Die Theorie des Number Field Sieve

Diplomarbeit

Jörg Zayer

Mai 1991

Angefertigt nach einem Thema von Prof. Dr. Johannes Buchmann
am Fachbereich 14, Informatik, Universität des Saarlandes, Saarbrücken

Hiermit versichere ich an Eides Statt, daß ich diese Arbeit selbständig verfaßt und keine anderen als die angegebenen Quellen benutzt habe.

Saarbrücken, 30.Mai 1991

Für das interessante Thema möchte ich Herrn Prof. Dr. Johannes Buchmann danken, ebenso Herrn Prof. Dr. Attila Pethö für seine Diskussionsbereitschaft und Anregungen.

Inhaltsverzeichnis

Kapitel 1

Einleitung

Die Faktorisierung großer natürlicher Zahlen beschäftigt die Mathematiker schon seit Jahrhunderten. Doch erst mit der Entwicklung von Kryptosystemen und der damit verbundenen Frage der Sicherheit erhielt die Zerlegung dieser Zahlen auch einen praktischen Wert.

Die vor allem in den 80er Jahren benutzten Verfahren, wie z.B. das quadratische Sieb oder die elliptische Kurvenmethode, ermöglichten die Faktorisierung bis zu hundert Stellen großer, natürlicher Zahlen.

Der hier vorgestellte Faktorisierungsalgorithmus 'Number Field Sieve' geht auf eine Arbeit von A.K.Lenstra, H.W.Lenstra Jr., M.S.Manasse und J.M.Pollard zurück. Er greift die Sicherheit von Kryptosystemen in einem noch größeren Ausmaß an, als dies die älteren Algorithmen schon taten, denn die Laufzeit des 'Number Field Sieve' für die Klasse der Zahlen, auf die das Verfahren anwendbar ist, verhält sich assymptotisch besser als die aller bisher bekannten Faktorisierungsalgorithmen. So gelang es 1990 mit dem 'Number Field Sieve' , die Zerlegung der 155-stelligen Zahl $F_9 = 2^{(2^9)} + 1$ zu bestimmen.

Die hier vorliegende Arbeit stellt meinen Versuch dar, dieses Verfahren von der mathematischen Seite her zu durchleuchten.
Die zu zerlegende natürliche Zahl n muß dabei im Bildbereich einer über $\mathbb{Q}$ irreduziblen Funktion $f : \mathbb{Z} \leftarrow \mathbb{Z}$

$$f(x) = x^d + c$$

mit $d \in \mathbb{N}$ und $c \in \mathbb{Z}$, $c \neq 0$, $|c|$ klein, liegen.

Meine Ausführungen beginne ich mit einer Sammlung mathematischer Definitionen und Sätze, deren Verbindung zum 'Number Field Sieve' sich in den anschließenden Kapiteln zeigen wird. Einen Großteil dieses mathematischen Einführungsabschnitts bilden dabei Erkenntnisse aus der algebraischen Zahlentheorie, was darauf zurückzuführen ist, daß dieses Verfahren einen Weg beschreitet, der über den algebraischen Zahlkörper $\mathbb{Q}[\rho]$ führt, wobei ρ eine Wurzel des oben angegebenen Polynoms $f(x)$ ist. Kapitel 3 beinhaltet die Darstellung des Grundgerüsts des 'Number Field Sieve' .

Alternativen und Ergänzungen zu den dort beschriebenen Prozeduren schließen sich daran an. Im Kapitel 6 analysiere ich die erwartete Laufzeit des Verfahrens.
Die Menge der Zahlen, die mit Hilfe der Grundversion des 'Number Field Sieve' faktorisiert werden können, ist durch die vorgegebene Form $x^d + c$ des Polynoms $f(x)$ und durch später genauer beschriebene Vorgaben an den benutzten Zahlkörper stark eingeschränkt. Inwieweit sich das Verfahren verallgemeinern läßt, zeige ich deshalb im vorletzten Teil meiner Arbeit auf. Den Abschluß bildet schließlich ein Ausblick in die Praxis des 'Number Field Sieve' .

In meinen Ausführungen gehe ich davon aus, daß dem Leser die algorithmischen Lösungen folgender Probleme bekannt sind :

- Lösen linearer Gleichungssysteme
  - über $\mathbb{R}$
  - über $\mathbb{Z}$
  - über $\mathbb{Z}/2\mathbb{Z}$
- Lösen linearer diophantischer Gleichungen
- Berechnung des größten gemeinsamen Teilers
  - zweier ganzer Zahlen
  - zweier Polynome mit Koeffizienten in einem Primkörper

Kapitel 2

Mathematische Grundlagen

In diesem Kapitel fasse ich die zur Beschreibung des 'Number Field Sieve' notwendigen mathematischen Grundlagen zusammen. Als Vorlagen dienten dazu die Arbeiten von Stewart & Tall ([13]) und Hecke ([3]).
Aussagen, die anderen Quellen entstammen, sind gesondert gekennzeichnet.

2.1 Polynome

1. **Bezeichnung**

   $\mathbb{N}$:= Menge der natürlichen Zahlen
   $\mathbb{Z}$:= Menge der ganzen Zahlen
   $\mathbb{P}$:= Menge der Primzahlen
   $\mathbb{Q}$:= Menge der rationalen Zahlen
   $\mathbb{R}$:= Menge der reellen Zahlen
   $\mathbb{C}$:= Menge der komplexen Zahlen
   $\mathbb{Z}/m\mathbb{Z}$:= Restklassenring modulo m

2. **Definition 1** [1]

   (a) Sei $\mathbb{K}$ ein Körper, x eine Unbestimmte über $\mathbb{K}$, so nennt man

   $$\mathbb{K}[x] := \{p_0 + p_1 \cdot x + \ldots + p_r \cdot x^r \mid r \in \mathbb{N},\, p_i \in \mathbb{K}\}$$

   den **Polynomring** über $\mathbb{K}$.

   (b) Ist $f(x) = f_0 + f_1 \cdot x + \ldots + f_m \cdot x^m \in \mathbb{K}[x]$, $f_m \neq 0$, so nennt man m den **Grad** von f.

   (c) Ein Element $x_0 \in \mathbb{K}$ mit $f(x_0) = 0$ heißt **Wurzel** von f in $\mathbb{K}$.

[1] [14], III.14 und V.28

3. **Satz 1**

   Sei $\mathbb{K}$ ein Körper, x eine Unbestimmte über $\mathbb{K}$, $f(x) \in \mathbb{K}[x]$ und x_0 eine Wurzel von f in $\mathbb{K}$,
   dann gibt es ein Polynom $g(x) \in \mathbb{K}[x]$ mit $grad(g) = grad(f) - 1$ und $f(x) = (x - x_0) \cdot g(x)$.

4. **Satz 2** [2]

   Sei $d \in \mathbb{N}$, $d \geq 2$ und $a \in \mathbb{K}\backslash\{0\}$.
   Das Polynom $f(x) = x^d - a$ ist genau dann irreduzibel über $\mathbb{K}$, wenn gilt

   (a) für alle $p \in \mathbb{P}$ mit $p|d$ gilt: $y^p \neq a, \quad \forall y \in \mathbb{K}$, und,

   (b) wenn $4|d$, dann $-4y^4 \neq a$, $\forall y \in \mathbb{K}$.

5. **Satz 3** (Eisensteinscher Satz) [3]

   Sei $f(x) = f_0 + f_1 \cdot x + \ldots + f_d \cdot x^d \in \mathbb{Z}[x]$, $f_d \neq 0$.
   Wenn es eine Primzahl $p \in \mathbb{P}$ gibt, so daß

   (a) $f_d \not\equiv 0 \bmod p$,

   (b) $f_i \equiv 0 \bmod p, \quad \forall i < d$, und

   (c) $f_0 \not\equiv 0 \bmod p^2$,

   dann ist $f(x)$ irreduzibel über $\mathbb{Q}$.

[2] [6], VIII.9, Theorem 16
[3] [14], V.31

2.2 Zahlkörper und algebraische Zahlen

1. **Definition 2**

   Sei $\mathbb{K}$ eine mindestens zweielementige Teilmenge der komplexen Zahlen.

   $$\text{Wenn gilt } \alpha, \beta \in \mathbb{K} \Rightarrow \begin{array}{ll} \text{a.)} & (\alpha + \beta) \in \mathbb{K} \\ \text{b.)} & (\alpha - \beta) \in \mathbb{K} \\ \text{c.)} & (\alpha \cdot \beta) \in \mathbb{K} \\ \text{d.)} & (\frac{\alpha}{\beta}) \in \mathbb{K}, \end{array}$$

   dann nennt man $\mathbb{K}$ einen **Zahlkörper (Number Field)**.

2. **Beispiel**

   $\mathbb{Q}$, die Menge der rationalen Zahlen, ist ein Zahlkörper.

3. **Definition 3**

   (a) Eine Zahl $\rho \in \mathbb{C}$, die die Wurzel eines Polynoms

   $$f(x) = f_0 + f_1 \cdot x + \ldots + f_{d-1} \cdot x^{d-1} + x^d$$

   mit $f_i \in \mathbb{Q}$ ist, heißt **algebraische Zahl**.

   (b) Ist $f(x)$ irreduzibel über $\mathbb{Q}$, dann heißt d der **Grad** der algebraischen Zahl $\rho =: \rho^{(1)}$, und die weiteren Wurzeln $\rho^{(2)}, \rho^{(3)}, \ldots, \rho^{(d)}$ von $f(x)$ heißen die **Konjugierten** von ρ.

4. **Beispiel**

   (a) Jede Zahl $q \in \mathbb{Q}$ ist eine algebraische Zahl vom Grad 1, denn q ist die Wurzel des über $\mathbb{Q}$ irreduziblen Polynoms $f(x) = x - q$.

   (b) i ist eine algebraische Zahl vom Grad 2, denn i ist die Wurzel des Polynoms $f(x) = x^2 + 1$. Die Konjugierte von i ist $-i$.

   (c) $\sqrt[3]{-2}$ ist eine algebraische Zahl vom Grad 3, denn $\sqrt[3]{-2}$ ist eine Wurzel des über $\mathbb{Q}$ irreduziblen Polynoms $f(x) = x^3 + 2$. Ihre Konjugierten sind
   $\rho^{(2)} = \sqrt[3]{-2e^{2\pi i}}$ und $\rho^{(3)} = \sqrt[3]{-2e^{4\pi i}}$.

5. **Satz 4**

   Sind α, β algebraische Zahlen, so sind auch $\alpha + \beta$, $\alpha - \beta$, $\alpha \cdot \beta$
   und, wenn $\beta \neq 0$, $\frac{\alpha}{\beta}$ algebraische Zahlen.

6. **Folgerung 1**

   Sei ρ eine algebraische Zahl vom Grad d. Dann bildet die Menge
   $\mathbb{Q}[\rho]:=\{\alpha \mid \alpha = a_0 + a_1 \cdot \rho + \ldots + a_{d-1} \cdot \rho^{d-1},\ a_i \in \mathbb{Q},\ i = 0, \ldots, d-1\ \}$
   einen Zahlkörper.

7. **Beispiel**

   $\mathbb{Q}[i]=\{a + b \cdot i \mid a, b \in \mathbb{Q}\}$ ist ein Zahlkörper.

8. **Definition 4**

   Sei ρ eine algebraische Zahl vom Grad d.
   Eine Menge von d Zahlen $\omega_1, \omega_2, \ldots, \omega_d$, *alle* $\omega_i \in \mathbb{Q}[\rho]$, heißt **Basis** von $\mathbb{Q}[\rho]$, wenn sich jede Zahl $\alpha \in \mathbb{Q}[\rho]$ in der Form $\alpha = \sum_{i=1}^{d} q_i \cdot \omega_i$ mit Koeffizienten $q_i \in \mathbb{Q}$ darstellen läßt.

9. **Bemerkung**

   Ist ρ eine algebraische Zahl vom Grad d, so ist $\mathbb{Q}[\rho]$ ein d-dimensionaler Vektorraum über $\mathbb{Q}$.

10. **Lemma 1**

    Sei ρ eine algebraische Zahl vom Grad d.
    Dann bildet $\{1, \rho, \rho^2, \ldots, \rho^{d-1}\}$ eine Basis von $\mathbb{Q}[\rho]$.

11. **Definition 5**

    Ist $f(x) = f_0 + f_1 \cdot x + \ldots + f_{d-1} \cdot x^{d-1} + x^d$, mit $f_i \in \mathbb{Z}$, irreduzibel über $\mathbb{Q}$ und ist ρ eine Wurzel von $f(x)$, dann heißt ρ eine **ganze algebraische Zahl** vom Grad d.

12. **Satz 5**

    Sind α, β ganze algebraische Zahlen, so sind auch $\alpha + \beta$, $\alpha - \beta$ und $\alpha \cdot \beta$ ganze algebraische Zahlen.

13. **Folgerung 2**

    Sei ρ eine ganze algebraische Zahl vom Grad d.
    Dann bildet sowohl die Menge G der ganzen algebraischen Zahlen in $\mathbb{Q}[\rho]$ als auch die Menge $\mathbb{Z}[\rho]$,
    $\mathbb{Z}[\rho]:=\{\alpha \mid \alpha = a_0 + a_1 \cdot \rho + \ldots + a_{d-1} \cdot \rho^{d-1}, a_i \in \mathbb{Z}, i = 1, \ldots, d-1\ \} \subseteq G \subseteq \mathbb{Q}[\rho]$,
    einen Integritätsbereich, d.h. einen kommutativen, nullteilerfreien Ring mit 1.

14. **Definition 6**

    Sei ρ eine algebraische Zahl vom Grad d.
    Eine Menge von d Zahlen $\omega_1, \omega_2, \ldots, \omega_d$, *alle* $\omega_i \in G$, heißt **Ganzheitsbasis** von $\mathbb{Q}[\rho]$, wenn sich jede Zahl $\alpha \in G$ eindeutig in der Form $\alpha = \sum_{i=1}^{d} z_i \cdot \omega_i$ mit Koeffizienten $z_i \in \mathbb{Z}$ darstellen läßt.

2.3 Teilbarkeit und Norm

1. **Voraussetzung**

   Im weiteren sollen folgende Voraussetzungen gelten :

   - $\rho = \rho^{(1)}$ ist eine ganze algebraische Zahl vom Grad d.
   - $\rho^{(2)}, \rho^{(3)}, \ldots, \rho^{(d)}$ seien die Konjugierten von ρ.
   - $\mathbb{K} := \mathbb{Q}[\rho]$ ist der von ρ erzeugte Zahlkörper.

2. **Definition 7**

   Seien α, β ganze algebraische Zahlen, $\beta \neq 0$.
   α heißt **teilbar** durch β, in Zeichen: $\beta | \alpha$, wenn es eine ganze algebraische Zahl γ gibt mit :

   $$\alpha = \beta \cdot \gamma.$$

3. **Definition 8**

   Eine ganze algebraische Zahl ϵ heißt **Einheit**, wenn es eine ganze algebraische Zahl ϵ_1 gibt mit :

   $$\epsilon \cdot \epsilon_1 = 1, \quad \text{d.h.} \quad \epsilon \text{ Einheit} \Leftrightarrow \epsilon | 1.$$

4. **Definition 9**

   Seien α, β, γ ganze algebraische Zahlen, $\alpha, \beta \neq 0$.

   (a) α heißt **assoziiert** zu β, in Zeichen: $\alpha \sim \beta$, wenn eine Einheit ϵ existiert mit

   $$\alpha = \beta \cdot \epsilon.$$

   (b) $\alpha \not\sim 1$ heißt **irreduzibel**, wenn gilt

   $$\alpha = \beta \cdot \gamma \Rightarrow \beta \sim 1 \text{ oder } \gamma \sim 1.$$

   (c) $\alpha \not\sim 1$ heißt **Primelement**, wenn gilt

   $$\alpha | (\beta \cdot \gamma) \Rightarrow \alpha | \beta \text{ oder } \alpha | \gamma.$$

5. **Satz 6**

   Sei ρ eine algebraische Zahl vom Grad d, $\rho^{(2)}, \ldots, \rho^{(d)}$ ihre Konjugierten und
   $$\alpha = a_0 + a_1 \cdot \rho + \ldots + a_{d-1} \cdot \rho^{d-1} \in \mathbb{Q}[\rho].$$
   Dann sind $\alpha^{(i)} = a_0 + a_1 \cdot \rho^{(i)} + \ldots + a_{d-1} \cdot {\rho^{(i)}}^{d-1}$, $i = 2, \ldots, d$, die Konjugierten von α.

6. **Definition 10**

   Für eine algebraische Zahl α heißt die Zahl
   $$N(\alpha) := \alpha^{(1)} \cdot \ldots \cdot \alpha^{(d)}$$
   die **Norm** von α.

7. **Satz 7**

   Seien α, β algebraische Zahlen, dann gilt :

   (a) $N(\alpha) \in \mathbb{Q}$.

   (b) Ist α eine ganze algebraische Zahl, dann ist $N(\alpha) \in \mathbb{Z}$.

   (c) Die Norm ist multiplikativ, d.h. $N(\alpha \cdot \beta) = N(\alpha) \cdot N(\beta)$.

8. **Lemma 2**

   Eine ganze algebraische Zahl ϵ ist genau dann eine Einheit, wenn $N(\epsilon) = \pm 1$.

9. **Satz 8**

   Die Menge E der Einheiten in dem Ring G der ganzen algebraischen Zahlen von $\mathbb{Q}[\rho]$ bildet bezüglich der Multiplikation eine abelsche Gruppe, die **Einheitengruppe**.

10. **Satz 9** [4]

    Ist eine der Konjugierten von ρ reell, so existieren in $\mathbb{K}$ nur die trivialen Einheitswurzeln -1 und 1, d.h. -1 und 1 sind die einzigen Zahlen ϵ aus $\mathbb{Q}[\rho]$, für die es einen Exponenten $k > 0 \in \mathbb{N}$ gibt, mit
    $$\epsilon^k = 1.$$

[4] [11], V.4, Lemma 4.1

11. **Satz 10**

    Unter den d Konjugierten von ρ seien r_1 reell und $2 \cdot r_2$ nicht reell, dann gibt es $l = r_1 + r_2$ Einheiten $\xi, \epsilon_1, \ldots, \epsilon_{l-1}$, wobei ξ eine w-te Einheitswurzel ist, derart, daß man in der Form

    $$\epsilon = \xi^{\,e} \cdot {\epsilon_1}^{e_1} \cdot \ldots \cdot {\epsilon_{l-1}}^{e_{l-1}}$$

    jede Einheit des Körpers genau einmal erhält, wenn $e_1, \ldots, e_{l-1}$ sämtliche Zahlen aus $\mathbb{Z}$ und e sämtliche Zahlen aus $\{0, \ldots, w-1\}$ durchlaufen.

2.4 Idealtheorie

1. **Definition 11**

   Eine Teilmenge I der ganzen algebraischen Zahlen heißt **Ideal** in $\mathbb{K}$, wenn für alle $\alpha, \beta \in I$ und alle ganzen algebraischen Zahlen λ, μ gilt :

   $$\lambda \cdot \alpha + \mu \cdot \beta \in I.$$

2. **Bezeichnung**

   Seien $\alpha_1, \ldots, \alpha_k$ ganze algebraische Zahlen. Dann nennt man

   $$I = \{\alpha \mid \alpha = \lambda_1 \cdot \alpha_1 + \ldots + \lambda_k \cdot \alpha_k, \text{ alle } \lambda_i \text{ ganze algebraische Zahlen } \}$$

   das von $\alpha_1, \ldots, \alpha_k$ erzeugte Ideal, und man schreibt: I=$(\alpha_1, \alpha_2, \ldots, \alpha_k)$.

3. **Definition 12**

   Ein Ideal I heißt **Hauptideal**, wenn es ein $\alpha \in I$ gibt mit $I = (\alpha)$.

4. **Satz 11**

   Sei I ein Ideal in $\mathbb{K}$, I$\neq (0)$.
   Dann existiert eine Menge von d Zahlen $\alpha_1, \ldots, \alpha_d$, so daß jede Zahl

   $$\alpha = z_1 \cdot \alpha_1 + \ldots + z_d \cdot \alpha_d \in I$$

   genau einmal dargestellt wird, wenn die Zahlen z_i, $i = 1, \ldots, d$, alle Zahlen aus $\mathbb{Z}$ durchlaufen.

5. **Definition 13**

   Die in Satz 11 beschriebene Menge von d Zahlen heißt **Basis** des Ideals.

6. **Definition 14**

   Sei G der Ring der ganzen algebraischen Zahlen in $\mathbb{K}$ und I ein Ideal in $\mathbb{K}$.
   Ist $\{\omega_1, \omega_2, \ldots, \omega_d\}$ eine Ganzheitsbasis von $\mathbb{K}$ und $\{\alpha_1, \alpha_2, \ldots, \alpha_d\}$ eine Basis von I mit

   $$\alpha_i = \sum_{j=1}^{d} a_{i,j} \cdot \omega_j, \quad a_{i,j} \in \mathbb{Z},$$

   dann heißt der Absolutbetrag der Determinante der Matrix $(a_{i,j})$, $i, j = 1, \ldots, d$, **Norm** von I :

   $$N(I) := |det(a_{i,j})|.$$

7. **Lemma 3**

   Wenn $I = (\alpha)$ ein Hauptideal ist, dann gilt
   $$N(I) = |N(\alpha)|.$$

8. **Folgerung 3**

   Ist $I = (p)$, $p \in \mathbb{P}$, so gilt $N(I) = p^d$.

9. **Definition 15**

   (a) Unter dem **Produkt** zweier Ideale $\mathcal{A} = (\alpha_1, \alpha_2, \ldots, \alpha_k)$ und $\mathcal{B} = (\beta_1, \beta_2, \ldots, \beta_r)$ versteht man das Ideal
   $$\mathcal{A} \cdot \mathcal{B} := (\alpha_1 \cdot \beta_1, \ldots, \alpha_1 \cdot \beta_r, \alpha_2 \cdot \beta_1, \ldots, \alpha_k \cdot \beta_r).$$

   (b) Ein Ideal $\mathcal{A}$ heißt **teilbar** durch ein Ideal $\mathcal{B}$, wenn es ein Ideal $\mathcal{C}$ gibt mit
   $$\mathcal{A} = \mathcal{B} \cdot \mathcal{C}.$$

   (c) Ein Ideal $\mathcal{P}$ heißt **Primideal**, wenn gilt

   i. $\mathcal{P} \neq (1)$ und

   ii. $\mathcal{A} | \mathcal{P} \Rightarrow \mathcal{A} = (1)$ oder $\mathcal{A} = \mathcal{P}$.

10. **Lemma 4**

    Sei $\mathcal{A}$ ein Ideal in $\mathbb{K}$.

    (a) Wenn die Norm $N(\mathcal{A})$ eine Primzahl ist, so ist $\mathcal{A}$ ein Primideal.

    (b) $N(\mathcal{A}) \in \mathcal{A}$.

    (c) Wenn $\mathcal{A}$ ein Primideal ist, so gilt

    i. $p \in \mathcal{A}$ für genau eine Primzahl $p \in \mathbb{P}$

    ii. $N(\mathcal{A}) = p^m$, für ein $m \leq d$.

11. **Satz 12**

    Seien α, β ganze algebraische Zahlen, dann gilt :
    $$\beta | \alpha \Leftrightarrow (\beta) | (\alpha).$$

12. **Lemma 5**

    Seien α, β ganze algebraische Zahlen, dann gilt :
    $(\alpha) = (\beta) \Leftrightarrow \alpha = \beta \cdot \epsilon$, ϵ ist eine Einheit.

13. **Satz 13**

    Jedes von (0) und (1) verschiedene Ideal läßt sich, bis auf die Reihenfolge der Faktoren, auf eindeutige Weise als Produkt von Primidealen darstellen.

14. **Definition 16**

    Ein Integritätsbereich R heißt ZPE-Ring, wenn jedes Element $\alpha \in$R, α keine Einheit, sich eindeutig als Produkt von Primelementen darstellen läßt.

15. **Satz 14**

    Ist in einem Integritätsbereich R jedes Ideal ein Hauptideal, d.h. R ist ein Hauptidealring, so ist R ein ZPE-Ring.

16. **Satz 15**

    Sei G der Ring der ganzen Zahlen in $\mathbb{K}$, dann ist die Zerlegung von Elementen aus G in Primelemente genau dann eindeutig, wenn jedes Ideal in G ein Hauptideal ist.

17. **Satz 16**

    Sei $f(x) = f_0 + f_1 \cdot x + \ldots + f_{d-1} \cdot x^{d-1} + x^d$ mit $f_i \in \mathbb{Z}$, irreduzibel über $\mathbb{Q}$, ρ eine Wurzel von $f(x)$, $\mathbb{K} = \mathbb{Q}[\rho]$ der zugehörige Zahlkörper vom Grad d, $p \in \mathbb{P}$ und G=$\mathbb{Z}[\rho]$ der Ring der ganzen Zahlen in $\mathbb{K}$.

    Zerfällt $f(x)$ über $\mathbb{Z}/p\mathbb{Z}$ in

    $$\overline{f}_1(x)^{e_1} \cdot \ldots \cdot \overline{f}_r(x)^{e_r},$$

    wobei $\overline{f}_i(x)^{e_i} \in \mathbb{Z}/p\mathbb{Z}[x]$ irreduzibel über $\mathbb{Z}/p\mathbb{Z}$ ist, dann zerfällt das Hauptideal (p) in

    $$(p) = \mathcal{P}_1^{e_1} \cdot \ldots \cdot \mathcal{P}_r^{e_r}$$

    wobei $\mathcal{P}_i$ das von $(p, g_i(\rho))$ erzeugte Primideal ist mit $g_i(x) \in \mathbb{Z}[x]$ so, daß $g_i(x) \bmod p = \overline{f}_i(x)$ und $N(\mathcal{P}_i) = p^{d_i}$, mit $d_i = grad(\overline{f}_i)$.

2.5 Gitter und Logarithmischer Raum

1. **Definition 17**

   Seien $v_1, \ldots, v_m$ linear unabhängige Vektoren aus $\mathbb{R}^n$.
   Die von den Vektoren $v_1, \ldots, v_m$ erzeugte, additive Untergruppe L von $(\mathbb{R}^n, +)$ heißt **Gitter der Dimension** m :

   $L := (v_1, \ldots, v_m).$

2. **Definition 18**

   Sei $f(x) \in \mathbb{Z}[x]$ über $\mathbb{Q}$ irreduzibel und seien

   $$\rho^{(1)}, \ldots, \rho^{(s)}, \rho^{(s+1)}, \ldots, \rho^{(s+t)}, \rho^{(s+t+1)}, \ldots, \rho^{(s+2t)} = \rho^{(d)}$$

   die Wurzeln von $f(x)$, wobei $\rho^{(1)}, \ldots, \rho^{(s)} \in \mathbb{R}$, $\rho^{(s+1)}, \ldots, \rho^{(s+2t)} \in \mathbb{C} \setminus \mathbb{R}$ und $\rho^{(s+t+i)}$ konjugiert komplex zu $\rho^{(s+i)}$, $i = 1, \ldots, t$.

   Seien σ_j die Monomorphismen $\mathbb{Q}[\rho] \to \mathbb{C}$, $j = 1, \ldots, d$,
   $\rho \mapsto \rho^{(j)}$

   dann definiert man

   (a) $L^{ST} := \{(x_1, \ldots, x_{s+t}) \mid x_i = \sigma_i(\alpha),\ \alpha \in \mathbb{Q}[\rho]\} \subseteq \mathbb{R}^s \times \mathbb{C}^t$.

   (b) $N(x) := |x_1| \cdot \ldots \cdot |x_s| \cdot |x_{s+1}|^2 \cdot \ldots \cdot |x_{s+t}|^2$.

   (c) $\sigma : \mathbb{Q}[\rho] \to L^{ST}$
   $\sigma(\alpha) := (\sigma_1(\alpha), \ldots, \sigma_{s+t}(\alpha))$.

   (d) $\lambda : L^{ST} \to \mathbb{R}^{s+t}$
   $x \mapsto \lambda(x) = (\lambda_1(x), \ldots, \lambda_{s+t}(x))$ mit

   $$\lambda_j(x) := \begin{cases} \log(|x_j|) & : \quad j = 1, \ldots, s \\ \log(|x_j|^2) & : \quad j = s+1, \ldots, s+t. \end{cases}$$

3. **Lemma 6**

   Für $x, y \in L^{ST}$ gilt :

   $\lambda(xy) = \lambda(x) + \lambda(y).$

4. **Definition 19**

   Definiert man für $\alpha \in \mathbb{Q}[\rho]$

   $\mu(\alpha) := \lambda(\sigma(\alpha)),$

   so heißt die Abbildung $\mu : \mathbb{Q}[\rho] \to \mathbb{R}^{s+t}$ **logarithmische Representation** von $\mathbb{Q}[\rho]$, und $\mathbb{R}^{s+t}$ heißt **logarithmischer Raum**.

5. **Satz 17**

   Für $\alpha, \beta \in \mathbb{Q}[\rho]$ gilt :
   $$\mu(\alpha \cdot \beta) = \mu(\alpha) + \mu(\beta).$$

6. **Satz 18**

   Definiert man $\mu_i(\alpha) := \lambda_i(\sigma(\alpha))$, so gilt
   $$\sum_{j=1}^{s+t} \mu_j(\alpha) = \log |N(\alpha)|.$$

7. **Satz 19**

   Sei E die Einheitengruppe von G, dem Ring der ganzen algebraischen Zahlen. Dann bildet $\mu(\mathrm{E})$ ein Gitter der Dimension $s+t-1$ in $\mathbb{R}^{s+t}$.

Kapitel 3

Der Algorithmus

3.1 Die Grundidee

Sei n eine zusammengesetzte natürliche Zahl und $B = \{b_1, \ldots, b_k\} \subseteq \mathbb{Z}$. Wenn man genügend viele Kongruenzen der Form

$$(3.1.1) \qquad \prod_{i=1}^{k} b_i^{\lambda_{ij}} \equiv \prod_{i=1}^{k} b_i^{\mu_{ij}} \mod n\ , \quad j = 1, 2, \ldots$$

findet, so kann man daraus eine Kongruenz der Form

$$(3.1.2) \qquad X^2 \equiv Y^2 \mod n$$

gewinnen. Ist dabei $X \not\equiv \pm Y \mod n$, so ist der $ggT(X+Y, n)$ ein echter Teiler von n.

Diese Idee, Faktoren von n mit Hilfe einer Faktorbasis B zu finden, ist Bestandteil mehrerer Faktorisierungsalgorithmen. Der Unterschied zwischen den einzelnen Verfahren besteht darin, wie man die Kongruenzen der Form (3.1.1) findet. Das 'Number Field Sieve' löst dieses Problem auf folgende Weise :

Sei $f(x) = x^d + c$ ein über $\mathbb{Q}$ irreduzibles Polynom aus $\mathbb{Z}[x]$, $c \neq 0$, $|c|$ klein, und sei $m \in \mathbb{N}$ so, daß $f(m) = m^d + c = n$, dann ist nach Definition 5 jede Wurzel ρ von $f(x)$ eine ganze algebraische Zahl und nach Folgerung 1 $\mathbb{Q}[\rho]$ ein Zahlkörper.

Sei weiter $G = \mathbb{Z}[\rho]$ ein Hauptidealring und sei

$$\begin{aligned} \Phi : \mathbb{Z}[\rho] &\longrightarrow \mathbb{Z}/n\mathbb{Z} \\ \rho &\longmapsto m \end{aligned}$$

dann ist Φ ein Ringhomomorphismus, was man leicht erkennt, wenn man bedenkt, daß auf der einen Seite, wegen $f(\rho) = 0$, die Reduktionsformel

$$\rho^d = -c,$$

und auf der anderen Seite, wegen $m^d + c = n$, die Reduktionsformel

$$m^d \equiv -c \mod n$$

gilt.

Für eine natürliche Zahl L definiert man folgende Mengen :

- $B_1 := \{p \in \mathbb{P} \mid p \leq L\}$
- $B_2 := \{\pi \in \mathbb{Z}[\rho] \mid \pi \text{ Primelement, } N(\pi) \in B_1\}$
- $U := \{\epsilon \in \mathbb{Z}[\rho] \mid \epsilon \text{ ist Einheit in } \mathbb{Z}[\rho]\}$

Die Idee des 'Number Field Sieve' besteht darin, teilerfremde Zahlen $a, b \in \mathbb{Z}$ zu finden, so daß man auf der einen Seite

$a + bm$ über der Faktorbasis B_1

und auf der anderen Seite

$a + b\rho$ über der Faktorbasis B_2 und der Einheitenmenge U

zerlegen kann. Also:

$$(3.1.3) \qquad a + bm = \prod_{p \in B_1} p^{\nu_p} \quad \text{und}$$

$$(3.1.4) \qquad a + b\rho = \left(\prod_{\pi \in B_2} \pi^{\mu_\pi}\right) \cdot \left(\prod_{\epsilon \in U} \epsilon^{\lambda_\epsilon}\right)$$

mit $\nu_p, \mu_\pi, \lambda_\epsilon \in \mathbb{Z}$.

Wenn man solch eine Zerlegung gefunden hat, dann kann man daraus folgende Kongruenz herstellen:

$$\begin{aligned}
\prod_{p \in B_1} p^{\nu_p} &= a + bm && (3.1.3) \\
&= \Phi\left(a + b\rho\right) && (\text{ Definition von } \Phi \text{ }) \\
&= \Phi\left[\left(\prod_{\pi \in B_2} \pi^{\mu_\pi}\right) \cdot \left(\prod_{\epsilon \in U} \epsilon^{\lambda_\epsilon}\right)\right] && (3.1.4) \\
&\equiv \prod_{\pi \in B_2} \Phi\left(\pi\right)^{\mu_\pi} \cdot \prod_{\epsilon \in U} \Phi(\epsilon)^{\lambda_\epsilon} \mod n && (\text{ } \Phi \text{ Homomorphismus })
\end{aligned}$$

Setzt man nun

$$(3.1.5) \qquad B := B_1 \cup \Phi(B_2) \cup \Phi(U),$$

so hat man eine Kongruenz der Form (3.1.1) erzeugt.

Findet man eine genügend große Anzahl t solcher Kongruenzen

$$\prod_{i=1}^{k} b_i^{\lambda_{ij}} \equiv \prod_{i=1}^{k} b_i^{\mu_{ij}} \mod n, \quad j = 1, 2, \ldots, t,$$

woraus für $ggT(b_i, n) = 1$ folgt

$$(3.1.6) \qquad \prod_{i=1}^{k} b_i^{\nu_{ij}} \equiv 1 \mod n, \quad j = 1, 2, \ldots, t,$$

mit $\nu_{ij} := \lambda_{ij} - \mu_{ij}$, so muß man nur noch Zahlen $\chi_j \in \{0, 1\}$ suchen, so daß

$$(3.1.7) \qquad \sum_{j=1}^{t} \chi_j \cdot \nu_{ij} \equiv 0 \text{ mod } 2, \text{ d.h. } \sum_{j=1}^{t} \chi_j \cdot \nu_{ij} = 2\eta_i, \qquad \forall i = 1, \ldots, k,$$

denn dann gilt

$$\begin{aligned}
\prod_{j=1}^{t} \left(\prod_{i=1}^{k} b_i^{\nu_{ij}} \right)^{\chi_j} &= \prod_{j=1}^{t} \prod_{i=1}^{k} b_i^{\nu_{ij} \cdot \chi_j} \\
&= \prod_{i=1}^{k} \prod_{j=1}^{t} b_i^{\nu_{ij} \cdot \chi_j} \\
&= \prod_{i=1}^{k} b_i^{\sum_{j=1}^{t} \nu_{ij} \cdot \chi_j} \\
&= \prod_{i=1}^{k} b_i^{2\eta_i}
\end{aligned}$$

und aus (3.1.6) folgt

$$\prod_{i=1}^{k} b_i^{2\eta_i} = \prod_{j=1}^{t} \left(\prod_{i=1}^{k} b_i^{\nu_{ij}} \right)^{\chi_j} \equiv \prod_{j=1}^{t} 1^{\chi_j} \equiv 1^2 \mod n.$$

Damit hat man eine Kongruenz der Form $X^2 \equiv Y^2 \mod n$ gefunden, wobei

$$X = \prod_{i=1}^{k} b_i^{\eta_i} \quad \text{und} \quad Y = 1 \text{ ist.}$$

Diese Überlegungen ergeben die Grundstruktur des 'Number Field Sieve' Algorithmus.

Algorithmus NFS

Input	: $n \in \mathbb{N}$ zusammengesetzt $L \in \mathbb{N}$, die obere Schranke der Primzahlbasis irreduzibles Polynom $f(x) \in \mathbb{Z}[x]$ $m \in \mathbb{N}$ mit $f(m) = n$
Output	: Faktoren n_1, n_2 von n

1.) überprüfe die Irreduzibilität von $f(x)$ über $\mathbb{Q}$,
wenn reduzibel : Output (Fehler : Polynom reduzibel); stop

2.) bestimme die Primzahlbasis B_1

3.) bestimme die Primelementbasis B_2

4.) bestimme die Einheitenmenge U

5.) für ausreichend viele Paare $(a, b) \in \mathbb{Z}^2$

- überprüfe, ob $ggT(a, b) = 1$
- überprüfe, ob $a + bm$ über B_1 zerfällt
- überprüfe, ob $a + b\rho$ über B_2 zerfällt
- wenn alle Überprüfungen mit WAHR zu beantworten, dann berechne aus den Zerlegungen von $a + bm$ und $a + b\rho$ die Kongruenzen

6.) kombiniere die gefundenen Kongruenzen zu Quadraten X^2, Y^2
und berechne $n_1 = ggT(X + Y, n)$; output ($n_1, n/n_1$)

7.) stop

3.2 Irreduzibilitätstests

Wie im letzten Abschnitt gesehen, geht man beim 'Number Field Sieve' davon aus, daß das Polynom $f(x) = x^d + c$, $c \neq 0$, irreduzibel über $\mathbb{Q}$ ist.

Ein Verfahren zur Überprüfung der Irreduzibilität von f liefert Satz 2. Man muß einfach testen, ob es für einen Primteiler p von d ein $x_0 \in \mathbb{Q}$ mit

$$(3.2.1) \qquad {x_0}^p = -c$$

oder, wenn $4|d$, ein $x_0 \in \mathbb{Q}$ mit

$$(3.2.2) \qquad 4{x_0}^4 = c$$

gibt.

Wenn kein solches x_0 gefunden werden kann, so ist $f(x)$ irreduzibel. Es stellt sich die Frage, wie man schnell testen kann, ob solche $x_0 \in \mathbb{Q}$ existieren. Eine Möglichkeit erhält man durch die Betrachtung der Polynome

$$(3.2.3) \qquad g(x) = x^p + c \in \mathbb{Z}[x]$$

und

$$(3.2.4) \qquad h(x) = 4x^4 - c \in \mathbb{Z}[x].$$

Die in Frage kommenden Werte für x_0 sind gerade die Wurzeln dieser Polynome. Um sie zu bestimmen, macht man folgende Fallunterscheidung:

- Ist $p = 2$ und $c > 0$, dann nimmt $g(x)$ nur positive Werte für $x \in \mathbb{R}$ an, d.h. $g(x)$ besitzt in $\mathbb{Q}$ keine Wurzeln.
- Für $p = 2$ und $c < 0$ existieren die beiden reellen Wurzeln
  $x_0 = \sqrt{-c} \quad$ und $\quad x_1 = -\sqrt{-c}$.
- Im Falle $p > 2$, d.h. $2 \nmid p$ (da p Primzahl), so hat $g(x)$ nur eine reelle Wurzel, nämlich $x_0 = \sqrt[p]{-c}$.

Die Wurzeln von $h(x)$ kann man auf gleiche Weise schnell bestimmen:

- Für $c < 0$ nimmt $h(x)$ nur positive Werte für $x \in \mathbb{R}$ an, d.h. es existieren keine reellen Wurzeln.
- Wenn $c > 0$, so gibt es zwei verschiedene reelle Wurzeln, die sich allerdings nur um den Faktor -1 unterscheiden, und zwar $x_0 = \sqrt[4]{\frac{c}{4}}$ und $x_1 = -\sqrt[4]{\frac{c}{4}}$.

Wenn keine der Wurzeln von $g(x)$ bzw. $h(x)$ in $\mathbb{Q}$ liegt, so ist nach Satz 2 $f(x)$ irreduzibel.

Wenn man allerdings eine rationale Wurzel x_0 von $g(x)$ findet, d.h. c läßt sich darstellen als $c = -{x_0}^p$, so gilt für

$$f(x) = x^d + c = (x^q)^p - (x_0)^p,$$

wobei $q = \frac{d}{p}$, daß $(x^q - x_0)$ ein Teiler von $f(x)$ und somit $n_1 = (m^q - x_0)$ ein Teiler von n ist.

Ist x_0 eine rationale Wurzel von $h(x)$, dann gilt wegen $c = 4{x_0}^4$:

$$f(x) = (x^q)^4 + 4 \cdot {x_0}^4 = \left(x^{(2q)} - 2 \cdot x_0 \cdot x^q + 2 \cdot {x_0}^2\right) \cdot (x^{2q} + 2 \cdot x_0 \cdot x^q + 2 \cdot {x_0}^2).$$

Somit hat man die Teiler :

$$n_1 = m^{2q} - 2 \cdot x_0 \cdot m^q + 2 \cdot {x_0}^2 \qquad \text{und}$$

$$n_2 = m^{2q} + 2 \cdot x_0 \cdot m^q + 2 \cdot {x_0}^2$$

von n gefunden. Sind die, auf diese Weise berechneten Zahlen, nichttriviale Teiler von n, so hat man eine Faktorisierung von n gefunden.

Algorithmus IRRED1

Input : $f(x) = x^d + c \in \mathbb{Z}[x]$, $c \neq 0$

Output : WAHR, wenn $f(x)$ irreduzibel
FALSCH, wenn $f(x)$ reduzibel ist

1.) Bestimme die Primteiler $p_1, \ldots, p_k$ von d
2.) For $i \leftarrow 1$ to k do
3.) if ($p_i = 2$ und $c < 0$ und $\sqrt{-c} \in \mathbb{Q}$) then
4.) output (FALSCH); stop
5.) if ($\sqrt[p_i]{|c|} \in \mathbb{Q}$) then output (FALSCH); stop
6.) if ($4|d$ und $\sqrt[4]{\frac{|c|}{4}} \in \mathbb{Q}$) then output (FALSCH); stop
7.) output (WAHR); stop

Eine weitere Möglichkeit, die Irreduzibilität von $f(x)$ zu überprüfen, geht auf Satz 3 (Eisensteinscher Satz) zurück. Dieser Test birgt allerdings sowohl Vor- als auch Nachteile. Der Vorteil besteht darin, daß man hier nicht auf eine spezielle Form des Polynoms $f(x)$ angewiesen ist. Nachteilig wirkt sich jedoch aus, daß dieser Test keinen Beweis für die Reduzibilität eines Polynoms liefert, d.h. wenn das Verfahren die Irreduzibilität eines Polynoms nicht zeigen kann, so bedeutet dies nicht, daß dieses Polynom über $\mathbb{Q}$ zerlegbar ist.

Sei nun $f(x) = f_0 + f_1 \cdot x + \ldots + f_d \cdot x^d \in \mathbb{Z}[x]$ mit $f_d, f_0 \neq 0$. Satz 3 sagt aus, daß $f(x)$ irreduzibel ist, wenn man eine Primzahl $p \in \mathbb{P}$ mit

1. $p \not| f_d$,
2. $p | f_i, \quad \forall i = 0, \ldots, d-1 \quad$ und
3. $p^2 \not| f_0$

findet.

Um sich von der Irreduzibilität von $f(x)$ zu überzeugen, muß man allerdings nicht für alle Primzahlen p die drei Bedingungen nachprüfen, denn aufgrund der zweiten Bedingung kommen nur solche Primzahlen in Frage, die alle Koeffizienten f_i, $i = 0, \ldots, d-1$, und somit den $ggT(f_0, \ldots, f_{d-1})$ teilen. Man muß also nur diesen ggT berechnen und für dessen Primteiler die erste und dritte Bedingung überprüfen.

Beispiel

Ein Beispiel dafür, daß das Verfahren die Irreduzibilität eines irreduziblen Polynoms nicht erkennt, ist $f(x) = x^2 + 2x + 4$. Die einzige in Frage kommende Primzahl ist 2. Aber 2 erfüllt die dritte Bedingung nicht, denn $2^2 | 4 = f_0$. Dennoch ist $x^2 + 2x + 4$ über $\mathbb{Q}$ irreduzibel, denn wäre $f(x)$ reduzibel, so wäre auch $f(x-1)$ reduzibel. Doch

$$\begin{aligned} f(x-1) &= (x-1)^2 + 2 \cdot (x-1) + 4 \\ &= x^2 + 3 \end{aligned}$$

und $x^2 + 3$ ist nach IRRED1 irreduzibel, da $\sqrt{|3|} \notin \mathbb{Q}$.

Algorithmus IRRED2

Input : $f(x) = f_0 + f_1 x + \ldots + f_d x^d \in \mathbb{Z}[x],\ f_0, f_d \neq 0$
Output : WAHR, wenn $f(x)$ irreduzibel
???, wenn die Irreduzibilität nicht bewiesen werden kann

1.) $d_1 \leftarrow ggT(f_0, \ldots, f_{d-1})$

2.) Bestimme die Primteiler $p_1, \ldots, p_k$ von d_1

3.) for $i \leftarrow 1$ to k do

4.) if (f_d mod $p_i = 0$) then goto 3

5.) if (f_0 mod ${p_i}^2 \neq 0$) then output(WAHR); stop

6.) Output(???)

7.) stop

3.3 Primzahlen

Im zweiten Schritt des NFS-Algorithmus sollen alle Primzahlen kleiner einer vorgegebenen Schranke L bestimmt werden. Der klassische Algorithmus, der dies leistet, heißt *Sieb des Eratosthenes* :

In einem Array werden alle Zahlen von $2, \ldots, L$ gespeichert. Dann streicht man nacheinander alle Vielfachen $v \cdot p_k$, $v \in \mathbb{N}$, der k-ten Primzahl aus dem Array. Dabei findet man $(v+1) \cdot p_k$ sehr schnell, indem man von $v \cdot p_k$ p_k Schritte im Array weitergeht, da ja $(v+1) \cdot p_k = v \cdot p_k + p_k$.

Die kleinste Zahl größer als p_k, die, nachdem alle Vielfachen von p_i, $i \leq k$ gestrichen worden sind, noch im Array vorkommt, ist p_{k+1}, da sie durch keine kleinere Primzahl teilbar ist (sonst wäre sie ausgestrichen worden).

Die auszustreichenden Vielfachen von p_k braucht man erst ab ${p_k}^2$ zu betrachten, da für $v < p_k$ $v \cdot p_k$ als Vielfaches eines Primteilers von v ausgestrichen wurde. Hieraus folgt auch, daß man das Verfahren dann abbrechen kann, wenn ${p_k}^2 \geq$L. Alle Zahlen, die zu diesem Zeitpunkt noch im Array stehen, sind Primzahlen.

Im ersten Schritt dieses Verfahrens, also $k = 1$ und $p_k = 2$, werden alle geraden Zahlen außer 2 aus dem Array gestrichen. Deshalb kann man sich von Beginn an auf ungerade Einträge im Array beschränken und das Verfahren mit $p_1 = 3$ starten.

Da die geraden Vielfachen von p_k im Array nicht mehr vorkommen, ist das nach $v \cdot p_k$, v ungerade, zu streichende Vielfache von p_k $(v+2) \cdot p_k$. Dies bedeutet aber auch, daß $(v+2) \cdot p_k$ p_k Stellen weiter im Array steht, weil eben die geraden Einträge fehlen.

Beispiel

Berechnen der Primzahlen kleiner 25.

$p_0 = 2$.
Im Array sind die ungeraden Zahlen kleiner gleich 25 gespeichert.
$*$ bedeutet : ausgestrichen.

	3	5	7	9	11	13	15	17	19	21	23	25
Sieben mit $p_1 = 3$:	*			*			*			*		
nächste Primzahl also $p_2 = 5$:	*	*		*			*			*		*

Da $5^2 \geq 25$ sind also alle noch im Array stehenden Zahlen Primzahlen. Diese sind

7, 11, 13, 17, 19, 23.

Algorithmus PRIMZAHLEN

Input : obere Schranke L
Output : alle Primzahlen kleiner L

1 .) if (L mod 2 = 0) then $M \leftarrow$ L/2
2 .) else $M \leftarrow$ (L+1)/2 (∗ $M =$ Anzahl der ungeraden Zahlen kleiner L ∗)
3 .) for $i \leftarrow 1$ to $M-1$ do
4 .) $\quad A[i] \leftarrow$ WAHR (∗ $A[i] \cong 2i+1$ ∗)
5 .) $j \leftarrow 1$
6 .) output (2)
7 .) while ($A[j]$ =FALSCH) do
8 .) $\quad j \leftarrow j+1$
9 .) $p \leftarrow 2j+1$
10 .) output (p)
11 .) $k \leftarrow \frac{p^2-1}{2}$ (∗ $A[k] \cong p^2$ ∗)
12 .) if $(k \geq M)$ then goto 17
13 .) while $(k < M)$ do
14 .) $\quad A[k] \leftarrow$FALSCH
15 .) $\quad k \leftarrow k+p$
16 .) goto 7
17 .) for $k \leftarrow j+1$ to $M-1$ do
18 .) $\quad$ if ($A[k]$ =WAHR) then output ($2k+1$)
19 .) stop

3.4 Primelemente

3.4.1 Der mathematische Hintergrund

In Schritt 5 des Algorithmus NFS wird zur Zerlegung der ganzen algebraischen Zahl[1] $\alpha + \beta\rho$ eine Menge von Primelementen aus $\mathbb{Z}[\rho]$ mit primer Norm benötigt. Bevor ich näher darauf eingehe, wie man diese berechnen kann, möchte ich zuerst einige mathematische Sachverhalte, die im Zusammenhang mit der Primelementberechnung stehen, aufzeigen.

Im folgenden sei die Grundvoraussetzung gegeben, daß G ein Hauptidealring ist. Mit Hilfe von Satz 14 läßt sich daraus schließen, daß G ein ZPE-Ring ist. Weiter sei E die Einheitengruppe in G.

Die Teilbarkeit in G ist durch Definition 7 festgelegt :

$$\beta|\alpha \Leftrightarrow \exists\ \gamma \quad \text{mit } \alpha = \beta \cdot \gamma\ , \qquad \alpha, \beta, \gamma \in \text{G}.$$

Ein Maß für ein Element aus G ist die Norm, die nach Satz 7 multiplikativ ist. Diese Eigenschaft der Norm bildet die Grundlage für nachstehendes Lemma :

Lemma 7

Seien $\alpha, \beta \in$G.
Dann gilt : $\beta|\alpha \Rightarrow N(\beta)|N(\alpha)$.

Beweis :

$$\begin{aligned} \beta|\alpha &\Rightarrow \exists\ \gamma \in \text{G mit } \alpha = \beta \cdot \gamma && \text{(Definition 7)} \\ &\Rightarrow N(\alpha) = N(\beta \cdot \gamma) = N(\beta) \cdot N(\gamma) && \text{(Multiplikativität der Norm)} \\ &\Rightarrow N(\beta)|N(\alpha) && \text{(Teilbarkeit in } \mathbb{Z}) \\ & && \square \end{aligned}$$

Die Umkehrung dieses Lemmas gilt jedoch nicht, wie man an folgendem Beispiel erkennen kann :

Beispiel

$f(x) = x^3 + 2$ ist über $\mathbb{Q}$ irreduzibel.
Eine Wurzel von $f(x)$ ist $\rho = -\sqrt[3]{2}$.
In diesem Fall gilt sogar G$= \mathbb{Z}[\rho]$. Die Norm $N(\alpha)$, für $\alpha = a_0 + a_1 \cdot \rho + a_2 \cdot \rho^2 \in$G, kann man mit Hilfe der Formel

$$N(\alpha) = {a_0}^3 - 2 \cdot {a_1}^3 + 4 \cdot {a_2}^3 + 6 \cdot a_0 \cdot a_1 \cdot a_2$$

berechnen. Wie man diese Formel errechnet, wird an späterer Stelle[2] gezeigt.

[1] siehe auch Kapitel 3.1 (3.1.4)
[2] Kapitel 5.2

Für $\alpha = 5 - 6\rho + 3\rho^2 \in$G beträgt die Norm $N(\alpha) = 5^3 + 2\cdot 6^3 + 4\cdot 3^3 - 6\cdot 5\cdot 6\cdot 3$
$= 125.$
Für $\beta = 5 \in$G beträgt die Norm ebenfalls $N(\beta) = 125$.

Also $N(\beta)|N(\alpha)$, aber ganz offensichtlich teilt β α nicht, da alle Komponenten der Vielfachen von β durch 5 teilbar sein müssen.

Unter der oben gemachten Voraussetzung, daß es sich bei G um einen ZPE-Ring handelt, gilt, daß ein Element aus G genau dann prim ist, wenn es irreduzibel ist. So kann man für Primelemente folgende Aussage treffen:

Lemma 8
Sei G ein ZPE-Ring.
Dann ist $\pi \in$G$\setminus$E genau dann ein Primelement, wenn aus

$$\pi = \alpha\cdot\beta \Rightarrow \alpha \in \mathrm{E} \quad \text{oder} \quad \beta \in \mathrm{E}.$$

Eine hinreichende Bedingung dafür, daß ein Element aus G ein Primelement ist, liefert das nächste Lemma.

Lemma 9
Sei $\pi \in$G und $N(\pi) = p \in \mathbb{P}$.
Dann ist π ein Primelement.

Beweis :

Zu zeigen ist : $\pi = \alpha\cdot\beta \Rightarrow \alpha \in$E oder $\beta \in$E.
Da $\pi = \alpha\cdot\beta$ und die Norm multiplikativ ist, folgt

$$N(\pi) = N(\alpha\cdot\beta) = N(\alpha)\cdot N(\beta).$$

Da nach Satz 7b) die Normen ganzer algebraischer Zahlen in $\mathbb{Z}$ liegen und nach Voraussetzung $N(\pi) = p \in \mathbb{P}$, folgt aus der Primzahleigenschaft von p, daß $N(\alpha)$ oder $N(\beta)$ eine Einheit in $\mathbb{Z}$ sein muß, d.h. $N(\alpha) = \pm 1$ oder $N(\beta) = \pm 1$.
Dann gilt aber nach Lemma 2, daß $\alpha \in$E oder $\beta \in$E und nach Lemma 8, daß π ein Primelement ist. $\square$

Eine weitere Möglichkeit, Eigenschaften von Primelementen aus G zu erhalten, beruht auf der Zurückführung der Teilbarkeit von Elementen in G auf die Teilbarkeit der von ihnen erzeugten Hauptideale. Denn nach Satz 12 gilt:

$$\beta|\alpha \Leftrightarrow (\beta)|(\alpha), \qquad \alpha, \beta \in \mathrm{G}.$$

Hieraus läßt sich ein Zusammenhang zwischen Primelementen in G und Primidealen in $\mathbb{Q}[\rho]$ herstellen.

Lemma 10
Sei $\pi \in$G.
π ist genau dann ein Primelement in G, wenn (π) ein Primideal ist.

<u>*Beweis*</u> :

"$\Rightarrow$"
Vor: π ist Primelement

Beh: (π) ist Primideal

Annahme: (π) ist zerlegbar
D.h. es existieren zwei Ideale $\mathcal{P}_1, \mathcal{P}_2$ mit $\mathcal{P}_1, \mathcal{P}_2 \neq (1)$ und $(\pi) = \mathcal{P}_1 \cdot \mathcal{P}_2$.
Da $\pi \in (\pi)$, muß es Elemente $\gamma_1 \in \mathcal{P}_1, \gamma_2 \in \mathcal{P}_2$ mit $\pi = \gamma_1 \cdot \gamma_2$ geben.
Da π Primelement, folgt aber: $\gamma_1 \in$E oder $\gamma_2 \in$E. Dies bedeutet, daß entweder $\mathcal{P}_1 = (1)$ oder $\mathcal{P}_2 = (1)$, was der Annahme widerspricht.

Also gilt die Behauptung.

"$\Leftarrow$"
Vor: (π) ist Primideal

Beh: π ist Primelement in G

Annahme: π ist zerlegbar in G
D.h. es existieren zwei Elemente $\alpha, \beta \in$ G$\backslash$E mit $\pi = \alpha \cdot \beta$, β teilt also π.
Nach Satz 12 gilt dann aber auch

$$(\beta)|(\pi).$$

Weil (π) Primideal ist, folgt daraus , daß $(\beta) = (1)$ oder $(\beta) = (\pi)$. Wenn $(\beta) = (1)$, dann $N(\,(\beta)\,) = 1$ und wegen $|N(\beta)| = N(\,(\beta)\,)$ gilt damit $N(\beta) = \pm 1$, also $\beta \in$E, was ein Widerspruch zur Annahme wäre.

Also gilt $(\beta) = (\pi)$ und damit auch $N(\,(\beta)\,) = N(\,(\pi)\,)$. Für die Norm von α bedeutet das :

$$|N(\alpha)| = N(\,(\alpha)\,) = N(\,(\pi)\,)\,/\,N(\,(\beta)\,) = 1.$$

Also gilt $N(\alpha) = \pm 1$ und damit $\alpha \in$E, was ein Widerspruch zur Annahme ist.

Also gilt die Behauptung. $\square$

Dieses Lemma zeigt, daß man die Suche nach Primelementen in G auf die Suche nach primen Hauptidealen zurückführen kann. Doch wie findet man prime Hauptideale?

Nach Lemma 4c) gilt : Wenn $\mathcal{P}$ ein Primideal ist, dann existiert genau eine Primzahl $p \in \mathbb{P}$ mit $p \in \mathcal{P}$. Zerlegt man also alle Hauptideale (p) mit $p \in \mathbb{P}$ in Primideale, so erhält man alle Primideale und damit auch alle Primideale primer Norm.
Jetzt bleibt nur noch die Frage zu klären, wie man das Hauptideal (p) zerlegen kann.

Für G= $\mathbb{Z}[\rho]$ hilft Satz 16 weiter. Berechnet man die über $\mathbb{Z}/p\mathbb{Z}$ irreduziblen Polynome $\overline{f_i}(x) \in \mathbb{Z}/p\mathbb{Z}[x]$, in die $f(x)$ modulo p zerfällt, und sind dabei $g_i(x) \in \mathbb{Z}[x]$ mit

$$g_i(x) \equiv \overline{f_i}(x) \bmod p,$$

so sind $\mathcal{P}_i := (p, g_i(\rho))$ Primideale mit der Norm

$$N(\mathcal{P}_i) = p^{grad\left(\overline{f_i}(x)\right)}.$$

Im 'Number Field Sieve' interessiert man sich allerdings nur für solche Primideale, die prime Norm besitzen [3], d.h. $N(\mathcal{P}_i) = p$, was bedeutet, daß $grad\left(\overline{f_i}(x)\right) = 1$.
$\overline{f_i}(x)$ muß in diesem Fall also ein Linearfaktor sein und deshalb die Form

$$\overline{f_i}(x) = x - cp, \qquad cp \in \{0, \ldots, p-1\},$$

besitzen. Dies bedeutet, daß cp eine Wurzel von $\overline{f_i}(x)$ und somit eine Wurzel von $f(x)$ mod p ist. Da man für jede Wurzel $cp^{(i)}$ von $f(x)$ mod p einen Linearfaktor $x - cp^{(i)}$ abspalten kann, folgt daraus, daß es zu jeder Wurzel $cp^{(i)}$ von $f(x)$ mod p ein Primideal $\mathcal{P}_i$ mit $N(\mathcal{P}_i) = p$ und $\mathcal{P}_i = (p, \rho - cp^{(i)})$ gibt.

Aus diesen Überlegungen folgt der

Satz 20
Sei $p \in \mathbb{P}$.
$\mathcal{P}$ ist genau dann ein Primideal der Norm p, wenn ein $cp \in \{0, \ldots, p-1\}$ existiert mit

$$f(cp) \equiv 0 \bmod p$$

und

$$\mathcal{P} = (p, \rho - cp).$$

Anders ausgedrückt besagt dieser Satz, daß sich jedem Paar (p, cp) mit $p \in \mathbb{P}$, $cp \in \{0, \ldots, p-1\}$ und $f(cp) \equiv 0 \bmod p$ genau ein Primideal $\mathcal{P} = (p, \rho - cp)$ zuordnen läßt und umgekehrt.

Im folgenden werde ich diese Primideale mit $\mathcal{P}_{p,cp}$ bezeichnen.

[3] siehe auch Kapitel 3.6

Alle Elemente $\alpha \in$G, die in einem solchen Primideal $\mathcal{P}_{p,cp}$ liegen, besitzen ein gemeinsames Merkmal:

Lemma 11
Sei $\mathcal{P}_{p,cp} = (p, \rho - cp)$ und sei $\varphi_{cp} : \mathbb{Z}[\rho] \to \mathbb{Z}/p\mathbb{Z}$ mit $\rho \mapsto cp$.
Dann gilt :

$$\alpha \in \mathcal{P}_{p,cp} \Leftrightarrow \alpha \in kern(\varphi_{cp}), \text{ d.h. } \varphi_{cp}(\alpha) \equiv 0 \bmod p.$$

<u>*Beweis*</u> :

1. "$\Rightarrow$"
   Zeige : $\alpha \in \mathcal{P}_{p,cp} \Rightarrow \varphi_{cp}(\alpha) \equiv 0 \bmod p$
   Aus $\alpha \in \mathcal{P}_{p,cp} \Rightarrow \alpha = \beta \cdot p + \gamma \cdot (\rho - cp)$, wobei
   $$\beta = b_0 + b_1 \cdot \rho + \ldots + b_{d-1} \cdot \rho^{d-1} \quad \text{und} \quad \gamma = c_0 + c_1 \cdot \rho + \ldots + c_{d-1} \cdot \rho^{d-1} \in \mathbb{Z}[\rho]$$
   $$\begin{aligned}\varphi_{cp}(\alpha) &= \varphi_{cp}(\beta \cdot p + \gamma \cdot (\rho - cp)) \\ &\equiv (b_0 + b_1 cp + \ldots + b_{d-1} cp^{d-1}) \underbrace{p}_{\equiv 0} + (c_0 + c_1 cp + \ldots + c_{d-1} cp^{d-1}) \cdot \underbrace{(cp - cp)}_{\equiv 0} \bmod p \\ &\equiv 0 \bmod p \\ &\Rightarrow \alpha \in kern(\varphi_{cp})\end{aligned}$$

2. "$\Leftarrow$"
   Zeige : $\alpha \in kern(\varphi_{cp}) \Rightarrow \alpha \in \mathcal{P}_{p,cp}$
   Sei $\alpha = a_0 + a_1\rho + \ldots + a_{d-1}\rho^{d-1}$ mit $\varphi_{cp}(\alpha) \equiv 0 \bmod p$.
   Dann existiert ein $k \in \mathbb{Z}$, so daß $\varphi_{cp}(\alpha) = k \cdot p$.
   $$\begin{aligned}\Rightarrow \alpha &= \alpha - \varphi_{cp}(\alpha) + k \cdot p \\ &= a_0 + a_1 \cdot \rho + \ldots + a_{d-1} \cdot \rho^{d-1} - (a_0 + a_1 \cdot cp + \ldots + a_{d-1} \cdot cp^{d-1}) + k \cdot p \\ &= a_1 \cdot (\rho - cp) + \ldots + a_{d-1} \cdot (\rho^{d-1} - cp^{d-1}) + k \cdot p \\ &= \underbrace{\left(a_1 + \sum_{i=2}^{d-1} a_i \cdot \sum_{j=0}^{i-1} (\rho^{i-j-1} \cdot cp^j) \right)}_{\beta :=} \cdot (\rho - cp) + \underbrace{k}_{\gamma :=} \cdot p\end{aligned}$$
   $$\Rightarrow \alpha = \beta \cdot (\rho - cp) + \gamma \cdot p$$
   $$\Rightarrow \alpha \in \mathcal{P}_{p,cp}$$

Aus 1.) und 2.) folgt die Behauptung. □

Sei das Primideal $\mathcal{P}_{p,cp}$ ein Hauptideal, d.h. $\mathcal{P}_{p,cp} = (\pi)$, so ist nach Lemma 10 π ein Primelement in G.
Ein erzeugendes Element π von $\mathcal{P}_{p,cp}$ besitzt folgende Eigenschaften :

(i) $\pi \in \mathcal{P}_{p,cp}$ und

(ii) $|N(\pi)| = N(\mathcal{P}_{p,cp})$.

Äquivalent zur ersten Eigenschaft ist nach Lemma 12 die folgende :
(i') $\varphi_{cp}(\pi) \equiv 0 \bmod p$.

Es gilt also :

Lemma 12
Ein zu einem Paar (p, cp) gehörendes Primideal $\mathcal{P}_{p,cp}$ wird genau dann von $\pi \in$G erzeugt, wenn

1. $N(\pi) = p$ und
2. $\varphi_{cp}(\pi) \equiv 0 \bmod p$.

Ist G= $\mathbb{Z}[\rho]$, wie zu Beginn dieses Kapitels vorausgesetzt, ein ZPE-Ring, so ist G nach Satz 15 auch ein Hauptidealring und somit werden alle Primideale primer Norm, durch Primelemente, die obigen Bedingungen genügen, erzeugt.

Aus den oben beschriebenen Zusammenhängen zwischen Primelementen und Primidealen ergibt sich der folgende Algorithmus zur Berechnung der Primelemente aus G mit primer Norm.

Algorithmus PRIMELEMENTE

Input : $p_1, \ldots, p_k \in B_1,\ f(x)$
Output : Erzeuger aller Primideale mit Norm p_i

1.) for all $p \in B_1$ do
2.) Bestimme alle Paare (p, cp) mit $f(cp) \equiv 0 \bmod p$
3.) Bestimme einen Erzeuger π von $\mathcal{P}_{p,cp}$
4.) output (π)
5.) stop

3.4.2 Die Paare (p, cp)

Bezeichnung

In diesem Abschnitt werde ich $/_p$ bzw. ggT_p für die Division bzw. den größten gemeinsamen Teiler modulo p verwenden.

Bei der Suche nach den Primidealen mit primer Norm, spielen die Paare (p, cp) mit $p \in \mathbb{P}$, $cp \in \{0, \ldots, p-1\}$ und $f(cp) \equiv 0 \bmod p$ eine sehr große Rolle.

Der unkomplizierteste Weg, den man zur Berechnung der (p, cp) beschreiten kann, ist das Einsetzen der möglichen Werte von cp in $f(x) \bmod p$ und das Kontrollieren, für welche dieser Werte

$$f(cp) \equiv 0 \bmod p$$

ist.

Die maximale Anzahl der $cp's$, die diese Kongruenz erfüllen können, ist gleich dem Grad d des Polynoms $f(x)$. Deshalb kann man das Verfahren nach d gefundenen cp's, für ein $p \in \mathbb{P}$, abbrechen.

Aus diesen Betrachtungen resultiert der erste Algorithmus zur Berechnung der Paare (p, cp):

Algorithmus PCP1

Input : $f(x)$, p

Output : alle $cp \in \{0, \ldots, p-1\}$ mit $f(cp) \equiv 0 \bmod p$

1.) $cnt \leftarrow 0$
2.) for $cp \leftarrow 0$ to $p-1$ do
3.) if $(f(cp) \equiv 0 \bmod p)$ then output(cp); $cnt \leftarrow cnt + 1$
4.) if $(cnt = d)$ then stop
5.) stop

Beispiel

Für $f(x) = x^3 + 2$ und $|B_1| = 500$ errechnet man 497 verschiedene Paare, z.B.

$p = 7$: es existiert kein cp mit $f(cp) \equiv 0 \bmod 7$,

$p = 5$: es existiert genau ein cp, nämlich $cp = 2$,

$p = 31$: es existieren drei cp's : $cp^{(1)} = 11$, $cp^{(2)} = 24$, $cp^{(3)} = 27$.

Ein zweites Verfahren zur Berechnung der Paare (p, cp), beruht auf dem *Kleinen Fermatschen Satz.*

Satz 21(Kleiner Fermatscher Satz) [4]
Sei $p \in \mathbb{P}$.
Dann gilt für alle $a \in \mathbb{Z}$ mit $p \not| a$

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

Für das Polynom $x^{p-1} - 1$ folgt aus diesem Satz, daß es modulo p in $p-1$ verschiedene Linearfaktoren zerfällt, denn alle Zahlen $cp \in \{1, \ldots, p-1\}$ sind nach dem Kleinen Fermatschen Satz Wurzeln von $x^{p-1} - 1$ mod p. Es gilt also die Kongruenz

$$x^{p-1} - 1 \equiv \prod_{cp=1}^{p-1} (x - cp) \mod p.$$

Bildet man den $ggT_p(x^{p-1} - 1, f(x))$, so besteht dieser aus dem Produkt aller verschiedenen Linearfaktoren mit Ausnahme von x, die $f(x)$ modulo p teilen.

Um zu überprüfen, ob x $f(x)$ mod p teilt, ist es nur notwendig, den konstanten Anteil f_0 von $f(x)$ zu untersuchen, denn $f(x) = f_0 + f_1 x + \ldots + f_{d-1} x^{d-1} + x^d$ ist genau dann modulo p durch x teilbar, wenn gilt $f_0 \equiv 0 \bmod p$.

Nach Ausführung dieses kleinen Tests kann man die restlichen Linearfaktoren von $f(x)$ modulo p durch Berechnung von

$$(3.4.1) \qquad h(x) = ggT_p(x^{p-1} - 1, f(x))$$

bestimmen.
Der Grad d_h von $h(x)$ gibt dabei an, in wieviele verschiedene Linearfaktoren, mit Ausnahme von x, $f(x)$ dabei zerfällt und somit, wieviele Paare (p, cp) existieren.

Man unterscheidet dabei drei Fälle :

$d_n = 0$: $f(x)$ besitzt modulo p keine Linearfaktoren.

$d_h = 1$: $f(x)$ besitzt modulo p genau einen Linearfaktor. Dieser ist $x - h_0$.

$d_h > 1$: $f(x)$ besitzt modulo p mehrere Linearfaktoren.

Tritt einer der beiden ersten Fälle ein, so hat man bereits alle Linearfaktoren des Polynoms $f(x)$ gefunden. Für $d_h > 1$ allerdings kennt man zwar die Anzahl der verschiedenen Linearfaktoren, nicht aber ihr Aussehen. Dieses versucht man nun durch die Berechnung des größten gemeinsamen Teilers von h(x) und $g(x)$ zu bestimmen, wobei $g(x)$ ein Teiler von $x^{p-1} - 1$ modulo p ist.

Für $p = 2$ kann dieser dritte Fall nicht eintreten, da $grad(ggT_p(x^{p-1} - 1, f(x)))$ höchstens 1 sein kann.

Es kann also davon ausgegangen werden, daß $p > 2$ und damit p ungerade ist.

[4][2], Corollary 3.2.1

Aus dem Kleinen Fermatschen Satz folgt

$$b^{p-1} \equiv 1 \mod p, \qquad \forall b \not\equiv 0 \mod p$$

und damit

$$(x-a)^{p-1} \equiv 1 \mod p, \qquad x \neq a.$$

Also zerfällt $(x-a)^{p-1}-1$ modulo p in die Linearfaktoren

$$(x-0)\cdot(x-1)\cdot\ldots\cdot(x-(a-1))\cdot(x-(a+1))\cdot\ldots\cdot(x-(p-1)).$$

Da p ungerade ist, gilt weiter :

$$(x-a)^{p-1}-1 = \left((x-a)^{\frac{p-1}{2}}-1\right)\cdot\left((x-a)^{\frac{p-1}{2}}+1\right).$$

Für verschiedene a zerfällt $\left((x-a)^{\frac{p-1}{2}}-1\right)$ auch in verschiedene Linearfaktoren. Zerfällt zum Beispiel

$$(x-0)^{\frac{p-1}{2}}-1$$

modulo p in

$$(x-c_1)\cdot\ldots\cdot\left(x-c_{\frac{p-1}{2}}\right)$$

so zerfällt

$$\left((x-a)^{\frac{p-1}{2}}-1\right)$$

modulo p in

$$(x-c_1+a)\cdot\ldots\cdot(x-c_{\frac{p-1}{2}}+a).$$

Berechnet man für ein $a \in \{0,\ldots,p-1\}$ die größten gemeinsamen Teiler

$$(3.4.2) \qquad h_1(x) = ggT_p\left((x-a)^{\frac{p-1}{2}}-1, h(x)\right), \qquad \text{und}$$

$$(3.4.3) \qquad h_2(x) = ggT_p\left((x-a)^{\frac{p-1}{2}}+1, h(x)\right),$$

so können mehrere wesentlich verschiedene Fälle auftreten.

Fall 1 $grad(h_1(x)) + grad(h_2(x)) \neq grad(h(x))$

Ein Linearfaktor von $h(x)$ taucht in $h_1(x), h_2(x)$ nicht mehr auf. Dabei kann es sich nur um $(x-a)$ handeln, da dieser in der Zerlegung von

$$(x-a)^{p-1}-1 \mod p$$

nicht vorkommt. $(x-a)$ ist also ein Linearfaktor von $h(x)$ und damit von $f(x)$.

Fall 2 $grad(h_1(x)) = 0$ oder $grad(h_2(x)) = 0$

In diesem Fall gewinnt man keine neuen Erkenntnisse über die Beschaffenheit der Linearfaktoren. Man muß es mit einem anderen a versuchen.

Fall 3 $grad(h_1(x)) = 1$ oder $grad(h_2(x)) = 1$

Man hat einen Linearfaktor von $h(x)$ und damit von $f(x)$ modulo p gefunden. Dieser ist $x - h_{1,0}$ bzw. $x - h_{2,0}$. $h_{i,0}$ ist dabei der konstante Anteil von $h_i(x)$.

Fall 4 $grad(h_1(x)) > 1$ und $grad(h_2(x)) > 1$

Dies bedeutet, daß man eine Zerlegung von $h(x)$ modulo p in zwei nicht lineare Faktoren gefunden hat. Diese beiden Faktoren versucht man nun durch Bildung von $ggT_p\left((x-a)^{\frac{p-1}{2}} - 1, h_i(x)\right)$, $i = 1, 2$, für andere Werte von a zu berechnen.

Auf diese Weise lassen sich alle Linearfaktoren von $f(x)$ bestimmen. Einen Linearfaktor $(x - cp)$ findet man dabei spätestens, wenn man den größten gemeinsamen Teiler mit $(x - cp)^{\frac{p-1}{2}} - 1$ bildet, weil dann Fall 1 eintreten muß .

Aus diesen Überlegungen ergibt sich der folgende Algorithmus zur Berechnung der Paare (p, cp).

Algorithmus PCP2

Input : $f(x)$, p

Output : alle $cp \in \{0, \dots, p-1\}$ mit $f(cp) \equiv 0 \bmod p$

if $(f_0 \equiv 0) \bmod p$ then output(0)
$h_1(x) \leftarrow ggT_p(f(x), x^{p-1} - 1)$ $(* \cong (3.4.1) *)$
$df \leftarrow grad(h_1)$
if $(df = 0)$ then stop
if $(df = 1)$ then output($-h_{1,0}$); stop
$e_1 \leftarrow df$
$jj \leftarrow 1$; $k \leftarrow 0$ (* $jj + k$ = Anzahl der verschiedenen Funktionen h_i *)
for $a \leftarrow 0$ to $p-1$ do
    for $j \leftarrow 1$ to jj do
        $i \leftarrow jj + k + 1$ (* erste unbesetzte Funktion *)
        $h_i(x) \leftarrow ggT_p\left(h_j(x), (x-a)^{\frac{p-1}{2}} - 1\right)$ $(* \cong (3.4.2) *)$
        $h_{i+1}(x) \leftarrow ggT_p\left(h_j(x), (x-a)^{\frac{p-1}{2}} + 1\right)$ $(* \cong (3.4.2) *)$
        if $(grad(h_i(x) + grad(h_{i+1}(x)) < grad(h_j(x))$ then (* Fall 1 *)
            output(a); $h_j(x) \leftarrow h_j(x) /_p (x-a)$;$e_j \leftarrow e_j - 1$; $df \leftarrow df - 1$
        $e_i \leftarrow grad(h_i)$
        case $(e_i) =$
                    1: output($-h_{i,0}$) (* Fall 3 *)
                        $df \leftarrow df - 1$
                        $h_j(x) \leftarrow h_j(x) /_p h_i(x)$
            $e_j - 1$: $g(x) \leftarrow h_j(x) /_p h_i(x)$ (* Fall 3 *)
                        output($-g_0$); $df \leftarrow df - 1$
                        $h_j(x) \leftarrow h_i(x)$; $e_j \leftarrow e_j - 1$
              sonst: $h_j(x) \leftarrow h_j(x) /_p h_i(x)$; $e_j \leftarrow e_j - e_i$ (* Fall 2 oder Fall 4 *)
                        $k \leftarrow k + 1$
        if $(e_j = 1)$ then
            output($-h_{j,0}$); $df \leftarrow df - 1$
            $h_j(x) \leftarrow h_i(x)$; $e_j \leftarrow e_i$
            if $(k = 0)$ then $jj \leftarrow jj - 1$; $j \leftarrow j - 1$
                        else $k \leftarrow k - 1$
        if $(df = 0)$ then stop
    $jj \leftarrow jj + k$; $k \leftarrow 0$
stop

3.4.3 Primidealerzeuger

Im Kapitel 3.4.1 wurde gezeigt, daß zu jedem Paar (p, cp) mit $f(cp) \equiv 0 \bmod p$ ein Primideal $\mathcal{P}_{p,cp} = (p, \rho - cp)$ existiert. Im Falle, G=$\mathbb{Z}[\rho]$ ist ein Hauptidealring, werden alle Primideale von einem Element π der Norm p erzeugt, wobei π die Eigenschaft

$$\varphi_{cp}(\pi) \equiv 0 \bmod p$$

besitzt.

In diesem Abschnitt werde ich zwei Möglichkeiten aufzeigen, wie man diese Primidealerzeuger berechnen kann. Beiden Verfahren geht die Erstellung der Liste aller Paare (p, cp) mit $p \in B_1$ voraus.

Die erste Version der Erzeugendenberechnung untersucht alle Elemente $\alpha \in \mathbb{Z}[\rho]$, deren Koeffizienten zwischen den Schranken UK und $OK \in \mathbb{Z}$ liegen. Für diese Elemente berechnet man die Norm $N(\alpha)$. Liegt diese in B_1, also $N(\alpha) = p_j \in B_1$, so hat man einen Primidealerzeuger gefunden.

Das zugehörige ${cp_j}^{(i)}$ findet man durch Überprüfen der Kongruenz

$$\varphi_{{cp_j}^{(i)}}(\alpha) \equiv 0 \bmod p_j$$

für alle Paare $(p_j, {cp_j}^{(i)})$ aus der Liste. Wenn ${cp_j}^{(i_1)}$ die Kongruenz erfüllt, dann ist α ein Erzeuger von $\mathcal{P}_{p_j,{cp_j}^{(i_1)}}$. Wählt man die Schranken UK klein und OK groß genug, so findet man zu jedem Paar (p, cp) der Liste einen Erzeuger.

Verbessern läßt sich dieses Verfahren, indem man nicht nur die Elemente mit primer Norm betrachtet, sondern auch solche, deren Norm ein kleines, quadratfreies Vielfaches einer Primzahl ist.

Dazu werden zu jedem Paar (p, cp) die Variablen $a(p, cp)$ und $g(p, cp)$ eingeführt. $a(p, cp)$ enthält das kleinste zu (p, cp) gefundene Vielfache und $g(p, cp)$ das entsprechende Element aus $\mathbb{Z}[\rho]$. $a(p, cp)$ wird mit ∞ vorbesetzt. Findet man ein Element $\alpha \in \mathcal{P}_{p,cp}$, dessen Norm $N(\alpha)$ sich als

$$|N(\alpha)| = |k| \cdot p$$

schreiben läßt, wobei

- k quadratfrei,
- $0 < |k| < p$ und
- $|k| < a(p, cp)$

ist, so wird α unter $g(p, cp)$ abgespeichert und $a(p, cp)$ durch $|k|$ ersetzt.

Wenn man zu jedem Paar (p, cp) der Liste ein solches α gefunden hat, was gleichbedeutend ist mit der Tatsache, daß für alle (p, cp) $\quad a(p, cp) \neq \infty$. so gilt für die kleinste Primzahl p_1, die in der Liste vorkommt :

$$a(p_1, cp_1{}^{(i)}) = 1, \qquad i = 1, \ldots,$$

denn sonst wäre das zugehörige Element $g(p_1, cp_1{}^{(i)})$ noch durch ein Element mit Norm $k \neq 1$ teilbar, was wegen der Minimalität von p_1 nicht möglich ist. Also gilt $N\left(g(p_1, cp_1{}^{(i)})\right) = p_1$ und somit ist $g(p_1, cp_1{}^{(i)})$ ein Erzeuger von $P_{p_1, cp_1{}^{(i)}}$.
Man dividiert nun alle $g(p, cp^{(m)})$ mit

$$p_1 | a(p, cp^{(m)}) \qquad \text{und}$$

$$\varphi_{cp^{(i)}}\left(g(p, cp^{(m)})\right) \equiv 0 \bmod p_1$$

durch die Primideale mit der Norm $g(p_1, cp_1{}^{(i)})$. Ist p_2 die zu p_1 nächstgrößere Primzahl in der Liste, so muß nun $|a(p_2, cp_2{}^{(i)})| = 1$ sein, da p_1 der einzige Faktor war, der in $|a(p_2, cp_2{}^{(i)})|$ vorkommen konnte, und dieser herausdividiert wurde. $p_1{}^k$, $k > 1$, konnte nicht auftreten, da $a(p, cp)$ als quadratfrei vorausgesetzt wurde. Aus diesem Grund können auch in den gefundenen $g(p, cp)$ keine Primfaktoren mit Norm p^k, $k > 1$, stecken.

Setzt man dieses Verfahren induktiv fort, so erhält man alle gesuchten Primidealerzeuger. In der Praxis wird sich herausstellen, daß bei gut gewählten Grenzen UK, OK für die meisten (p, cp) $\quad |a(p, cp)| = 1$ sein wird.

Im nachstehenden Algorithmus werden die obigen Überlegungen verarbeitet. Dabei werden folgende neuen Prozeduren benutzt :

- $factor(n)$, liefert den größten Primteiler p von n und n/p.
- $nextelement(\alpha)$, liefert das nach α kommende Element auf dem Weg von $(UK + UK \cdot \rho + \ldots + UK \cdot \rho^{d-1})$ nach $(OK + OK \cdot \rho + \ldots + OK \cdot \rho^{d-1})$.
- $teiler(k, \alpha)$, liefert ein Paar (p, cp) mit $p|k$ und $\varphi_{cp}(\alpha) \equiv 0 \bmod p$.
- $quadfrei(k)$, liefert WAHR, wenn k quadratfrei; FALSCH, sonst.

Algorithmus ERZEUGER1

Input : Liste PCP aller Paare (p, cp)
UK, OK

Output : die Primidealerzeuger von $\mathcal{P}_{p,cp}$

for all $(p, cp) \in$ PCP do
  $a(p, cp) \leftarrow \infty$
$\alpha \leftarrow UK + UK \cdot \rho + \ldots + UK \cdot \rho^{d-1}$
while $(\alpha \neq OK + OK \cdot \rho + \ldots + OK \cdot \rho^{d-1})$
  $n \leftarrow N(\alpha)$
  $(k, q) \leftarrow factor(n)$
  if $(|k| < q$ und $quadfrei(k) =$WAHR$)$ then
    for all $(p, cp) \in PCP$ mit $p = q$ do
      if $(\varphi_{cp}(\alpha) \equiv 0 \bmod p$ und $a(q, cp) > |k|)$ then
        $a(q, cp) \leftarrow |k|$
        $g(q, cp) \leftarrow \alpha$
  $\alpha \leftarrow nextelement(\alpha)$
for all $(p, cp) \in$PCP, p in aufsteigender Reihenfolge do
  if $(a(p, cp) = \infty)$ then output ("Grenzen UK, OK zu klein gewählt")
  while $(|k| \neq 1)$ do
    $(q, c) \leftarrow teiler(k, g(p, cp))$
    $k \leftarrow k/q$
    $g(p, cp) \leftarrow g(p, cp)/g(q, c)$
  output ($g(q, c)$)
stop

Das zweite Verfahren zur Bestimmung der Primidealerzeuger sucht nicht wie das erste in ganz $\mathbb{Z}[\rho]$, sondern nur in den entsprechenden Primidealen.

In Kapitel 3.4.1 wurde gezeigt, daß das zu (p, cp) gehörende Primideal $\mathcal{P}_{p,cp}$ durch p und $\rho - cp$ erzeugt wird. Dies bedeutet, daß die Elemente

$$p,\ p \cdot \rho, \ldots,\ p \cdot \rho^{d-1} \quad \text{und}$$
$$(\rho - cp),\ (\rho - cp) \cdot \rho, \ldots,\ (\rho - cp) \cdot \rho^{d-1}$$

ein Erzeugendensystem von $\mathcal{P}_{p,cp}$ bilden.

Stellt man die Elemente dieses Erzeugendensystems so als Spaltenvektoren dar, daß der $(i+1)$-te Eintrag der Vektoren dem Koeffizienten von ρ^i entspricht, dann bilden diese eine $d \times 2d$ Matrix M mit ganzzahligen Einträgen.

Diese Matrix läßt sich durch die elementaren Spaltenoperationen

- Tauschen zweier Spalten
- Negieren einer Spalte
- Addieren ganzer Vielfachen einer Spalte zu einer anderen

in eine solche Form bringen, daß die letzten d Spalten Nullvektoren sind. Die anderen Spalten bilden eine $d \times d$ Matrix $\tilde{M}$.
Möglich ist dies deshalb, weil eine Basis von $\mathbb{Q}[\rho]$ aus d linear unabhängigen Elementen besteht und $\mathcal{P}_{p,cp} \subseteq \mathbb{Q}[\rho]$ ist. Da aber $p, p \cdot \rho, \ldots, p \cdot \rho^{d-1}$ d linear unabhängige Elemente aus $\mathcal{P}_{p,cp}$ sind und $\mathcal{P}_{p,cp}$ als Teilmenge von $\mathbb{Q}[\rho]$ höchstens d Basiselemente besitzen kann, besteht eine Basis von $\mathcal{P}_{p,cp}$ aus genau d Elementen. Somit enthält M genau d linear unabhängige Spalten.

Die Spalten der Matrix $\tilde{M}$ entsprechen also den Basiselementen von $\mathcal{P}_{p,cp}$. Mit Hilfe weiterer elementarer Spaltenoperationen läßt sich $\tilde{M}$ in die Matrix $H = (h_{ij})$ überführen, wobei für H gilt :

- $h_{ij} = 0, \quad \forall j > i,$
- $h_{ii} > 0, \quad \forall i,$
- $h_{ij} < 0 \quad \text{und} \quad |h_{ij}| < h_{ii}, \quad \forall j < i.$

H nennt man die Hermit Normalform von $\tilde{M}$ [5].

Aus den Basisvektoren $H[j]$ bestimmt man dann mit Hilfe des LLL-Reduktionsalgorithmus LLL-reduzierte Basisvektoren $R[j]$ [6]. Diese besitzen betragsmäßig relativ kleine Einträge, was zu Elementen aus $\mathcal{P}_{p,cp}$ führt, deren Normen mit großer Wahrscheinlichkeit klein sind.

[5] siehe dazu [1]
[6] vgl. dazu [11]

Für die auf diese Weise entstandenen Basisvektoren $R[j]$ berechnet man die Norm der entsprechenden Basiselemente. Beträgt eine dieser Normen p, so hat man damit einen Erzeuger von $\mathcal{P}_{p,cp}$ gefunden. Sollte keines dieser Elemente prime Norm besitzen, so bildet man kleine Linearkombinationen

$$v = \sum_{j=1}^{d} \lambda_j \cdot R[j] \quad \text{mit } \lambda_j \in \mathbb{Z},\ |\lambda_j| \text{ klein},$$

und berechnet die Norm der entsprechenden Elemente.

Da $\{R[1], \ldots, R[d]\}$ einer Basis von $\mathcal{P}_{p,cp}$ entspricht, und da $\mathcal{P}_{p,cp}$ ein Hauptideal primer Norm ist, muß es eine Linearkombination geben, die zu einem Erzeuger von $\mathcal{P}_{p,cp}$ führt.

Aus diesen Überlegungen heraus ergibt sich der folgende Algorithmus, der die nachstehenden Prozeduren benutzt.

- $vektordarstellung(\alpha)$, erzeugt den Vektor v mit den Einträgen
$$v[i+1] = \text{Koeffizient von } \rho^i \text{ in } \alpha.$$
- $vektordarstellung^{-1}(v)$, Umkehrfunktion von $vektordarstellung(\alpha)$
- $Hermit(MAT)$, bringt Matrix MAT in Hermit Normalform.
- $LLL(MAT)$, berechnet die LLL-reduzierte Matrix von MAT.
- $linkomb(MAT)$, liefert neue Linearkombination der Spaltenvektoren von MAT.

Algorithmus ERZEUGER2

Input : ein Paar (p, cp)

Output : die Erzeuger von $\mathcal{P}_{p,cp}$

```
for i ← 0 to d − 1 do
    M[i + 1] ← vektordarstellung(p · ρ^i)
    M[d + i + 1] ← vektordarstellung((ρ − cp) · ρ^i)
H ← Hermit(M)
R ← LLL(H)
for i ← 1 to d do
    π ← vektordarstellung^(−1)(R[i])
    if (N(π) = p) then output ( π ); stop
while (N(π) ≠ p)
    π ← linkomb(R)
output ( π )
stop
```

Beispiel

Für $f(x) = x^3 + 2$ erhält man mit $UK = -15$ und $OK = 15$ für $|B_1| = 500$ alle gesuchten Primidealerzeuger. So ist zum Beispiel

$$\mathcal{P}_{5,2} = (1 - \rho - \rho^2)$$
$$\mathcal{P}_{31,11} = (-1 + 4 \cdot \rho + 4 \cdot \rho^2)$$
$$\mathcal{P}_{31,24} = (-1 - 2 \cdot \rho + \rho^2)$$
$$\mathcal{P}_{31,27} = (-1 + 2 \cdot \rho^2)$$

Beispiel für ERZEUGER2:

Erzeugendensystem von $\mathcal{P}_{31,24}$: $\{31,\ 31 \cdot \rho,\ 31 \cdot \rho^2,\ (\rho - 24),\ (\rho - 24) \cdot \rho,\ (\rho - 24) \cdot \rho^2\}$.
Hieraus ergeben sich nach Hermit und LLL-Reduktion die folgenden Basiselemente :

$-1 - 2 \cdot \rho + \rho^2$ mit der Norm 31,

$-2 - \rho - 2 \cdot \rho^2$ mit der Norm -62,

$4 - 2 \cdot \rho - \rho^2$ mit der Norm 124.

Also ist $-1 - 2 \cdot \rho + \rho^2$ der gesuchte Erzeuger.

3.5 Die Einheiten

Die Einheiten in G bilden nach Satz 8 eine abelsche Gruppe E. Geht man davon aus, daß nur die trivialen Einheitswurzeln 1 und -1 in $\mathbb{Q}[\rho]$ existieren, so muß man nur die Menge $B_E = \{\epsilon_1, \ldots, \epsilon_l\}$, der in Satz 10 beschriebenen $l := r_1 + r_2 - 1$ Einheiten bestimmen, um alle auftretenden Einheiten ϵ in der Form

$$\epsilon = (-1)^e \cdot {\epsilon_1}^{e_1} \cdot \ldots \cdot {\epsilon_{l-1}}^{e_{l-1}}$$

darstellen zu können.
Dabei ist r_1 die Anzahl der reellen Wurzeln von $f(x)$ und $2 \cdot r_2$ die Anzahl der komplexen Wurzeln von $f(x)$.

Wegen $r_2 = \frac{d-r_1}{2}$ kann man l auch in der Form

$$l = r_1 + \frac{d-r_1}{2} - 1 = \frac{d+r_1}{2} - 1$$

ausdrücken.

Liegt $f(x)$ in der speziellen Form $f(x) = x^d + c$ vor, so kommen nur drei Möglichkeiten für r_1 in Betracht, nämlich

$$\begin{array}{ll} r_1 = 0, \text{ wenn } d \text{ gerade und } c > 0 & \Rightarrow l = \frac{d-2}{2}, \\ r_1 = 1, \text{ wenn } d \text{ ungerade} & \Rightarrow l = \frac{d-1}{2}, \\ r_1 = 2, \text{ wenn } d \text{ gerade und } c < 0 & \Rightarrow l = \frac{d}{2}. \end{array}$$

Definition 20

Seien $\epsilon_1, \ldots, \epsilon_m$ Einheiten aus E.
$\epsilon_1, \ldots, \epsilon_m$ heißen **multiplikativ unabhängig**, wenn aus

$${\epsilon_1}^{e_1} \cdot {\epsilon_2}^{e_2} \cdot \ldots \cdot {\epsilon_m}^{e_m} = 1$$

folgt

$$e_1 = e_2 = \ldots = e_m = 0.$$

Wählt man die zur Zerlegung von $a + b\rho$ benötigte Einheitenmenge U $= B_E \cup \{-1\}$, so kann man damit alle auftretenden Einheiten zerlegen. Doch wie findet man die Einheiten aus B_E, bzw. wie findet man überhaupt Einheiten ?

Erkennen kann man Einheiten sehr leicht an ihrer Norm, die bekanntlich 1 oder -1 betragen muß (Lemma 2). Deshalb kann man die Suche nach Einheiten während der Suche nach Primidealerzeugern im ERZEUGER1-Algorithmus miterledigen. Dazu muß man nur einen Zwischenschritt einfügen :

5.5) if ($|n| = 1$) then output ("Einheit :" α).

Auf diese Weise lassen sich viele Einheiten berechnen, doch möchte man noch jene herausfinden, die B_E bilden. Dazu betrachtet man einige Eigenschaften von B_E.

Sei $B_E = \{\epsilon_1, \ldots, \epsilon_l\}$. Man definiert die Abbildung

$$\mu : \mathbb{Q}[\rho] \to \mathbb{R}^{r_1+r_2}$$

$$\text{mit } \mu(\alpha) = (\mu_1(\alpha), \ldots, \mu_{r_1+r_2}(\alpha)),$$

wobei für $\mu_j(\alpha)$ gemäß Definition 18 und Satz 18 gilt

$$\mu_j(\alpha) := \begin{cases} \log(|\sigma_j(\alpha)|) & : \quad j = 1, \ldots, r_1 \\ \log(|\sigma_j(\alpha)|^2) & : \quad j = r_1 + 1, \ldots, r_1 + r_2, \end{cases}$$

dann gilt nach Satz 17

$$\mu(\alpha \cdot \beta) = \mu(\alpha) + \mu(\beta).$$

Sieht man die Vektoren $\mu(\epsilon_i)$ als Spaltenvektoren an, so bilden sie eine Matrix $\tilde{M}$ aus $\mathbb{R}^{l+1 \times l}$. Da $\mu(E)$ nach Satz 19 ein Gitter der Dimension $l = r_1 + r_2 - 1$ in $\mathbb{R}^{r_1+r_2}$ bildet, folgt, daß $\tilde{M}$ den Rang l besitzen muß.

Unter Anwendung von Satz 18 gilt für einen Spaltenvektor $\mu(\epsilon_i)$ von $\tilde{M}$

$$0 = \log(|N(\epsilon_i)|) = \sum_{j=1}^{l} \mu_j(\epsilon_i) + \mu_{l+1}(\epsilon_i)$$

$$\Rightarrow \mu_{l+1}(\epsilon_i) = -\sum_{j=1}^{l} \mu_j(\epsilon_i).$$

Für die Zeilenvektoren z_j der Matrix M gilt somit

$$z_{l+1} = -\sum_{j=1}^{l} z_j,$$

d.h. die letzte Zeile läßt sich als Summe der anderen darstellen, ist also linear abhängig. Streicht man sie nun aus der Matrix $\tilde{M}$, so erhält man eine $l \times l$ Matrix M, die ebenfalls den Rang l wie $\tilde{M}$ besitzt, da nur eine linear abhängige Zeile gestrichen wurde. Da $M \in \mathbb{R}^{l \times l}$ und den Rang l hat, ist M regulär und somit invertierbar. Betrachtet man die Abbildung

$$\tilde{\mu} : \mathbb{Q}[\rho] \to \mathbb{R}^{l}$$

$$\text{mit } \tilde{\mu}(\alpha) = (\mu_1(\alpha), \ldots, \mu_l(\alpha)),$$

so bilden die Spaltenvektoren $\tilde{\mu}(\epsilon_i)$ gerade die Matrix M.

Die multiplikative Abhängigkeit von l Einheiten kann man also daran erkennen, ob die von ihnen definierte Matrix M den Rang l besitzt. Um B_E zu bestimmen kann man deshalb wie folgt vorgehen. Man berechnet l Einheiten [7] und die daraus resultierende Matrix M. Die Einheiten werden in der Menge U zusammengefaßt. Aus der Matrix M entfernt man die linear abhängigen Spalten und streicht die entsprechenden Einheiten aus U. Die freigewordenenen Stellen ersetzt man durch neuberechnete Einheiten bzw. durch die daraus errechneten Spaltenvektoren. Dieses Verfahren wird solange durchgeführt bis l linear unabhängige Spalten und damit multiplikativ unabhängige Einheiten gefunden wurden.

[7] ERZEUGER1-Algorithmus

Der aus diesen Überlegungen resultierende Algorithmus zur Bestimmung von B_E benutzt die Prozeduren

- $nexteinheit()$, liefert eine noch nicht betrachtete Einheit
- $linabhspalte(M)$, liefert eine linear abhängige Spalte der Matrix M

Algorithmus EINHEITEN

Input : Anzahl der reellen Wurzeln r
Grad d von $f(x)$

Output : B_E

$l \leftarrow \frac{d+r}{2} - 1$
for $i \leftarrow 1$ to l do
    $B_E[i] \leftarrow nexteinheit()$
    $M[i] \leftarrow \tilde{\mu}(B_E[i])$
while $(rang(M) \neq l)$ do
    $j \leftarrow linabhspalte(M)$
    $B_E[j] \leftarrow nexteinheit()$
    $M[j] \leftarrow \tilde{\mu}(B_E[j])$
output ($B_E[1], \ldots, B_E[l]$)
stop

3.6 Die Zerlegung

Als Voraussetzung sei im weiteren gegeben, daß G= $\mathbb{Z}[\rho]$ ein ZPE-Ring ist.
Das Kernstück des 'Number Field Sieve' ist die Zerlegung von

$$a + bm \quad \text{über } B_1$$

und

$$a + b\rho \quad \text{über } B_2 \cup U$$

mit $ggT(a, b) = 1$.

Definition 21
Paare $(a, b) \in \mathbb{Z}^2$ mit

$$(3.6.1) \qquad a + bm = \prod_{p \in B_1} p^{\nu_p},$$

$$(3.6.2) \qquad a + b\rho = \prod_{\pi \in B_2} \pi^{\mu_\pi} \cdot \epsilon \ , \ \epsilon \in E, \qquad \text{und}$$

$$(3.6.3) \qquad ggT(a, b) = 1$$

heißen **gute Paare**.

Schritt 5 des 'Number Field Sieve' -Algorithmus verlangt die Suche nach genügend vielen dieser guten Paare. Dadurch ergeben sich mehrere Probleme :

- Wie findet man ein gutes Paar (a, b) ?
- Wie läßt sich $a + bm$ über B_1 zerlegen ?
- Wie sieht die Zerlegung von $a + b\rho$ über $B_2 \cup U$ aus ?

Die Suche nach Paaren $(a, b) \in \mathbb{Z}^2$ mit den Eigenschaften (3.6.1) bis (3.6.3) erstreckt sich über eine vorgegebene Teilmenge von $\mathbb{Z}^2$. Die möglichen Werte von a und b werden dabei durch obere und untere Schranken begrenzt:

$$A_{min} \leq a \leq A_{max},$$

$$B_{min} \leq b \leq B_{max}.$$

Dabei läßt man für a auch negative Werte zu, während b nur aus der Menge der natürlichen Zahlen gewählt wird. Begründet ist diese Wahl in der Tatsache, daß die Paare (a, b) und $(-a, -b)$ die gleiche Kongruenz der Form (3.1.6) liefern.

Erzeugt nämlich (a, b) die Kongruenz

$$(3.6.4) \qquad a + bm = \prod_{i=1}^{k} b_i^{\nu_i} \equiv \prod_{i=1}^{k} b_i^{\mu_i} \equiv \prod_{\pi \in B_2} \Phi(\pi)^{\mu_\pi} \cdot \prod_{\epsilon \in U} \Phi(\epsilon)^{\lambda_\epsilon} \qquad \text{mod } n$$

und damit die Kongruenz

$$\prod_{i=1}^{k} b_i^{\nu_i - \mu_i} \equiv 1 \qquad \text{mod } n$$

der Form (3.1.6), so erzeugt $(-a, -b)$, da

$$-a - bm = -(a + bm) \qquad \text{und}$$

$$-a - b\rho = -(a + b\rho),$$

die Kongruenz

$$-\prod_{i=1}^{k} b_i^{\nu_i} \equiv -\prod_{i=1}^{k} b_i^{\mu_i} \qquad \text{mod } n$$

und somit ebenfalls

$$\prod_{i=1}^{k} b_i^{\nu_i - \mu_i} \equiv 1 \qquad \text{mod } n.$$

Also ergeben Paare mit negativem b keine Kongruenzen, die nicht auch von Paaren mit positivem b erzeugt werden könnten. Man kann deshalb $B_{min} > 0$ voraussetzen. Da für kleine b $a + bm$ relativ klein bleibt und somit die Wahrscheinlichkeit der Zerlegung über B_1 relativ groß wird, wählt man $B_{min} = 1$. Wie man die anderen Schranken $B_{max}, A_{min}, A_{max}$ wählen soll, wird im Kapitel 6 behandelt.

Eine Möglichkeit der Überprüfung, ob ein gegebenes Paar (a, b) gut ist, besteht im einfachen Nachrechnen der Eigenschaften (3.6.1) bis (3.6.3). Zuerst kontrolliert man, ob a, b zueinander relativ prim sind. Die Notwendigkeit a, b teilerfremd zu wählen, liegt darin , daß, wenn

$$a = g \cdot \tilde{a} \quad \text{und} \quad b = g \cdot \tilde{b} \quad \text{mit } g = ggT(a, b),$$

(a, b) und $(\tilde{a}, \tilde{b})$ die gleiche Kongruenz der Form (3.1.6) erzeugen. Denn erzeugt $(\tilde{a}, \tilde{b})$ analog zu (3.6.4) die Kongruenz

$$\prod_{i=1}^{k} b_i{}^{\nu_i - \mu_i} \equiv 1 \qquad \text{mod } n,$$

so erzeugt (a, b), da

$$a + bm = g \cdot (\tilde{a} + \tilde{b}m) \qquad \text{und}$$

$$a + b\rho = g \cdot (\tilde{a} + \tilde{b}\rho),$$

die Kongruenz

$$g \cdot \prod_{i=1}^{k} b_i^{\nu_i} \equiv g \cdot \prod_{i=1}^{k} b_i^{\mu_i} \qquad \text{mod } n$$

und somit für $ggT(g, n) = 1$ ebenfalls

$$\prod_{i=1}^{k} b_i^{\nu_i - \mu_i} \equiv 1 \mod n.$$

Man kann also auf Paare (a, b) mit $ggT(a, b) \neq 1$ verzichten.

Besteht ein Paar (a, b) die '$ggT-$Probe', so muß die Zerlegbarkeit von $a + bm$ über B_1 getestet werden. Dies ist mittels Probedivision möglich, d.h. für alle Primzahlen p_i aus der Basis B_1 überprüft man, ob p_i $a + bm$ teilt. Wenn dies der Fall ist, so wird $a + bm$ durch die höchste in $a + bm$ aufgehende Potenz von p_i dividiert.

Hat man auf diese Weise alle Primfaktoren aus B_1 von $a + bm$ herausdividiert, so ist $a + bm$ genau dann über B_1 zerlegbar, wenn nur noch 1 oder -1 übrig geblieben ist.

Algorithmus ABM

Input : $a, b, m, B_1 = \{p_1, \ldots, p_k\}$

Output : Array $E[1..k]$ ($E[i]$ ist der maximale Exponent von p_i, der in der Zerlegung von $a + bm$ vorkommt.)
WAHR , wenn $a + bm$ über B_1 zerlegbar ist
FALSCH , sonst

```
n ← |a + bm|
for j ← 1 to k do
    E[j] ← 0
    while (n ≡ 0 mod p_j)
        n ← n/p_j
        E[j] ← E[j] + 1
if (n = 1) then output ( E , WAHR )
            else output( E , FALSCH );
stop
```

Erfüllt ein Paar (a, b) neben dem ggT-Test auch noch die Bedingung der Zerlegbarkeit von $a + bm$ über B_1, so muß die Eigenschaft (3.6.2) überprüft werden.

Es ist allerdings nur notwendig zu testen, ob $a + b\rho$ so durch Elemente aus B_2 teilbar ist, daß nur noch eine Einheit übrig bleibt. Denn geht man davon aus, daß die Elemente von U eine Basis von E bilden[8], so kann man die bei der Zerlegung berechnete Einheit natürlich über U zerlegen.

Also bleibt zu prüfen, welche Primelemente in welcher Potenz $a + b\rho$ teilen.
Wenn $a + b\rho$ über B_2 bis auf eine Einheit zerfällt, so läßt sich $a + b\rho$ schreiben als

$$a + b\rho = \prod_{\pi \in B_2} \pi^{\mu_\pi} \cdot \epsilon\,, \qquad \epsilon \in E \text{ Einheit.}$$

Aus der Multiplikativität der Norm ergibt sich dann

$$N(a + b\rho) = N\left(\prod_{\pi \in B_2} \pi^{\mu_\pi} \cdot \epsilon\right) = \prod_{\pi \in B_2} N(\pi^{\mu_\pi}) \cdot N(\epsilon) = \pm \prod_{\pi \in B_2} N(\pi)^{\mu_\pi}.$$

Dies bedeutet :

(3.6.5) Ist $a + b\rho$ über B_2 zerlegbar, dann zerfällt $|N(a + b\rho)|$ über den Normen der Primelemente aus B_2.

Auch die Umkehrung gilt, denn

Lemma 13

Seien $f(x)$, ρ und $\mathbb{Q}[\rho]$ wie in Kapitel 3.1 definiert und sei G=$\mathbb{Z}[\rho] \subseteq \mathbb{Q}[\rho]$ die Menge der ganzen algebraischen Zahlen in $\mathbb{Q}[\rho]$.
Sei $g(x) \in \mathbb{Z}[x]$ mit $g(x)|f(x) \bmod p$, $p \in \mathbb{P}$, und seien weiter $a, b \in \mathbb{Z}$ mit $ggT(a, b) = 1$.
Ist $grad(g) > 1$, dann gilt $a + b\rho \notin \mathcal{P} = (p, g(\rho))$.

Beweis :

Annahme: $a + b\rho \in \mathcal{P}$

$$\Rightarrow \exists \alpha(\rho), \beta(\rho) \in \mathbb{Z}[\rho] \text{ mit}$$

$$\text{(3.6.6)} \qquad \mathbb{Z}[\rho] \ni a + b\rho = \alpha(\rho) \cdot p + \beta(\rho) \cdot g(\rho)$$

Betrachtet man den natürlichen Homomorphismus

$$\phi : \mathbb{Z}[\rho] \to \mathbb{Z}[x]/f(x)\mathbb{Z}[x]$$

$$\text{mit } \phi(\alpha(\rho)) \mapsto \alpha(x) \bmod f(x),$$

dann folgt aus (3.6.6)

$$\text{(3.6.7)} \qquad a + bx \equiv \alpha(x) \cdot p + \beta(x) \cdot g(x) \qquad \bmod f(x)$$

[8]so wurde U in Kapitel 3.5 konstruiert

Wendet man hierauf den natürlichen Homomorphismus

$$\psi: \mathbb{Z}[x]/f(x)\mathbb{Z}[x] \to (\mathbb{Z}/p\mathbb{Z}[x])/f(x)(\mathbb{Z}/p\mathbb{Z}[x])$$
$$\text{mit } h(x) \bmod f(x) \mapsto h(x) \mod f(x), p$$

an, so gilt

$$a + bx \equiv \alpha(x) \cdot \underbrace{p}_{\equiv 0} + \beta(x) \cdot g(x) \mod f(x), p\ , \tag{3.6.8}$$

also

$$a + bx \equiv \beta(x) \cdot g(x) \mod f(x), p. \tag{3.6.9}$$

Da nach Voraussetzung $g(x)|f(x) \bmod p$, gilt auch die Kongruenz

$$a + bx \equiv \beta(x) \cdot g(x) \mod g(x), p\ , \tag{3.6.10}$$

also

$$a + bx \equiv 0 \mod g(x), p$$

Dies ist aber nur dann möglich, wenn

$$a + bx \equiv 0 \bmod g(x) \quad \text{oder} \quad a + bx \equiv 0 \bmod p\ .$$

Aus $a + bx \equiv 0 \bmod g(x)$ folgt aber

$$\exists q(x) \in \mathbb{Z}[x] \quad \text{mit } g(x) \cdot q(x) = a + bx.$$

Da aber $grad(g(x)) > 1$ und $grad(a + bx) = 1$, kann es ein solches $q(x)$ nicht geben.
Also muß $a + bx \equiv 0 \bmod p$ gelten und somit

$$p|a \quad \text{und} \quad p|b.$$

Also auch $p|ggT(a, b)$, was allerdings ein Widerspruch zu $ggT(a, b) = 1$ ist. Also gilt die Behauptung. □

Dieses Lemma zeigt also, daß Zahlen der Form $a + b\rho$ mit $ggT(a, b) = 1$ in keinem Primideal der Norm p^k, $k > 1$, liegen und somit auch von keinem Primelement dieser Norm geteilt werden können. Dies bedeutet, daß alle Primelemente, die $a + b\rho$ teilen, prime Norm besitzen müssen, d.h. bis zu einer gewissen Größe der Norm findet man alle potentiellen Primteiler von $a + b\rho$ in B_2. Im Fall, daß die Norm von $a + b\rho$ über die Normen der Primelemente aus B_2 zerfällt, bedeutet das, daß $a + b\rho$ über B_2 zerfällt.

Da zu jedem $\pi \in B_2$ mit

$$\pi | a + b\rho \Leftrightarrow a + b\rho \in (\pi)$$

nach Konstruktion von B_2 ein Paar (p, cp) mit $(\pi) = \mathcal{P}_{p,cp}$ existieren muß, folgt aus Lemma 11

$$\pi | a + b\rho \Leftrightarrow \varphi_{cp}(a + b\rho) \equiv 0 \mod p.$$

Allerdings gibt es zu jedem $p \in B_1$ höchstens ein cp mit $\varphi_{cp}(a + b\rho) \equiv 0 \bmod p$, denn

Lemma 14

Seien $a, b \in \mathbb{Z}$, $ggT(a, b) = 1$, $p \in \mathbb{P}$ und $cp \not\equiv \tilde{cp} \bmod p$.
Dann gilt : $\varphi_{cp}(a + b\rho) \equiv 0 \bmod p \Rightarrow \varphi_{\tilde{cp}}(a + b\rho) \not\equiv 0 \bmod p$.

<u>*Beweis*</u> :

Annahme: $\varphi_{\tilde{cp}}(a + b\rho) \equiv 0 \bmod p$

$$\varphi_{cp}(a + b\rho) \equiv 0 \bmod p \quad \Rightarrow \quad a + b \cdot cp \equiv 0 \bmod p \quad \Rightarrow \quad b \cdot cp \equiv -a \bmod p$$

$$\varphi_{\tilde{cp}}(a + b\rho) \equiv 0 \bmod p \quad \Rightarrow \quad a + b \cdot \tilde{cp} \equiv 0 \bmod p \quad \Rightarrow \quad b \cdot \tilde{cp} \equiv -a \bmod p$$

$$\Rightarrow b \cdot cp \equiv b \cdot \tilde{cp} \bmod p$$

1.Fall $ggT(b, p) \neq 1 \Rightarrow p|b$, da p Primzahl

$$\Rightarrow b \cdot cp \equiv b \cdot \tilde{cp} \equiv 0 \bmod p$$

$$\Rightarrow a + \underbrace{b \cdot cp}_{\equiv 0} \equiv 0 \bmod p$$

$$\Rightarrow a \equiv 0 \bmod p$$

$$\Rightarrow p|a$$

$$\Rightarrow p|a, b$$

$\Rightarrow p|ggT(a, b)$, was ein Widerspruch zu $ggT(a, b) = 1$ ist.

2.Fall $ggT(b, p) = 1$

$$\Rightarrow b \cdot cp \equiv b \cdot \tilde{cp} \bmod p$$

$\Rightarrow cp \equiv \tilde{cp} \bmod p$, was ein Widerspruch zur Voraussetzung ist.

Also gilt die Behauptung. $\square$

Nach Aussage dieses Lemmas wird $a + b\rho$ also höchstens von einem Primideal (π) der Norm p geteilt.
Da nach Lemma 13 nur Primelemente primer Norm $a + b\rho$ teilen können, folgt schließlich

Folgerung 4

Seien $a, b \in \mathbb{Z}$ mit $ggT(a, b) = 1$ und sei $\pi \in \mathbb{Z}[\rho]$ mit $|N(\pi)| = p$.
Teilt π $a + b\rho$ und geht p^k in $N(a + b\rho)$ auf, so teilt π^k $a + b\rho$.

Aus den obigen Überlegungen ergibt sich ein Algorithmus zur Berechnung der Zerlegung von $a + b\rho$ über B_2.

Algorithmus ABRHO

Input : a, b
alle Paare (p, cp) und ihre korrespondierenden Primidealerzeuger $\pi_{p,cp}$

Output : Array $E[p, cp]$ ($E[p, cp]$ maximale Potenz von $\pi_{p,cp}$, die in $a + b\rho$ aufgeht.)
WAHR , wenn $a + b\rho$ über B_2 zerlegbar ist
FALSCH , sonst

$n \leftarrow |N(a + b\rho)|$
for all (p, cp) do
  $E[p, cp] \leftarrow 0$
  if $(\varphi_{cp}(a + b\rho) \equiv 0 \bmod p$) then
    while $(n \equiv 0 \bmod p)$ do
      $n \leftarrow n/p$
      $E[p, cp] \leftarrow E[p, cp] + 1$
  if $(n = 1)$ then output (E , WAHR); stop
output (E , FALSCH)
stop

Mit Hilfe dieses Algorithmus kann man die Eigenschaft (3.6.2) für ein Paar (a, b) überprüfen. Verbindet man den ggT-Test mit den beiden Algorithmen ABM und ABRHO, so erhält man ein Verfahren zum Testen der 'gut'-Eigenschaft eines Paares (a, b).

Algorithmus ABGUT

Input	:	a, b
		Primzahlbasis B_1
		Liste PCP der Paare (p, cp)
Output	:	WAHR , wenn (a, b) gut
		FALSCH , sonst

```
1 .) if (ggT(a,b) = 1) then
2 .)     if (ABM(a,b,B_1) = (*,WAHR)) then
3 .)         if (ABRHO(a,b,PCP) = (*,WAHR)) then
4 .)             output ( WAHR ); stop
5 .) output ( FALSCH );
6 .) stop
```

Wendet man ABGUT auf alle Paare (a, b) mit $A_{min} \leq a \leq A_{max}$, $1 \leq b \leq B_{max}$ an, so erhält man alle guten Paare aus diesem Bereich. Allerdings hat dieses Verfahren den großen Nachteil, daß für jedes neue Paar (a, b) die Berechnung von vorne begonnen werden muß, da man keinen Nutzen aus vorherigen Berechnungen ziehen kann. Dies schlägt dann in der Laufzeit des Algorithmus negativ zu Buche. Deshalb ist dieser Art, gute Paare zu berechnen, eine andere vorzuziehen. Bei dieser neuen Methode handelt es sich um ein Siebverfahren[9], also ein Verfahren, das, ausgehend von einer gewissen Grundmenge, nacheinander die Elemente aussortiert, die eine der Eigenschaften (3.6.1) bis (3.6.3) nicht erfüllen.

[9]daher auch der Name 'Number Field Sieve'

Die angesprochene Grundmenge M_b ist hier die Menge aller Paare (a,b) mit $a \in I_a = [A_{min}, A_{max}] \subseteq \mathbb{Z}$ und $b \in I_b = [1, B_{max}] \subseteq \mathbb{Z}$. Für ein festes b, sind nun die $a \in I_a$ gesucht, für die (a,b) gut ist.
Sei $GUT[A_{min}, \ldots, A_{max}]$ ein Array, vorbesetzt mit

$$GUT[a] \leftarrow \text{WAHR}.$$

$GUT[a]$ =FALSCH bedeutet, daß das Paar (a,b) mindestens eine der drei Eigenschaften (3.6.1) bis (3.6.3) nicht erfüllt.

Nacheinander werden nun die drei Eigenschaften überprüft, allerdings nicht mehr wie früher für jedes a einzeln, sondern für das gesamte Intervall I_a.

Das ggT–Sieb

Sei $b = {q_1}^{e_1} \cdot \ldots \cdot {q_k}^{e_k}$, $q_i \in \mathbb{P}$, $e_i > 0$.
Dann gilt

$$\begin{aligned} ggT(a,b) = 1 &\Leftrightarrow ggT(a,q_i) = 1, \qquad \forall i = 1,\ldots,k \\ \text{(3.6.11)} \qquad &\Leftrightarrow q_i \not| a, \qquad \forall i = 1,\ldots,k, \end{aligned}$$

d.h. $GUT[a]$ wird genau dann auf FALSCH gesetzt, wenn (3.6.11) für mindestens ein q_i nicht erfüllt ist.
Setzt man $r := A_{min} \bmod q_i$, $0 \le r \le q_i$, $i \in \{1,\ldots,k\}$, so kann man daraus das kleinste Element $a_0 \in I_a$ berechnen, für das $q_i | a_0$ gilt, denn

$$\begin{aligned} r = 0 &\Rightarrow q_i | A_{min} \\ &\Rightarrow a_0 := A_{min} \quad \text{erfüllt (3.6.11) nicht} \\ r \neq 0 &\Rightarrow a_0 := A_{min} + q_i - r \quad \text{ist der gesuchte Wert,} \\ &\quad \text{denn } a_0 \equiv A_{min} + q_i - r \mod q_i \\ &\qquad\qquad \equiv r - r \mod q_i \\ &\qquad\qquad \equiv 0 \mod q_i \\ &\qquad\quad \Rightarrow q_i | a_0. \\ &\quad \text{Da } a_0 - A_{min} = q_i - r < q_i, \\ &\qquad\quad \Rightarrow a_0 \text{ besitzt die geforderte Minimaleigenschaft.} \end{aligned}$$

Der nächst größere Wert a_1 mit $q_i | a_1$ ist $a_0 + q_i$ und der darauffolgende $a_2 = a_0 + 2 \cdot q_i$, u.s.w. Das heißt, von der ersten Stelle a_0 aus kann man alle anderen a_j, mit $q_i | a_j$, durch Addieren der Vielfachen von q_i erreichen. Es gilt also die Formel :

$$\text{(3.6.12)} \qquad a_i = a_{i-1} + p = a_0 + i \cdot p, \quad \forall i = 0, \ldots$$

Aus diesen Betrachtungen ergibt sich der GGT-SIEB-Algorithmus. Dieser benutzt eine Prozedur $nextdivisor(b)$, welche beim i-ten Aufruf den i-ten Primteiler q_i von b liefert. Sind alle Primteiler ausgegeben worden, so liefert $nextdivisor(b)$ 1.

Algorithmus GGT-SIEB

Input : A_{min}, A_{max}, b

Output : Array $GUT[A_{min}, \ldots, A_{max}]$ mit
$GUT[a]$= WAHR, wenn $ggT(a, b) = 1$
= FALSCH, sonst

```
for a ← A_min to A_max do
    GUT[a] ← WAHR
q_i ← nextdivisor(b)
while (q_i ≠ 1) do
    r ← A_min mod q_i
    if (r ≠ 0) then r ← q_i − r
    r ← r + A_min
    for a ← r to A_max step q_i do
        GUT[a] ← FALSCH
    q_i ← nextdivisor(b)
output ( GUT )
stop
```

Das $a + bm$-Sieb

Ausgehend von der Primzahlbasis $B_1 =\{p_1, \dots, p_k\}$ sollen die Werte für $a \in I_a$ aussortiert werden, für die $a + bm$ nicht über B_1 zerlegbar ist.

Dazu betrachtet man das Array $A[A_{min}, \dots, A_{max}]$, das mit

$$A[a] := |a + bm|,$$

für alle $a \in I_a$, vorbesetzt wurde. Für eine Primzahl $p_i \in B_1$ sucht man nun, ähnlich wie im GGT-Sieb-Algorithmus, die erste Stelle $a_0 \in I_a$, für die gilt :

$$p_i | A[a_0].$$

Dazu bildet man

$$r := \underbrace{A[A_{min}]}_{A_{min}+bm} \bmod p_i \ , \ 0 \le r < p_i.$$

$$\begin{aligned}
\text{Ist } r = 0 &\Rightarrow p_i | A[A_{min}] \\
&\Rightarrow a_0 := A_{min} \text{ ist die gesuchte Stelle.} \\
\text{Ist } r \neq 0 &\Rightarrow a_0 := A_{min} + p_i - r \text{ ist der gesuchte Wert,} \\
&\quad \text{denn } A[a_0] \equiv A[A_{min} + p_i - r] \quad \bmod p_i \\
&\qquad\qquad \equiv \underbrace{A_{min} + bm}_{r} - r + \underbrace{p_i}_{0} \quad \bmod p_i \\
&\qquad\qquad \equiv 0 \quad \bmod p_i \\
&\qquad \Rightarrow p_i | A[a_0]. \\
&\quad \text{Da } a_0 - A_{min} = p_i - r < p_i \\
&\qquad \Rightarrow a_0 \text{ besitzt also die geforderte Minimaleigenschaft.}
\end{aligned}$$

$A[a_0]$ wird nun durch $A[a_0]/{p_i}^t$ ersetzt, wobei ${p_i}^t$ die höchste Potenz von p_i ist, die in $A[a_0]$ aufgeht.

Der nächst größere Wert a_1 mit $p_i | A[a_1]$ ist $a_0 + p_i$ und der darauffolgende $a_2 = a_0 + 2 \cdot p_i$, u.s.w., denn

$$A[a_0 + p_i] = a_0 + p_i + bm = \underbrace{a_0 + bm}_{\equiv 0} + p_i \equiv 0 \bmod p_i$$

Das bedeutet, daß man auch hier von der ersten Stelle a_0 aus alle anderen a_j, mit $p_i | A[a_j]$, durch Addieren der Vielfachen von p_i erreichen kann. Man erhält also wiederum die gleiche Formel (3.6.12) für a_i.

Wurde auf diese Weise das gesamte Intervall I_a für p_i durchgearbeitet, so führt man das gleiche Verfahren mit p_{i+1} aus. Wenn die Primzahlbasis B_1 vollständig abgearbeitet wurde, sind genau die Zahlen $a + bm$ über B_1 zerlegbar, für die gilt

$$A[a] = 1.$$

Betrachtet man das im GGT-SIEB-Algorithmus erarbeitete Array GUT, so kann man dort an den Stellen a mit $A[a] \neq 1$ FALSCH einsetzen.

Algorithmus ABMSIEB

Input : $B_1 = \{p_1, \ldots, p_k\}$, A_{min}, A_{max}, b, m, GUT
Output : GUT (bearbeitet)

1 .) for $a \leftarrow A_{min}$ to A_{max} do
2 .) $\quad A[a] \leftarrow |a + bm|$
3 .) for $i \leftarrow 1$ to k do
4 .) $\quad r \leftarrow A[A_{min}] \bmod p_i$
5 .) $\quad$ if $(r \neq 0)$ then $r \leftarrow p_i - r$
6 .) $\quad r \leftarrow A_{min} + r$
7 .) $\quad$ for $a \leftarrow r$ to A_{max} step p_i do
8 .) $\qquad$ while $(A[a] \equiv 0 \bmod p_i)$ do
9 .) $\qquad\quad A[a] \leftarrow A[a]/p_i$
10 .) for $a \leftarrow A_{min}$ to A_{max} do
11 .) $\quad$ if $(A[a] \neq 1)$ then $GUT[a] \leftarrow$ FALSCH
12 .) output (GUT)
13 .) stop

Dieser Algorithmus läßt sich mit Hilfe einer Methode, die für die meisten $a \in I_a$ ohne Schritt 8 auskommt, noch verbessern.

Sei dazu l_{I_a} die Länge des Intervalls I_a. Dann erweitert man B_1 zu $\tilde{B}_1$ derart, daß

$$\tilde{B}_1 := \{{p_i}^{t_i} \mid p_i \in B_1,\ t_i \in \mathbb{N} \text{ mit } t_i < \log_{p_i}(l_{I_a}) + 1\}.$$

Sei $m_i := \lfloor \log_{p_i}(l_{I_a}) + 1 \rfloor$ der maximale Exponent von p_i mit ${p_i}^{m_i} \in \tilde{B}_1$, dann findet man für eine Primzahl $p_i \in B_1$ die Elemente a aus I_a mit $p_i | a + bm$ auf folgende Weise:

Hat man die p_i-Potenzen ${p_i}^j$, $j = 1, \ldots, m_i$, berechnet, so findet man die Stellen a mit ${p_i}^j | A[a]$ analog zum ABMSIEB-Algorithmus, d.h. man berechnet die erste Stelle a_0, an der ${p_i}^j | A[a_0]$ und findet die weiteren durch Aufaddieren von ${p_i}^j$. Der Unterschied zu obigem Verfahren besteht nun darin, daß man an diesen Stellen den Arrayeintrag nicht durch ${p_i}^j$ dividiert, sondern nur durch p_i. Denn ist $A[a]$ durch ${p_i}^j$ teilbar, dann auch durch ${p_i}^t$ mit $t = 1, \ldots, j-1$. Für jeden dieser Exponenten t wurde $A[a]$ aber schon einmal durch p_i dividiert, also insgesamt durch ${p_i}^{j-1}$. Teilt man nun $A[a]$ wieder durch p_i, so wurde wie gewünscht $A[a]$ durch ${p_i}^j$ dividiert.

Ist $j = m_i$, so kann man maximal an einer Stelle a aus dem Intervall I_a $A[a]$ durch ${p_i}^{m_i}$ teilen, denn die nächste in Frage kommende Stelle wäre $a + {p_i}^{m_i}$, welche aber nicht mehr im Intervall I_a liegt, da nach Konstruktion ${p_i}^{m_i} \geq l_{I_a}$. An dieser einzigen verbleibenden Stelle a muß man nun solange wie möglich durch p_i dividieren, da $A[a]$ durch eine noch höhere Potenz von p_i als ${p_i}^{m_i}$ teilbar sein kann.

Hieraus ergibt sich die zweite Version des ABMSIEB-Algorithmus.

Algorithmus ABMSIEB2

Input : $\tilde{B}_1$, A_{min}, A_{max}, b, m, GUT
Output : GUT (bearbeitet)

for $a \leftarrow A_{min}$ to A_{max} do
    $A[a] \leftarrow |a + bm|$
for $i \leftarrow 1$ to k do
    for $j \leftarrow 1$ to m_i do
        $r \leftarrow A[A_{min}] \bmod {p_i}^j$
        if $(r \neq 0)$ then $r \leftarrow {p_i}^j - r$
        $r \leftarrow A_{min} + r$
        for $a \leftarrow r$ to A_{max} step ${p_i}^j$ do
            $A[a] \leftarrow A[a]/p_i$
        if $(j = m_i$ und $r \leq A_{max})$ then
            while $(A[r] \equiv 0 \bmod p_i)$ do
                $A[a] \leftarrow A[a]/p_i$
for $a \leftarrow A_{min}$ to A_{max} do
    if $(A[a] \neq 1)$ then $GUT[a] \leftarrow$ FALSCH
output (GUT)
stop

Das $a + b\rho$-Sieb

Analog zum $a+bm$-Sieb kann man die Paare (a, b) aussieben, für die $a+b\rho$ nicht über $B_2 \cup E$ zerlegbar ist.

Dazu benutzt man wieder das Array $A[A_{min}, \ldots, A_{max}]$. Diesmal wird $A[a]$ allerdings mit $|N(a+b\rho)|$ vorbesetzt. Nach (3.6.5) entspricht ja die Zerlegung von $a+b\rho$ über B_2 der Zerlegung von $|N(a+b\rho)|$ über den Normen der Primelemente aus B_2. Weiter existiert bekanntlich zu jedem $\pi \in B_2$ ein Paar (p, cp), so daß $(\pi) = \mathcal{P}_{p,cp} = (p, \rho - cp)$, und es gilt

$$\pi | a + b\rho \Leftrightarrow a + b \cdot cp \equiv 0 \bmod p.$$

Sei $\pi \in B_2$ und (p, cp) das zu π korrespondierende Paar. Gesucht wird wieder die erste Stelle a_0 im Intervall I_a, für die $\pi|(a+b\rho)$ und damit auch

$$\underbrace{N(\pi)}_{=p} \;|\; \underbrace{A[a_0]}_{=|N(a+b\rho)|}$$

gilt. Setzt man

$$r := A_{min} + b \cdot cp \bmod p,$$

dann gilt

$$\begin{aligned}
r = 0 &\Rightarrow \pi|(A_{min} + b\rho) \\
&\Rightarrow a_0 := A_{min} \text{ ist die gesuchte Stelle.} \\
r \neq 0 &\Rightarrow a_0 := A_{min} + p - r \text{ ist der gesuchte Wert,} \\
&\quad \text{denn } a_0 + b \cdot cp \equiv A_{min} + p - r + b \cdot cp \quad \bmod p \\
&\qquad\qquad\qquad \equiv \underbrace{A_{min} + b \cdot cp}_{r} - r + \underbrace{p}_{0} \quad \bmod p \\
&\qquad\qquad\qquad \equiv 0 \quad \bmod p \\
&\qquad\quad \Rightarrow \pi|(a_0 + b\rho). \\
&\quad \text{Da } a_0 - A_{min} = p - r < p \\
&\qquad\quad \Rightarrow a_0 \text{ besitzt die geforderte Minimaleigenschaft.}
\end{aligned}$$

Analog zum $a+bm$-Sieb ersetzt man $A[a_0]$ durch $A[a_0]/p^t$, wobei p^t die höchste Potenz von p ist, die in $A[a_0]$ aufgeht.

Die nächste Stelle a_1 mit $p|(a_1 + b \cdot cp)$ ist $a_0 + p$, denn

$$a_1 + b \cdot cp \equiv \underbrace{p}_{0} + \underbrace{a_0 + b \cdot cp}_{0} \equiv 0 \quad \bmod p.$$

Induktiv fortgesetzt erhält man für a_i wiederum die Rekursionsformel (3.6.12) :

$$a_i = a_{i-1} + p = a_0 + i \cdot p.$$

Führt man dieses Verfahren für alle $\pi \in B_2$ durch, so sind genau die Elemente $a + b\rho$ über B_2 bis auf eine Einheit zerlegbar, für die $A[a] = 1$ gilt. Man erhält also einen zum ABMSIEB analogen Algorithmus.

Algorithmus ABRHOSIEB

Input : alle Paare (p, cp), A_{min}, A_{max}, b, m, GUT

Output : GUT (bearbeitet)

for $a \leftarrow A_{min}$ to A_{max} do
$A[a] \leftarrow |N(a + b\rho)|$
for all (p, cp) do
$r \leftarrow A_{min} + b \cdot cp \bmod p$
if $(r \neq 0)$ then $r \leftarrow p - r$
$r \leftarrow A_{min} + r$
for $a \leftarrow r$ to A_{max} step p do
while $(A[a] \equiv 0 \bmod p)$ do
$A[a] \leftarrow A[a]/p$
for $a \leftarrow A_{min}$ to A_{max} do
if $(A[a] \neq 1)$ then $GUT[a] \leftarrow$ FALSCH
output (GUT)
stop

Mit Hilfe der oben beschriebenen Algorithmen läßt sich nun leicht das Verfahren zum Testen der Paare (a, b), für $b \in \{1, \ldots, B_{max}\}$, b fest, und a innerhalb der vorgegebenen Grenzen, angeben :

Algorithmus ABGUTSIEB

Input : A_{min}, A_{max}, b, m
B_1, (p, cp)-Liste

Output : alle guten Paare (a, b)

1 .) $GUT \leftarrow GGTSIEB(A_{min}, A_{max}, b)$
2 .) $GUT \leftarrow ABMSIEB(B_1, A_{min}, A_{max}, GUT)$
3 .) $GUT \leftarrow ABRHOSIEB$ ((p, cp)-Liste, A_{min}, A_{max}, GUT)
4 .) for $a \leftarrow A_{min}$ to A_{max} do
5 .) if ($GUT[a]$ = WAHR) then output ((a, b))
6 .) stop

Hat man mit Hilfe des Algorithmus ABGUTSIEB ein gutes Paar (a, b) gefunden, so weiß man, daß einerseits $a + bm$ über B_1 und andererseits $a + b\rho$ über $B_2 \cup U$ zerlegbar ist. Wie diese Zerlegung explizit aussieht, ist an dieser Stelle noch nicht bekannt. Wendet man allerdings auf ein gutes Paar die Algorithmen ABM und ABRHO an, so erhält man sowohl die Zerlegung von $a + bm$ über B_1 als auch die Zerlegung von $a + b\rho$ über B_2. Man muß also nur noch die in den Zerlegungen unberücksichtigten Einheiten berechnen.

Für $a + bm$ findet man diese Einheit sehr schnell. Sie ist nämlich 1, wenn $(a + bm) \geq 0$, und -1 sonst. In der Praxis allerdings wird m gegenüber a und b so groß sein, daß lediglich die 1 als Einheit vorkommt.

Im Fall $a + b\rho$ verhält es sich mit den Einheiten nicht mehr so einfach:
Sei $\tilde{\nu_\pi}$ der Exponent von π, den ABRHO angewandt auf (a, b) liefert, dann definiert man

$$\alpha := \prod_{\pi \in B_2} \pi^{\tilde{\nu_\pi}}.$$

Somit gilt

$$a + b\rho = \alpha \cdot \epsilon \qquad \text{mit } \epsilon \in E,$$

also

$$\epsilon = \tfrac{a+b\rho}{\alpha}.$$

In welcher Potenz die Basisvektoren aus U in ϵ vorkommen, berechnet man mittels der in Kapitel 3.5 beschriebenen regulären Matrix M. Man muß dazu das Gleichungssystem

$$(3.6.13) \qquad M \cdot x = b$$

mit $b = \tilde{\mu}(\epsilon)$ lösen. Da M regulär ist, gilt

$$x = M^{-1} \cdot b.$$

Die i-te Komponente $x[i]$ liefert dann die Exponenten e_i der Darstellung

$$\epsilon = {\epsilon_0}^{e_0} \cdot {\epsilon_1}^{e_1} \cdot \ldots \cdot {\epsilon_l}^{e_l}.$$

Um das korrekte Vorzeichen ${\epsilon_0}^{e_0} = (-1)^{e_0}$ zu berechnen, muß man kontrollieren, ob

$$\prod_{i=1}^{l} {\epsilon_i}^{e_l} = \epsilon \quad \text{oder} \quad \prod_{i=1}^{l} {\epsilon_i}^{e_l} = -\epsilon.$$

Im zweiten Fall ist $e_0 = 1$, im ersten 0.

Beispiel

Betrachtet man $f(x) = x^3 + 2$, $m = 2^{43}$ und $B_1 = \{2, 3, 5, \ldots, 3571\}$, so ist das Paar $(1693, 749)$ ein gutes.

$$a + bm = 6588273673635485 = 5 \cdot 13 \cdot 19 \cdot 449 \cdot 1567 \cdot 2477 \cdot 3061$$

$$N(a + b \cdot \rho) = a^3 - 2b^3 = 401280059 = 3 \cdot 43 \cdot 157 \cdot 397 \cdot 499$$

$1693 + 749 \cdot \rho$ zerfällt in

$$(1 - \rho) \cdot (3 - 2 \cdot \rho) \cdot (5 + \rho^2) \cdot (7 - 3 \cdot \rho) \cdot (5 + \rho + 4 \cdot \rho^2) \cdot (1 + 2 \cdot \rho + \rho^2) \cdot (-1).$$

Dabei ist $(1 + 2 \cdot \rho + \rho^2)$ eine Einheit.

3.7 Die Quadratbildung

An dieser Stelle möchte ich untersuchen, wie man aus den Zerlegungen von $a + bm$ und $a + b\rho$ für genügend viele gute Paare (a, b) die zur Faktorisierung von n notwendigen Quadrate berechnet.

Seien

- $B_1 = \{p_1, \ldots, p_{m_1}\}$
- $B_2 = \{\pi_1, \ldots, \pi_{m_2}\}$
- $E = \{-1, \epsilon_1, \ldots, \epsilon_l\}$

Das $j-$te gute Paar (a_j, b_j) liefert

$$a_j + b_j m = \prod_{i=1}^{m_1} p_i{}^{e_{i,j}} \qquad \text{und}$$

$$a_j + b_j\rho = (-1)^{g_{0,j}} \cdot \prod_{i=1}^{m_2} \pi_i{}^{f_{i,j}} \cdot \prod_{i=1}^{l} \epsilon^{g_{i,j}}.$$

Dies führt zur Kongruenz

$$(3.7.1) \qquad \prod_{i=1}^{m_1} p_i{}^{e_{i,j}} \equiv (-1)^{g_{0,j}} \cdot \prod_{i=1}^{m_2} \Phi(\pi_i{}^{f_{i,j}}) \cdot \prod_{i=1}^{l} \Phi(\epsilon^{g_{i,j}}) \qquad \mathrm{mod}\ n$$

und damit zu

$$(3.7.2) \qquad (-1)^{g_{0,j}} \cdot \prod_{i=1}^{m_1} p_i{}^{e_{i,j}} \cdot \prod_{i=1}^{m_2} \Phi(\pi_i)^{-f_{i,j}} \cdot \prod_{i=1}^{l} \Phi(\epsilon)^{-g_{i,j}} \equiv 1 \qquad \mathrm{mod}\ n.$$

Definiert man B wie in (3.1.5), so gilt

$B = \{b_1, \ldots, b_{m_1+m_2+l+1}\}$

mit

$$b_i := \begin{cases} p_i & : \quad i = 1, \ldots, m_1 \\ \Phi(\pi_{i-m_1}) & : \quad i = m_1 + 1, \ldots, m_1 + m_2 \\ \Phi(\epsilon_{i-m_1-m_2}) & : \quad i = m_1 + m_2 + 1, \ldots, m_1 + m_2 + l \\ -1 & : \quad i = m_1 + m_2 + l + 1. \end{cases}$$

Somit läßt sich (3.7.2) auch in der Form

$$\prod_{i=1}^{k} b_i{}^{\nu_{i,j}} \equiv 1 \qquad \mathrm{mod}\ n$$

darstellen, wobei

$$\nu_{i,j} := \begin{cases} e_{i,j} & : \quad i = 1, \ldots, m_1 \\ -f_{i-m_1,j} & : \quad i = m_1 + 1, \ldots, m_1 + m_2 \\ -g_{i-m_1-m_2,j} & : \quad i = m_1 + m_2 + 1, \ldots, m_1 + m_2 + l \\ -g_{0,j} & : \quad i = m_1 + m_2 + l + 1. \end{cases}$$

Gesucht wird nun eine Teilmenge $T \subseteq \{1, \ldots, t\}$, so daß das Produkt der linken Seiten der j-ten Kongruenzen, $j \in T$, ein Quadrat wird. Das Produkt der rechten Seiten ist natürlich immer ein Quadrat, da dort ja nur 1 aufmultipliziert wird. Also :

$$(3.7.3) \qquad \prod_{j \in T} \prod_{i=1}^{k} {b_i}^{\nu_{i,j}} = \prod_{i=1}^{k} {b_i}^{2\eta_i} \ , \qquad \eta_i \in \mathbb{Z}.$$

Betrachtet man die Exponentenvektoren $v_j \in \mathbb{Z}^k$ mit

$$v_j[i] := \nu_{i,j} \qquad i = 1, \ldots, k$$

und den Vektor $v \in \mathbb{Z}^k$ mit

$$v[i] := \sum_{j \in T} v_j[i] = \sum_{j \in T} \nu_{i,j},$$

so hat man genau dann ein Quadrat in der Form (3.7.3) erzeugt, wenn alle Einträge in v gerade sind. Da man sich nur dafür interessiert, ob die Einträge in v gerade oder ungerade sind, genügt es, die Vektoren modulo 2 zu betrachten. Deshalb setzt man

$$\bar{v}_j[i] := v_j[i] \mod 2$$

und

$$\bar{v}[i] := \sum_{j \in T} v_j[i] \mod 2$$

Ein Quadrat der Form (3.7.3) läßt sich damit genau dann berechnen, wenn der Nullvektor als nichttriviale Linearkombination der Vektoren $\bar{v}_j$ über $\mathbb{Z}/2\mathbb{Z}$ darstellbar ist.

Betrachtet man die Matrix Q mit Einträgen aus $\mathbb{Z}/2\mathbb{Z}$, deren Spalten von den Vektoren $\bar{v}_j$ gebildet werden, so geben die Lösungen x des linearen Gleichungssystems über $\mathbb{Z}/2\mathbb{Z}$,

$$Q \cdot x = 0 \ ,$$

die möglichen Linearkombinationen der $\bar{v}_j$ an. Wenn x nicht die triviale Lösung ist, so hat man eine Teilmenge T bestimmt, aus der sich eine Kongruenz der Gestalt (3.7.3) bilden läßt :

$$T := \{j \mid x[j] = 1\}, \qquad j = 1, \ldots, k.$$

Eine nichttriviale Lösung des Systems $Q \cdot x = 0$ findet man natürlich spätestens dann, wenn die Matrix Q aus mehr Spalten als Zeilen besteht, d.h. t muß größer als k sein, also

$$(3.7.4) \qquad t > k = m_1 + m_2 + l + 1 = |B_1| + |B_2| + |E|.$$

Für eine Lösung x und die ihr entsprechende Menge T gilt

$$\prod_{j \in T} \prod_{i=1}^{k} {b_i}^{\nu_{i,j}} = \prod_{i=1}^{k} {b_i}^{v[i]} \qquad \text{mit } v[i] \equiv 0 \bmod 2.$$

Somit erhält man die Kongruenz

$$(3.7.5) \qquad \prod_{i=1}^{k} {b_i}^{v[i]} \equiv 1 \mod n.$$

Setzt man

$$(3.7.6) \qquad X = \left(\prod_{i=1}^{k} {b_i}^{\frac{v[i]}{2}} \ mod\ n \right) \qquad \text{und} \qquad Y = 1,$$

so kann man mit

$$n_1 = ggT(X - Y, n)$$

einen Teiler von n ausrechnen.

Allerdings besteht die Möglichkeit, daß es sich bei n_1 um einen trivialen Teiler von n handelt, was genau dann vorkommt, wenn $X \equiv \pm 1 \mod n$ ist . In diesem Fall muß man eine andere Lösung von $Q \cdot x = 0$ betrachten.

Im ungünstigsten Fall findet man für alle Lösungen des Gleichungssystems nur die trivialen Teiler von n. Dann muß man neue gute Paare (a, b) bestimmen und die Matrix Q um die daraus errechneten Spaltenvektoren erweitern.

Es besteht aber auch die Möglichkeit, Lösungen zu finden, bevor man k Spaltenvektoren erarbeitet hat. Deshalb empfiehlt sich ein Verfahren, das die Erweiterung der Matrix Q um linear unabhängige Spaltenvektoren erlaubt. Das Hinzufügen von linear abhängigen Spalten ist dabei unnötig, denn aus einer solchen Spalte lassen sich, wie oben beschrieben, Teiler von n bestimmen. Ist einer dieser Teiler nichttrivial, so kann man das Verfahren abbrechen, andernfalls ist eine Aufnahme der linear abhängigen Spalte in die Matrix Q sinnlos, da diese Spalte ja als Linearkombination der anderen darstellbar ist und somit keinen neuen Beitrag leistet.

Der folgende Algorithmus zur Berechnung der Quadrate aus den gefundenen Kongruenzen beruht auf den oben angestellten Überlegungen. Er verwendet dabei (Q, v) für eine Matrix, die durch die Erweiterung der Matrix Q durch den Spaltenvektor v enstanden ist. Weiterhin ist zu beachten, daß man durch die Hinzunahme einer linear abhängigen Spalte zu einer Matrix, die aus linear unabhängigen Spalten besteht, ein Gleichungssystem $M \cdot x = 0$ erhält, das über $\mathbb{Z}/2\mathbb{Z}$ genau eine nichttriviale Lösung besitzt. Der dem linear abhängigen Vektor entsprechende Eintrag im Lösungsvektor x ist dann 1, denn wäre er 0, so erhielte man nur die triviale Lösung.

Folgende neue Prozeduren werden verwendet :

- $nextvek()$, liefert den Exponentenvektor einer neuen Kongruenz.
- $lsg_2(M, b)$, liefert die Lösung von $M \cdot x = b$ über $\mathbb{Z}/2\mathbb{Z}$

Algorithmus QUADRATE

Input : $n,\ B = \{b_1, \ldots, b_k\}$
Output : n_1 mit $n_1 | n$

1 .) $spalten \leftarrow 0$
2 .) $Q \leftarrow ()$ (* leere Matrix *)
3 .) $\bar{Q} \leftarrow ()$ (* leere Matrix *)
4 .) $v \leftarrow nextvek()$
5 .) $\bar{v} \leftarrow v \bmod 2$
6 .) if $(rang(\bar{Q}) = rang(\bar{Q}, \bar{v}))$ then (* v linear abhängig *)
7 .) $\quad x \leftarrow lsg_2((\bar{Q}, \bar{v}), \vec{0})$
8 .) $\quad w \leftarrow v$
9 .) $\quad$ for $i \leftarrow 1$ to $spalten + 1$ do
10 .) $\qquad$ if $(x[i] = 1)$ then $w \leftarrow w + Q[i]$
11 .) $\quad X \leftarrow 1$
12 .) $\quad$ for $i \leftarrow 1$ to $|B|$ do
13 .) $\qquad X \leftarrow X \cdot b_i^{\frac{w[i]}{2}}$
14 .) $\quad n_1 \leftarrow ggT(X - 1, n)$
15 .) $\quad$ if $(n_1 \neq 1$ und $n_1 \neq n)$ then
16 .) $\qquad$ output (n_1); stop
17 .) else (* v linear unabhängig *)
18 .) $\quad \bar{Q} \leftarrow (\bar{Q}, \bar{v})$
19 .) $\quad Q \leftarrow (Q, v)$
20 .) $\quad spalten \leftarrow spalten + 1$
21 .) goto 4

Kapitel 4

Varianten

Nachdem ich im letzten Kapitel das Grundgerüst des 'Number Field Sieve' vorgestellt habe, möchte ich nun einige Varianten aufzeigen, die dazu beitragen, schnell viele gute Paare und damit Zerlegungen zu finden.

4.1 Die Large Prime Variante

4.1.1 Large Prime in $a + bm$ (LP)

Bei der Large Prime Variante erlaubt man in der Zerlegung von $a+bm$ noch einen Faktor

$$q \in \mathbb{P},\ L < q \leq S,\ S \in \mathbb{N}^{\,1}\ .$$

Die Zerlegung von $a + bm$ läßt sich dann in der Form

$$(4.1.1) \qquad a + bm = q^t \cdot \prod_{p \in B_1} p^{\nu_p}$$

mit $t \in \{0, 1\}$ schreiben. q nennt man im Fall $t = 1$ eine **Large Prime (LP)**.

Erkennt man auch solche Paare (a, b) als gute Paare an, die anstatt der Eigenschaft (3.6.1) die Eigenschaft (4.1.1) erfüllen, so erhält man für $A_{min} \leq a \leq A_{max}$ und $1 \leq b \leq B_{max}$ wesentlich mehr gute Paare (a, b).

Eine Möglichkeit, aus den guten Paaren mit einer Large Prime in der Zerlegung von $a+bm$ eine Kongruenz der Form (3.1.1) herzustellen, wäre die Erweiterung der Faktorbasis B_1 und damit der Basis B um die Primzahlen p, $L < p \leq S$. Durch diese Erweiterung der Basis würden aber auch entsprechend mehr Kongruenzen zum Faktorisieren von n gebraucht (3.7.4). Deshalb gewinnt man mit dieser Vorgehensweise nichts.

[1] L war die obere Schranke der Primzahlbasis B_1

Dagegen verspricht folgende Behandlung von Large Prime mehr Erfolg : Berechnet man für zwei gute Paare die gleiche LP q, also die Kongruenzen

$$(4.1.2) \qquad q \cdot \prod_{i=1}^{k} {b_i}^{\nu_i} \equiv 1 \mod n$$

und

$$(4.1.3) \qquad q \cdot \prod_{i=1}^{k} {b_i}^{\mu_i} \equiv 1 \mod n,$$

so kann man durch Division daraus die Kongruenz

$$(4.1.4) \qquad \prod_{i=1}^{k} {b_i}^{\nu_i - \mu_i} \equiv 1 \mod n$$

berechnen. Diese Kongruenz besitzt wieder die gewünschte Gestalt (3.1.6).

Findet man ein weiteres gutes Paar mit derselben Large Prime, so läßt sich die daraus berechnete Kongruenz wiederum mit der Kongruenz (4.1.2) zu einer Kongruenz der Form (3.1.6) verknüpfen. Findet man also $s+1$ gute Paare mit der gleichen Large Prime, so kann man daraus s Kongruenzen der Gestalt (3.1.6) bilden.

Es bleibt nun noch die Frage zu klären, wie man eine Large Prime erkennt. Im Schritt 11 des ABMSIEB-Algorithmus wird überprüft, ob $a + bm$ total über B_1 zerfällt, d.h. ob $A[a] = 1$ ist. Dieser Test muß nun so geändert werden, daß für $1 < A[a] \leq S$ untersucht wird, ob $A[a]$ eine Large Prime ist.

Sei p_{m_1} die größte in B_1 vorkommende Primzahl.

- Gilt $A[a] < {p_{m_1}}^2$, so ist $A[a]$ eine Primzahl. Denn wäre dem nicht so, dann gäbe es einen Teiler $p < p_{m_1}$ von $A[a]$. Aber alle Teiler kleiner als p_{m_1} wurden im ABM-SIEB herausdividiert.

- Gilt ${p_{m_1}}^2 < A[a] \leq S$, so muß man durch Suchen in einer Primzahltabelle überprüfen, ob $A[a]$ eine Primzahl ist.

In der Praxis werden allerdings nur solche Large Prime betrachtet mit ${p_{m_i}}^2 > S$, denn sonst wäre der Large Prime Bereich zu groß, um noch nützlich zu sein. Dies bedeutet, daß man den Schritt 11 des ABMSIEB-Algorithmus durch

11.) if $(A[a] > S)$ then $GUT[a] \leftarrow$FALSCH

ersetzen kann.

4.1.2 Large Prime in $a + b\rho$ (LPI)

Analog zur Large Prime in der Zerlegung von $a + bm$ kann man auch eine Large Prime in der Zerlegung von $a + b\rho$ zulassen, d.h.

$$(4.1.5) \qquad a + b\rho = \tau^t \cdot \prod_{\pi \in B_2} \pi^{\mu_\pi} \cdot \epsilon$$

mit $N(\tau) \in \mathbb{P},\ N(\tau) \leq S,\ t \in \{0,1\},\ \epsilon \in E$.

τ nennt man im Fall $t = 1$ **Large Prime** und (τ) **Large Prime Ideal (LPI)**. Ersetzt man in der Definition guter Paare die Eigenschaft (3.6.2) durch (4.1.5), so erhält man für ein gutes Paar mit der Large Prime τ in der Zerlegung von $a + b\rho$ die Kongruenz

$$\Phi(\tau)^{-1} \cdot \prod_{i=1}^{k} b_i^{\nu_i} \equiv 1 \mod n.$$

Findet man eine weitere Kongruenz mit gleicher Large Prime τ, so läßt sich daraus analog zur Behandlung von LP's durch Division eine Kongruenz der Form (3.1.6) herstellen. Die zum Erkennen von LPI im ABRHOSIEB-Algorithmus notwendige Änderung ist die gleiche wie im ABMSIEB-Algorithmus, da ja nicht Elemente aus $\mathbb{Z}[\rho]$ in A abgespeichert sind, sondern deren Normen. Also :

11.) if $(A[a] > S)$ then $GUT[A] \leftarrow$FALSCH.

4.1.3 Large Prime in beiden Zerlegungen (LPPI)

Neben einer Large Prime in nur einer der beiden Zerlegungen

$$a + bm = q^t \cdot \prod_{p \in B_1} p^{\nu_p} \qquad \text{bzw.}$$

$$a + b\rho = \tau^t \cdot \prod_{\pi \in B_2} \pi^{\nu_\pi} \cdot \epsilon$$

läßt man auch je eine Large Prime in beiden Zerlegungen zu.
Auf diese Weise erhält man eine Kongruenz

$$(4.1.6) \qquad \Phi(\tau)^{-1} \cdot q \cdot \prod_{i=1}^{k} b_i^{\nu_i} \equiv 1 \mod n.$$

Findet man eine LP- oder LPI-Kongruenz mit der gleichen Large Prime q oder τ, also entweder

$$(4.1.7) \qquad q \cdot \prod_{i=1}^{k} b_i^{\mu_i} \equiv 1 \mod n$$

oder

$$(4.1.8) \qquad \Phi(\tau)^{-1} \cdot \prod_{i=1}^{k} b_i^{\lambda_i} \equiv 1 \mod n,$$

so kann man aus (4.1.6) entweder eine LPI-Kongruenz

$$(4.1.9) \qquad \Phi(\tau)^{-1} \cdot \prod_{i=1}^{k} b_i^{\nu_i-\mu_i} \equiv 1 \mod n$$

oder eine LP-Kongruenz

$$(4.1.10) \qquad q \cdot \prod_{i=1}^{k} b_i^{\nu_i-\lambda_i} \equiv 1 \mod n$$

herstellen und entsprechend weiterverarbeiten.

Beispiel

Daß es im Normalfall mehr gute Paare mit Large Primes gibt als ohne, versteht sich von selbst. Wie das Verhältnis in der Praxis tatsächlich aussieht, zeigt folgendes Beispiel.[2]

$n = 7^{149} + 1$ und $f(x) = x^5 + 7$.

Gesucht wurden die guten Paare in den Grenzen :

$$-5000000 \le a \le 5000000 \quad \text{und}$$
$$1 \le b \le 1136000.$$

B_1 enthielt die ersten 25000 Primzahlen.

Gefunden wurden

10688 Paare ohne Large Primes,

103692 Paare mit LP's,

116410 Paare mit LPI's,

1138617 Paare mit LPPI's.

[2]vgl. [1], 3, results

4.2 Logarithmisches Sieben

In den Siebalgorithmen ABMSIEB, ABMSIEB2, und ABRHOSIEB wurden die im Array A abgespeicherten Werte $|a + bm|$ bzw. $|N(a + b\rho)|$ durch ihre Primteiler aus B_1 dividiert. In der Praxis benötigt aber eine Division gegenüber einer Addition bzw. Subtraktion wesentlich mehr Zeit. Deshalb benutzt man ein Verfahren, das anstatt der Division nur eine Subtraktion ausführt. Dazu speichert man allerdings nicht wie zuvor $|a + bm|$ bzw. $|N(a + b\rho)|$ ab, sondern

$$\log |a + bm| \text{ bzw. } \log |N(a + b\rho)|.$$

Immer dann, wenn früher $A[a]$ durch eine Primzahl p dividiert wurde, subtrahiert man nun $\log(p)$ von $A[a]$. Die Schritte 8 und 9 im ABMSIEB und im ABRHOSIEB sowie die Schritte 11 und 12 im ABMSIEB2 entfallen, da die Behandlung von höheren Potenzen wieder Divisionen mit sich führt. Die Bedingung in Schritt 11 der Algorithmen ABMSIEB und ABRHOSIEB bzw. Schritt 14 des ABMSIEB2 muß geändert werden. Sie lautet nun nicht mehr

if $(A[a] \neq 1)$ then ...

sondern

if $(A[a] \neq 0)$ then ... ,

da $\log(1) = 0$. Benutzt man die Large Prime Variante, so lautet Schritt 11 analog

if $(A[a] > \log(S))$ then ...

Der Nachteil dieser Variante liegt im Wegfallen der oben angegebenen Schritte, was zur Folge hat, daß nur durch die erste Potenz der Primzahlen aus B_1 dividiert wird. Dadurch erkennt ABMSIEB und ABRHOSIEB nur solche Paare als gut an, bei denen in der Zerlegung von $a + bm$ bzw. $a + b\rho$ nur Exponenten aus $\{0, 1\}$ vorkommen. ABMSIEB2 schwächt dagegen diesen Nachteil durch die Tatsache ab, daß die Faktorbasis B_1 durch Hinzunahme von höheren Potenzen der Primzahlen zur Faktorbasis $\tilde{B}_1$ vergrößert wurde und somit auch höhere Potenzen berücksichtigt werden. Kombiniert man diese Variante mit der Large Prime Variante, dann werden die gemachten Fehler, solange sie kleiner als S sind, durch die Abfrage if $(A[a] > S)$ bzw. durch die logarithmische Version der Abfrage absorbiert.
Ist der erzeugte Fehler größer als S, so erkennt das Verfahren ein gutes Paar (a, b) nicht als solches. Hier opfert man eine Kongruenz der Schnelligkeit des Algorithmus. Das Anwenden der Large Prime Variante und die damit verbundene großzügigere Bedingung $A[a] \leq S$ für gute Paare hilft auch beim Zerlegen von $a + b\rho$ über das Manko hinweg, nur erste Potenzen im ABRHOSIEB-Algorithmus zu betrachten, zumal die in der Zerlegung der Normen vorkommenden höheren Potenzen meistens von kleinen Primzahlen stammen.

4.3 Logarithmisches Zerlegen

Zur Berechnung der Einheit, die übrig bleibt, wenn man $a + b\rho$ über B_2 zerlegt, multipliziert man die Primteiler von $a + b\rho$ auf und erhält die fehlende Einheit durch Division von $a + b\rho$ durch das Primteilerprodukt. Anstatt die Primelemente aufzumultiplizieren, kann man auch hier mit der Addition bzw. Subtraktion von Logarithmen auskommen.

Sei dazu die Abbildung

$$\mu_1 : \mathbb{Q}[\rho] \to \mathbb{R}$$

gemäß Definition 18 und Satz 18 gegeben. Dann gilt

$$\mu_1(\alpha \cdot \beta) = \mu_1(\alpha) + \mu_1(\beta). \tag{4.3.1}$$

Aus

$$a + b\rho = \prod_{\pi \in B_2} \pi^{\nu_\pi} \cdot \epsilon \qquad \epsilon \in E \tag{4.3.2}$$

folgt

$$\begin{aligned}
\mu_1(a + b\rho) &= \mu_1\left(\prod_{\pi \in B_2} \pi^{\nu_\pi} \cdot \epsilon\right) && \text{nach (4.3.2)}\\
&= \sum_{\pi \in B_2} \mu_1\left(\pi^{\nu_\pi}\right) + \mu_1(\epsilon) && \text{nach (4.3.1)}\\
&= \sum_{\pi \in B_2} \mu_1(\pi) \cdot \nu_\pi + \mu_1(\epsilon) && \text{nach (4.3.1)}
\end{aligned}$$

Für die gesuchte Einheit ϵ gilt somit

$$\mu_1(\epsilon) = \mu_1(a + b\rho) - \sum_{\pi \in B_2} \mu_1(\pi) \cdot \nu_\pi.$$

Aus dem Wert $\mu_1(\epsilon)$ läßt sich aber ϵ selbst nicht berechnen. Nimmt aber μ_1 für verschiedene Elemente α auch verschiedene Werte $\mu_1(\alpha)$ an, so kann man die Einheiten analog zu den Large Primes behandeln, d.h. sobald eine weitere Zerlegung mit dem gleichen Wert für die Einheit gefunden wird, dividiert man die entsprechenden Kongruenzen und erhält so eine Kongruenz der Form (3.1.6). Leider gibt es keine injektiven μ_1, was daran liegt, daß

$$\mu_1(-\alpha) = \mu_1(-1) + \mu_1(\alpha) = \mu_1(\alpha).$$

Sorgt man aber dafür, daß die gesuchte Einheit eine positive Norm besitzt, was nur dann immer möglich ist, wenn $N(-1) = -1$, so kann man durch Berechnung von

$$N(\alpha + \beta\rho)/N\left(\prod_{\pi \in B_2} \pi^{\nu_\pi}\right)$$

das Vorzeichen der Norm der Einheit bestimmen. Ist $r = \rho^{(1)} \in \mathbb{R}$ und $N(-1) = -1$, dann ist μ_1, eingeschränkt auf Elemente positiver Norm, injektiv, da $1, r, r^2, \ldots, r^{d-1}$ über $\mathbb{Q}$ linear unabhängig sind.

Der Nachteil dieser Variante besteht darin, daß man einige Kongruenzen verschenkt, denn auch hier gilt wie in der Large Prime Variante :

Aus $s + 1$ Zerlegungen mit der gleichen Einheit werden s Kongruenzen erzeugt.

Andererseits muß aber die Einheitenmenge U nicht mehr berechnet werden. Außerdem wird durch das Ersetzen des Aufmultiplizierens der Primelemente durch das Aufaddieren der entsprechenden Logarithmen und das Wegfallen der Division von $a + b\rho$ durch das Primteilerprodukt Laufzeit eingespart.

4.4 Verwendung mehrerer Zahlkörper

Betrachtet man zu der zu faktorisierenden Zahl n mehrere über $\mathbb{Q}$ irreduzible Polynome

$$f_i(x) \in \mathbb{Z}[x]$$

mit den Werten

$$m_i \in \mathbb{N} \qquad \text{mit } f_i(m_i) = n$$

und die zugehörigen Zahlkörper $\mathbb{K}_i$, $i = 1, \ldots, t$.
Seien ${B_1}^{(i)}$, ${B_2}^{(i)}$, $U^{(i)}$, $\Phi^{(i)}$ analog zum Fall eines Zahlkörpers definiert, dann gilt für die Gesamtfaktorbasis B

$$\begin{aligned} B = & \bigcup_{i=1}^{t} \left({B_1}^{(i)} \cup \Phi^{(i)}({B_2}^{(i)}) \cup \Phi^{(i)}(U^{(i)})\right) \\ & \bigcup_{i=1}^{t} {B_1}^{(i)} \cup \bigcup_{i=1}^{t} \Phi^{(i)}({B_2}^{(i)}) \cup \bigcup_{i=1}^{t} \Phi^{(i)}(U^{(i)}). \end{aligned}$$

Für ${B_1}^{(i)}$ gilt ja

$${B_1}^{(i)} = \{p \mid p \in \mathbb{P},\ p \leq L^{(i)}\}, \text{ für die oberen Schranken } L^{(i)} \in \mathbb{N}.$$

Sei $j \in 1, \ldots t$ so, daß

$$L^{(i)} \leq L^{(j)} \quad \forall i = 1, \ldots, t,$$

dann gilt

$$\bigcup_{i=1}^{t} {B_1}^{(i)} = {B_1}^{(j)}.$$

Damit benötigt man etwa

$$(4.4.1) \qquad s = |{B_1}^{(j)}| + \sum_{i=1}^{t} |{B_2}^{(i)}| + \sum_{i=1}^{t} |U^{(i)}|.$$

Kongruenzen zur Faktorisierung von n. Damit entfallen auf jeden Zahlkörper etwa $\frac{s}{t}$ Kongruenzen. Der Vorteil dieser Variante ist darin zu sehen, daß für die einzelnen Zahlkörper weniger Kongruenzen gefunden werden müssen als beim Verfahren mit nur einem Zahlkörper. Man kann sich deshalb bei der Suche nach guten Paaren auf betragsmäßig kleine Werte a, b beschränken.

Kapitel 5

Anmerkungen

5.1 Freie Kongruenzen

Besitzt $f(x)$ modulo $p \in B_1$ d Nullstellen $cp^{(i)}$, ist also $f(x)$ modulo p total in Linearfaktoren $f_i(x)$ zerlegbar, so zerfällt (p) nach Satz 16 in

$$(p) = \prod_{i=1}^{d} (p, \rho - cp^{(i)})$$

und somit gilt

$$p = \prod_{i=1}^{d} \pi_i \cdot \epsilon$$

mit $(\pi_i) = (p, \rho - cp^{(i)})$ und $\epsilon \in E$.
Auf der anderen Seite zerfällt p natürlich über B_1, so daß man dadurch die Kongruenz

$$p \equiv \prod_{i=1}^{d} \Phi(\pi_i) \cdot \Phi(\epsilon) \qquad \bmod n$$

erhält. Eine solche Kongruenz, die durch das totale Zerlegen eines Hauptideals (p) enstanden ist, nennt man **freie Kongruenz**. Diese kann auch zur Faktorisierung von n mitbenutzt werden.

Beispiel
Für $n = 2^{43} + 1$ mit $f(x) = x^3 + 2$ und $|B_1| = 500$ findet man 81 freie Kongruenzen, was immerhin etwa 8 Prozent der zur Faktorisierung von n notwendigen Kongruenzen ausmacht.

Zum Beispiel zerfällt 41 in

$$41 = (1 + \rho + 2 \cdot \rho^2) \cdot (3 - 2 \cdot \rho + 3 \cdot \rho^2) \cdot (-3 - \rho + 2 \cdot \rho^2).$$

5.2 Die Norm

Bei der Suche nach guten Paaren spielt die Norm eine erhebliche Rolle. Bisher habe ich mich aber darüber ausgeschwiegen, wie man die Norm berechnet. Das Versäumte will ich an dieser Stelle nachholen.
Die einfachste Methode der Normberechnung geht auf die Definition 10 zurück. Mit Hilfe der Monomorphismen σ_i aus Definition 18 läßt sich die Norm schreiben als

$$\begin{aligned} N(\alpha) &= \alpha^{(1)} \cdot \ldots \cdot \alpha^{(d)} \\ &= \sigma_1(\alpha) \cdot \ldots \cdot \sigma_d(\alpha). \end{aligned}$$

Man erhält also die Norm durch einfaches Einsetzen der einzelnen Wurzeln von $f(x)$ in α und Aufmultiplizieren der dadurch berechneten komplexen Zahlen.

Diese Normberechnung führt allerdings dazu, daß in der Praxis die Normen aufgrund von Rundungsfehlern nur angenähert berechnet werden können. Da die Norm ganzer algebraischer Zahlen, und nur solche interessieren ja beim 'Number Field Sieve' , nach Satz 7 in $\mathbb{Z}$ liegt, verwendet man im Fall $G = \mathbb{Z}[\rho]$ besser ein Verfahren, das nur in $\mathbb{Z}$ rechnet. Dies geschieht durch Anwendung von Lemma 3 und Definition 14, wonach für eine gegebene Zahl $\alpha = a_0 + a_1 \cdot \rho + \ldots + a_{d-1} \cdot \rho^{d-1} \in \mathbb{Z}[\rho]$ gilt

$$|N(\alpha)| = N(\mathcal{I}) \quad \text{mit } \mathcal{I} = (\alpha).$$

Betrachtet man die in Definition 14 erwähnten Basen

$$\{\omega_1, \ldots, \omega_d\}=\{1, \rho, \ldots, \rho^{d-1}\} \quad \text{von } \mathbb{Q}[\rho]$$

und

$$\{\alpha, \alpha \cdot \rho, \ldots, \alpha \cdot \rho^{d-1}\} \quad \text{von } \mathcal{I},$$

und definiert man die Matrix (a_{ij}) so, daß die Zeile a_j aus den Koeffizienten von ρ^{i-1} der Zahlen $\alpha, \alpha{\cdot}\rho, \ldots, \alpha{\cdot}\rho^{d-1}$ besteht, dann entspricht (a_{ij}) der in Definition 14 beschriebenen Matrix, und es folgt:

$$|N(\alpha)| = |det(a_{ij})|$$

Beispiel
Für $f(x) = x^3 + 2$ ergibt sich

- $\alpha = a_0 + a_1 \cdot \rho + a_2 \cdot \rho^2$
- $\alpha \cdot \rho = -2 \cdot a_2 + a_0 \cdot \rho + a_1 \cdot \rho^2$
- $\alpha \cdot \rho^2 = -2 \cdot a_1 - 2 \cdot a_2 \cdot \rho + a_0 \cdot \rho^2$

Hieraus ergeben sich die Spaltenvektoren

$$b_1 = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix}, \qquad b_2 = \begin{pmatrix} -2 \cdot a_2 \\ a_0 \\ a_1 \end{pmatrix}, \qquad b_3 = \begin{pmatrix} -2 \cdot a_1 \\ -2 \cdot a_2 \\ a_0 \end{pmatrix},$$

und damit die Matrix

$$(a_{ij}) = \begin{pmatrix} a_0 & -2 \cdot a_2 & -2 \cdot a_1 \\ a_1 & a_0 & -2 \cdot a_2 \\ a_2 & a_1 & a_0 \end{pmatrix}.$$

Es gilt also

$$|N(\alpha)| = |det(a_{ij})| = |{a_0}^3 - 2 \cdot {a_1}^3 + 4 \cdot {a_2}^3 + 6 \cdot a_0 \cdot a_1 \cdot a_2|.$$

Wie man am Beispiel sieht, läßt sich aus der Normberechnung mittels Determinante noch eine dritte Version, die Norm zu bestimmen, herleiten, nämlich die Normberechnung durch eine Formel. Diese Formel entspricht der Determinante eines formalen Elementes aus dem Ring der ganzen Zahlen, wie es im Beispiel gezeigt ist, und wird zu Beginn des Algorithmus berechnet.

Die allgemeine Normberechnung wird im 'Number Field Sieve' lediglich in den Teilalgorithmen ERZEUGER1 und ERZEUGER2 benötigt, denn bei der Zerlegung von $a + b\rho$ muß nur noch die Norm von $a + b\rho$ berechnet werden. Diese läßt sich sehr leicht herleiten:

Sei $f(x) = f_0 + f_1 \cdot x + \ldots + f_{d-1} \cdot x^{d-1} + x^d$. Aus obigen Betrachtungen folgt dann

$$|N(a + b\rho)| = \left| det \begin{pmatrix} a & 0 & 0 & \cdots & 0 & -b \cdot f_0 \\ b & a & 0 & \cdots & 0 & -b \cdot f_1 \\ 0 & \cdot & \cdot & \cdots & 0 & -b \cdot f_2 \\ 0 & \cdot & \cdot & \cdot & & \vdots \\ \vdots & & \cdot & \cdot & a & -b \cdot f_{d-2} \\ 0 & \cdot & \cdot & 0 & b & a - b \cdot f_{d-1} \end{pmatrix} \right|.$$

Entwickelt man nach der letzten Spalte, so erhält man für die Norm

$$\begin{aligned} |N(a + b\rho)| &= (-1)^{d+1}(-b \cdot f_0)b^{d-1} + (-1)^{d+2}(-b \cdot f_1)ab^{d-2} + \ldots + (-1)^{2d}(a - b \cdot f_{d-1})a^{d-1} \\ &= a^d + \sum_{i=0}^{d-1} (-1)^{d+i+1}(-f_i \cdot a^i \cdot b^{d-i}) \\ &= a^d + \sum_{i=0}^{d-1} (-1)^{d+i}(f_i \cdot a^i \cdot b^{d-i}) \end{aligned}$$

Für den Fall $f(x) = x^d + c$ gilt damit

$$|N(a + b\rho)| = a^d + (-1)^d \cdot c \cdot b^d = a^d + c \cdot (-b)^d.$$

Beispiel
Sei $f(x) = x^3 + 2$
Dann gilt nach obiger Formel :

$$N(a + b\rho) = a^3 - 2 \cdot b^3.$$

5.3 Division in $\mathbb{Z}[\rho]$

In der 'Number Field Sieve' Version ohne Varianten benötigt man zum Berechnen der Einheit in der Zerlegung von $a + b\rho$ und im ERZEUGER1-Algorithmus eine Division in $\mathbb{Z}[\rho]$. Die Division zweier Zahlen aus $\mathbb{Z}[\rho]$ geht zwar im allgemeinen nicht auf, doch wird sie im 'Number Field Sieve' nur dann benötigt, wenn klar ist, daß sie erfolgreich sein wird.

Seien $\alpha, \beta \in \mathbb{Z}[\rho]$ mit

$$\alpha = a_0 + a_1 \cdot \rho + \ldots + a_{d-1} \cdot \rho^{d-1} \quad \text{und}$$
$$\beta = b_0 + b_1 \cdot \rho + \ldots + b_{d-1} \cdot \rho^{d-1}.$$

Gilt $\beta | \alpha$, dann existiert nach Definition 7 ein $\gamma \in \mathbb{Z}[\rho]$

$$\gamma = c_0 + c_1 \cdot \rho + \ldots + c_{d-1} \cdot \rho^{d-1},$$

so daß $\alpha = \beta \cdot \gamma$. Dieses Element γ soll nun bestimmt werden.
Es gilt

$$(5.3.1) \qquad \begin{aligned} \alpha &= \beta \cdot \gamma \\ &= (b_0 + b_1 \cdot \rho + \ldots + b_{d-1} \cdot \rho^{d-1}) \cdot (c_0 + c_1 \cdot \rho + \ldots + c_{d-1} \cdot \rho^{d-1}) \\ &= a_0 + a_1 \cdot \rho + \ldots + a_{d-1} \cdot \rho^{d-1}. \end{aligned}$$

Multipliziert man das Produkt aus und sortiert nach Potenzen von ρ, so gilt für den Koeffizienten von ρ^i, der nach (5.3.1) gleich a_i sein muß :

$$a_i = \sum_{j=0}^{d-1} \left(c_j \cdot \sum_{k=0}^{d-1} b_k \cdot \xi_{i,j,k} \right),$$

wobei sich die Faktoren $\xi_{i,j,k} \in \mathbb{Z}$ aus der Multiplikation $\beta \cdot \gamma$ ergeben. Somit läßt sich das lineare Gleichungssystem $\quad M \cdot c = a$

mit $$a := \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{d-1} \end{pmatrix}, \qquad c := \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_{d-1} \end{pmatrix} \quad \text{und} \quad (m_{i,j}) = \sum_{k=0}^{d-1} b_k \cdot \xi_{i,j,k}$$

erstellen, dessen Lösung gerade die Vektordarstellung der gesuchten ganzen algebraischen Zahl γ ist.

Da $\beta|\alpha$ und die Division in $\mathbb{Z}[\rho]$ eindeutig ist, folgt, daß das Gleichungssystem $M \cdot c = a$ genau eine Lösung $c \in \mathbb{Z}^d$ besitzt. Diese Lösung läßt sich leicht durch sukzessives Lösen der linearen diophantischen Gleichungen

$$\sum_{j=0}^{d-1} m_{i,j} \cdot c_j = a_i \qquad i = 0, \ldots, d-1$$

berechnen.

5.4 Dividieren von Kongruenzen

Sowohl die Large Prime Variante als auch das Logarithmische Zerlegen macht die Division zweier Kongruenzen modulo n erforderlich. Aus

$$(5.4.1) \qquad \eta \cdot \prod_{b_i \in B} {b_i}^{\nu_i} \equiv 1 \mod n$$

und

$$(5.4.2) \qquad \eta \cdot \prod_{b_i \in B} {b_i}^{\mu_i} \equiv 1 \mod n$$

mit $\eta \in \mathbb{Z}$, wurde die Kongruenz

$$(5.4.3) \qquad \prod_{b_i \in B} b_i^{\nu_i - \mu_i} \equiv 1 \mod n$$

berechnet. Dies ist allerdings nur bedingt korrekt, denn wenn

$$n_1 = ggT(\eta \cdot \prod_{b_i \in B} {b_i}^{\mu_i}, n) \neq 1,$$

so muß (5.1.4) nicht korrekt sein, da aus

$$g \cdot a \equiv g \cdot b \mod g \cdot c$$

nicht

$$a \equiv b \mod g \cdot c, \qquad g, c \in \mathbb{N},\ a, b \in \mathbb{Z}$$

folgt. Allerdings gilt stets :

$$a \equiv b \mod c.$$

Man berechnet deshalb vor der Division der Kongruenzen den ggT n_1. Ist dieser ungleich 1, so hat man einen nichttrivialen Teiler von n, eben n_1, gefunden. Setzt man nun $\tilde{n} \leftarrow n/n_1$, so gelten alle bisher gefundenen Kongruenzen

$$\prod_{b \in B} b_i^{\nu_i - \mu_i} \equiv 1 \mod n$$

auch modulo $\tilde{n}$. Ebenso ist die Kongruenz

$$X^2 \equiv Y^2 \mod n$$

auch modulo $\tilde{n}$ korrekt. Deshalb kann man mit $\tilde{n}$ statt mit n weiterrechnen und erhält dennoch alle Teiler von n. Anstatt vor jeder Division die ggT-Berechnung auszuführen, kann man natürlich auch zu Beginn des Verfahrens den $ggT(b_i, n)$ für alle Faktorbasiselemente b_i bilden und $\tilde{n}$ entsprechend berechnen.

5.5 Einheitswurzeln

Ein weiteres Problem, das ich bisher noch nicht angesprochen habe, entsteht durch Einheitswurzeln, und zwar nicht triviale Einheitswurzeln (also $\neq \pm 1$). Diese treten zwar nach Satz 9 nur dann auf, wenn $f(x)$ keine reelle Wurzel besitzt, verursachen dann aber einige Probleme. Sie besitzen nämlich die unangenehme Eigenschaft, daß

$$(5.5.1) \qquad \mu_1(\epsilon_1) = 0$$

für μ_1 wie in Definition 18 bzw. Satz 18 und eine Einheitswurzel ϵ_1. Die Korrektheit von (5.5.1) ist leicht einzusehen, denn für ϵ_1 gilt :

$$\exists k \in \mathbb{N} : {\epsilon_1}^k = 1 \Rightarrow 0 = \mu_1(1) = \mu_1({\epsilon_1}^k) = k \cdot \mu_1(\epsilon_1),$$

woraus (5.5.1) folgt. Dies bedeutet für ein Element $\epsilon \in E$:

$$\tilde{\mu}(\epsilon) = \tilde{\mu}(\epsilon) + \tilde{\mu}(\epsilon_1) = \tilde{\mu}(\epsilon \cdot \epsilon_1)$$

mit $\tilde{\mu}$ wie in Kapitel 3.5. Ist

$$\epsilon = (a + b\rho) / \prod_{\pi \in B_2} \pi^{\nu_\pi},$$

so erhält man durch Lösen des Gleichungssystems (3.6.13), zu welchen Anteilen x_i die Basisvektoren ϵ_i aus U in ϵ vorkommen. Allerdings ist es so nicht möglich festzustellen, welche Einheitswurzel $\tilde{\epsilon}$ noch fehlt. Diese läßt sich aber anhand

$$\tilde{\epsilon} = \epsilon / \prod_{i=1}^{l} {\epsilon_i}^{x_i}$$

ermitteln. Jede so gefundene Einheitswurzel wird zu U hinzugenommen und die Faktorbasis somit erweitert. Eine zweite Möglichkeit der Behandlung einer Einheitswurzel entspricht der Behandlung von Large Primes, d.h. beim zweiten Vorkommen werden die entsprechenden Kongruenzen dividiert.

5.6 Der Übergang von b nach $b+1$

Der Siebalgorithmus ABMSIEB liefert für ein festes $b \in \{1, \ldots, B_{max}\}$ die Zahlen $a \in \{A_{min}, \ldots, A_{max}\}$: mit $a + bm$ ist über B_1 zerlegbar. [1]

Eine große Rolle spielt dabei die Berechnung der ersten Stelle im Array A, an der eine Primzahl $p_i \in B_1$ $A[a]$ teilt. Dieses geschieht durch die Bestimmung des kleinsten positiven Restes

$$r_i = A_{min} + b \cdot m \mod p_i.$$

Speichert man sich nun, für alle $p_i \in B_1$, die Werte

$$\tilde{A}[i] := A_{min} \bmod p_i$$

und

$$\tilde{M}[i] := m \bmod p_i$$

ab, so lassen sich damit für die verschiedenen b die Reste $r_{b,i}$ durch

$$r_{b,i} := \tilde{A}[i] + b \cdot \tilde{M}[i] \bmod p_i$$

berechnen. Da

$$\begin{aligned} r_{b,i} &= \tilde{A}[i] + b \cdot \tilde{M}[i] \bmod p_i \\ &= \tilde{A}[i] + (b-1) \cdot \tilde{M}[i] + \tilde{M}[i] \bmod p_i \\ &= r_{b-1,i} + \tilde{M}[i] \bmod p_i \end{aligned}$$

gilt, erhält man ausgehend von $r_{0,i} = 0$ alle benötigten Werte von $r_{b,i}$ durch Addieren von $\tilde{M}[i] \bmod p_i$.

Da $0 \leq \tilde{M}[i], r_{b-1,i} < p_i$, kann man $r_{b,i}$ wie folgt berechnen:

$$r_{b,i} \leftarrow r_{b-1,i} + \tilde{M}[i]$$

$$\text{if } (r_{b,i} > p_i) \text{ then } r_{b,i} \leftarrow r_{b,i} - p_i.$$

Für die Variante des Logarithmischen Siebens bedeutet dieser Übergang von b zu $b+1$, daß man nach der Vorberechnung von $\tilde{A}$ und $\tilde{M}$ ganz ohne Multiplikation und Division auskommt.

Analog angewandt, findet diese Berechnung der $r_{b,i}$ auch beim ABMSIEB2 und beim ABRHOSIEB Verwendung.

[1] siehe Kapitel 3.8

5.7 Verschiedenes

1. Die Berechnung von X in (3.7.6) erfordert für $v[i] < 0$ die Bestimmung des multiplikativen Inversen von b_i mod n. Dieses existiert natürlich nur dann, wenn $ggT(b_i, n) = 1$ ist, wovon man allerdings aufgrund der Überlegungen in Kapitel 5.4 ausgehen kann. ${b_i}^{-1}$ läßt sich mit dem erweiterten Euklidischen Algorithmus[2] errechnen.
   Man kann aber auch ohne Inversenbestimmung auskommen. Sei dazu :
   $$P, M, N \subseteq \{1, \ldots, k\} \text{ mit } P \cup N \cup M = \{1, \ldots, k\} \quad \text{und}$$
   $$v[i] \begin{cases} > 0 & : \ i \in P \\ = 0 & : \ i \in N \\ < 0 & : \ i \in M \end{cases} \text{, dann folgt aus (3.7.5)}$$
   $$X^2 \equiv \prod_{i=1}^{k} {b_i}^{v[i]} \equiv \prod_{i \in P} {b_i}^{v[i]} \cdot \prod_{j \in M} {b_j}^{-|v[j]|} \equiv 1 \mod n.$$
   Hieraus folgt aber
   $$\prod_{i \in P} {b_i}^{v[i]} \equiv \prod_{j \in M} {b_j}^{|v[j]|} \mod n,$$
   wobei ja $v[i], |v[j]|$ gerade sind. Setzt man
   $$\tilde{X} := \left(\prod_{i \in P} {b_i}^{\frac{v[i]}{2}}\right) \quad \text{und} \quad \tilde{Y} := \left(\prod_{j \in M} {b_j}^{\frac{|v[j]|}{2}}\right),$$
   so kann man mit $ggT(\tilde{X} + \tilde{Y}, n)$ einen Teiler von n berechnen.

2. Läßt sich für die zu faktorisierende Zahl n kein günstiges[3] Polynom $f(x)$ mit $f(m) = n$ finden, so kann man auch ein kleines Vielfaches von n zur Auswahl des Polynoms verwenden, d.h.
   $$r \cdot n = f(m) \quad \text{mit } r > 0.$$
   Den 'Number Field Sieve' Algorithmus kann man dagegen mit n starten, da alle gefundenen Kongruenzen nicht nur modulo $r \cdot n$, sondern auch modulo n, korrekt sind. [4]
   Ausgenutzt wurde diese Möglichkeit, z.B. bei der Faktorisierung von $n = 2^{128} + 1$.[5] Da
   $$2 \cdot n = 2 \cdot (2^{128} + 1) = (2^{43})^3 + 2$$
   ist, konnte das irreduzible Polynom $f(x) = x^3 + 2$ zur Faktorisierung von n herangezogen werden.

[2] siehe dazu [4]
[3] d.h irreduzibel und mit kleinen Koeffizienten
[4] siehe Kapitel 5.4
[5] siehe [12]

Kapitel 6

Die Laufzeit

Das Kernstück des 'Number Field Sieve' besteht aus

- der Berechnung der guten Paare (p, cp)
- der Berechnung der Kongruenzen
- dem Lösen eines Gleichungssystems über $\mathbb{Z}/2\mathbb{Z}$

Ich möchte nun die erwartete Laufzeit dieses Kernstücks analysieren. Dazu verwende ich folgende Definitionen und Sätze aus dem Artikel 'Algorithms in Number Theorie' von A.K. und H.W.Lenstra :

Definition 22
Sei $\gamma, \nu \in \mathbb{R}$ mit $0 \leq \nu \leq 1$.
Man definiert

$$L_x[\nu, \gamma] := e^{(\gamma + O(1)) \cdot (\log x)^{\nu} \cdot (\log\log x)^{(1-\nu)}}$$

für $x \to \infty$.

Definition 23
Seien $k, m \in \mathbb{N}$.
m heißt **k-smooth**, wenn alle Primteiler von m kleiner oder gleich k sind. In Zeichen:

$$m \text{ ist } k\text{-smooth} \Leftrightarrow \forall p \in \mathbb{P} \text{ mit } p|m \text{ gilt} : p \leq k.$$

Satz 22
Für feste $\gamma, \delta, \nu, \omega \in \mathbb{R}$ mit $\gamma, \delta > 0$ und $0 < \omega < \nu \leq 1$ beträgt die Wahrscheinlichkeit P, daß eine zufällig gewählte positive ganze Zahl $m \leq L_x[\nu, \gamma]$ $L_x[\omega, \delta]$-smooth ist :

$$P = L_x[\nu - \omega, -\tfrac{\gamma}{\delta} \cdot (\nu - \omega)] \quad \text{für } x \to \infty.$$

Definition 24

Die Funktion $\Pi : \mathbb{N} \to \mathbb{N}$
mit $\Pi(n) = \#\{p \in \mathbb{P} \mid p \leq n\}$
heißt Primzahlzählfunktion.

Lemma 15
$\Pi(L_x[\nu, \gamma]) = L_x[\nu, \gamma]$.

Wählt man für

- $f(x) = x^d + c$, $|c|$ klein und
- $n = f(m)$

die Parameter des 'Number Field Sieve' wie folgt :

- $d := \left(\frac{3 \log n}{(2 \log \log n)}\right)^{\frac{1}{3}}$,
- $L := L_n[\frac{1}{3}, (\frac{2}{3})^{\frac{2}{3}}]$,
- $a, b \leq L_n[\frac{1}{3}, (\frac{2}{3})^{\frac{2}{3}}]$,

so kann man die Laufzeit des Verfahrens gut abschätzen. Es gilt

Lemma 16
Unter obigen Voraussetzungen ist

$$m = L_n[\tfrac{2}{3}, (\tfrac{2}{3})^{\frac{1}{3}}].$$

<u>*Beweis*</u> :
Für kleine $|c|$ gilt:

$$\begin{aligned} m \approx n^{\frac{1}{d}} &= n^{\left(\frac{3 \log n}{(2 \log \log n)}\right)^{-\frac{1}{3}}} \\ &= n^{(2 \cdot \log \log n / (3 \cdot \log n))^{\frac{1}{3}}} \\ &= e^{\log n \cdot 2^{\frac{1}{3}} \cdot (\log \log n)^{\frac{1}{3}} / (3^{\frac{1}{3}} \cdot (\log n)^{\frac{1}{3}})} \\ &= e^{(\frac{2}{3})^{\frac{1}{3}} \cdot (\log n)^{\frac{2}{3}} \cdot (\log \log n)^{\frac{1}{3}}} \\ &= L_n[\tfrac{2}{3}, (\tfrac{2}{3})^{\frac{1}{3}}] \qquad \square \end{aligned}$$

Gesucht werden im 'Number Field Sieve' solche Paare (a, b), für die sowohl

- $|a + bm|$ als auch
- $|N(a + b\rho)| = |a^d - c \cdot (-b)^d|$

über der Faktorbasis B_1 zerfällt.
$|a + bm|$ und $|a^d - c \cdot (-b)^d|$ sind dabei nach oben durch $L_n[\frac{2}{3}, (\frac{2}{3})^{\frac{1}{3}}]$ begrenzt.

<u>*Beweis*</u> :

$$\begin{aligned}
a^d &\leq L_n[\tfrac{1}{3}, (\tfrac{2}{3})^{\frac{2}{3}}]^d \\
&= \left(e^{(\frac{2}{3})^{\frac{2}{3}} \cdot (\log n)^{\frac{1}{3}} \cdot (\log\log n)^{\frac{2}{3}}}\right)^{\left(\frac{3 \log n}{(2 \log\log n)}\right)^{\frac{1}{3}}} \\
&= e^{\frac{(\frac{2}{3})^{\frac{2}{3}} \cdot (\log n)^{\frac{1}{3}} \cdot (\log\log n)^{\frac{2}{3}} \cdot 3^{\frac{1}{3}} \cdot (\log n)^{\frac{1}{3}}}{2^{\frac{1}{3}} \cdot (\log n)^{\frac{1}{3}}}} \\
&= e^{(\frac{2}{3})^{\frac{1}{3}} \cdot (\log n)^{\frac{2}{3}} \cdot (\log\log n)^{\frac{1}{3}}} \\
&= L_n[\tfrac{2}{3}, (\tfrac{2}{3})^{\frac{1}{3}}].
\end{aligned}$$

Für $|c|$ klein, gilt also $|N(a + b\rho)| \leq L_n[\frac{2}{3}, (\frac{2}{3})^{\frac{1}{3}}]$.

$$\begin{aligned}
m \cdot b &\leq L_n[\tfrac{2}{3}, (\tfrac{2}{3})^{\frac{1}{3}}] \cdot L_n[\tfrac{1}{3}, (\tfrac{2}{3})^{\frac{2}{3}}] \\
&= e^{(\frac{2}{3})^{\frac{1}{3}} \cdot (\log n)^{\frac{2}{3}} \cdot (\log\log n)^{\frac{1}{3}} + (\frac{2}{3})^{\frac{2}{3}} \cdot (\log n)^{\frac{1}{3}} \cdot (\log\log n)^{\frac{2}{3}}}.
\end{aligned}$$

Da der zweite Summand assymptotisch zu langsam wächst, folgt

$$\begin{aligned}
m \cdot b &\leq e^{(\frac{2}{3})^{\frac{1}{3}} \cdot (\log n)^{\frac{2}{3}} \cdot (\log\log n)^{\frac{1}{3}}} \\
&= L_n[\tfrac{2}{3}, (\tfrac{2}{3})^{\frac{1}{3}}].
\end{aligned}$$

Somit ist auch $|a + bm|$ durch $L_n[\frac{2}{3}, (\frac{2}{3})^{\frac{1}{3}}]$ begrenzt.

Unter der heuristischen Annahme, daß $|N(a+b\rho)|$ und $|a+bm|$ zufällige ganze Zahlen sind, gilt damit

Satz 23
Die Wahrscheinlichkeit, daß $|N(a+b\rho)|$ bzw. $|a+bm|$ L-smooth sind, beträgt jeweils

$$L_n[\tfrac{1}{3}, -(\tfrac{1}{18})^{\frac{1}{3}}].$$

<u>*Beweis*</u> :

Für L gilt

$$L = L_n[\tfrac{1}{3}, (\tfrac{2}{3})^{\frac{2}{3}}].$$

Damit errechnet sich die Wahrscheinlichkeit, daß $|a+bm|$ L-smooth ist, nach Satz 22:

$$\begin{aligned} P &= L_n[\tfrac{2}{3} - \tfrac{1}{3}, -\frac{(\frac{2}{3})^{\frac{1}{3}}}{(\frac{2}{3})^{\frac{2}{3}}} \cdot (\tfrac{2}{3} - \tfrac{1}{3})] \\ &= L_n[\tfrac{1}{3}, -(\tfrac{2}{3})^{(-\frac{1}{3})} \cdot \tfrac{1}{3}] \\ &= L_n[\tfrac{1}{3}, -(\tfrac{3}{2})^{\frac{1}{3}} \cdot \tfrac{1}{3}] \\ &= L_n[\tfrac{1}{3}, -(\tfrac{1}{18})^{\frac{1}{3}}] \end{aligned}$$

Die gleiche Wahrscheinlichkeit errechnet man für $N(a+b\rho)$ □

Die Wahrscheinlichkeit, daß sowohl $|a+bm|$ als auch $|N(a+b\rho)|$ zerfallen, beträgt demnach also

$$\begin{aligned} L_n[\tfrac{1}{3}, -(\tfrac{1}{18})^{\frac{1}{3}}] \cdot L_n[\tfrac{1}{3}, -(\tfrac{1}{18})^{\frac{1}{3}}] &= L_n[\tfrac{1}{3}, -(\tfrac{1}{18})^{\frac{1}{3}}]^2 \\ &= e^{-2\cdot(\frac{1}{18})^{\frac{1}{3}} \cdot (\log n)^{\frac{1}{3}} \cdot (\log\log n)^{\frac{2}{3}}} \\ &= e^{-(\frac{4}{9})^{\frac{1}{3}} \cdot (\log n)^{\frac{1}{3}} \cdot (\log\log n)^{\frac{2}{3}}} \\ &= e^{-(\frac{2}{3})^{\frac{2}{3}} \cdot (\log n)^{\frac{1}{3}} \cdot (\log\log n)^{\frac{2}{3}}} \\ &= L_n[\tfrac{1}{3}, -(\tfrac{2}{3})^{\frac{2}{3}}] \end{aligned}$$

In $L_n[\frac{1}{3}, (\frac{2}{3})^{\frac{2}{3}}]$ Paaren (a,b) findet man also mit hoher Wahrscheinlichkeit ein gutes Paar. Um die zur Faktorisierung von n benötigten $L_n[\frac{1}{3}, (\frac{2}{3})^{\frac{2}{3}}]$ guten Paare zu finden, müssen deshalb

$$L_n[\tfrac{1}{3}, (\tfrac{2}{3})^{\frac{2}{3}}] \cdot L_n[\tfrac{1}{3}, (\tfrac{2}{3})^{\frac{2}{3}}] = L_n[\tfrac{1}{3}, 2 \cdot (\tfrac{2}{3})^{\frac{2}{3}}]$$

Paare untersucht werden.
Damit beträgt die erwartete Laufzeit zur Berechnung der Kongruenzen

$$L_n[\tfrac{1}{3}, 2 \cdot (\tfrac{2}{3})^{\frac{2}{3}}].$$

Nach Berechnungen von D.H.Wiedemann [1] benötigt man zum Berechnen der linearen Abhängigkeit dünn besetzter Exponentenvektoren die gleiche Laufzeit. Damit beträgt die Gesamtlaufzeit des 'Number Field Sieve'

$$L_n[\tfrac{1}{3}, 2 \cdot (\tfrac{2}{3})^{\frac{2}{3}}].$$

Vergleicht man diesen Wert mit der erwarteten Laufzeit des Quadratischen Siebs

$$e^{(1+O(1))\cdot(\log n)^{\frac{1}{2}}\cdot(\log\log n)^{\frac{1}{2}}},$$

so sieht man, daß die Laufzeit des 'Number Field Sieve' sich assymptotisch wesentlich besser verhält. Man vermutet, daß das 'Number Field Sieve' ab den 100-stelligen Zahlen auch in der Praxis schneller arbeitet als das Quadratische Sieb.

[1]vgl[15]

Kapitel 7

Verallgemeinerung

Das 'Number Field Sieve' , so wie ich es beschrieben habe, geht von folgenden Voraussetzungen aus

1. $f(x) = x^d + c$, $|c|$ klein,
2. $G = \mathbb{Z}[\rho]$,
3. G Hauptidealring.

Inwieweit man den Algorithmus verallgemeinern kann, soll nun untersucht werden.

ad 1. Gegeben sei eine zusammengesetzte Zahl n. Ein Polynom $f(x) \in \mathbb{Z}[x]$ mit $f(m) = n$ für ein $m \in \mathbb{N}$ gewinnt man durch folgendes Verfahren :

1. Wähle den Grad d, gemäß Kapitel 6
2. Berechne $m := \lfloor \sqrt[d]{n} \rfloor$
3. Bestimme die Koeffizienten von $f(x)$ wie folgt:

$$\begin{aligned} f_d &:= 1 \\ f_{d-1} &:= \lfloor (n - f_d \cdot m^d)/m^{d-1} \rfloor \\ f_{d-2} &:= \lfloor (n - f_d \cdot m^d - f_{d-1} \cdot m^{d-1})/m^{d-2} \rfloor \quad \text{u.s.w.} \end{aligned}$$

Es ergibt sich also für f_i die Formel:

$$f_i := \lfloor (n - \sum_{j=i+1}^{d} f_j \cdot m^j)/m^i \rfloor, \quad i = 0, \ldots, d-1.$$

Man sieht leicht, daß das auf diese Weise bestimmte Polynom

$$f(x) = f_0 + f_1 \cdot x + \ldots + f_{d-1} \cdot x^{d-1} + x^d$$

die gewünschte Eigenschaft $f(m) = n$ besitzt.

Ist $f(x)$ über $\mathbb{Q}$ irreduzibel, ρ eine Wurzel von $f(x)$ und $G = \mathbb{Z}[\rho]$ ein Hauptidealring, so ist das Verfahren wie im Fall $f(x) = x^d + c$ durchführbar. Natürlich läßt sich die Irreduzibilität nicht mehr mit IRRED1 überprüfen, aber man kann sie zumindest mit IRRED2 testen. Auch muß man beachten, daß die Norm von $a + b\rho$ nicht mehr $a^d + c \cdot (-b)^d$ beträgt, sondern durch die etwas komplizierte Formel aus Kapitel 5.2 berechnet werden muß. Sind die Koeffizienten f_i klein, so ergibt sich aber die gleiche Laufzeit für das Verfahren wie im Fall $f(x) = x^d + c$.

ad 2. Ist $\mathbb{Z}[\rho]$ nicht mit dem Ring der ganzen algebraischen Zahlen G identisch, so kann man Satz 16 nicht mehr uneingeschränkt anwenden, d.h. die zu den Paaren (p, cp) gehörenden Ideale $\mathcal{P}_{p,cp}$ sind nicht mehr unbedingt Primideale.
Man definiert

Definition 25

Sei $\mathbb{Q}[\rho]$ ein algebraischer Zahlkörper vom Grad d und $W = w_1, \ldots, w_d$ eine Menge von d Elementen aus $\mathbb{Q}[\rho]$.

$D(W) := det(\sigma_i(w_j))^2$, $\quad 1 \le i, j \le d$, heißt **Diskriminante** von W.

Aus [5] [1] geht hervor, daß die Zerlegung von (p) mit Satz 16 in Primideale für $p \in \mathbb{P}$ mit

$$(7.1.1) \qquad p \not| \frac{D(\mathbb{Z}[\rho])}{D(G)},$$

wobei $\mathbb{Z}[\rho] = (1, \rho, \ldots, \rho^{d-1})$ und $G = (\omega_1, \ldots, \omega_{d-1})$, auch im Falle G$\neq \mathbb{Z}[\rho]$ möglich ist.

Für das 'Number Field Sieve' bedeutet diese Erkenntnis, daß für die p, die (7.1.1) nicht erfüllen, $\mathcal{P}_{p,cp}$ kein Primideal der Norm p sein muß, d.h. die ERZEUGER Algorithmen finden keinen Erzeuger der Norm p.
In diesem Fall läßt man einfach die Paare (p, cp), mit den Primzahlen p weg, die (7.1.1) nicht erfüllen, d.h. man verzichtet auf solche guten Paare (a, b), für die $N(a + b\rho)$ durch p teilbar ist. Da die Anzahl der Primzahlen, die (7.1.1) nicht erfüllen, im allgemeinen gering ist, geht auch nur eine geringe Anzahl von Primelementen und damit von guten Paaren verloren, so daß das Verfahren in ansonsten gewohnter Weise die Teiler von n berechnet.

ad 3. Ist G kein Hauptidealring, dann kommt es vor, daß das zu einem Paar (p, cp) gehörende Primideal $\mathcal{P}_{p,cp}$ kein Hauptideal ist, d.h. die ERZEUGER Algorithmen werden in diesem Fall keinen Erfolg haben. Ein Weg auch solche Primideale zu

[1] I.3 Bemerkung zu Prop. 25 und III.3 Prop. 16

nutzen, führt über die Klassenzahl des betrachteten Zahlkörpers:

Definition 26[2]

Ist $\mathcal{A}$ ein Ideal und $c \neq 0 \in$G, dann nennt man $c^{-1} \cdot \mathcal{A}$ ein **gebrochenes Ideal**.

Definition 27[3]

Sei F die Gruppe der gebrochenen Ideale bezüglich der Multiplikation und H die Menge der gebrochenen Hauptideale in einem Zahlkörper, so nennt man die Zahl $h = |F/H|$ die **Klassenzahl** des Zahlkörpers.

Satz 24[4]

Sei h die Klassenzahl von $\mathbb{Q}[\rho]$ und $\mathcal{A}$ ein Ideal von G.
Dann gilt : $\mathcal{A}^h$ ist ein Hauptideal.

Ist h die Klassenzahl des gerade betrachteten Zahlkörpers $\mathbb{Q}[\rho]$, so gilt also für die Primideale $\mathcal{P}_{p,cp}$, die kein Hauptideal sind, daß ${\mathcal{P}_{p,cp}}^h$ ein Hauptideal ist. Deshalb ändert man den ERZEUGER2-Algorithmus derart um, daß die Erzeuger des Ideals ${\mathcal{P}_{p,cp}}^h$ berechnet werden. Diese speichert man gemeinsam mit dem Paar (p, cp) ab. Stellt man dann bei der Zerlegung von $a + b\rho$ fest, daß $a + b\rho$ in $\mathcal{P}_{p,cp}$ liegt, so berechnet man den Exponenten r mit

$$N(a + b\rho) = p^r \cdot q \quad \text{und} \quad ggT(p, q) = 1.$$

Gilt $h|r$, so setzt man den Arrayeintrag [5] $E(p, cp)$ auf $\frac{r}{h}$, da der Erzeuger des zu (p, cp) korrespondierenden Ideals die Norm p^h hat.
Ist h kein Teiler von r, dann betrachtet man nicht mehr die Zerlegungen von $a+bm$ und $a + b\rho$, sondern die von $(a + bm)^h$ und $(a + b\rho)^h$. Gilt nämlich die Kongruenz

$$a + bm \equiv \Phi(a + b\rho) \mod n,$$

so ist

$$(a + bm)^h \equiv (\Phi(a + b\rho))^h \equiv \Phi((a + b\rho)^h) \mod n$$

ebenfalls gültig. In der Zerlegung von $a + bm$, die ABM liefert, wird jeder Eintrag im Array E, also jeder Exponent, mit h multipliziert. Ähnlich verfährt man im ABRHO-Algorithmus. Dort werden alle Einträge außer $E(p, cp)$ mit h multipliziert. $E(p, cp)$ erhält den Wert r, was allerdings der Norm $p^{r \cdot h}$ entspricht.
Die anderen Schritte des 'Number Field Sieve' laufen in gewohnter Weise ab.

[2][13], Seite 116,117
[3][13], Seite 165
[4][13], IX, Proposition 9.8
[5]vgl. Kapitel 3.6 ABRHO-Algorithmus

Kapitel 8

Die Praxis

Die erste Faktorisierung mit dem 'Number Field Sieve' gelang J.M.Pollard. Er bestimmte die, allerdings schon bekannten, Teiler der 39-stelligen Zahl $F_7 = 2^{(2^7)} + 1$. Hier sind einige interessante Daten dieser Faktorisierung :

$$n = 2 \cdot F_7 = 2 \cdot 340282366920938463463374607431768211457$$

$$f(x) = x^3 + 2$$

$$m = 2^{43}$$

$$|B_1| = 500 \Rightarrow B_1 = \{2, \ldots, 3571\}.$$

Hieraus ergab sich :

$$|B_2| = 497$$

$$U = \{-1, 1 + \rho\} \Rightarrow |U| = 2.$$

Die gesamte Faktorbasis B hatte somit eine Größe von

$$|B| = |B_1| + |B_2| + |U| = 999.$$

Die Grenzen für die guten Paare waren

$$A_{min} = -4800\,,\ A_{max} = 4800 \quad \text{und } B_{max} = 2000.$$

Pollard benutzte zum Faktorisieren die LP- nicht aber die LPI- bzw. die LPPI-Variante.[1] Dabei wurden nur solche Large Prime berücksichtigt, die kleiner als 10000 waren.

Insgesamt berechnete Pollard auf einem Philips P2012 in 20 Stunden 1018 Kongruenzen, die zum Bestimmen der beiden Faktoren

$$59649589127497217 \quad \text{und} \quad 5704689200685129054721.$$

ausreichten.

[1]siehe auch Kapitel 4.1, die Large Prime Variante

Ein Aspekt des 'Number Field Sieve' in der Praxis soll nicht unerwähnt bleiben. Es handelt sich hierbei um die gute Eignung des 'Number Field Sieve' für verteilte Anwendung :

Hat man mehrere Rechner zur Verfügung, so ist es möglich, die Berechnung der guten Paare, gesteuert von einem Zentralprozessor, auf andere Prozessoren zu verteilen. Dazu wird die Menge $\{1, \ldots, B_{max}\}$ in mehrere paarweise disjunkte Teilmengen T_i unterteilt. Jeder der beteiligten Rechner siebt nun für eine solche Teilmenge T_i alle guten Paare (a, b) heraus. Diese werden dem Zentralprozessor mitgeteilt, welcher die entsprechenden Kongruenzen berechnet, sammelt und schließlich daraus die Teiler ermittelt.

Analog kann man die verteilte Anwendung auch beim Erzeugen der Paare (p, cp) und der zugehörigen Primidealerzeuger mit den PCP-Algorithmen und dem ERZEUGER2-Algorithmus benutzen. Hierbei wird allerdings nicht $\{ 1, \ldots, B_{max}\}$, sondern B_1 unterteilt.

Mit Hilfe der verteilten Anwendung (mehrere hundert CVAX's wurden benutzt) gelang es, die 107-stellige Zahl

$$n = 3^{239} - 1$$

zu faktorisieren. Die Daten dieser Faktorisierung sind:

$$f(x) = x^5 - 3$$

$$|B_1| = 40000$$

$$B_{max} = 120000$$

$$A_{min} = -5000000$$

$$A_{max} = 5000000$$

Die benötigte Gesamtlaufzeit betrug etwa 16 Tage, dann waren die 5 Faktoren gefunden:

* 2
* 479
* 17209
* 43301964055635533333945745533106128044213
* 1507825692437133671959981388935564985777556063472074560073441273431.

Betrachtet man sich die Parameter dieses Programms genauer, so stößt man auf ein Problem, das bei der Umsetzung des 'Number Field Sieve' in die Praxis vorkommt. Ich meine das Speicherplatzproblem. Das Siebarray im obigen Beispiel hat eine Länge von zehn Millionen Einträgen. Selbst wenn man logarithmisch siebt und die dabei vorkommenden Einträge durch Runden auf Short-Format bringt, bedeutet dies bei zwei Byte pro Short immer noch einen Speicherbedarf von zwanzig Megabyte. Dieses Problem bekommt man aber in den Griff, wenn man das Siebintervall in mehrere Abschnitte unterteilt und diese einzelnen Abschnitte nacheinander durchsiebt.

Beschließen möchte ich meine Ausführungen mit einem Beispiel, das ich mit Jürgen Loho an der Universität des Saarlandes auf einer Sun 4/60 Sparc Station 1 durchgerechnet habe.
Hier die Daten der Faktorisierung:

$$n = (3^{29})^5 + 3$$

$$f(x) = x^5 + 3$$

$$|B_1| = 6000 \Rightarrow |B_2| = 5881$$

Die Grenzen für die guten Paare waren

$A_{min} = -55000$, $A_{max} = 55000$, $B_{max} = 25000$.

Wir benutzten die LPPI-Variante, Logarithmisches Sieben und Logarithmisches Zerlegen. Nach 21 Stunden 20 Minuten waren 12047 Kongruenzen der Form (3.1.6) berechnet. Diese führten zu den Teilern:

- 6,
- 76801,
- 21523361,
- 24127552321,
- 77826286120321,
- 8175496684114031849318756820384 1.

Anhang A

Literaturverzeichnis

[1] **P.Domich, R.Kannan, L.Trotter**
Hermite normal form computation using modulo determinant arithmetic, Mathematics of operations research Vol.12, Februar 1987

[2] **H.M.Edgar**
A first course in number theorie, Wadsworth Publishing Company

[3] **E.Hecke**
Algebraische Zahlen, Chelsea Publishing Company, Bronx, New York

[4] **D.Knuth**
The art of computer programming vol.2, second edition, Addison-Wesley

[5] **S.Lang**
Algebraic number theory, Graduate Texts in Mathematics, Springer Verlag

[6] **S.Lang**
Algebra, Addison-Wesley

[7] **A.K.Lenstra, H.W.Lenstra Jr.**
Algorithms in number theorie, erschienen in J.van Leeuwen's Handbook of theoretical computer science, North-Holland, Amsterdam

[8] **A.K.Lenstra, H.W.Lenstra Jr., M.S.Manasse, J.M.Pollard**
The number field sieve, Symposium on Theory of Computing 1990, Seite 564 ff.

[9] **J.Loho**
Die Implementierung des 'Number Field Sieve', Manuskript

[10] **A.Pethö**
Vorlesungsmitschrift Computeralgebra I, Wintersemester 1990/91, Universität des Saarlandes, Saarbrücken

[11] **Pohst & Zassenhaus**
Algorithmic algebraic number theorie, Encyclopedia of mathematics and it's applications, Cambridge University Press

[12] **J.M.Pollard**
Factoring with cubic integers, Manuskript

[13] **I.N.Stewart, D.O.Tall**
Algebraic number theory, second edition, Chapman and Hall

[14] **B.L. van der Waerden**
Algebra I, 8.Auflage, Springer Verlag

[15] **D.H.Wiedemann**
Solving sparse linear equations over finite fields, IEEE Trans. Inform. Theory, IT-32, 1986, Seiten 54-62