zeit1)<<" sec\n";
  double zeit=GetTime();
```

```
  cout<<"Zeit: "; PrintTime(cout,zeit); cout<<"\n";
}
```

5. ft2.java

```
/* ft2.java - 28.10.2001
   Die n groessten wahrscheinlichen Primzahlen mit e Dezimalstellen
   werden bestimmt. Dabei wird ein Fermattest zur Basis 2 gemacht.
   Uebersetzung mit javac ft2.java
*/

import java.math.*;
import java.util.Date;

class ft2
{
  public static void main( String args[])
  {
    Date zeit1 = new Date();

    System.out.print(
      "Die n groessten wahrscheinlichen Primzahlen mit e " +
      "Dezimalstellen werden \nbestimmt. " +
      "Dabei wird jeweils ein Fermattest zur Basis 2 gemacht.\n");

    int e = Integer.parseInt( args[0]);
    int n = Integer.parseInt( args[1]); int n0=0;
    BigInteger b = BigInteger.valueOf(2);

    int r=1;
    BigInteger p = BigInteger.valueOf(10);
    p = p.pow(e);
    p = p.subtract( BigInteger.valueOf(r)); // p=10^e-r

    BigInteger eins = BigInteger.valueOf(1);
    BigInteger zwei = BigInteger.valueOf(2);

    while (n0<n)
    {
      if ((b.modPow(p.subtract(eins),p)).compareTo(eins)==0)
      {
        System.out.print("p=10^"+e+"-"+r+"\n");
        n0++;
      }
      r=r+2; p=p.subtract(zwei);
    }
    Date zeit2 = new Date();

    System.out.print(
      "Zeit: " + (zeit2.getTime()-zeit1.getTime())/1000
      + " sec\n");
  }
}
```

6. pp_ntl.c

```
/* pp_ntl.c - Anzahlen von Pseudoprimzahlen
   Uebersetzung mit g++ pp_ntl.c -lntl -lgmp
   Version vom 27.10.2001
*/

#include <NTL/ZZ.h>

int main()
{
  long azp=0, az2=0, az3=0, az5=0, az7=0, n_max;

  cout<<"Anzahlen von Pseudoprimzahlen <n_max:\n";
  cout<<"n_max="; cin>>n_max;

  for (int n=2;n<n_max;n++)
  {
    if (ProbPrime(n)==1)
      azp++;
    else
    {
      if (PowerMod(2,n-1,n)==1) az2++;
      if (PowerMod(3,n-1,n)==1) az3++;
      if (PowerMod(5,n-1,n)==1) az5++;
      if (PowerMod(7,n-1,n)==1) az7++;
    }
  }
  cout<<"Primzahlen: "<<azp<<"\n";
  cout<<"Pseudoprimzahlen zur Basis 2: "<<az2<<"\n";
  cout<<"Pseudoprimzahlen zur Basis 3: "<<az3<<"\n";
  cout<<"Pseudoprimzahlen zur Basis 5: "<<az5<<"\n";
  cout<<"Pseudoprimzahlen zur Basis 7: "<<az7<<"\n";
}
```

7. pzsu_ntl.c

```
/* pzsu_ntl.c - Version vom 27.10.2001
   Die n groessten wahrscheinlichen Primzahlen mit e Dezimalstellen
   werden bestimmt. Dabei werden zunaechst kleine Primteiler
   ausgeschlossen, dann wird ein Fermattest zur Basis 2 gemacht.
   Uebersetzung mit gcc pzsu_ntl.c -lntl -lgmp
*/

#include <NTL/ZZ.h>
#include <time.h>

int main()
{
  long e, m, M, n, n0=0, r, t, kT;
  ZZ p, b;
  time_t zeit1, zeit2;
  b=2;

  cout<<"Die n groessten wahrscheinlichen Primzahlen mit e ";
  cout<<"Dezimalstellen werden \nbestimmt. ";
```

```
  cout<<"Dabei werden zunaechst kleine Primteiler <10^m ";
  cout<<"ausgeschlossen, dann \nwird ein Fermattest zur Basis 2 ";
  cout<<"gemacht.\n";

  cout<<"e="; cin>>e;
  cout<<"m="; cin>>m; M=power_long(10,m);
  cout<<"n="; cin>>n; n0=0;

  time(&zeit1);

  r=1; p=power_ZZ(10,e)-r;

  while (n0<n)
  {
    cout<<"10^"<<e<<"-"<<r<<": ";
    kT=0;
    for (t=3;t<M;t=t+2)
      if (p%t==0)
      {
        cout<<t<<"\n";
        kT=1; break;
      }

    if (kT==0)
    {
      if (PowerMod(b,p-1,p)==1)
      {
        cout<<"Fermat-Test zur Basis 2 bestanden\n";
        n0++;
      }
      else
        cout<<"Fermat-Test: negativ\n";
    }

    r=r+2; p=p-2;
  }

  time(&zeit2); cout<<"Zeit: "<<difftime(zeit2,zeit1)<<" sec\n";
  double zeit=GetTime();
  cout<<"Zeit: "; PrintTime(cout,zeit); cout<<"\n";
}
```

8. spp_ntl.c

```
/* spp_ntl.c - Starke Pseudoprimzahlen - 29.10.2001, 8.2.2002 */

#include <NTL/ZZ.h>

int MillerRabin( ZZ n, ZZ a)
{
  ZZ q=n-1;
  int l=MakeOdd(q); // n-1=2^l*q
  ZZ b=PowerMod(a,q,n);
  if (b==1) return 1;
  for (int i=0; i<l; i++)
```

```
  {
    if (b==n-1) return 1;
    b=(b*b)%n;
  }
  return 0;
}

int main()
{
  ZZ n, n_max, a;
  cout<<"Bestimme der starken Pseudoprimzahlen <n_max zur Basis a:\n";
  cout<<"n_max="; cin>>n_max; cout<<"a="; cin>>a;

  for (n=3; n<n_max; n=n+2)
    if (ProbPrime(n)==0)
    {
      if (MillerRabin(n,a)==1)
        cout<<n<<"\n";
    }
}
```

9. rsa_ma

```
# ‘bf:=convert(zk,bytes);’ wandelt eine Zeichenkette ‘zk’ in eine
# Bytefolge ‘bf’ um.
# ‘bf:=readbytes("datei",infinity);’ liefert die Bytefolge ‘bf’ der
# Datei ‘datei’.
# ‘zk:=convert(bf,bytes);’ wandelt eine Bytefolge ‘bf’ in eine
# Zeichenkette ‘zk’ um.
# writebytes("datei",bf);’ schreibt die Bytefolge ‘bf’ in eine Datei
# ‘datei’.

rsa_encrypt:=proc(bytefolge,N,e)
  local bl, bf, zuviel, i, zf, a, j, b;
  # bl ist die Blocklaenge
  bl:=0; while 256^(bl+1)<N do bl:=bl+1; od; if bl>255 then bl:=255; fi;

  bf:=bytefolge;
  # Die Bytefolge wird ergaenzt, damit Anzahl=0 mod Blocklaenge ist.
  # ‘zuviel’ Bytes muessen spaeter wieder weggestrichen werden.
  zuviel:=bl-(nops(bf) mod bl);
  for i from 1 to zuviel-1 do bf:=[op(bf),10]; od;
  bf:=[op(bf),zuviel];

  zf:=[]; # zf wird die auszugebende Zahlenfolge
  for i from 1 to nops(bf)/bl do
    # jeweils bl Bytes werden in eine Zahl a umgewandelt
    a:=0;for j from 1 to bl do a:=256*a+bf[(i-1)*bl+j]; od;
    b:=Power(a,e) mod N; # Verschluesselung b=a^e mod N
    zf:=[op(zf),b];
  od;
  zf;
end;

rsa_decrypt:=proc(zahlenfolge,N,d)
```

```
  local bl, bf, i, b, a, bf1, j, zuviel;
  bl:=0; while 256^(bl+1)<N do bl:=bl+1; od; if bl>255 then bl:=255; fi;

  bf:=[]; # bf wird die auszugebende Bytefolge
  for i from 1 to nops(zahlenfolge) do
    b:=zahlenfolge[i];
    a:=Power(b,d) mod N;
    bf1:=[]; # a wird in die bl-elementige Bytefolge bf1 umgewandelt
    for j from 1 to bl do
      bf1:=[a mod 256,op(bf1)];
      a:=iquo(a,256);
    od;
    bf:=[op(bf),op(bf1)];
  od;
  zuviel:=bf[nops(bf)]; # 'zuviel' Bytes werden weggestrichen
  for i from 1 to zuviel do bf:=subsop(nops(bf)=NULL,bf); od;
  bf;
end;
```

10. kb_ma

```
# kb_ma: Maple-Programme in Zusammenhang mit Kettenbruechen
# Version vom 14.7.2001
#
# 1. kbe
#    Kettenbruchentwicklung einer rationalen Zahl
# 2. kbe_intervall
#    Kettenbruchentwicklung der Zahlen zwischen a und b
# 3. kb2r
#    Umwandlung eines Kettenbruchs in eine rationale Zahl
# 4. kbe_nb
#    Kettenbruchentwicklung einer rationalen Zahl mit Naeherungsbruechen
# 5. kbe_wd
#    Kettenbruchentwicklung von sqrt(d)
# 6. kb_rsa
#    Versuch, N mit dem oeffentlichen RSA-Schluessel (N,e) zu
#    faktorisieren

# Kettenbruchentwicklung einer rationalen Zahl
kbe:=proc()
  b0:=numer(args[1]); b1:=denom(args[1]); k:=[];
  while b1>0 do
    b2:=b0 mod b1; a:=(b0-b2)/b1;
    k:=[op(k),a];
    b0:=b1; b1:=b2;
  od;
  k;
end;

kbe_intervall:=proc()
  printf("Kettenbruchentwicklung der Zahlen zwischen %a und %a\n",
         args[1],args[2]);
  a:=kbe(args[1]); b:=kbe(args[2]);
  printf("Kettenbruchentwicklung von %a: %a\n",args[1],a);
  printf("Kettenbruchentwicklung von %a: %a\n",args[2],b);
```

```
  k:=[]; i:=1;
  while i<=nops(a) and i<=nops(b) and a[i]=b[i] do
    k:=[op(k),a[i]]; i:=i+1;
  od;
  if i>nops(a) then al:=b[i]; ar:=infinity;
  elif i>nops(b) then al:=a[i]; ar:=infinity;
  else al:=min(a[i],b[i]); ar:=max(a[i],b[i]);
  fi;
  printf("Gemeinsame Teilquotienten: %a\n",k);
  printf("Der naechste Teilquotient liegt zwischen %a und %a\n",
         al,ar);
end;

# Umwandlung eines Kettenbruchs in eine rationale Zahl
kb2r:=proc()
  k:=args[1];
  while nops(k)>1 do
    printf("%a\n",k);
    i:=nops(k)-1;
    k:=subsop(i=k[i]+1/k[i+1],k);
    k:=subsop(i+1=NULL,k);
  od;
  op(k);
end;

# Kettenbruchentwicklung einer rationalen Zahl mit Naeherungsbruechen
kbe_nb:=proc()
  b0:=numer(args[1]); b1:=denom(args[1]); aa:=[]; pp:=[]; qq:=[];
  p_2:=0; p_1:=1;
  q_2:=1; q_1:=0;
  while b1>0 do
    b2:=b0 mod b1; a:=(b0-b2)/b1; b0:=b1; b1:=b2; aa:=[op(aa),a];
    p:=a*p_1+p_2; p_2:=p_1; p_1:=p; pp:=[op(pp),p];
    q:=a*q_1+q_2; q_2:=q_1; q_1:=q; qq:=[op(qq),q];
  od;
  [aa,pp,qq]
end;

# Kettenbruchentwicklung von sqrt(d)
kbe_wd:=proc()
  d:=args[1];
  wd:=isqrt(d); if wd^2>d then wd:=wd-1; fi;
  aa:=[wd]; b:=wd; c:=d-wd^2; # Beginn mit b_1 und c_1
  while c>1 do
    a:=iquo(b+wd,c); aa:=[op(aa),a];
    b:=a*c-b;
    c:=(d-b^2)/c;
  od;
  a:=iquo(b+wd,c); aa:=[op(aa),a];
  aa;
end;

# kb_rsa versucht aus dem oeffentlichen RSA-Schluessel (N,e) mittels
# Kettenbruechen die Faktorisierung N=pq zu bestimmen. Gelingt dies,
```

```
# wird [p,q,d] mit dem privaten Schluessel d ausgegeben, sonst [].
kb_rsa:=proc()
  local N, e;
  N:=args[1]; e:=args[2];
  w4N:=isqrt(4*N); if w4N^2>4*N then w4N:=w4N-1; fi;
  b0:=e; b1:=N-w4N;
  ki:=1; ki1:=0;
  di:=0; di1:=1;
  i:=-1;
  while b1>0 do
    i:=i+1;
    b2:=b0 mod b1; ai:=(b0-b2)/b1; b0:=b1; b1:=b2;
    k:=ai*ki+ki1; ki1:=ki; ki:=k;
    d:=ai*di+di1; di1:=di; di:=d;
    if i>0 and (e*di-1) mod ki=0 then
      si:=N+1-(e*di-1)/ki;
      Di:=si^2-4*N;
      if issqr(Di) then
        printf("e=%d d%d=%d k%d=%d s%d=%d\n",e,i,di,i,ki,i,si);
        p:=(si-isqrt(Di))/2; q:=(si+isqrt(Di))/2;
        printf("ln(d)/ln(N)=%f\n",ln(di)/ln(N));
        return([p,q,di]);
      fi;
    fi;
  od;
  [];
end;
```

11. algo4_ma

```
# algo4_ma
# Kapitel: Periodische Kettenbrueche und reellquadratische Zahlen
# Version: 8.2.2002
# Funktionen:
# qz_abD(q1,q2,d)
# kbe_uvwd(u,v,w,d)
# Red(D)
# kbe_wd(d)
# pell(d)

unprotect(D);

# Umwandlung einer quadratischen Zahl alpha=q_1+q_2*sqrt(d) in
# Diskriminantenform (b+sqrt(D))/(2*a): alpha genuegt der Gleichung
# alpha^2-2*q1*alpha+(q1^2-d*q2^2)=0, multipliziert man mit dem Nenner,
# so erhaelt man a*alpha^2-b*alpha+c=0.
# Die Zahl d muss nicht quadratfrei sein.
# Eingabe: q_1, q_2, d
# Ausgabe: [a,b,D]
qz_abD:=proc()
  local q1, q2, d, a, b, c, D;
  q1:=args[1]; q2:=args[2]; d:=args[3];
  b:=2*q1; c:=q1^2-d*q2^2;
  a:=ilcm(denom(b),denom(c)); if q2<0 then a:=-a; fi;
  b:=a*b; c:=a*c; D:=b^2-4*a*c;
```

```
  [a,b,D];
end;

# Kettenbruchentwicklung von (u+v*sqrt(d))/w fuer Vorlesungsskript
# Eingabe: u, v, w, d
# Ausgabe: Kettenbruchentwicklung von (u+v*sqrt(d))/w in TeX-Format
kbe_uvwd_tex:=proc()
  local u, v, w, d, z, D, F, a, b, c, aa, bb, cc, g, wD, alpha,
        alphap, m, uu, vv, ww, zz, i, U, abD;
  u:=args[1]; v:=args[2]; w:=args[3]; d:=args[4];
  if issqr(d) then error "%1 ist eine Quadratzahl!", d; fi;
  if d<0 then error "%1 ist negativ!", d; fi;
  abD:=qz_abD(u/w,v/w,d); a:=abD[1]; b:=abD[2]; D:=abD[3];
  m:=isqrt(D/d); c:=(b^2-D)/(4*a);
  printf("$\\frac{%d+%d\\sqrt{%d}}{%d}$ ",u,v,d,w);
  printf("hat Diskriminante %d\n",D);
  wD:=isqrt(D); if wD^2>D then wD:=wD-1; fi;
  F:={}; g:={[a,b,c]}; i:=0; U:=[];

  printf("$$\\begin{tabular}{|c|c|c|c|c|c|c|}\\hline \n");
  printf("$i$ & $a$ & $b$ & $c$ & $\\alpha_i$ & $u_i$ & $\\alpha_i$ ");
  printf("& $\\alpha'_i$\\\\ \\hline\n");

  while F<>F union {g} do
    F:=F union {g};
    alpha:=(b+m*sqrt(d))/(2*a); alphap:=(b-m*sqrt(d))/(2*a);
    if a>0 then u:=floor((b+wD)/(2*a));
      else u:=floor((b+wD+1)/(2*a));
    fi;
    U:=[op(U),u];
    uu:=b; vv:=m; ww:=2*a; zz:=igcd(uu,vv,ww);
    uu:=uu/zz; vv:=vv/zz; ww:=ww/zz;
    if ww<0 then uu:=-uu; vv:=-vv; ww:=-ww; fi;
    printf("%d & %d & %d & %d & ",i,a,b,c);
    printf("$\\frac{%d+%d\\sqrt{%d}}{%d}$ & ",uu,vv,d,ww);
    printf("%d & ",u);
    printf("%1.3f & %1.3f\\\\ \\hline\n",evalf(alpha),evalf(alphap));

    aa:=-a*u^2+b*u-c;
    bb:=2*a*u-b;
    cc:=-a;
    a:=aa; b:=bb; c:=cc; g:=[a,b,c]; i:=i+1;
  od;
  printf("\\end{tabular}$$\n");
  printf("Wiederholung mit (%d,%d,%d)\n",a,b,c);
  printf("Kettenbruchentwicklung: %a\n",U);
end;

# Auflistung der reduzierten reellquadratischen Zahlen
# alpha=(b+sqrt(D))/(2*a) mit Diskriminante D in der Gestalt [a,b,D]
# Eingabe: D
# Ausgabe: Menge mit reduzierten [a,b,D]
Red:=proc()
  local D, wD, R, b, mac, a;
```

```
  D:=args[1];
  if D<=0 or isdisc(D)=false then error "D ist keine Diskriminante"; fi;
  wD:=isqrt(D); if wD^2>D then wD:=wD-1; fi;
  R:={};
  for b from 2-(D mod 2) by 2 to wD do
    mac:=(D-b^2)/4;
    for a from iquo(wD-b,2)+1 to iquo(wD+b,2) do
      if mac mod a=0 and igcd(a,b,mac/a)=1 then
        R:=R union {[a,b,D]};
      fi;
    od;
  od;
  R;
end;

# Kettenbruchentwicklung von sqrt(d)
# Eingabe: d
# Ausgabe: [u_0,u_1,...,u_k], wobei dann
#          [u_0,u_1,...,u_k,u_1,...,u_k,u_1,...,u_k,...] die
#          Kettenbruchentwicklung von sqrt(d) ist
kbe_wd:=proc()
  local d, wd, v, a, u, kbe;
  d:=args[1];
  wd:=isqrt(d); if wd^2>d then wd:=wd-1; fi;
  v:=0; a:=1; kbe:=[];
  do
    u:=iquo(v+wd,a); kbe:=[op(kbe),u];
    v:=a*u-v;
    a:=(d-v^2)/a;
    if a=1 then break; fi;
  od;
  kbe:=[op(kbe),2*wd];
end;

# Kleinste Loesung der Pellschen Gleichung x^2-d*y^2=1 mit x,y>0
# Eingabe: d
# Ausgabe: [x,y]
pell:=proc()
  local d, wd, v, a, u, p0, p1, p2, q0, q1, q2, k, x, y;
  d:=args[1];
  wd:=isqrt(d); if wd^2>d then wd:=wd-1; fi;
  v:=0; a:=1;
  p2:=0; p1:=1;
  q2:=1; q1:=0;
  k:=0;
  do
    u:=iquo(v+wd,a);
    p0:=u*p1+p2; p2:=p1; p1:=p0;
    q0:=u*q1+q2; q2:=q1; q1:=q0;
    k:=k+1;
    v:=a*u-v;
    a:=(d-v^2)/a;
    if a=1 then break; fi;
  od;
```

```
  if k mod 2=0 then x:=p1; y:=q1; else x:=p1^2+d*q1^2; y:=2*p1*q1; fi;
  [x,y];
end;
```

12. rsa_bsp_gen_ma

```
# rsa_bsp_gen_ma - 9.2.2002
# Erzeugung von RSA-Beispielen gewuenschter Stellenzahl

Digits:=1000:

# st(N) liefert die Anzahl der Dezimalstellen von N
st:=proc() trunc(ln(args[1])/ln(10)+1); end;

# Z ist global definiert
Z:=78465867892689763498934564564626583765849378569384576927592759234 9;

# Naechste Zufallszahl
nzz:=proc()
  local A, B;
  A:=32647853264785632785623874653287645832764857236487526845213264510 1;
  B:=34756893782764859476893478276895656743768927658273647852634853624 1;
  return (A*args[1]+B) mod 2^256;
end;

rsabsp:=proc()
  local stz, p, q, z;
  global Z;
  z:=0;
  stz:=args[1];
  while z<>stz do
    Z:=nzz(Z); p:=Z mod 10^iquo(stz,2); p:=nextprime(p);
    Z:=nzz(Z); q:=Z mod 10^iquo(stz,2); q:=nextprime(q);
    z:=st(p*q);
  od;
  [p*q,p,q];
end;

# Beispiel: Erzeugung von 10 40-stelligen RSA-Zahlen
# zahl:=array(1..10);
# for i from 1 to 10 do zahl[i]:=rsabsp(40); od;
```

13. cfr_ma

```
# cfrac_bsp_ma
# Zum Erstellen von Beispielen fuer CRAC
# Eingabe: N, anzahl, Faktorbasis, Ausgabedatei,
#          wobei anzahl die Anzahl der Beispielzeilen angibt
cfrac_bsp_tex:=proc()
  local N, wN, anzahl, fb, i, u, v, P, a, rest, ex, j, aus, r, x, y;
  N:=args[1]; anzahl:=args[2]; fb:=args[3]; aus:=args[4];
  wN:=isqrt(N); if wN^2>N then wN:=wN-1; fi;
  u:=array(0..anzahl-1);
  v:=array(0..anzahl);
  P:=array(-2..anzahl-1);
  a:=array(0..anzahl);
```

```
  x:=array(0..anzahl);
  y:=array(0..anzahl);
  r:=-1;
  fprintf(aus,"$N=%d$\nFaktorbasis: $%a$\n",N,fb);
  fprintf(aus,"$$\\begin{tabular}{|c|c|c|c|c|c|}\\hline\n");
  P[-2]:=0; P[-1]:=1; v[0]:=0; a[0]:=1;
  for i from 0 to anzahl-1 do
    u[i]:=iquo(v[i]+wN,a[i]);  fprintf(aus,"$u_{%d}=%d$ & ",i,u[i]);
    P[i]:=(u[i]*P[i-1]+P[i-2]) mod N;
                               fprintf(aus,"$P_{%d}=%d$ & ",i,P[i]);
    v[i+1]:=a[i]*u[i]-v[i];    fprintf(aus,"$v_{%d}=%d$ & ",i+1,v[i+1]);
    a[i+1]:=(N-v[i+1]^2)/a[i]; fprintf(aus,"$a_{%d}=%d$ & ",i+1,a[i+1]);
    fprintf(aus,"\n$(-1)^{%d}a_{%d}=",i+1,i+1);
    if i mod 2=0 then fprintf(aus,"-"); fi;
    rest:=a[i+1];
    for j from 1 to nops(fb) do
      e:=0;
      while rest mod fb[j]=0 do e:=e+1; rest:=rest/fb[j]; od;
      if e>0 then fprintf(aus,"%d",fb[j]);
        if e>1 then fprintf(aus,"^{%d}",e); fi;
        if rest>1 then fprintf(aus,"\\cdot "); fi;
      fi;
    od;
    if rest>1 then fprintf(aus,"%d",rest); fi;
    fprintf(aus,"$ & ");
    if rest=1 then
      fprintf(aus,"* ");
      r:=r+1; x[r]:=P[i]; y[r]:=(-1)^(i+1)*a[i+1];
    fi;
    fprintf(aus,"\\\\ \\hline\n");
  od;
  fprintf(aus,"\\end{tabular}$$\n\n");
  fprintf(aus,"$$\\begin{tabular}{|l|l|l|l|}\\hline\n");
  for i from 0 to r do
    rest:=y[i];
    for j from 1 to nops(fb) do
      while rest mod fb[j]^2=0 do rest:=rest/fb[j]^2; od;
    od;
    fprintf(aus,"$x_{%d}=%d$ & $y_{%d}=%d=",i,x[i],i,y[i]);
    if y[i]<0 then fprintf(aus,"(-1)\\cdot "); fi;
    for j from 1 to nops(fb) do
      if rest mod fb[j]=0 then fprintf(aus,"%d\\cdot ",fb[j]); fi;
    od;
    fprintf(aus,"\\mathrm{Quadrat}$ & $c_{%d}=(",i);
    if y[i]<0 then fprintf(aus,"1,"); else fprintf(aus,"0,"); fi;
    for j from 1 to nops(fb) do
      if rest mod fb[j]=0 then fprintf(aus,"1"); else
        fprintf(aus,"0"); fi;
      if j<nops(fb) then fprintf(aus,","); fi;
    od;
    fprintf(aus,")$ \\\\ \\hline\n");
  od;
  fprintf(aus,"\\end{tabular}$$\n");
end;
```

14. cfrac_gmp.c

```
/* cfrac_gmp.c - 30.11.2001, 9.2.2002
   Uebersetzung mit gcc cfrac_gmp.c -lgmp

   Funktionen:
   int nofo( int *e, int *w, int m, int n) - Matrizennormalform
   void cfrac( mpz_t N, mpz_t N1, int n) - CFRAC, wobei die
     Faktorisierbarkeit von a mit Basisprimzahlen durch sukzessive
     ggT-Bildung mit dem Produkt der Basisprimzahlen getestet wird.
   int dez( mpz_t N) - Anzahl der Dezimalstellen
   void cfrac_mul( mpz_t N, mpz_t N1, int n, int k) - Erweiterung von
     cfrac um Multiplikator
   void cfrac_eas( mpz_t N, mpz_t N1, int n) - CFRAC mit early abort
     strategy, wobei die Faktorisierbarkeit von a mit Basisprimzahlen
     durch sukzessives Herausteilen getestet wird.
*/

#include <stdio.h>
#include <gmp.h>
#include <time.h>

#define M 1001 // x[M], y[M]

// Rueckgabewert 0, falls Zeile m identisch 0, sonst 1
int nofo( int *e, int *w, int m, int n)
{
  int i=0, j=0, k, v;
  while (i<m)
  {
    if (e[n*i+j]==1)
    {
      if (e[n*m+j]==1) // Addiere Zeile i zu Zeile m
      {
        for ( k=0; k<n; k++) e[n*m+k]^=e[n*i+k];
        for ( k=0; k<2*n; k++) w[2*n*m+k]^=w[2*n*i+k];
      }
      i++; j++;
    }
    else
    {
      if (e[n*m+j]==1) // Vertausche Zeile i mit Zeile j
      {
        for ( k=0; k<n; k++)
        { v=e[n*i+k]; e[n*i+k]=e[n*m+k]; e[n*m+k]=v; }
        for ( k=0; k<2*n; k++)
        { v=w[2*n*i+k]; w[2*n*i+k]=w[2*n*m+k]; w[2*n*m+k]=v; }
        i++; j++;
      }
      else
      {
        j++;
      }
    }
  }
```

```
  for ( j=0; j<n; j++)
    if (e[n*m+j]==1) return 1;
  return 0;
}

void cfrac( mpz_t N, mpz_t N1, int n)
{
  int i, j, *p, q, *e, *b, *w, m=-1, r=-1, azi=0, azr=0, azt=0, sgn=0;
  mpz_t hi, prodp, x[M], y[M], xx, yy, ggT, aa, aag;
  mpz_t d, wd, v, a, u, P0, P1;
  mpz_init(hi); mpz_init(prodp); mpz_init(xx); mpz_init(yy);
  mpz_init(ggT); mpz_init(aa); mpz_init(aag); mpz_init(d);
  mpz_init(wd); mpz_init(v); mpz_init(a); mpz_init(u);
  mpz_init(P0); mpz_init(P1);

  // Faktorbasis ***************************************************
  p=(int *) malloc(n*sizeof(int));
  p[0]=-1; p[1]=2; mpz_init_set_ui(hi,2);
  for (i=2;i<n;)
  {
    mpz_nextprime(hi,hi);
    if (mpz_legendre(N,hi)==1) p[i++]=mpz_get_ui(hi);
  }
  //printf("Faktorbasis: ");
  //for (i=0;i<n;i++) printf("%d ",p[i]);
  //printf("\n");

  // Produkt der Basisprimzahlen
  mpz_set_ui(prodp,1);
  for (i=1;i<n;i++) mpz_mul_ui(prodp,prodp,p[i]);
  //printf("prodp="); mpz_out_str(stdout,10,prodp); printf("\n");

  // Zum Abspeichern der Relationen ********************************
  e=(int *) malloc((n+1)*n*sizeof(int));
  b=(int *) malloc(n*sizeof(int));
  w=(int *) malloc(2*n*2*n*sizeof(int));
  for (i=0;i<(n+1)*n;i++) e[i]=0;
  for (i=0;i<2*n*2*n;i++) w[i]=0;

  // Initialisierung der Kettenbruchentwicklung ********************
  // Kettenbruchentwicklung mit v_i, a_i, u_i=floor((v_i+sqrt(d))/a_i)
  // Zaehler der Naeherungsbrueche P_i mit P_(-2)=0, P_(-1)=1
  mpz_set(d,N); mpz_sqrt(wd,d);
  mpz_set_ui(v,0); mpz_set_ui(a,1);   // v_0=0, a_0=1
  mpz_set_ui(P0,0); mpz_set_ui(P1,1); // P_(-2)=0, P_(-1)=1

  for (;;)
  {
    sgn^=1;
    // Start mit v_i, a_i, P_(i-2), P_(i-1). Dann wird mit
    // u_i=floor((v_i+wd)/a_i), v_(i+1)=a_i*u_i-v_i,
    // a_(i+1)=(d-v_(i+1)^2)/a_i, P_i=u_i*P_(i-1)+P_(i-2) die Werte
    // v_(i+1), a_(i+1), P_(i-1), P_i berechnet. Dann sollte gelten:
    // P_i^2=(-1)^(i+1)*a_(i+1) mod d bzw. P1^2=(-1)^(i+1)*a mod d
```

```
mpz_add(u,v,wd); mpz_fdiv_q(u,u,a);  // u=(v+wd)/a
mpz_mul(hi,a,u); mpz_sub(v,hi,v);    // v=a*u-v
mpz_mul(hi,v,v); mpz_sub(hi,d,hi);
mpz_divexact(a,hi,a);  // a=(d-v*v)/a;
mpz_mul(hi,u,P1); mpz_add(hi,hi,P0);
mpz_set(P0,P1); mpz_mod(P1,hi,d); // P2=(u*P1+P0)%d; P0=P1; P1=P2;

if (mpz_cmp_ui(a,1)==0) { printf("1"); fflush(stdout); }

azi++;
//printf("."); fflush(stdout);
mpz_set(aa,a); mpz_set_ui(aag,0);
while (mpz_cmp_ui(aag,1)!=0)
{
   mpz_gcd(aag,aa,prodp);
   mpz_divexact(aa,aa,aag);
}
if (mpz_cmp_ui(aa,1)==0)
{
  azr++;
  //printf("*"); fflush(stdout);
  mpz_set(aa,a);
  for (j=1;j<n;j++)
  {
    b[j]=0;
    while (mpz_fdiv_q_ui(hi,aa,p[j])==0)
    { b[j]^=1; mpz_set(aa,hi); }
  }
  // Neue Relation gefunden!
  //printf("."); fflush(stdout);
  m++; r++;
  mpz_init_set(x[r],P1); mpz_init_set(y[r],a);
  e[n*m]=sgn; for ( j=1; j<n; j++) e[n*m+j]=b[j];
  w[2*n*m+r]=1; for (j=0;j<r;j++) w[2*n*m+j]=0;
  //pmat(e,w,m,n);

  // Nun wird die Normalform gebildet
  if (nofo(e,w,m,n)==0)
  {
    azt++;
    //printf("t"); fflush(stdout);
    //printf("*"); fflush(stdout);

    mpz_set_ui(xx,1); mpz_set_ui(yy,1);
    for (j=0;j<=r;j++)
      if (w[2*n*m+j]==1)
      {
        mpz_mul(xx,xx,x[j]); mpz_mod(xx,xx,N);
        mpz_mul(yy,yy,y[j]);
      }

    mpz_sqrt(hi,yy); mpz_add(hi,xx,hi);
    mpz_gcd(ggT,hi,N); // ggT=GCD(xx+SqrRoot(yy),N);
```

```
        if (mpz_cmp_ui(ggT,1)>0 && mpz_cmp(ggT,N)<0)
        {
          printf("(%d,%d,%d)\n",azi,azr,azt);
          mpz_init_set(N1,ggT);
          return;
        }
        m--;
      }
    }
  }
  return;
}

int dez( mpz_t N)
{
  int az=0;
  mpz_t h;
  mpz_init_set(h,N);
  while (mpz_sgn(h)!=0) { az++; mpz_fdiv_q_ui(h,h,10); }
  return az;
}

void cfrac_mul( mpz_t N, mpz_t N1, int n, int k)
{
  int i, j, *p, q, *e, *b, *w, m=-1, r=-1, azi=0, azr=0, azt=0, sgn=0;
  mpz_t hi, prodp, x[M], y[M], xx, yy, ggT, aa, aag;
  mpz_t d, wd, v, a, u, P0, P1;
  mpz_init(hi); mpz_init(prodp); mpz_init(xx); mpz_init(yy);
  mpz_init(ggT); mpz_init(aa); mpz_init(aag); mpz_init(d);
  mpz_init(wd); mpz_init(v); mpz_init(a); mpz_init(u);
  mpz_init(P0); mpz_init(P1);

  mpz_mul_ui(d,N,k);

  // Faktorbasis ********************************************************
  p=(int *) malloc(n*sizeof(int));
  p[0]=-1; p[1]=2; mpz_init_set_ui(hi,2);
  for (i=2;i<n;)
  {
    mpz_nextprime(hi,hi);
    if (mpz_legendre(d,hi)!=-1) p[i++]=mpz_get_ui(hi);
  }
  //printf("Faktorbasis: ");
  //for (i=0;i<n;i++) printf("%d ",p[i]);
  //printf("Letztes Element der Faktorbasis: %d\n",p[n-1]);

  // Produkt der Basisprimzahlen
  mpz_set_ui(prodp,1);
  for (i=1;i<n;i++) mpz_mul_ui(prodp,prodp,p[i]);
  //printf("prodp="); mpz_out_str(stdout,10,prodp); printf("\n");
  printf("Das Produkt der Faktorbasisprimzahlen hat %d ",dez(prodp));
  printf("Stellen.\n");

  // Zum Abspeichern der Relationen *************************************
```

```
  e=(int *) malloc((n+1)*n*sizeof(int));
  b=(int *) malloc(n*sizeof(int));
  w=(int *) malloc(2*n*2*n*sizeof(int));
  for (i=0;i<(n+1)*n;i++) e[i]=0;
  for (i=0;i<2*n*2*n;i++) w[i]=0;

  // Initialisierung der Kettenbruchentwicklung **********************
  // Kettenbruchentwicklung mit v_i, a_i, u_i=floor((v_i+sqrt(d))/a_i)
  // Zaehler der Naeherungsbrueche P_i mit P_(-2)=0, P_(-1)=1
  // mpz_set(d,N);
  mpz_sqrt(wd,d);
  mpz_set_ui(v,0); mpz_set_ui(a,1);   // v_0=0, a_0=1
  mpz_set_ui(P0,0); mpz_set_ui(P1,1); // P_(-2)=0, P_(-1)=1

  for (;;)
  {
    sgn^=1;
    // Start mit v_i, a_i, P_(i-2), P_(i-1). Dann wird mit
    // u_i=floor((v_i+wd)/a_i), v_(i+1)=a_i*u_i-v_i,
    // a_(i+1)=(d-v_(i+1)^2)/a_i, P_i=u_i*P_(i-1)+P_(i-2) die Werte
    // v_(i+1), a_(i+1), P_(i-1), P_i berechnet. Dann sollte gelten:
    // P_i^2=(-1)^(i+1)*a_(i+1) mod d bzw. P1^2=(-1)^(i+1)*a mod d
    mpz_add(u,v,wd); mpz_fdiv_q(u,u,a);  // u=(v+wd)/a
    mpz_mul(hi,a,u); mpz_sub(v,hi,v);    // v=a*u-v
    mpz_mul(hi,v,v); mpz_sub(hi,d,hi);
    mpz_divexact(a,hi,a);  // a=(d-v*v)/a;
    mpz_mul(hi,u,P1); mpz_add(hi,hi,P0);
    mpz_set(P0,P1); mpz_mod(P1,hi,N); // P2=(u*P1+P0)%d; P0=P1; P1=P2;

    if (mpz_cmp_ui(a,1)==0) { printf("1"); fflush(stdout); }

    azi++;
    //printf("."); fflush(stdout);
    mpz_set(aa,a); mpz_set_ui(aag,0);
    while (mpz_cmp_ui(aag,1)!=0)
    {
       mpz_gcd(aag,aa,prodp);
       mpz_divexact(aa,aa,aag);
    }
    if (mpz_cmp_ui(aa,1)==0)
    {
      azr++;
      // printf("*"); fflush(stdout);
      mpz_set(aa,a);
      for (j=1;j<n;j++)
      {
        b[j]=0;
        while (mpz_fdiv_q_ui(hi,aa,p[j])==0)
        { b[j]^=1; mpz_set(aa,hi); }
      }
      // Neue Relation gefunden!
      //printf("."); fflush(stdout);
      m++; r++;
      mpz_init_set(x[r],P1); mpz_init_set(y[r],a);
```

```
      e[n*m]=(i+1)%2; for (j=1;j<n;j++) e[n*m+j]=b[j];
      w[2*n*m+r]=1; for (j=0;j<r;j++) w[2*n*m+j]=0;
      //pmat(e,w,m,n);

      // Nun wird die Normalform gebildet
      if (nofo(e,w,m,n)==0)
      {
        azt++;
        //printf("t"); fflush(stdout);
        //printf("*"); fflush(stdout);

        mpz_set_ui(xx,1); mpz_set_ui(yy,1);
        for (j=0;j<=r;j++)
          if (w[2*n*m+j]==1)
          {
            mpz_mul(xx,xx,x[j]); mpz_mod(xx,xx,N);
            mpz_mul(yy,yy,y[j]);
          }

        mpz_sqrt(hi,yy); mpz_add(hi,xx,hi);
        mpz_gcd(ggT,hi,N); // ggT=GCD(xx+SqrRoot(yy),N);

        if (mpz_cmp_ui(ggT,1)>0 && mpz_cmp(ggT,N)<0)
        {
          printf("(%d,%d,%d)\n",azi,azr,azt);
          mpz_init_set(N1,ggT);
          return;
        }
        m--;
      }
    }
  }
  return;
}

void cfrac_eas( mpz_t N, mpz_t N1, int n)
{
  int i, j, *p, q, *e, *b, *w, m=-1, r=-1, azi=0, azr=0, azt=0, sgn=0;
  mpz_t hi, x[M], y[M], xx, yy, ggT, aa, wN, R1, R2;
  mpz_t d, wd, v, a, u, P0, P1;
  mpz_init(hi); mpz_init(xx); mpz_init(yy);
  mpz_init(wN); mpz_init(R1); mpz_init(R2);
  mpz_init(ggT); mpz_init(aa); mpz_init(d);
  mpz_init(wd); mpz_init(v); mpz_init(a); mpz_init(u);
  mpz_init(P0); mpz_init(P1);

  // Faktorbasis ********************************************************
  p=(int *) malloc(n*sizeof(int));
  p[0]=-1; p[1]=2; mpz_init_set_ui(hi,2);
  for (i=2;i<n;)
  {
    mpz_nextprime(hi,hi);
    if (mpz_legendre(N,hi)==1) p[i++]=mpz_get_ui(hi);
  }
```

```
if (n<96)
{
  printf("n=%d ist zu klein fuer early abort strategy!\n",n);
  return;
}

// Early abort strategy ******************************************
mpz_sqrt(wN,N);
mpz_fdiv_q_ui(R1,wN,500); mpz_fdiv_q_ui(R2,wN,20000000);

// Zum Abspeichern der Relationen ********************************
e=(int *) malloc((n+1)*n*sizeof(int));
b=(int *) malloc(n*sizeof(int));
w=(int *) malloc(2*n*2*n*sizeof(int));
for (i=0;i<(n+1)*n;i++) e[i]=0;
for (i=0;i<2*n*2*n;i++) w[i]=0;

// Initialisierung der Kettenbruchentwicklung ********************
// Kettenbruchentwicklung mit v_i, a_i, u_i=floor((v_i+sqrt(d))/a_i)
// Zaehler der Naeherungsbrueche P_i mit P_(-2)=0, P_(-1)=1
mpz_set(d,N); mpz_sqrt(wd,d);
mpz_set_ui(v,0); mpz_set_ui(a,1);   // v_0=0, a_0=1
mpz_set_ui(P0,0); mpz_set_ui(P1,1); // P_(-2)=0, P_(-1)=1

for (;;)
{
  sgn^=1;
  // Start mit v_i, a_i, P_(i-2), P_(i-1). Dann wird mit
  // u_i=floor((v_i+wd)/a_i), v_(i+1)=a_i*u_i-v_i,
  // a_(i+1)=(d-v_(i+1)^2)/a_i, P_i=u_i*P_(i-1)+P_(i-2) die Werte
  // v_(i+1), a_(i+1), P_(i-1), P_i berechnet. Dann sollte gelten:
  // P_i^2=(-1)^(i+1)*a_(i+1) mod d bzw. P1^2=(-1)^(i+1)*a mod d
  mpz_add(u,v,wd); mpz_fdiv_q(u,u,a);  // u=(v+wd)/a
  mpz_mul(hi,a,u); mpz_sub(v,hi,v);    // v=a*u-v
  mpz_mul(hi,v,v); mpz_sub(hi,d,hi);
  mpz_divexact(a,hi,a);  // a=(d-v*v)/a;
  mpz_mul(hi,u,P1); mpz_add(hi,hi,P0);
  mpz_set(P0,P1); mpz_mod(P1,hi,d); // P2=(u*P1+P0)%d; P0=P1; P1=P2;

  if (mpz_cmp_ui(a,1)==0) { printf("1"); fflush(stdout); }

  azi++;
  //printf("."); fflush(stdout);
  mpz_set(aa,a);
  for (j=1;j<16;j++)
  {
    b[j]=0;
    while (mpz_fdiv_q_ui(hi,aa,p[j])==0)
    { b[j]^=1; mpz_set(aa,hi); }
  }
  if (mpz_cmp(aa,R1)<=0)
  {
    for (j=16;j<96;j++)
```

```
{
  b[j]=0;
  while (mpz_fdiv_q_ui(hi,aa,p[j])==0)
  { b[j]^=1; mpz_set(aa,hi); }
}
if (mpz_cmp(aa,R2)<=0)
{
  for (j=96;j<n;j++)
  {
    b[j]=0;
    while (mpz_fdiv_q_ui(hi,aa,p[j])==0)
    { b[j]^=1; mpz_set(aa,hi); }
  }
  if (mpz_cmp_ui(aa,1)==0)
  {
    azr++;
    //printf("*"); fflush(stdout);
    mpz_set(aa,a);
    // Neue Relation gefunden!
    //printf("."); fflush(stdout);
    m++; r++;
    mpz_init_set(x[r],P1); mpz_init_set(y[r],a);
    e[n*m]=sgn; for ( j=1; j<n; j++) e[n*m+j]=b[j];
    w[2*n*m+r]=1; for (j=0;j<r;j++) w[2*n*m+j]=0;
    //pmat(e,w,m,n);

    // Nun wird die Normalform gebildet
    if (nofo(e,w,m,n)==0)
    {
      azt++;
      //printf("t"); fflush(stdout);
      //printf("*"); fflush(stdout);

      mpz_set_ui(xx,1); mpz_set_ui(yy,1);
      for (j=0;j<=r;j++)
        if (w[2*n*m+j]==1)
        {
          mpz_mul(xx,xx,x[j]); mpz_mod(xx,xx,N);
          mpz_mul(yy,yy,y[j]);
        }

      mpz_sqrt(hi,yy); mpz_add(hi,xx,hi);
      mpz_gcd(ggT,hi,N); // ggT=GCD(xx+SqrRoot(yy),N);

      if (mpz_cmp_ui(ggT,1)>0 && mpz_cmp(ggT,N)<0)
      {
        printf("(%d,%d,%d)\n",azi,azr,azt);
        mpz_init_set(N1,ggT);
        return;
      }
      m--;
    }
  }
}
```

```
    }
  }
  return;
}

int main( int argc, char *argv[])
{
  int i, n;
  mpz_t N, N1, N2;
  time_t zeit1, zeit2;
  mpz_init(N); mpz_init(N1); mpz_init(N2);

  for (i=0;i<argc;i++)
    printf("%s\n",argv[i]);

  printf("N="); mpz_inp_str(N,stdin,10);
  printf("n="); scanf("%d",&n);

  printf("\nN="); mpz_out_str(stdout,10,N);
  printf("\nn=%d\n",n);

  time(&zeit1);
  cfrac_eas(N,N1,n);
  mpz_divexact(N2,N,N1);
  printf("N=");
  mpz_out_str(stdout,10,N1); printf("*");
  mpz_out_str(stdout,10,N2); printf("\n");
  time(&zeit2);
  printf("Zeit: %.0f sec\n",difftime(zeit2,zeit1));
  return 0;
}
```

15. cfrac_ntl.c

```
/* cfrac_ntl.c - 28.11.2001, 29.11.2001, 30.11.2001, 1.12.2001, 9.2.2002
   Uebersetzung mit g++ cfrac_ntl.c -lntl -lgmp
   Anmerkung: Die Funktionen dienten der Beispielerstellung.

   Funktionen:
   void pmat( int *e, int *w, int m, int n) - Zur Matrizenausgabe
   int nofo( int *e, int *w, int m, int n) - Matrizennormalform
   ZZ cfrac_1( ZZ N, int n) - CFRAC, wobei die Faktorisierbarkeit
     von a mit Basisprimzahlen wird durch sukzessives Herausteilen
     getestet wird.
   ZZ cfrac_2( ZZ N, int n) - CFRAC, wobei die Faktorisierbarkeit
     von a mit Basisprimzahlen wird durch sukzessive ggT-Bildung mit
     dem Produkt der Basisprimzahlen getestet wird.
   ZZ cfrac_1_eas( ZZ N, int n) - cfrac_1 erweitert mit early abort
     strategy
   ZZ cfrac_2_eas( ZZ N, int n) - cfrac_2 erweitert mit early abort
     strategy

   Experimentell:
   cfrac_2 ist schneller als cfrac_1
   cfrac_1_eas ist (meist) schneller als cfrac_2_eas
```

```
   cfrac_1_eas ist (meist) am schnellsten
*/

#include <stdio.h>
#include <NTL/ZZ.h>

#define M 1001 // x[M], y[M]

// Zur Matrizenausgabe
void pmat( int *e, int *w, int m, int n)
{
  int i, j;
  for ( i=0; i<=m; i++)
  {
    for ( j=0; j<n; j++) cout << e[n*i+j] << " ";
    cout << "\t";
    for ( j=0; j<2*n; j++) cout << w[2*n*i+j] << " ";
    cout << "\n";
  }
  return;
}

// Rueckgabewert 0, falls Zeile m von e identisch 0 ist, sonst 1.
int nofo( int *e, int *w, int m, int n)
{
  int i=0, j=0, k, v;
  while (i<m)
  {
    if (e[n*i+j]==1)
    {
      if (e[n*m+j]==1) // Addiere Zeile i zu Zeile m
      {
        for ( k=0; k<n; k++) e[n*m+k]^=e[n*i+k];
        for ( k=0; k<2*n; k++) w[2*n*m+k]^=w[2*n*i+k];
      }
      i++; j++;
    }
    else
    {
      if (e[n*m+j]==1) // Vertausche Zeile i mit Zeile j
      {
        for ( k=0; k<n; k++)
        { v=e[n*i+k]; e[n*i+k]=e[n*m+k]; e[n*m+k]=v; }
        for ( k=0; k<2*n; k++)
        { v=w[2*n*i+k]; w[2*n*i+k]=w[2*n*m+k]; w[2*n*m+k]=v; }
        i++; j++;
      }
      else
      {
        j++;
      }
    }
  }
  for ( j=0; j<n; j++)
```

```
    if (e[n*m+j]==1) return 1;
  return 0;
}

// Faktorisierbarkeit von a mit Basisprimzahlen wird durch sukzessives
// Herausteilen getestet.
ZZ cfrac_1( ZZ N, int n)
{
  int azi=0, azr=0, azt=0;
  int i, j;

  // Faktorbasis ********************************************************
  int *p; p=(int *) malloc(n*sizeof(int));
  PrimeSeq s; s.next();
  p[0]=-1; p[1]=2;
  for ( i=2; i<n; )
    if (Jacobi(N,to_ZZ(p[i]=s.next()))==1) i++;
  //cout << "Faktorbasis: ";
  //for ( i=0; i<n; i++) cout << p[i] << " ";
  //cout << "\n";

  // Zum Abspeichern der Relationen *************************************
  ZZ x[M], y[M], xx, yy, ggT, aa, aag;
  int *e, *b, *w, m=-1, r=-1;
  e=(int *) malloc((n+1)*n*sizeof(int));
  b=(int *) malloc(n*sizeof(int));
  w=(int *) malloc(2*n*2*n*sizeof(int));
  for (i=0;i<(n+1)*n;i++) e[i]=0;
  for (i=0;i<2*n*2*n;i++) w[i]=0;

  // Initialisierung der Kettenbruchentwicklung *********************
  // Kettenbruchentwicklung mit v_i, a_i, u_i=floor((v_i+sqrt(d))/a_i)
  // Zaehler der Naeherungsbrueche P_i mit P_(-2)=0, P_(-1)=1
  ZZ d=N, wd=SqrRoot(d), v, a, u, P0, P1, P2;
  v=0; a=1; P0=0; P1=1; // v_0=0, a_0=1, P_(-2)=0, P_(-1)=1
  int sgn=0;

  for (;;)
  {
    sgn^=1;
    // Start mit v_i, a_i, P_(i-2), P_(i-1). Dann wird mit
    // u_i=floor((v_i+wd)/a_i), v_(i+1)=a_i*u_i-v_i,
    // a_(i+1)=(d-v_(i+1)^2)/a_i, P_i=u_i*P_(i-1)+P_(i-2) die Werte
    // v_(i+1), a_(i+1), P_(i-1), P_i berechnet. Dann sollte gelten:
    // P_i^2=(-1)^(i+1)*a_(i+1) mod d bzw. P1^2=(-1)^(i+1)*a mod d

    u=(v+wd)/a; v=a*u-v; a=(d-v*v)/a; P2=(u*P1+P0)%d; P0=P1; P1=P2;

    //if (a==1) cout << "1";

    aa=a;
    for (j=1;j<n;j++)
    {
      b[j]=0;
```

```
      while (aa%p[j]==0) { b[j]^=1; aa/=p[j]; }
    }
    azi++;
    if (aa==1)
    {
      azr++;
      // Neue Relation gefunden!
      m++; r++;
      //cout << "m=" << m << " r=" << r << "\n";
      //cout << "r=" << r << "\n";
      x[r]=P1; y[r]=a;
      //cout << "x[" << r << "]=" << x[r] << " ";
      //cout << "y[" << r << "]=" << y[r] << "\n";
      e[n*m]=sgn; for ( j=1; j<n; j++) e[n*m+j]=b[j];
      w[2*n*m+r]=1; for (j=0;j<r;j++) w[2*n*m+j]=0;
      //pmat(e,w,m,n);

      // Nun wird die Normalform gebildet
      if (nofo(e,w,m,n)==0)
      {
        azt++;
        //cout << "*"; fflush(stdout);
        //pmat(e,w,m,n);

        xx=1; yy=1;
        for (j=0;j<=r;j++)
          if (w[2*n*m+j]==1)
          {
            xx=(xx*x[j])%N; yy*=y[j];
          }

        ggT=GCD(xx+SqrRoot(yy),N);

        if (ggT>1 && ggT<N)
        {
          //cout << "\n";
          cout << "(" << azi << "," << azr << "," << azt << ")\n";
          flush(cout);
          return ggT;
        }
        m--;
      }
    }
  }
}

// Faktorisierbarkeit von a mit Basisprimzahlen wird durch sukzessive
// ggT-Bildung mit dem Produkt der Basisprimzahlen getestet.
ZZ cfrac_2( ZZ N, int n)
{
  int azi=0, azr=0, azt=0;
  int i, j;

  // Faktorbasis ***************************************************
```

```
int *p; p=(int *) malloc(n*sizeof(int));
PrimeSeq s; s.next();
p[0]=-1; p[1]=2;
for ( i=2; i<n; )
  if (Jacobi(N,to_ZZ(p[i]=s.next()))==1) i++;

ZZ prodp; // Produkt der Basisprimzahlen
prodp=1; for ( i=1; i<n; i++) prodp*=p[i];

// Zum Abspeichern der Relationen ********************************
ZZ x[M], y[M], xx, yy, ggT, aa, aag;
int *e, *b, *w, m=-1, r=-1;
e=(int *) malloc((n+1)*n*sizeof(int));
b=(int *) malloc(n*sizeof(int));
w=(int *) malloc(2*n*2*n*sizeof(int));
for (i=0;i<(n+1)*n;i++) e[i]=0;
for (i=0;i<2*n*2*n;i++) w[i]=0;

// Initialisierung der Kettenbruchentwicklung **********************
// Kettenbruchentwicklung mit v_i, a_i, u_i=floor((v_i+sqrt(d))/a_i)
// Zaehler der Naeherungsbrueche P_i mit P_(-2)=0, P_(-1)=1
ZZ d=N, wd=SqrRoot(d), v, a, u, P0, P1, P2;
v=0; a=1; P0=0; P1=1; // v_0=0, a_0=1, P_(-2)=0, P_(-1)=1
int sgn=0;

for (i=0;;i++)
{
  sgn^=1;
  // Start mit v_i, a_i, P_(i-2), P_(i-1). Dann wird mit
  // u_i=floor((v_i+wd)/a_i), v_(i+1)=a_i*u_i-v_i,
  // a_(i+1)=(d-v_(i+1)^2)/a_i, P_i=u_i*P_(i-1)+P_(i-2) die Werte
  // v_(i+1), a_(i+1), P_(i-1), P_i berechnet. Dann sollte gelten:
  // P_i^2=(-1)^(i+1)*a_(i+1) mod d bzw. P1^2=(-1)^(i+1)*a mod d

  u=(v+wd)/a; v=a*u-v; a=(d-v*v)/a; P2=(u*P1+P0)%d; P0=P1; P1=P2;

  //if (a==1) cout << "1";

  aa=a;
  while ((aag=GCD(aa,prodp))!=1) aa/=aag;
  azi++;
  if (aa==1)
  {
    azr++;
    aa=a;
    for (j=1;j<n;j++)
    {
      b[j]=0;
      while (aa%p[j]==0) { b[j]^=1; aa/=p[j]; }
    }
    // Neue Relation gefunden!
    m++; r++;
    x[r]=P1; y[r]=a;
    e[n*m]=sgn; for ( j=1; j<n; j++) e[n*m+j]=b[j];
```

```
      w[2*n*m+r]=1; for (j=0;j<r;j++) w[2*n*m+j]=0; // XXXXXXXXXXXXX
      // Nun wird die Normalform gebildet
      if (nofo(e,w,m,n)==0)
      {
        azt++;
        xx=1; yy=1;
        for (j=0;j<=r;j++)
          if (w[2*n*m+j]==1)
          {
            xx=(xx*x[j])%N; yy*=y[j];
          }
        ggT=GCD(xx+SqrRoot(yy),N);
        if (ggT>1 && ggT<N)
        {
          cout << "(" << azi << "," << azr << "," << azt << ")\n";
          flush(cout);
          return ggT;
        }
        m--;
      }
    }
  }
}

// cfrac_1 erweitert mit early abort strategy
ZZ cfrac_1_eas( ZZ N, int n)
{
  int azi=0, azr=0, azt=0;
  int i, j;

  // Faktorbasis ********************************************************
  int *p; p=(int *) malloc(n*sizeof(int));
  PrimeSeq s; s.next();
  p[0]=-1; p[1]=2;
  for ( i=2; i<n; )
    if (Jacobi(N,to_ZZ(p[i]=s.next()))==1) i++;

  // Zum Abspeichern der Relationen *************************************
  ZZ x[M], y[M], xx, yy, ggT, aa, aag;
  int *e, *b, *w, m=-1, r=-1;
  e=(int *) malloc((n+1)*n*sizeof(int));
  b=(int *) malloc(n*sizeof(int));
  w=(int *) malloc(2*n*2*n*sizeof(int));
  for (i=0;i<(n+1)*n;i++) e[i]=0;
  for (i=0;i<2*n*2*n;i++) w[i]=0;

  // Early abort strategy ***********************************************
  ZZ wN=SqrRoot(N), R1=wN/500, R2=wN/20000000;

  // Initialisierung der Kettenbruchentwicklung *************************
  // Kettenbruchentwicklung mit v_i, a_i, u_i=floor((v_i+sqrt(d))/a_i)
  // Zaehler der Naeherungsbrueche P_i mit P_(-2)=0, P_(-1)=1
  ZZ d=N, wd=SqrRoot(d), v, a, u, P0, P1, P2;
  v=0; a=1; P0=0; P1=1; // v_0=0, a_0=1, P_(-2)=0, P_(-1)=1
```

```
  int sgn=0;

  for (;;)
  {
    sgn^=1;
    // Start mit v_i, a_i, P_(i-2), P_(i-1). Dann wird mit
    // u_i=floor((v_i+wd)/a_i), v_(i+1)=a_i*u_i-v_i,
    // a_(i+1)=(d-v_(i+1)^2)/a_i, P_i=u_i*P_(i-1)+P_(i-2) die Werte
    // v_(i+1), a_(i+1), P_(i-1), P_i berechnet. Dann sollte gelten:
    // P_i^2=(-1)^(i+1)*a_(i+1) mod d bzw. P1^2=(-1)^(i+1)*a mod d

    u=(v+wd)/a; v=a*u-v; a=(d-v*v)/a; P2=(u*P1+P0)%d; P0=P1; P1=P2;

    //if (a==1) cout << "1";

    azi++;
    aa=a;
    for (j=1;j<=15;j++)
    {
      b[j]=0;
      while (aa%p[j]==0) { b[j]^=1; aa/=p[j]; }
    }
    if (aa<=R1)
    {
      for (j=16;j<=95;j++)
      {
        b[j]=0;
        while (aa%p[j]==0) { b[j]^=1; aa/=p[j]; }
      }
      if (aa<=R2)
      {
        for (j=96;j<n;j++)
        {
          b[j]=0;
          while (aa%p[j]==0) { b[j]^=1; aa/=p[j]; }
        }
        // Um Beispiele fuer large prime variation zu erhalten, kann man
        // folgende Zeile einfuegen:
        // if (aa>1 && aa<p[n-1]*p[n-1])
        //   cout << "aa=" << aa << " P1=" << P1 << " a=" << a << "\n";
        if (aa==1)
        {
          azr++;
          m++; r++;
          x[r]=P1; y[r]=a;
          e[n*m]=sgn; for ( j=1; j<n; j++) e[n*m+j]=b[j];
          w[2*n*m+r]=1; for (j=0;j<r;j++) w[2*n*m+j]=0;
          // Nun wird die Normalform gebildet
          if (nofo(e,w,m,n)==0)
          {
            azt++;
            xx=1; yy=1;
            for (j=0;j<=r;j++)
              if (w[2*n*m+j]==1)
```

```
              {
                xx=(xx*x[j])%N; yy*=y[j];
              }
            ggT=GCD(xx+SqrRoot(yy),N);
            if (ggT>1 && ggT<N)
            {
              cout << "(" << azi << "," << azr << "," << azt << ")\n";
              flush(cout);
              return ggT;
            }
            m--;
          }
        }
      }
    }
  }
}

// cfrac_2 erweitert mit early abort strategy
ZZ cfrac_2_eas( ZZ N, int n)
{
  int azi=0, azr=0, azt=0;
  int i, j;

  // Faktorbasis ********************************************************
  int *p; p=(int *) malloc(n*sizeof(int));
  PrimeSeq s; s.next();
  p[0]=-1; p[1]=2;
  for ( i=2; i<n; )
    if (Jacobi(N,to_ZZ(p[i]=s.next()))==1) i++;

  ZZ prodp1, prodp2, prodp3; // Produkt von Basisprimzahlen
  prodp1=1; for (i=1;i<=15;i++) prodp1*=p[i];
  prodp2=1; for (i=16;i<=95;i++) prodp2*=p[i];
  prodp3=1; for (i=96;i<n;i++) prodp3*=p[i];

  // Zum Abspeichern der Relationen *************************************
  ZZ x[M], y[M], xx, yy, ggT, aa, aag;
  int *e, *b, *w, m=-1, r=-1;
  e=(int *) malloc((n+1)*n*sizeof(int));
  b=(int *) malloc(n*sizeof(int));
  w=(int *) malloc(2*n*2*n*sizeof(int));
  for (i=0;i<(n+1)*n;i++) e[i]=0;
  for (i=0;i<2*n*2*n;i++) w[i]=0;

  // Early abort strategy ***********************************************
  ZZ wN=SqrRoot(N), R1=wN/500, R2=wN/20000000;

  // Initialisierung der Kettenbruchentwicklung *************************
  // Kettenbruchentwicklung mit v_i, a_i, u_i=floor((v_i+sqrt(d))/a_i)
  // Zaehler der Naeherungsbrueche P_i mit P_(-2)=0, P_(-1)=1
  ZZ d=N, wd=SqrRoot(d), v, a, u, P0, P1, P2;
  v=0; a=1; P0=0; P1=1; // v_0=0, a_0=1, P_(-2)=0, P_(-1)=1
  int sgn=0;
```

```
for (;;)
{
  sgn^=1;
  // Start mit v_i, a_i, P_(i-2), P_(i-1). Dann wird mit
  // u_i=floor((v_i+wd)/a_i), v_(i+1)=a_i*u_i-v_i,
  // a_(i+1)=(d-v_(i+1)^2)/a_i, P_i=u_i*P_(i-1)+P_(i-2) die Werte
  // v_(i+1), a_(i+1), P_(i-1), P_i berechnet. Dann sollte gelten:
  // P_i^2=(-1)^(i+1)*a_(i+1) mod d bzw. P1^2=(-1)^(i+1)*a mod d

  u=(v+wd)/a; v=a*u-v; a=(d-v*v)/a; P2=(u*P1+P0)%d; P0=P1; P1=P2;

  azi++;
  aa=a;
  while ((aag=GCD(prodp1,aa))!=1) aa/=aag;
  if (aa<=R1)
  {
    while ((aag=GCD(prodp2,aa))!=1) aa/=aag;
    if (aa<=R2)
    {
      while ((aag=GCD(prodp3,aa))!=1) aa/=aag;
      if (aa==1)
      {
        azr++;
        aa=a;
        for (j=1;j<n;j++)
        {
          b[j]=0;
          while (aa%p[j]==0) { b[j]^=1; aa/=p[j]; }
        }
        // Neue Relation gefunden!
        m++; r++;
        x[r]=P1; y[r]=a;
        e[n*m]=sgn; for ( j=1; j<n; j++) e[n*m+j]=b[j];
        w[2*n*m+r]=1; for (j=0;j<r;j++) w[2*n*m+j]=0; // XXXXXXXXXXXXX
        // Nun wird die Normalform gebildet
        if (nofo(e,w,m,n)==0)
        {
          azt++;
          xx=1; yy=1;
          for (j=0;j<=r;j++)
            if (w[2*n*m+j]==1)
            {
              xx=(xx*x[j])%N; yy*=y[j];
            }
          ggT=GCD(xx+SqrRoot(yy),N);
          if (ggT>1 && ggT<N)
          {
            cout << "(" << azi << "," << azr << "," << azt << ")\n";
            flush(cout);
            return ggT;
          }
          m--;
        }
```

```
        }
      }
    }
  }
}

int main( int argc, char *argv[])
{
  ZZ N, N1, N2; int n;

  cout << "\n";
  cout << "N="; cin >> N;
  cout << "n="; cin >> n;
  cout << "\n";
  cout << "N=" << N << "\n";
  cout << "n=" << n << "\n";

  N1=cfrac_1_eas(N,n); N2=N/N1;
  cout << "N=" << N1 << "*" << N2 << "\n";

  cout << "Zeit: "; PrintTime(cout,GetTime()); cout << "\n";
  return 0;
}
```

16. algo6_ma

```
# algo6_ma
# Kapitel: Endlich erzeugte Z-Moduln in quadratischen Zahlkoerpern
# Version: 12.2.2002
#
# Funktionen:
# isdisc(D)
# qz_abD(q1,q2,D)
# ggT_lk([a_1,a_2,...,a_r])
# modul_hnf([[x_1,y_1],[x_2,y_2],...,[x_r,y_r]])
# modul_nf([[x_1,y_1],...,[x_r,y_r]],d)
# modul_kon(D,a)
# ggT3(a1,a2,a3)
# modul_multD([A1,a1,b1,D],[A2,a2,b2,D])
# modul_mult([A1,a1,b1,D1],[A1,a2,b2,D2])
# fund_disc(D)
# modul_test([A,a,b,D])
# rq_einheit(D)
# hauptideal(x,y,D)
# modul_kon_p(D,p)

unprotect(D);

# Test, ob D Diskriminante einer quadratischen Zahl ist
# Eingabe: D
# Ausgabe: true oder false
isdisc:=proc()
  local D;
  D:=args[1];
  if D mod 4=2 or D mod 4=3 then return false; fi;
```

```
  if issqr(D) then return false fi;
  true;
end;

# Umwandlung einer quadratischen Zahl alpha=q_1+q_2*sqrt(d) in
# Diskriminantenform (b+sqrt(D))/(2*a). (alpha genuegt der Gleichung
# alpha^2-2*q1*alpha+(q1^2-d*q2^2)=0, multipliziert man mit dem Nenner,
# so erhaelt man a*alpha^2-b*alpha+c=0.) Die Zahl d muss nicht
# quadratfrei sein.
# Eingabe: q_1,q_2,d
# Ausgabe: [a,b,D]
qz_abD:=proc()
  local q1, q2, d, a, b, c, D;
  q1:=args[1]; q2:=args[2]; d:=args[3];
  b:=2*q1; c:=q1^2-d*q2^2;
  a:=ilcm(denom(b),denom(c)); if q2<0 then a:=-a; fi;
  b:=a*b; c:=a*c; D:=b^2-4*a*c;
  [a,b,D];
end;

# Erweiterter euklidischer Algorithmus
# Eingabe: [a_1,a_2,...,a_r]
# Ausgabe: [x_1,x_2,...,x_r,ggT] mit
# ggT=ggT(a_1,a_2,a_3,...)=a_1*x_1+a_2*x_2+...a_r*x_r
ggT_lk:=proc()
  local a, x, ggT, i, j, s, t;
  a:=args[1];
  x:=[1]; ggT:=a[1]; if ggT<0 then ggT:=-ggT; x:=[-1]; fi;
  for i from 2 to nops(a) do
    ggT:=igcdex(ggT,a[i],'s','t');
    for j from 1 to i-1 do
      x[j]:=s*x[j];
    od;
    x:=[op(x),t];
  od;
  [op(x),ggT];
end;

# Hermitesche Normalform eines Moduls
# M=\Z(x_1+y_1*sqrt(d))+...\Z(x_r+y_r*sqrt(d)) in Q(sqrt(d))
# Eingabe: [[x_1,y_1],[x_2,y_2],...,[x_r,y_r]] mit x_i,y_i aus Q
# Ausgabe: [q1,q2,q3] mit M=Z*q1+Z*(q2+q3*sqrt(d))
modul_hnf:=proc()
  local r, n, u, v, w, uu, vv, ww, i, x;
  r:=nops(args[1]);
  n:=1; vv:=[]; ww:=[];
  for i from 1 to r do
    vv:=[op(vv),op(i,args[1])[1]]; ww:=[op(ww),op(i,args[1])[2]];
    n:=ilcm(n,denom(vv[i]),denom(ww[i]));
  od;
  vv:=map(x->n*x,vv); ww:=map(x->n*x,ww);
  x:=ggT_lk(ww);
  w:=x[r+1]; v:=0; for i from 1 to r do v:=v+vv[i]*x[i]; od;
  if w>0 then
```

```
    uu:=[]; for i from 1 to r do uu:=[op(uu),vv[i]-ww[i]/w*v]; od;
  else
    uu:=vv;
  fi;
  u:=ggT_lk(uu)[r+1]; if u>0 then v:=v mod u; fi;
  [u/n,v/n,w/n];
end;

# Normalform [A,a,b,D]=A*(Z*a+Z*(b+sqrt(D))/2) eines Moduls
# M=\Z(x_1+y_1*sqrt(d))+...\Z(x_r+y_r*sqrt(d)) in Q(sqrt(d)) mit
# rationalen Zahlen x_i,y_i
# Eingabe: [[x_1,y_1],...,[x_r,y_r]],d
# Ausgabe: [A,a,b,D]
modul_nf:=proc()
  local d, q, abD, a, b, D, A;
  d:=args[2];
  q:=modul_hnf(args[1]); # M=Z*q1+Z*(q2+q3*sqrt(d))
  #printf("d=%d q=%a\n",d,q);
  abD:=qz_abD(q[2]/q[1],q[3]/q[1],d);
  a:=abD[1]; b:=abD[2]; D:=abD[3];
  b:=b mod (2*a); if b>a then b:=b-2*a; fi;
  A:=q[1]/a;
  [A,a,b,D];
end;

# Zu vorgegebener Diskriminante und Norm a werden alle primitiven
# Moduln mit Diskriminante D und Norm a bestimmt
# Eingabe: D, a
# Ausgabe: {[1,a,b_1,D],...,[1,a,b_r,D]}
modul_kon:=proc()
  local a, b, c, D, S, b0;
  D:=args[1];
  a:=args[2];
  if isdisc(D)=false then error "D ist keine Diskriminante"; fi;
  S:={}; b0:=D mod 2;
  for b from b0 to a by 2 do
    if (b^2-D) mod (4*a)=0 and igcd(a,b,(b^2-D)/(4*a))=1 then
      if b=a then S:=S union {[1,a,b,D]};
        else S:=S union {[1,a,b,D],[1,a,-b,D]};
      fi;
    fi;
  od;
  S;
end;

# Erweiterter euklidischer Algorithmus
# Eingabe: a1,a2,a3
# Ausgabe: [x1,x2,x3] mit ggT(a1,a2,a3)=a1*x1+a2*x2+a3*x3
ggT3:=proc()
  local a1, a2, a3, a12, b1, b2, a123, c, b3;
  a1:=args[1]; a2:=args[2]; a3:=args[3];
  a12:=igcdex(a1,a2,'b1','b2');
  a123:=igcdex(a12,a3,'c','b3');
  [b1*c,b2*c,b3];
```

```
end;

# Multiplikation von Moduln mit gleicher Diskriminante
# Eingabe: [A1,a1,b1,D],[A2,a2,b2,D]
# Ausgabe: [A,a,b,D] mit [A,a,b,D]=[A1,a1,b1,D]*[A2,a2,b2,D]
modul_multD:=proc()
  local M1, M2, A1, A2, a1, a2, b1, b2, D, xyz, x, y, z, A, b, a;
  M1:=args[1]; A1:=M1[1]; a1:=M1[2]; b1:=M1[3]; D:=M1[4];
  M2:=args[2]; A2:=M2[1]; a2:=M2[2]; b2:=M2[3];
  xyz:=ggT3(a1,a2,(b1+b2)/2); x:=xyz[1]; y:=xyz[2]; z:=xyz[3];
  A:=a1*x+a2*y+(b1+b2)/2*z;
  b:=(a1*b2*x+a2*b1*y+(b1*b2+D)/2*z)/A;
  a:=a1*a2/A^2;
  b:=b mod (2*a); if b>a then b:=b-2*a; fi;
  [A1*A2*A,a,b,D];
end;

# Multiplikation von Moduln im gleichen Zahlkoerper
# Eingabe: [A1,a1,b1,D1],[A1,a2,b2,D2]
# Ausgabe: [A,a,b,D] mit [A,a,b,D]=[A1,a1,b1,D1]*[A1,a2,b2,D2]
modul_mult:=proc()
  local M1, M2, A1, A2, a1, a2, b1, b2, D1, D2, m1, m2, d;
  M1:=args[1]; A1:=M1[1]; a1:=M1[2]; b1:=M1[3]; D1:=M1[4];
  M2:=args[2]; A2:=M2[1]; a2:=M2[2]; b2:=M2[3]; D2:=M2[4];
  # if D1=D2 then return modul_multD(M1,M2); fi;
  # D1=m1^2*d, D2=m2^2*d, D1/D2=(m1/m2)^2
  m2:=denom(D1/D2); m1:=m2*D1/D2;
  if issqr(m1)=false or issqr(m2)=false then
    error "Die Moduln liegen nicht im gleichen Zahlkoerper";
  fi;
  m1:=isqrt(m1); m2:=isqrt(m2); d:=D1/m1^2;
  modul_nf([[A1*A2*a1*a2,0],
            [A1*A2*a1*b2/2,A1*A2*a1*m2/2],
            [A1*A2*a2*b1/2,A1*A2*a2*m1/2],
            [A1*A2*(b1*b2+m1*m2*d)/4,A1*A2*(b1*m2+b2*m1)/4]],d);
end;

# Zu einer Diskriminante D wird die Fundamentaldiskriminante D1 sowie
# der Fuehrer f bestimmt, D=f^2*D1, was die Faktorisierbarkeit von D
# voraussetzt.
# Eingabe: D
# Ausgabe: [D1,f]
fund_disc:=proc()
  local D, P, f, D1, i, j;
  D:=args[1];
  if isdisc(D)=false then error "D ist keine Diskriminante"; fi;
  P:=ifactors(D)[2]; f:=1;
  for i from 1 to nops(P) do
    j:=iquo(P[i][2],2); f:=f*P[i][1]^j;
  od;
  D1:=D/f^2;
  if D1 mod 4=2 or D1 mod 4=3 then D1:=4*D1; f:=f/2; fi;
  [D1,f];
end;
```

```
# Test, ob [A,a,b,D] tatsaechlich einen Modul beschreibt
# Eingabe: [A,a,b,D]
# Ausgabe: true, false [A,a,b,D] Modul ist, sonst false
modul_test:=proc()
  local M, A, a, b, D, c;
  M:=args[1];
  a:=M[2]; b:=M[3]; D:=M[4];
  if (b^2-D) mod (4*a)<>0 then return false; fi;
  c:=(b^2-D)/(4*a);
  if igcd(a,b,c)>1 then return false; fi;
  true;
end;

# Grundeinheit der reellquadratischen Ordnung mit Diskriminante D.
# Start mit einer reduzierten Zahl (b+sqrt(D)/2=[1,b,D],
# naemlich b=wD bzw. b=wD-1. Dann ist epsilon=x+y*omega mit
# x=q_(k-2)+(b-(D mod 2)/2*q_(k-1), y=q_(k-1),
# omega=((D mod 2)+sqrt(D))/2 eine Grundeinheit.
# Eingabe: D
# Ausgabe: [x,y] mit epsilon=x+y*((D mod 2)+sqrt(D))/2 (Grundeinheit)
rq_einheit:=proc()
  local a, b, D, wD, u, aa, bb, kbe;
  D:=args[1];
  if D<0 or isdisc(D)=false then
    error "D ist keine positive Diskriminante";
  fi;
  wD:=isqrt(D); if wD^2>D then wD:=wD-1; fi;
  a:=1; b:=wD-((D-wD) mod 2);
  aa:=a; bb:=b; qi2:=1; qi1:=0; # kbe:=[];
  do
    u:=iquo(bb+wD,2*aa); # kbe:=[op(kbe),u];
    qi:=u*qi1+qi2; qi2:=qi1; qi1:=qi;
    bb:=2*aa*u-bb;
    aa:=(D-bb^2)/(4*aa);
    if aa=a and bb=b then break; fi;
  od;
  x:=qi2+qi1*(b-(D mod 2))/2; y:=qi1;
  test:=x^2+(D mod 2)*x*y+((D mod 2)-D)/4*y^2;
  if test<>1 and test<>-1 then error "keine Einheit!"; fi;
  [x,y];
end;

# Mit delta=D mod 2 sei omega=(delta+sqrt(D))/2 und R_D=Z+Z*omega
# Eingabe: x, y, D zu alpha=x+y*omega
# Ausgabe: [A,a,b,D] = alpha*R_D
hauptideal:=proc()
  local x, y, D, delta, M;
  x:=args[1]; y:=args[2]; D:=args[3]; delta:=D mod 2;
  if isdisc(D)=false then error "D ist keine Diskriminante"; fi;
  modul_nf([[x+y*delta/2,y/2],
            [x*delta/2+y*(delta+D)/4,(x+delta*y)/2]],D);
end;
```

```
# Konstruktion eines primitiven Moduls mit Norm p
# Eingabe: D,p
# Ausgabe: Menge der primitiven Moduln mit Diskriminante D und Norm p
modul_kon_p:=proc()
  local D, p, S, b;
  D:=args[1]; p:=args[2]; S:={};
  if isdisc(D)=false then error "D ist keine Diskriminante"; fi;
  if isprime(p)=false then error "p ist keine Primzahl"; fi;
  if D mod p=0 then
    if p=2 then
      if D mod 16=8 then S:=S union {[1,2,0,D]}; fi;
      if D mod 16=12 then S:=S union {[1,2,2,D]}; fi;
    else
      if D mod p^2>0 and D mod 4=0 then S:=S union {[1,p,0,D]}; fi;
      if D mod p^2>0 and D mod 4=1 then S:=S union {[1,p,p,D]}; fi;
    fi;
  else
    if p=2 then
      if D mod 8=1 then S:=S union {[1,2,1,D],[1,2,-1,D]}; fi;
    else
      if numtheory[legendre](D,p)=1 then
        b:=numtheory[msqrt](D,p);
        if (D+b) mod 2>0 then b:=p-b; fi;
        S:=S union {[1,p,b,D],[1,p,-b,D]};
      fi;
    fi;
  fi;
  S;
end;
```

17. algo7_ma

```
# algo7_ma
# Kapitel: Imaginaerquadratische Klassengruppen
# Version: 12.2.2002
#
# Funktionen:
# isdisc(D)
# modul_test([a,b,D])
# HD_red([a,b,D])
# HD_red_tex([a,b,D])
# HD_inv([a,b,D])
# modul_next(D,p0)
# Red(D)
# hD(D)
# HD_mult([a1,b1,D],[a2,b2,D])
# HD_quad([a,b,D])
# HD_pot([a,b,D],k)
# shanks(D,n)
# md5:=proc(S)
# gq_signatur(S,K,z)
# gq_key_gen:=proc(p,q,e,J)
# gq_signatur_test(T,Kpub,sig)
# gq_signatur_angriff(T,Kpub)
# iq_gq_key_gen(D,a_A,e)
```

```
# iq_gq_signatur(S,K,z)
# iq_gq_signatur_test(T,Kpub,sig)
# iq_gq_signatur_angriff(T,Kpub)

unprotect(D);

# Eingabe: D
# Ausgabe: true, falls D quadratische Diskriminante ist, sonst false
isdisc:=proc()
  local D;
  D:=args[1];
  if D mod 4=2 or D mod 4=3 or issqr(D) then return false; fi;
  true;
end;

# Eingabe: [a,b,D]
# Ausgabe: true, falls Z.a+Z.(b+sqrt(D))/2 Modul mit negativer
#          Diskriminante D ist, sonst false
modul_test:=proc()
  local C, a, b, D;
  C:=args[1]; a:=C[1]; b:=C[2]; D:=C[3];
  if D>=0 or D mod 4=2 or D mod 4=3 or
     a<=0 or (b^2-D) mod (4*a)>0 or igcd(a,b,(b^2-D)/(4*a))>1 then
     return false;
  fi;
  true;
end;

# Reduktionsverfahren
# Eingabe: [a,b,D], wobei -a<b<=a gelten sollte
# Ausgabe: [aa,bb,D] - reduzierter Modul in der Klasse von [a,b,D]
HD_red:=proc()
  local C, a, b, D, c;
  C:=args[1]; a:=C[1]; b:=C[2]; D:=C[3]; c:=(b^2-D)/(4*a);
  # b:=mods(b,2*a);
  while a>c do
    a:=c; b:=mods(-b,2*a); c:=(b^2-D)/(4*a);
  od;
  if a=c and b<0 then b:=-b; fi;
  [a,b,D];
end;

# Eingabe: [a,b,D] imaginaerquadratischer Modul
# Ausgabe: Reduktionsprozess in TeX-Format
HD_red_tex:=proc()
  local C, a, b, D, c;
  C:=args[1]; a:=C[1]; b:=C[2]; D:=C[3]; c:=(b^2-D)/(4*a);
  printf("$$\\begin{tabular}{|c|c|c|}\\hline\n");
  printf("$a$ & $b$ & $c$ \\\\ \\hline\\hline\n");
  printf("%d & %d & %d \\\\ \\hline\n",a,b,c);
  while a>c do
    a:=c; b:=mods(-b,2*a); c:=(b^2-D)/(4*a);
    printf("%d & %d & %d \\\\ \\hline\n",a,b,c);
  od;
```

```
  if a=c and b<0 then
    b:=-b;
    printf("%d & %d & %d \\\\ \\hline\n",a,b,c);
  fi;
  printf("\\end{tabular}$$\n");
  [a,b,D];
end;

# Eingabe: Modul M=[a,b,D]
# Ausgabe: Zu M inverser Modul in der Klassengruppe
HD_inv:=proc()
  local C, a, b, D, c;
  C:=args[1]; a:=C[1]; b:=C[2]; D:=C[3]; c:=(b^2-D)/(4*a);
  b:=-b;
  if b=-a then b:=a; fi;
  if a=c and b<0 then b:=-b; fi;
  [a,b,D];
end;

# Fuer Diskrimiante D und p0 wird der naechste Modul mit Primzahlnorm p
# konstruiert
# Eingabe: D,p0
# Ausgabe: [p,b,D] mit p>=p0 minimal
modul_next:=proc()
  local D, p, C, b;
  D:=args[1]; p:=args[2]-1; C:=[];
  if isdisc(D)=false then error "D ist keine Diskriminante"; fi;
  while C=[] do
    p:=nextprime(p);
    if D mod p=0 then
      if p=2 then
        if D mod 16=8 then C:=[2,0,D]; fi;
        if D mod 16=12 then C:=[2,2,D]; fi;
      else
        if D mod p^2>0 and D mod 4=0 then C:=[p,0,D]; fi;
        if D mod p^2>0 and D mod 4=1 then C:=[p,p,D]; fi;
      fi;
    else
      if p=2 then
        if D mod 8=1 then C:=[2,1,D]; fi;
      else
        if numtheory[legendre](D,p)=1 then
          b:=numtheory[msqrt](D,p);
          if (D+b) mod 2>0 then b:=p-b; fi;
          C:=[p,b,D];
        fi;
      fi;
    fi;
  od;
  C;
end;

# Red(D) - Die Menge der reduzierten Moduln mit Diskriminante D<0
# Eingabe: imaginaerquadratische Diskriminante D
```

```
# Ausgabe: Menge der reduzierten [a,b,D] mit Diskriminante D
Red:=proc()
  local D, a, b, c, ac, wac, S;
  D:=args[1]; S:={};
  b:=D mod 2;
  do
    ac:=(b^2-D)/4; wac:=isqrt(ac); if wac^2>ac then wac:=wac-1; fi;
    if b>wac then return S; fi;
    a:=b; if a=0 then a:=1; fi;
    do
      if ac mod a=0 then
        c:=ac/a;
        if igcd(a,b,c)=1 then
          S:=S union {[a,b,D]};
          if b>0 and b<a and a<c then S:=S union {[a,-b,D]}; fi;
        fi;
      fi;
      a:=a+1;
      if a>wac then break; fi;
    od;
    b:=b+2;
  od;
  S;
end;

# Eingabe: imaginaerquadratische Diskriminante D
# Ausgabe: Klassenzahl h(D)
hD:=proc()
  local D, a, b, c, ac, wac, h;
  D:=args[1]; h:=0;
  b:=D mod 2;
  do
    ac:=(b^2-D)/4; wac:=isqrt(ac); if wac^2>ac then wac:=wac-1; fi;
    if b>wac then return h; fi;
    a:=b; if a=0 then a:=1; fi;
    do
      if ac mod a=0 then
        c:=ac/a;
        if igcd(a,b,c)=1 then
          h:=h+1;
          if b>0 and b<a and a<c then h:=h+1; fi;
        fi;
      fi;
      a:=a+1;
      if a>wac then break; fi;
    od;
    b:=b+2;
  od;
end;

# Multiplikation in der Klassengruppe
# Eingabe: [a1,b1,D], [a2,b2,D]
# Ausgabe: [a,b,D] (reduziert) mit [a,b,D]=[a1,b1,D]*[a2,b2,D] in H(D)
HD_mult:=proc()
```

```
  local C1, C2, a1, a2, b1, b2, D, a12, x, y, a, b, A, u, v;
  C1:=args[1]; a1:=C1[1]; b1:=C1[2]; D:=C1[3];
  C2:=args[2]; a2:=C2[1]; b2:=C2[2];
  a12:=igcdex(a1,a2,'x','y');
  a:=a1*a2; b:=a1*b2*x+a2*b1*y;
  if a12>1 then
    A:=igcdex(a12,(b1+b2)/2,'u','v');
    b:=b*u+(b1*b2+D)/2*v;
    if A>1 then a:=a/A^2; b:=b/A; fi;
  fi;
  b:=mods(b,2*a);
  HD_red([a,b,D]);
end;

# Quadrieren in der Klassengruppe
# Eingabe: [a,b,D]
# Ausgabe: [aa,bb,D]=[a,b,D]^2 in H(D)
HD_quad:=proc()
  local C, a, b, D, A, aa, bb, x, y;
  C:=args[1]; a:=C[1]; b:=C[2]; D:=C[3];
  A:=igcdex(a,b,'x','y');
  aa:=a^2/A^2; bb:=mods((a*b*x+(b^2+D)/2*y)/A,2*aa);
  HD_red([aa,bb,D]);
end;

# Potenzieren in der Klassengruppe
# Eingabe: C, k
# Ausgabe: C^k in H(D)
HD_pot:=proc()
  local C, k, D, X, Y;
  C:=args[1]; k:=args[2]; D:=C[3];
  X:=C;
  if k mod 2=0 then Y:=[1,D mod 2,D]; else Y:=C; fi;
  while k>1 do
    k:=iquo(k,2);
    X:=HD_quad(X);
    if k mod 2=1 then Y:=HD_mult(X,Y); fi;
  od;
  Y;
end;

# Berechnet wird sqrt(-D)/pi*prod
# Eingabe: D, n
# Ausgabe: sqrt(-D)/pi*prod(1/(1-(D/p_i)/p_i),i=1..n), konvergiert fuer
#          n gegen unendlich gegen die Klassenzahl h(D)
shanks:=proc()
  local D, n, Zaehler, Nenner, i, p;
  Digits:=50;
  D:=args[1]; n:=args[2];
  if D>=0 or D mod 4=2 or D mod 4=3 then
    error "D ist keine Diskriminante";
  fi;
  Zaehler:=1; Nenner:=1;
  for i from 1 to n do
```

```
    p:=ithprime(i);
    Zaehler:=Zaehler*p;
    Nenner:=Nenner*(p-numtheory[jacobi](D,p));
    printf("i=%d: %f\n",i,sqrt(-D)*Zaehler/Nenner/Pi);
  od;
  evalf(sqrt(-D)*Zaehler/Nenner/Pi);
end;

# MD5-Hashfunktion: Berechnung durch Abspeichern der Zeichenkette in
# eine Datei ‘md5_tmp1’, dann Aufruf ‘md5 md5_tmp1 > md5_tmp2’, dann
# Einlesen der Datei ‘md5_tmp2’.
# Eingabe: Zeichenkette S
# Ausgabe: md5-Hashwert von S in Dezimaldarstellung
md5:=proc()
  local S;
  S:=args[1];
  if type(S,string)=false then
    error "Eingabe ist keine Zeichenkette"; fi;
  fprintf(md5_tmp1,"%s",S); fclose(md5_tmp1);
  system("md5 md5_tmp1 > md5_tmp2");
  hbytes:=readbytes(md5_tmp2,infinity); fclose(md5_tmp2);
  fremove(md5_tmp1); fremove(md5_tmp2);
  # hbytes ist Liste mit Bytes, letzter Eintrag ist 10 bzw. \n
  hbytes:=subsop(nops(hbytes)=NULL,hbytes);
  # Die letzten 32 Eintraege liefern den Hashwert
  while nops(hbytes)>32 do hbytes:=subsop(1=NULL,hbytes); od;
  h:=convert(hbytes,bytes); # Hexadezimaldarstellung
  # printf("md5-Hashwert: %s\n",h);
  h:=convert(h,decimal,hex); # Umwandlung in Dezimalzahl
end;

# GQ-Signatur - 30.1.2002
# Zunaechst fuer Zeichenketten
# Will man das auf Dateien anwenden, bildet man mit
#  Sdatei:=convert(readbytes(datei,infinity),bytes);
# die zugehoerige Zeichenkette.

# GQ-Signatur
# Eingabe: S,K,z - S Datei oder Zeichenkette, K GQ-Schluessel
#           K=[N,e,J,a]
#           z Zufallszahl
# Ausgabe: GQ-Signatur
gq_signatur:=proc()
  local T, K, N, e, J, a, r, Tr, t, s;
  if nargs<>3 then
    error "Eingabe: Zeichenkette, privater GQ-Schluessel, Zufallszahl";
  fi;
  T:=args[1]; K:=args[2]; z:=args[3];
  N:=K[1]; e:=K[2]; J:=K[3]; a:=K[4];
  if (J*(Power(a,e) mod N)) mod N<>1 then
    error "2. Argument ist kein privater GQ-Schluessel"; fi;
  if type(T,string)=false then
    error "1. Argument ist keine Zeichenkette"; fi;
  r:=Power(z,e) mod N;
```

```
  Tr:=cat(T,convert(r,string));
  t:=md5(Tr);
  s:=(z*(Power(a,t) mod N)) mod N;
  [s,t];
end;

# GQ-Schluessel-Erzeugung
# Eingabe: p, q, e, J;
# Ausgabe: [N,e,J,a] - privater GQ-Schluessel
gq_key_gen:=proc()
  local p, q, e, J, N, d, a;
  p:=args[1]; q:=args[2]; e:=args[3]; J:=args[4];
  N:=p*q;
  while igcd(e,(p-1)*(q-1))>1 do e:=e+1; od;
  d:=1/e mod (p-1)*(q-1);
  a:=Power(1/J mod N,d) mod N; # a=J^(-d) mod N
  [N,e,J,a];
end;

# Ueberpruefen einer GQ-Signatur
# Eingabe: T, Kpub, sig
# Ausgabe: true oder false
gq_signatur_test:=proc()
  local T, Kpub, N, e, J, sig, s, t, rr, tt;
  if nargs<>3 then
    error "Eingabe: Zeichenkette, oeffentlicher GQ-Schluessel, Signatur";
  fi;
  T:=args[1]; Kpub:=args[2]; sig:=args[3];
  N:=Kpub[1]; e:=Kpub[2]; J:=Kpub[3]; s:=sig[1]; t:=sig[2];
  rr:=((Power(s,e) mod N)*(Power(J,t) mod N)) mod N;
  tt:=md5(cat(T,convert(rr,string)));
  if t=tt then true else false; fi;
end;

# Zu einer Zeichenkette T, einem oeffentlichen GQ-Schluessel Kpub wird
# versucht, eine (gefaelschte) GQ-Signatur zu konstruieren.
# Eingabe: T, Kpub
# Ausgabe: GQ-Signatur von T fuer GQ-Schluessel Kpub
gq_signatur_angriff:=proc()
  local T, Kpub, N, e, J, x, rx, tx, y, s, t, zeit1, zeit2;
  T:=args[1]; Kpub:=args[2];
  N:=Kpub[1]; e:=Kpub[2]; J:=Kpub[3]; x:=-1; zeit1:=time();
  do
    x:=x+1;
    rx:=Power(J,x) mod N;
    tx:=md5(cat(T,convert(rx,string)));
    if (tx-x) mod e=0 then
      y:=(tx-x)/e;
      s:=Power(1/J mod N,y) mod N; t:=tx;
      zeit2:=time();
      printf("x=%d (%.2f sec)\n",x,zeit2-zeit1);
      return [s,t];
    fi;
  od:
```

```
end;

# IQ-GQ-Signatur-Schluesselerzeugung
# Eingabe: D, a_A, e - naeherungsweise
# [D,e,J,A]
iq_gq_key_gen:=proc()
  local D, a_A, e, J, A;
  if nargs<>3 then error "Eingabe: D, a_A, e"; fi;
  D:=args[1]; a_A:=args[2]; e:=args[3];
  while D>=0 or D mod 4=2 or D mod 4=3 do D:=D-1; od;
  A:=modul_next(D,a_A); A:=HD_red(A);
  J:=HD_pot(HD_inv(A),e);
  [D,e,J,A];
end;

# IQ-GQ-Signatur
# Eingabe: S,K,z - S Datei oder Zeichenkette, K IQ-GQ-Schluessel
#           K=[D,e,J,A]
#           z Zufallszahl
# Ausgabe: IQ-GQ-Signatur
iq_gq_signatur:=proc()
  local T, K, N, e, J, A, R, a_R, b_R, TR, t, S, z, Z;
  if nargs<>3 then
    error "Eingabe: Zeichenkette, privater IQ-GQ-Schluessel, Zufallszahl";
  fi;
  T:=args[1]; K:=args[2]; z:=args[3];
  D:=K[1]; e:=K[2]; J:=K[3]; A:=K[4];
  if HD_mult(J,HD_pot(A,e))[1]<>1 then
    error "2. Argument ist kein privater IQ-GQ-Schluessel"; fi;
  if type(T,string)=false then
    error "1. Argument ist keine Zeichenkette"; fi;
  Z:=HD_red(modul_next(D,z));
  R:=HD_pot(Z,e); a_R:=R[1]; b_R:=R[2];
  TR:=cat(T,convert(a_R,string),convert(b_R,string));
  t:=md5(TR);
  S:=HD_mult(Z,HD_pot(A,t));
  [S,t];
end;

# Ueberpruefen der IQ-GQ-Signatur sig einer Zeichenkette T mit
# oeffentlichem Schluessel Kpub
# Eingabe: T, Kpub, sig
# Ausgabe: true oder false
iq_gq_signatur_test:=proc()
  local T, Kpub, N, e, J, sig, s, t, rr, tt;
  if nargs<>3 then
    error "Eingabe: Zeichenkette, oeffentlicher IQ-GQ-Schluessel, Signatur";
  fi;
  T:=args[1]; Kpub:=args[2]; sig:=args[3];
  D:=Kpub[1]; e:=Kpub[2]; J:=Kpub[3]; S:=sig[1]; t:=sig[2];
  RR:=HD_mult(HD_pot(S,e),HD_pot(J,t)); a_RR:=RR[1]; b_RR:=RR[2];
  tt:=md5(cat(T,convert(a_RR,string),convert(b_RR,string)));
  if t=tt then true else false; fi;
end;
```

```
# Versuch, fuer eine Zeichenkette T und einen IQ-GQ-Schluessel Kpub eine
# Signatur zu faelschen
# Eingabe: T, Kpub
# Ausgabe: IQ-GQ-Signatur
iq_gq_signatur_angriff:=proc()
  local T, Kpub, N, e, J, x, rx, tx, y, s, t, zeit1, zeit2;
  T:=args[1]; Kpub:=args[2];
  D:=Kpub[1]; e:=Kpub[2]; J:=Kpub[3]; x:=-1; zeit1:=time();
  do
    x:=x+1;
    Rx:=HD_pot(J,x);
    tx:=md5(cat(T,convert(Rx[1],string),convert(Rx[2],string)));
    if (tx-x) mod e=0 then
      y:=(tx-x)/e;
      S:=HD_pot(HD_inv(J),y); t:=tx;
      zeit2:=time();
      printf("x=%d (%.2f sec)\n",x,zeit2-zeit1);
      return [S,t];
    fi;
  od:
end;
```

18. iqkg_gmp.c

```
/* iqkg_gmp.c
 * 14.12.2001, 16.1.2002, 17.1.2002, 20.1.2002, 26.1.2002, 12.2.2002
 *
 * int pprim( int n)
 * int ppotenz( int p, int k)
 * int eratosthenes( int p[], int k)
 * void mpz_gcdext2( mpz_t d, mpz_t u, mpz_t v, mpz_t a, mpz_t b)
 * int pm1( mpz_t d, mpz_t N, int k, int a)
 * int isdisc( mpz_t D)
 * void hD( mpz_t h, mpz_t D)
 * void hd_red( mpz_t a, mpz_t b, mpz_t D)
 * void hd_ausgabe( mpz_t a, mpz_t b, mpz_t D)
 * void hd_mult( mpz_t a, mpz_t b,
 * void hd_quad( mpz_t a2, mpz_t b2, mpz_t a, mpz_t b, mpz_t D)
 * void hd_pot( mpz_t ka, mpz_t kb, mpz_t a, mpz_t b, mpz_t D, mpz_t k)
 * void hd_pot_ui( mpz_t ka, mpz_t kb, mpz_t a, mpz_t b, mpz_t D, int k)
 * int modul_kon( mpz_t a, mpz_t b, mpz_t D)
 * void modul_next( mpz_t a, mpz_t b, mpz_t D)
 * void modul_next_p( mpz_t a, mpz_t b, mpz_t D)
 * void hd_kgv( mpz_t ka, mpz_t kb, mpz_t a, mpz_t b, mpz_t D, int k)
 * void hd_fac( mpz_t N1, mpz_t N, int k)
 */

#include <stdio.h>
#include <gmp.h>
#include <time.h>

/* pprim(n) testet, ob ein kleines ungerades n prim ist oder nicht */
int pprim( int n)
```

```
{
  int t;
  for (t=3;t*t<=n;t=t+2) if (n%t==0) return 0;
  return 1;
}

/* ppotenz(p,k) liefert die groesste p-Potenz <=k (12.12.2000) */
int ppotenz( int p, int k)
{
  int pp=1;
  while (k/pp>=p) pp*=p;
  return pp;
}

/* Das Sieb des Eratosthenes: Eine Liste aller Primzahlen <=k wird
 * erstellt. Rueckgabewert ist die Anzahl der Primzahlen.
 * (12.12.2000) */
int eratosthenes( int p[], int k)
{
  int i, j, az=0;

  for (i=2;i<=k;i++) p[i]=1;

  for (i=2;i*i<=k;i++)
    if (p[i]==1) /* alle echten Vielfachen von i werden 0 gesetzt */
      for (j=2*i;j<=k;j+=i) p[j]=0;

  az=0; for (i=2;i<=k;i++) if (p[i]==1) p[az++]=i;
  return az;
}

/* Erweiterter euklidischer Algorithmus - Cohen, Algorithmus 1.3.6
   (16.1.2002)
*/
void mpz_gcdext2( mpz_t d, mpz_t u, mpz_t v, mpz_t a, mpz_t b)
{
  mpz_t aa, bb, v1, v3, t1, t3, q;

  mpz_init_set(aa,a); mpz_init_set(bb,b);
  if (mpz_sgn(a)<0) mpz_neg(aa,a);
  if (mpz_sgn(b)<0) mpz_neg(bb,b);
  mpz_set_ui(u,1); mpz_set(d,aa); // u=1, d=a
  mpz_init_set_ui(v1,0); mpz_init_set(v3,bb); // v1=0, v3=b
  mpz_init(t1); mpz_init(t3); mpz_init(q);

  while (mpz_sgn(v3)>0)
  {
    mpz_fdiv_q(q,d,v3); mpz_mod(t3,d,v3); // q=[d/v3], t3=d mod v3
    mpz_mul(t1,q,v1); mpz_sub(t1,u,t1); // t1=u-q*v1
    mpz_set(u,v1); // u=v1
    mpz_set(d,v3); // d=v3
    mpz_set(v1,t1); // v1=t1
    mpz_set(v3,t3); // v3=t3
  }
```

```
  if (mpz_sgn(bb)==0)
    mpz_set_ui(v,0);
  else
  {
    // v=(d-a*u)/b
    mpz_mul(v,aa,u); mpz_sub(v,d,v); mpz_divexact(v,v,bb);
  }
  if (mpz_sgn(a)<0) mpz_neg(u,u);
  if (mpz_sgn(b)<0) mpz_neg(v,v);

  mpz_clear(aa); mpz_clear(bb); mpz_clear(v1); mpz_clear(v3);
  mpz_clear(t1); mpz_clear(t3); mpz_clear(q);

  return;
}

/* (p-1)-Methode: d=ggT(a^kgV(1..k)-1,N)
 * Rueckgabewert 0 fuer d=1, 1 fuer 1<d<n, 2 fuer d=n
 * (12.12.2000, 26.1.2002) */
int pm1( mpz_t d, mpz_t N, int k, int a)
{
  int *p, azp, i;

  p=(int *) malloc((k+1)*sizeof(int));
  azp=eratosthenes(p,k);

  mpz_set_ui(d,a); // d=a^kgV(1..k) mod N
  for (i=0;i<azp;i++) mpz_powm_ui(d,d,ppotenz(p[i],k),N);

  mpz_sub_ui(d,d,1); mpz_gcd(d,N,d); // d=ggT(a^kgV(1..k)-1,N)

  if (mpz_cmp_ui(d,1)==0) return 0;
    else if (mpz_cmp(d,N)==0) return 2;
    else return 1;
}

/* isdisc(D) testet, ob D eine imaginaerquadratische Diskriminante ist
 * oder nicht */
int isdisc( mpz_t D)
{
  int r;
  mpz_t rr;
  if (mpz_sgn(D)>=0) return 0;
  mpz_init(rr); r=mpz_mod_ui(rr,D,4); mpz_clear(rr);
  if (r==0 || r==1) return 1; else return 0;
}

/* Bestimmung der Klassenzahl (20.1.2002) */
void hD( mpz_t h, mpz_t D)
{
  mpz_t a, b, c, absD, ac, wac, r, ggT;

  mpz_init(a); mpz_init(b); mpz_init(c); mpz_init(absD);
  mpz_init(ac); mpz_init(wac); mpz_init(r); mpz_init(ggT);
```

```
  mpz_mod_ui(b,D,2); mpz_neg(absD,D); mpz_set_ui(h,0);

  do
  {
    mpz_mul(ac,b,b); mpz_add(ac,ac,absD); mpz_fdiv_q_ui(ac,ac,4);
    mpz_sqrt(wac,ac);

    if (mpz_cmp(b,wac)>0)
    {
      mpz_clear(a); mpz_clear(b); mpz_clear(c); mpz_clear(absD);
      mpz_clear(ac); mpz_clear(wac); mpz_clear(r); mpz_clear(ggT);
      return;
    }

    mpz_set(a,b); if (mpz_sgn(b)==0) mpz_set_ui(a,1);

    do
    {
      mpz_fdiv_qr(c,r,ac,a);
      if (mpz_sgn(r)==0)
      {
        mpz_gcd(ggT,a,b); mpz_gcd(ggT,c,ggT);
        if (mpz_cmp_ui(ggT,1)==0)
        {
          mpz_add_ui(h,h,2);
          if (mpz_cmp(a,b)==0 || mpz_sgn(b)==0 || mpz_cmp(a,c)==0)
            mpz_sub_ui(h,h,1);
        }
      }
      mpz_add_ui(a,a,1);
    } while (mpz_cmp(a,wac)<=0);

    mpz_add_ui(b,b,2);
  } while (1);
}

/* Reduktionprozess auf [a,b,D] angewandt */
void hd_red( mpz_t a, mpz_t b, mpz_t D)
{
  mpz_t c, hi;
  mpz_init(c); mpz_init(hi);

  mpz_mul(c,b,b); mpz_sub(c,c,D);
  mpz_divexact(c,c,a); mpz_fdiv_q_2exp(c,c,2); // c=(b^2-D)/(4*a)

  while (mpz_cmp(a,c)>0)
  {
    mpz_set(a,c);
    mpz_mul_2exp(hi,a,1); mpz_neg(b,b); mpz_mod(b,b,hi);
    if (mpz_cmp(b,a)>0) mpz_sub(b,b,hi);
    mpz_mul(c,b,b); mpz_sub(c,c,D); mpz_divexact(c,c,a);
    mpz_fdiv_q_2exp(c,c,2);
  }
```

```
  if (mpz_cmp(a,c)==0 && mpz_sgn(b)<0) mpz_neg(b,b);
  mpz_clear(c); mpz_clear(hi);
  return;
}

/* Zum Ausdruck von [a,b,D] */
void hd_ausgabe( mpz_t a, mpz_t b, mpz_t D)
{
  printf("["); mpz_out_str(stdout,10,a); printf(",");
  mpz_out_str(stdout,10,b); printf(",");
  mpz_out_str(stdout,10,D); printf("]");
  return;
}

/* Multiplikation in der Klassengruppe */
void hd_mult( mpz_t a, mpz_t b,
              mpz_t a1, mpz_t b1, mpz_t a2, mpz_t b2, mpz_t D)
{
  mpz_t a12, x, y, A, hi, aa, bb;
  mpz_init(a12); mpz_init(x); mpz_init(y); mpz_init(A); mpz_init(hi);
  mpz_init(aa); mpz_init(bb);
  // printf("C1="); hd_ausgabe(a1,b1,D); printf("\t");
  // printf("C2="); hd_ausgabe(a2,b2,D); printf("\n");
  //mpz_gcdext2(a12,x,y,a1,a2);
  mpz_gcdext(a12,x,y,a1,a2);
  mpz_mul(aa,a1,a2); // a=a1*a2
  mpz_mul(hi,a1,b2); mpz_mul(hi,hi,x); mpz_mul(bb,a2,b1);
  mpz_mul(bb,bb,y); mpz_add(bb,hi,bb); // b=a1*b2*x+a2*b1*y
  if (mpz_cmp_ui(a12,1)>0)
  {
    mpz_add(hi,b1,b2); mpz_fdiv_q_2exp(hi,hi,1); // (b1+b2)/2
    //mpz_gcdext2(A,x,y,a12,hi); // A=a12*x+(b1+b2)/2*y
    mpz_gcdext(A,x,y,a12,hi); // A=a12*x+(b1+b2)/2*y
    mpz_mul(hi,b1,b2); mpz_add(hi,hi,D); mpz_fdiv_q_2exp(hi,hi,1);
    mpz_mul(hi,hi,y); // (b1*b2+D)/2*y
    mpz_mul(bb,bb,x); mpz_add(bb,bb,hi); // b=b*x+(b1*b2+D)/2*y
    if (mpz_cmp_ui(A,1)>0)
    {
      mpz_divexact(aa,aa,A); mpz_divexact(aa,aa,A);
      mpz_divexact(bb,bb,A);
    }
  }
  mpz_mul_2exp(hi,aa,1); mpz_mod(bb,bb,hi);
  if (mpz_cmp(bb,aa)>0) mpz_sub(bb,bb,hi); // b=b mod 2*a mit -a<b<=a
  hd_red(aa,bb,D); mpz_set(a,aa); mpz_set(b,bb);
  mpz_clear(a12); mpz_clear(x); mpz_clear(y); mpz_clear(A);
  mpz_clear(hi); mpz_clear(aa); mpz_clear(bb);
  return;
}

/* Quadrieren in der Klassengruppe */
void hd_quad( mpz_t a2, mpz_t b2, mpz_t a, mpz_t b, mpz_t D)
{
  mpz_t A, x, y, hi, aa, bb;
```

```
  mpz_init(A); mpz_init(x); mpz_init(y); mpz_init(hi);
  mpz_init(aa); mpz_init(bb);
  //mpz_gcdext2(A,x,y,a,b);
  mpz_gcdext(A,x,y,a,b);
  mpz_mul(aa,a,a);
  mpz_mul(hi,b,b); mpz_add(hi,hi,D); mpz_fdiv_q_2exp(hi,hi,1);
  mpz_mul(hi,hi,y); mpz_mul(bb,a,b); mpz_mul(bb,bb,x);
  mpz_add(bb,bb,hi);
  if (mpz_cmp_ui(A,1)!=0)
  {
    mpz_divexact(aa,aa,A); mpz_divexact(aa,aa,A);
    mpz_divexact(bb,bb,A);
  }
  mpz_mul_2exp(hi,aa,1);
  mpz_mod(bb,bb,hi);
  if (mpz_cmp(bb,aa)>0) mpz_sub(bb,bb,hi);
  hd_red(aa,bb,D); mpz_set(a2,aa); mpz_set(b2,bb);
  mpz_clear(A); mpz_clear(x); mpz_clear(y); mpz_clear(hi);
  mpz_clear(aa); mpz_clear(bb);
  return;
}

/* Potenzieren [a,b,D]^k in der Klassengruppe */
void hd_pot( mpz_t ka, mpz_t kb, mpz_t a, mpz_t b, mpz_t D, mpz_t k)
{
  mpz_t xa, xb, ya, yb, kk, hi;
  mpz_init_set(kk,k); mpz_init(hi);
  mpz_init_set(xa,a); mpz_init_set(xb,b);
  if (mpz_mod_ui(hi,kk,2)==0)
  {
    mpz_init_set_ui(ya,1); mpz_init(yb); mpz_mod_ui(yb,D,2);
  }
  else
  {
    mpz_init_set(ya,a); mpz_init_set(yb,b);
  }
  while (mpz_cmp_ui(kk,1)>0)
  {
    mpz_fdiv_q_2exp(kk,kk,1);
    hd_quad(xa,xb,xa,xb,D);
    if (mpz_mod_ui(hi,kk,2)==1) hd_mult(ya,yb,xa,xb,ya,yb,D);
  }
  mpz_set(ka,ya); mpz_set(kb,yb);
  mpz_clear(kk); mpz_clear(hi);
  mpz_clear(xa); mpz_clear(xb); mpz_clear(ya); mpz_clear(yb);
  return;
}

/* Potenzieren [a,b,D]^k in der Klassengruppe mit int k */
void hd_pot_ui( mpz_t ka, mpz_t kb, mpz_t a, mpz_t b, mpz_t D, int k)
{
  mpz_t xa, xb, ya, yb, hi;
  mpz_init(hi); mpz_init_set(xa,a); mpz_init_set(xb,b);
  if (k%2==0)
```

```
  {
    mpz_init_set_ui(ya,1); mpz_init(yb); mpz_mod_ui(yb,D,2);
  }
  else
  {
    mpz_init_set(ya,a); mpz_init_set(yb,b);
  }
  while (k>1)
  {
    k>>=1;
    hd_quad(xa,xb,xa,xb,D);
    if (k%2==1) hd_mult(ya,yb,xa,xb,ya,yb,D);
  }
  mpz_set(ka,ya); mpz_set(kb,yb);
  mpz_clear(xa); mpz_clear(xb); mpz_clear(ya); mpz_clear(yb);
  mpz_clear(hi);
  return;
}

/* Zu gegebenen D, a wird getestet, ob ein Modul [a,b,D] existiert. Wenn
 * ja, wird 1 zurueckgegeben mit dem Modul, wenn nein, 0. */
int modul_kon( mpz_t a, mpz_t b, mpz_t D)
{
  mpz_t c, hi;
  mpz_init(c); mpz_init(hi);

  mpz_mod_ui(b,D,2);
  while (mpz_cmp(b,a)<=0)
  {
    mpz_mul(hi,b,b); mpz_sub(hi,D,hi); mpz_fdiv_q_2exp(c,hi,2);
    mpz_fdiv_qr(c,hi,c,a);
    if (mpz_sgn(hi)==0)
    {
      mpz_gcd(hi,a,b); mpz_gcd(hi,hi,c);
      if (mpz_cmp_ui(hi,1)==0)
      {
        mpz_clear(c); mpz_clear(hi);
        return 1;
      }
    }
    mpz_add_ui(b,b,2);
  }
  mpz_clear(c); mpz_clear(hi);
  return 0;
}

/* Zu D, a=a0 wird der ‘naechste’ Modul [a,b,D] mit a>=a0 bestimmt. */
void modul_next( mpz_t a, mpz_t b, mpz_t D)
{
  while (modul_kon(a,b,D)==0)
  {
    printf("a="); mpz_out_str(stdout,10,a); printf("\n");
    mpz_add_ui(a,a,1);
  }
```

```
  return;
}

/* Zu D, a0 wird der ‘naechste Modul [p,b,D] mit p>a0 bestimmt, wobei p
 * eine Primzahl ist */
void modul_next_p( mpz_t a, mpz_t b, mpz_t D)
{
  do
  {
    mpz_nextprime(a,a);
  } while (modul_kon(a,b,D)==0);
  return;
}

/* Berechnung von [a,b,D]^kgV(3,5,7,9,11,...,k) */
void hd_kgv( mpz_t ka, mpz_t kb, mpz_t a, mpz_t b, mpz_t D, int k)
{
  int p, q, qp, t;
  mpz_set(ka,a); mpz_set(kb,b);
  for (p=3;p<=k;p=p+2)
  {
    if (pprim(p)==1)
    {
      q=1;
      while ((qp=q*p)<=k && qp>0) q=qp;
      hd_pot_ui(ka,kb,ka,kb,D,q);
      // printf("q=%d\t",q); hd_ausgabe(ka,kb,D); printf("\n");
    }
  }
}

/* Faktorisierung mit Klassengruppe, Eingabe N, k (23.1.2002) */
void hd_fac( mpz_t N1, mpz_t N, int k)
{
  int *pp, azp, i, multiplikator, kk;
  mpz_t D, a, b, ka, kb, ka2, kb2, ggT;

  mpz_init(D); mpz_init(a); mpz_init(b); mpz_init(ka); mpz_init(kb);
  mpz_init(ka2); mpz_init(kb2); mpz_init(ggT);

  // Hier wird eine Liste aller Primzahlen p<=k erzeugt.
  pp=(int *) malloc((k+1)*sizeof(int));
  azp=eratosthenes(pp,k);
  for (i=0;i<azp;i++) pp[i]=ppotenz(pp[i],k);

  for (multiplikator=1;;multiplikator++)
  {
    mpz_mul_si(D,N,-multiplikator);
    if (isdisc(D)==1)
    {
      // printf("Multiplikator=%d\n",multiplikator);
      mpz_set_ui(a,multiplikator);
      do
      {
```

```
        modul_next_p(a,b,D);
        mpz_set(ka,a); mpz_set(kb,b);
        for (i=1;i<azp;i++) hd_pot_ui(ka,kb,ka,kb,D,pp[i]);

        kk=k;
        while ((kk/=2)>0)
        {
          hd_quad(ka2,kb2,ka,kb,D);
          if (mpz_cmp_ui(ka2,1)==0)
          {
            if (mpz_sgn(kb)==0 || mpz_cmp(ka,kb)==0) // b=0 oder a=b
              mpz_set(ggT,ka);
            else // a=c
            {
              mpz_mul_2exp(ggT,ka,1); mpz_sub(ggT,ggT,kb); // 2a-b
            }
            mpz_gcd(ggT,ggT,N);
            if (mpz_cmp_ui(ggT,1)!=0)
            {
              mpz_set(N1,ggT);
              printf("k=%d Multiplikator=%d\n",k,multiplikator);
              gmp_printf("Teiler=%Zd\n",ggT);
              mpz_clear(D); mpz_clear(a); mpz_clear(b); mpz_clear(ka);
              mpz_clear(kb); mpz_clear(ka2); mpz_clear(kb2);
              mpz_clear(ggT);
              return;
            }
            kk=0;
          }
          mpz_set(ka,ka2); mpz_set(kb,kb2);
        }
      } while (mpz_cmp_ui(ka,1)==0);
    }
  }
}
```

19. iqkg_hd_gmp.c

```
/* iqkg_hd_gmp.c - 20.1.2002, 12.2.2002 */

#include "iqkg_gmp.c"

main()
{
  int d1, d2;
  mpz_t D, h;
  time_t zeit1, zeit2;
  mpz_init(D); mpz_init(h);

  printf("Auflistung von imaginaerquadratischen Klassenzahlen ");
  printf("mit d1<=abs(D)<=d2.\n");
  printf("d1="); scanf("%d",&d1);
  printf("d2="); scanf("%d",&d2);

  time(&zeit1);
```

```
  mpz_set_si(D,-d1);

  while (mpz_cmp_si(D,-d2)>=0)
  {
    if (isdisc(D)==1)
    {
      hD(h,D);
      gmp_printf("h(%Zd)=%Zd\n",D,h);
    }
    mpz_sub_ui(D,D,1);
  }

  time(&zeit2);
  printf("Zeit: %.0f sec\n",difftime(zeit2,zeit1));
}
```

20. iqkg_fac_gmp.c

```
/* iqkg_fac_gmp.c - 19.1.2002
 * Faktorisierung von N mit imaginaerquadratischen Klassengruppen.
 * Weder werden kleine Teiler abgespalten noch wird ein Primzahltest
 * gemacht. */

#include "iqkg_gmp.c"

main()
{
  int k;
  mpz_t N, N1, N2;
  time_t zeit1, zeit2;
  mpz_init(N); mpz_init(N1); mpz_init(N2);

  printf("N="); mpz_inp_str(N,stdin,10);
  printf("k="); scanf("%d",&k); printf("\n");

  printf("N="); mpz_out_str(stdout,10,N); printf(" ");

  time(&zeit1);
  hd_fac(N1,N,k);
  time(&zeit2);
  printf("Zeit: %.0f sec\n",difftime(zeit2,zeit1));
  mpz_divexact(N2,N,N1);
  printf("\n"); mpz_out_str(stdout,10,N); printf(" = ");
  mpz_out_str(stdout,10,N1); printf(" * ");
  mpz_out_str(stdout,10,N2); printf("\n");
}
```

21. algo8_ma

```
# algo8_ma
# Kapitel: Reellquadratische Klassengruppen
# Version: 12.2.2002
#
# Funktionen:
# isdisc(D)
```

```
# rho([a,b,D])
# Red(D)
# HD(D)
# HD_tex(D)
# HD_tex_tabelle(d1,d2)
# hD(D)
# liste_shift(a)
# liste_vergleich(a,b)
# liste_nf(a)
# kbe_red([a,b,D])
# Red_Z(D)
# Red_Z_tex(D)

unprotect(D);

# Test, ob D Diskriminante einer quadratischen Zahl ist
# Eingabe: D
# Ausgabe: true oder false
isdisc:=proc()
  local D;
  D:=args[1];
  if D mod 4=2 or D mod 4=3 or issqr(D) then return false; fi;
  true;
end;

# rho(alpha)=1/(alpha-floor(alpha)) - Kettenbruchnachfolgeabbildung
# Bei haeufiger Anwendung sollte man wD=floor(sqrt(D)) als Parameter mit
# uebergeben.
# Eingabe: [a,b,D]
# Ausgabe: rho([a,b,D])
rho:=proc()
  local a, b, D, wD, u;
  a:=args[1][1]; b:=args[1][2]; D:=args[1][3];
  wD:=isqrt(D); if wD^2>D then wD:=wD-1; fi;
  if a>0 then
    u:=(b+wD)/(2*a);
  else
    u:=(b+wD+1)/(2*a);
  fi;
  u:=floor(u);
  bb:=2*a*u-b;
  aa:=(D-bb^2)/(4*a);
  [aa,bb,D];
end;

# Auflistung der reduzierten reellquadratischen Zahlen
# alpha=(b+sqrt(D))/(2*a) mit Diskriminante D in der Gestalt [a,b,D]
Red:=proc()
  local D, wD, R, b, mac, a;
  D:=args[1];
  if D<=0 or isdisc(D)=false then error "D ist keine Diskriminante"; fi;
  wD:=isqrt(D); if wD^2>D then wD:=wD-1; fi;
  R:={};
  for b from 2-(D mod 2) by 2 to wD do
```

```
    mac:=(D-b^2)/4;
    for a from iquo(wD-b,2)+1 to iquo(wD+b,2) do
      if mac mod a=0 and igcd(a,b,mac/a)=1 then
        R:=R union {[a,b,D]};
      fi;
    od;
  od;
  R;
end;

# Beschreibung der Klassengruppe durch Zykel in Red(D)
# Eingabe: D
# Ausgabe: {[M1,rho(M1),rho^2(M1),...,rho^(-1)(M_1)],[M2,rho(M2),...]}
HD:=proc()
  local D, R, H, zykel, h;
  D:=args[1];
  R:=Red(D); H:={}; h:=0;
  while R<>{} do
    M:=op(1,R); R:=R minus {M}; zykel:=[M];
    do
      M:=rho(M); R:=R minus {M};
      if M=zykel[1] then break; else zykel:=[op(zykel),M]; fi;
    od;
    h:=h+1;
    # printf("%d. Zykel: %a\n",h,zykel);
    H:=H union {zykel};
  od;
  H;
end;

# Zur Ausgabe von H(D) im TeX-Format fuer die Funktion HD_tex_tabelle
# Eingabe: D, Ausgabefilename
HD_tex:=proc()
  local D, H, zykel, M, h, i, j;
  D:=args[1]; aus:=args[2];
  H:=HD(D); h:=nops(H);
  fprintf(aus,"\\hline & $D=%d$, $h(D)=%d$ \\\\ \\hline\n",D,h);
  for i from 1 to h do
    zykel:=op(i,H);
    fprintf(aus,"%d & [",i);
    for j from 1 to nops(zykel) do
      M:=zykel[j];
      fprintf(aus,"$\\frac{%d+\\sqrt{%d}}{%d}$",M[2],D,2*M[1]);
      if j<nops(zykel) then fprintf(aus,",");
        else fprintf(aus,"]\\\\ \\hline\n"); fi;
    od;
  od;
  H;
end;

# Zur Ausgabe einer Liste von Klassengruppen H(D) im TeX-Format fuer das
# Skript im Bereich d1<=D<=d2
# Eingabe: d1, d2, Ausgabefilename
HD_tex_tabelle:=proc()
```

```
  d1:=args[1]; d2:=args[2]; aus:=args[3];
  fprintf(aus,"$$\\begin{tabular}{|r|l|} \\hline\n");
  for d from d1 to d2 do
    if isdisc(d) then
      HD_tex(d,aus):
    fi;
  od;
  fprintf(aus,"\\end{tabular}$$\n");
end;

# Bestimmung der Klassenzahl h(D)
# Eingabe: D
# Ausgabe: h(D)
hD:=proc()
  local D;
  D:=args[1];
  nops(HD(D));
end;

# Die folgenden Listenoperationen dienen dazu in jedem Zykel reduzierter
# Zahlen einen Vertreter auszuwaehlen durch Betrachtung der
# Kettenbruchentwicklung

# Zyklischer Linksshift von Listen
liste_shift:=proc()
  local a;
  a:=args[1];
  [op(subsop(1=NULL,a)),a[1]];
end;

# Lexikgraphischer Vergleich zweier Listen mit Rueckgabewert 1, 0, -1
# bei a<b, a=b, a>b.
liste_vergleich:=proc()
  local a, b, i;
  a:=args[1]; b:=args[2];
  i:=1;
  while i<=nops(a) do
    if a[i]<b[i] then return 1;
    elif a[i]>b[i] then return -1;
    else i:=i+1;
    fi;
  od;
  0;
end;

# Normalform: bei zyklischen Shifts wird das kleinste Element gesucht
liste_nf:=proc()
  local a, b, c, i;
  a:=args[1]; b:=a; c:=a;
  for i from 1 to nops(a)-1 do
    b:=liste_shift(b);
    if liste_vergleich(b,c)=1 then c:=b; fi;
  od;
  c;
```

```
end;

# Kettenbruchentwicklung reduzierter Zahlen [a,b,D] = (b+sqrt(D))/(2a)
# Eingabe: [a,b,D]
# Ausgabe: [u_0,u_1,....,u_(k-1)] - Kettenbruchentwicklung
kbe_red:=proc() # Eingabe [a,b,D]
  local a, b, D, wD, u, aa, bb, kbe;
  a:=args[1][1]; b:=args[1][2]; D:=args[1][3];
  wD:=isqrt(D); if wD^2>D then wD:=wD-1; fi;
  aa:=a; bb:=b; kbe:=[];
  do
    u:=iquo(bb+wD,2*aa); kbe:=[op(kbe),u];
    bb:=2*aa*u-bb;
    aa:=(D-bb^2)/(4*aa);
    if aa=a and bb=b then break; fi;
  od;
  kbe;
end;

# Aus jedem Zykel in Red(D) wird ein Element ausgewaehlt und die
# Kettenbruchentwicklung gebildet.
Red_Z:=proc()
  local D, H;
  D:=args[1];
  H:=Red(D);
  H:=map(x->kbe_red(x),H);
  H:=map(x->liste_nf(x),H);
end;

# Zykel im TeX-Format mit Kettenbruchentwicklung
Red_Z_tex:=proc()
  local D, wD, b, mac, a, R, H, Hkb, Z, h, i, j, aus;
  D:=args[1]; aus:=args[2];
  if D<=0 or isdisc(D)=false then error "D ist keine Diskriminante"; fi;
  wD:=isqrt(D); if wD^2>D then wD:=wD-1; fi;
  R:={}; H:=[]; Hkb:=[]; h:=0;
  for b from 2-(D mod 2) by 2 to wD do
    mac:=(D-b^2)/4;
    for a from iquo(wD-b,2)+1 to iquo(wD+b,2) do
      if mac mod a=0 and igcd(a,b,mac/a)=1 then
        alpha:=[a,b,D];
        if member(alpha,R)=false then
          h:=h+1; alpha_h:=alpha; R:=R union {alpha};
          Z:=[alpha];
          Hkb:=[op(Hkb),kbe_red(alpha)];
          alpha:=rho(alpha_h);
          while alpha<>alpha_h do
            R:=R union {alpha}; Z:=[op(Z),alpha];
            alpha:=rho(alpha);
          od;
          H:=[op(H),Z];
        fi;
      fi;
    od;
```

```
  od;
  fprintf(aus,"$$\\begin{tabular}{|r|l|l|}\\hline\n");
  fprintf(aus,"$i$ & $Z(\\alpha_i)$ ($D=%d$, $h(D)=%d$) & ",D,h);
  fprintf(aus,"Kettenbruchentwicklung von $\\alpha_i$ \\\\ \\hline\n");
  for i from 1 to h do
    Z:=H[i]; kbe:=Hkb[i];
    fprintf(aus,"%d & $\\{",i);
    for j from 1 to nops(Z) do
      M:=Z[j];
      fprintf(aus,"\\frac{%d+\\sqrt{%d}}{%d}",M[2],D,2*M[1]);
      if j<nops(Z) then fprintf(aus,", ");
      else fprintf(aus,"\\}$ & ");
      fi;
    od;
    fprintf(aus,"$\\overline{%a}$ \\\\ \\hline\n",kbe);
  od;
  fprintf(aus,"\\end{tabular}$$\n");
end;
```

Literaturverzeichnis

[BorewiczSafarevic] S. I. Borewicz, I. R. Safarevic, Zahlentheorie, Birkhäuser 1966.

[Buchmann] J. Buchmann, Einführung in die Kryptographie, Springer 1999.

[BuchmannHamdy] J. Buchmann, S. Hamdy, A survey on IQ cryptography, in K. Alster, J. Urbanowicz, H. C. Williams (Eds.), Public-Key Cryptography and Computational Number Theory, de Gruyter 2001.

[CanfieldErdösPomerance] E. Canfield, P. Erdös, C. Pomerance, On a Problem of Oppenheim concerning 'Factorisatio Numerorum', J. Number Theory **17** (1983), 1-28.

[Cohen] H. Cohen, A Course in Computational Algebraic Number Theory, Springer 1993.

[Forster] O. Forster, Algorithmische Zahlentheorie, Vieweg 1996.

[HardyWright] G. H. Hardy, E. M. Wright, An Introduction to the Theory of Numbers, Oxford University Press, Fifth Edition, Clarendon Press, Oxford 1979.

[Hua] L. K. Hua, Introduction to Number Theory, Springer-Verlag 1982.

[Knuth] D. E. Knuth, The Art of Computer Programming, Volume 2, Seminumerical Algorithms, Second Edition, Addison-Wesley 1981.

[Koblitz] N. Koblitz, A Course in Number Theory and Cryptography, Second Edition, Springer 1994.

[MenezesOorschotVanstone] A. Menezes, P. van Oorschot, S. Vanstone, Handbook of Applied Cryptography, CRC Press, 1996.

[Riesel] H. Riesel, Prime Numbers and Computer Methods for Factorization, Birkhäuser 1985.

[ScharlauOpolka] W. Scharlau, H. Opolka, Von Fermat bis Minkowski, Springer-Verlag 1980.

[Schmitz] Th. Schmitz, Abschätzung der Lösung der Pellschen Gleichung, Archiv der Mathematik und Physik, 3. Reihe, **24** (1916), 87-89.

[Schneier] B. Schneier, Angewandte Kryptographie, Addison-Wesley 1996.

[Wiener] M. J. Wiener, Cryptanalysis of short RSA secret exponents, IEEE Transactions on Information Theory **36** (1990), 553-558.

Version vom 13.2.2001