Faktorisierung von Polynomen über endlichen Körpern

Gutachter:
Dr. Michael Hortmann
Prof. Dr. Jens Gamst

Diplomarbeit im Fachbereich Mathematik, Universität Bremen,
vorgelegt von Wenke Sietas im Januar 1999.

Ich bedanke mich bei meinen beiden Betreuern, meinen Freundinnen und Freunden, Kommilitoninnen und Kommilitonen und bei meinen Eltern.

Inhaltsverzeichnis

Verzeichnis der Algorithmen

Einleitung

Die Frage nach der Faktorisierung von Polynomen in einer Variablen über endlichen Körpern stellt sich in vielen Zusammenhängen. Man benutzt diese Faktorisierungsverfahren zum Beispiel, um Polynome über $\mathbb{Z}$ zu faktorisieren, siehe dazu Beauzamy, Trevisan und Wang in [1].

Eine zweite Anwendung finden die von mir vorgestellten Algorithmen bei Verfahren zur Faktorisierung natürlicher Zahlen, siehe dazu Cohen in [10]. Dies ist für die Beurteilung der Public Key Kryptosysteme von nicht zu unterschätzender Bedeutung.

Die dritte zu erwähnende, weiterführende Anwendung ist die Unterstützung bei der Faktorisierung von Polynomen in mehreren Veränderlichen über endlichen Körpern, siehe dazu zum Beispiel von zur Gathen und Kaltofen in [59], Wan in [61] oder Geddes, Czapor und Labahn in [14]. Damit werden auch Polynome über rationalen Zahlkörpern faktorisierbar.

Es gibt noch weitere Anwendungsmöglichkeiten: Meine Arbeit entstand an der Universität Bremen in der Gründungsphase der wissenschaftlichen Einheit „Algorithmische Zahlentheorie, Algebraische Geometrie und Kryptographie“ (AlZAGK). Ich möchte daher im ersten Abschnitt als Motivation auf eine kryptographische Anwendung näher eingehen, das Public Key Verschlüsselungsverfahren von Chor und Rivest [9]. Außerdem gehe ich der Frage nach, wieviele irreduzible Polynome eines gewissen Grades es gibt. In Abschnitt zwei stelle ich die Werkzeuge zur Verfügung, die ein Polynom in die zur weitergehenden Faktorisierung notwendige Form bringen. Der dritte Abschnitt ist Polynomen gewidmet, die in lineare Faktoren zerfallen. Für solche Polynome stelle ich spezielle Verfahren vor, welche die Nullstellen finden. Abschnitt vier behandelt die Berlekamp–Unteralgebra, auf dieser Theorie bauen die verschiedenen Faktorisierungsmethoden aus Abschnitt fünf auf. Abschnitt sechs stellt den innovativen Zugang von Niederreiter vor, der sich aber als dem Zugang von Berlekamp ähnlich erweisen wird. Eine abschließende Betrachtung mit einem Vergleich der Laufzeiten findet in Abschnitt sieben statt.

In der Literatur zu meinem Thema waren für mich Lidl und Niederreiter [31], Menezes [35], Gao und Gathen [13] und von zur Gathen und Shoup [60] wichtige Quellen.

Die vorgestellten Verfahren sollen in der Praxis angewendet werden können, darum habe ich sie mit dem Computer–Algebra–System Maple implementiert, die Programme befinden sich auf der beigelegten CD–ROM. Um auch denjenigen Leserinnen und Lesern, die gerade keinen Computer zur Hand haben, einen Einblick in die Umsetzung der vorgestellten Theorien geben zu können, habe ich die Algorithmen außerdem in Pseudo–Code nach dem Vorbild von Cohen [10] aufgeschrieben. Zu Beginn eines jeden Pseudo–Codes verweise ich auf das entsprechende Programm auf der CD. Durch die Programmierung in Maple ist die Effizienz der Algorithmen nicht optimal. Ein Vergleich der Güte der verschiedenen Verfahren ist aber auch so möglich.

1 Rucksack–Verschlüsselungsverfahren

Es gibt in der Kryptographie die gegensätzlichen Konzepte der symmetrischen Verschlüsselungsverfahren und der Public Key Kryptographie. Bei symmetrischen Verfahren einigen sich zwei Teilnehmerinnen bei jedem Datenaustausch erneut über den zu verwendenden Schlüssel, den beide kennen. In der Public Key Kryptographie existieren für jede Teilnehmerin ein öffentlicher und ein geheimer Schlüssel: will Beatrice an Anna eine codierte Nachricht schicken, so benutzt sie den öffentlichen Schlüssel von Anna, ohne Annas geheimen Schlüssel zu kennen.

Bei beiden Konzepten werden sogenannte Einwegfunktionen benutzt. Man bezeichnet als Einwegfunktionen Abbildungen, die sich relativ leicht berechnen lassen, deren Umkehrung jedoch erheblich schwieriger ist. Man vermutet, daß dem in der Komplexitätstheorie bekannten NP–vollständigen Rucksack–Problem eine Einwegfunktion zugrunde liegt:

Rucksack–Problem: Eine Menge $A = \{a_i \mid 0 \leq i < k, k \in \mathbb{N} \setminus 0\}$ nicht negativer ganzer Zahlen[1] sei gegeben, außerdem $S \in \mathbb{N} \setminus \{0\}$. Gibt es eine Lösung für

$$\sum_{i=0}^{k-1} y_i a_i = S \text{ mit } y_i \in \{0,1\}? \tag{1}$$

Variante des Rucksack–Problems: (siehe dazu Chor und Rivest in [9])
Eine Menge $A = \{a_i \mid 0 \leq i < k, k \in \mathbb{N} \setminus \{0\}\}$ nicht negativer ganzer Zahlen sei gegeben, außerdem $S, s \in \mathbb{N} \setminus \{0\}$. Gibt es eine Lösung für

$$\sum_{i=0}^{k-1} y_i a_i = S \text{ mit } y_i \in \{0,1\} \text{ und } \sum_{i=0}^{k-1} y_i = s? \tag{2}$$

Die Einwegfunktion ist hier also das Summieren, sie ordnet einem k–Tupel von Nullen und Einsen eine natürliche Zahl S zu. Die Umkehrung ist schwierig, da es schwer fällt, bei Kenntnis der Summe S die einzelnen Summanden zu erraten.

Bemerkung 1.1
Das in der Variante des Rucksack–Problems genannte s heißt Gewicht.

Bemerkung 1.2
Eine schöne Einführung in kryptographische Konzepte und Verfahren, zum Beispiel Rucksack–Verschlüsselungsverfahren, gibt B. Schneier in [50], darum reiße ich das Thema nur an. Die Klasse der NP–vollständigen Probleme wird bei N. Koblitz in [22] beschrieben.

Rucksack–Verschlüsselungsverfahren beruhen auf der Schwierigkeit, bei bekanntem S ohne Zusatzinformationen eine Lösung für (1) beziehungsweise (2) zu finden, während die Lösung mit einer geheimen Zusatzinformation leicht sein soll. Jede Teilnehmerin veröffentlicht eine Menge A und ein Gewicht s. Hierbei werden die Elemente $a_i \in A$ so gewählt, daß (1) beziehungsweise (2) leicht lösbar ist, wenn die Teilnehmerin einen geheimen Schlüssel kennt. Die Lösung soll eindeutig sein, da sich hinter den y_i die übermittelte Nachricht m verbirgt.

Will Beatrice mit einem Verschlüsselungsverfahren, welches auf der Variante des Rucksack–Problems beruht, eine im Binärcode gegebene Nachricht m verschlüsseln, so muß sie nur, wie ich im folgenden Abschnitt 1.1 beschreiben werde, die Nachricht in die Form von k–Tupeln $y = (y_0, y_1, \ldots, y_{k-1})$ mit Gewicht s bringen und danach die Summen

$$E(y) = \sum_{i=0}^{k-1} y_i a_i$$

[1] Notation: $\mathbb{N} = \{0, 1, 2, \ldots\}$

bilden.

Eine Lauscherin kann die übermittelten k–Tupel y nicht erraten, da sie das Rucksack–Problem nicht lösen kann. Sie kann nur versuchen, an die Zusatzinformationen, das heißt den geheimen Schlüssel, zu gelangen und so die Nachricht zu entschlüsseln. Schlechte Rucksack–Verschlüsselungsverfahren arbeiten mit einer Menge A, aus der übermittelte Block y zu entnehmen ist, siehe dazu bei Schneier in [50] unter dem Stichwort „Superincreasing Rucksack".

1.1 Die Umwandlung binärer Texte in Folgen von Tupeln mit Gewicht s

Eine Nachricht ist im Normalfall eine Folge von Bits, die man jederzeit in k–Tupel unterteilen kann. Diese k–Tupel haben aber nicht notwendigerweise die Eigenschaft, daß ihr Gewicht nicht größer als s ist. Beatrice muß die geheime Botschaft m, die sie Anna schicken möchte, also erst mit einer umkehrbaren Abbildung in die entsprechende Form bringen. In [9] schlagen Chor und Rivest dazu das folgende Verfahren vor:

Anna setzt $k = p$ mit p prim. Beatrice teilt den binären Text zuerst auf in Blöcke von $\lfloor \log_2 \binom{p}{s} \rfloor$ Bits. Diese Blöcke interpretiert sie als die binäre Darstellung einer Zahl z mit $0 \leq z < \binom{p}{s}$. Um die Zahl z auf ein p–Tupel vom Gewicht s abzubilden, führt Beatrice für $i = 1, \ldots, p$ die folgenden Schritte durch: Ist z größer als $\binom{p-i}{s}$, so setzt Beatrice das i–te Bit im Vektor auf 1, sonst auf 0. Falls $z > \binom{p-i}{s}$ war, so aktualisiert sie z auf $z - \binom{p-i}{s}$ und s auf $s - 1$.

Algorithmus 1
Eingabe: Eine natürliche Zahl z mit $0 \leq z < \binom{p}{s}$, die Primzahl p und das Gewicht s.
Ausgabe: Ein binärer Block y der Länge p mit Gewicht s.
Programm: umwandeln.mws: umwandeln

1. Setze $i \leftarrow 1$.
2. Ist $z < \binom{p-i}{s}$, so setze $y_i \leftarrow 0$, sonst setze $y_i \leftarrow 1$, $z \leftarrow z - \binom{p-i}{s}$ und $s \leftarrow s - 1$.
3. Ist $i < p$, so setze $i \leftarrow i + 1$ und gehe zu Schritt 2.
4. Gib y aus.

Um eine Vorstellung von der Wirkungsweise des Verfahrens zu bekommen, gebe ich hier ein Beispiel:

Beispiel 1.1
Sei $p = 7$, $s = 3$, dann muß Beatrice ihre binäre Nachricht in $\lfloor \log_2 \binom{p}{s} \rfloor = \lfloor 5.129283016 \rfloor = 5$ Zeichen lange Blöcke aufteilen. Beatrice möchte die Zahl drei, die dem Block $z = 00011$ entspricht, an Anna schicken. Beim Umwandeln in einen Block y der Länge $p = 7$ mit dem Gewicht $s = 3$ berechnet Beatrice:

- $\binom{p-1}{s} = 20 \geq z$, *also ist* $y_1 = 0$,
- $\binom{p-2}{s} = 10 \geq z$, *also ist* $y_2 = 0$,
- $\binom{p-3}{s} = 4 \geq z$, *also ist* $y_3 = 0$,
- $\binom{p-4}{s} = 1 < z$, *also ist* $y_4 = 1$, *setze* $z \leftarrow z - 1 = 2$ *und* $s \leftarrow s - 1 = 2$,
- $\binom{p-5}{s} = 1 < z$, *also ist* $y_5 = 1$, *setze* $z \leftarrow z - 1 = 1$ *und* $s \leftarrow s - 1 = 1$,
- $\binom{p-6}{s} = 1 < z$, *also ist* $y_6 = 1$, *setze* $z \leftarrow z - 1 = 0$ *und* $s \leftarrow s - 1 = 0$,
- $\binom{p-7}{s} = 1 \geq z$, *also ist* $y_7 = 0$.

Beatrice wird also den umgewandelten Nachrichtenblock $y = 0001110$ der Länge sieben vom Gewicht drei verschlüsseln.

Ein derart veränderter Text muß natürlich wieder in seine ursprüngliche Form gebracht werden, bevor Anna ihn lesen kann. Die inverse Prozedur geht wie folgt: Ist der i–te Eintrag des Vektors 1, so berechnet Anna $g_i = \binom{p-i}{s}$, ist der i–te Eintrag 0, so setzt sie $g_i = 0$. Die Nachricht von Beatrice ist nun die Binärdarstellung der Summe der g_i.

Algorithmus 2
Eingabe: Ein binärer Block y der Länge p und vom Gewicht s, die Primzahl p und das Gewicht s.
Ausgabe: Eine natürliche Zahl z.
Programm: umwandeln.mws: rueckwandeln

1. Setze $z \leftarrow 0$ und $i \leftarrow 1$.
2. Ist $y_i = 1$, so setze $z \leftarrow z + \binom{p-i}{s}$ und $s \leftarrow s - 1$.
3. Ist $i < p$, so setze $i \leftarrow i + 1$ und gehe zu Schritt 2.
4. Gib z aus.

Da ich den Beweis dafür, daß dieser Algorithmus das Inverse zum vorherigen ist, hier nicht führen möchte, hoffe ich, das Vertrauen in diese Tatsache durch das folgende Beispiel verstärken zu können.

Beispiel 1.2
Anna entschlüsselt die von Beatrice geschickte Botschaft und erhält den Vektor $y = 0001110$. Anna weiß, daß $p = 7$ und $s = 3$ ist, das Ergebnis ihrer Invertierung muß also ein binärer Block der Länge $\lfloor \log_2 \binom{p}{s} \rfloor = \lfloor 5.129283016 \rfloor = 5$ sein.

Da y_4, y_5 und y_6 nicht 0 sind, startet Anna mit $z = 0$ und berechnet

- $\binom{p-4}{s} = 1$, *sie setzt also* $z \leftarrow z + 1 = 1$ *und* $s \leftarrow s - 1 = 2$,
- $\binom{p-5}{s} = 1$, *sie setzt also* $z \leftarrow z + 1 = 2$ *und* $s \leftarrow s - 1 = 1$,
- $\binom{p-6}{s} = 1$, *sie setzt also* $z \leftarrow z + 1 = 3$.

Anna stellt z binär dar als $z = 00011$, den von Beatrice gesandten Block der Länge fünf der geheimen Botschaft.

Wie Anna und Beatrice einen Text in die für Rucksack–Verschlüsselungsverfahren notwendige Form und wieder zurück umwandeln, ist somit geklärt. Jetzt interessieren sie sich für ein spezielles Rucksack–Verschlüsselungsverfahren, nämlich das von Chor und Rivest.

1.2 Das Rucksack–Verschlüsselungsverfahren von Chor und Rivest

In [9] stellen B. Chor und R. L. Rivest das von ihnen entwickelte Rucksack–Verschlüsselungsverfahren vor, das ich hier beschreiben möchte. Leider wurde es vor kurzem in [57] gebrochen. Dennoch möchte ich es als ein Beispiel für die Anwendung der Faktorisierung von Polynomen über endlichen Körpern anführen.

Chor und Rivest entwickeln ihr Verfahren auf der Basis der Variante des Rucksack–Problems.

So wird das System erzeugt:

1. Anna wählt eine Primzahl p und eine natürliche Zahl $s \geq 1$, so daß $\mathbb{F}_{p^s}$ für die Berechnung diskreter Logarithmen geeignet ist, also so, daß $p^s - 1$ in kleine Primfaktoren zerfällt, siehe dazu zum Beispiel S. C. Pohlig und M. E. Hellmann in [44]. Sie setzt $q = p^s$.
2. Anna wählt weiterhin ein zufälliges primitives Element $g \in \mathbb{F}_q$, wobei $\mathbb{F}_q = \mathbb{F}_p[x]/f\mathbb{F}_p[x]$ für ein normiertes irreduzibles Polynom $f \in \mathbb{F}_p[x]$ mit $\mathrm{Grad}\, f = s$ ist. g wird also repräsentiert durch ein Polynom aus $\mathbb{F}_p[x]$ mit $\mathrm{Grad}(g) < s$.

3. Sie bestimmt nun für jede natürliche Zahl $0 \leq i < p$ den Exponenten $a_i \in \mathbb{N}$ mit $0 \leq a_i < p^s$, für den gilt

$$(x - i) \equiv g(x)^{a_i} \mod f(x).$$

4. Anna wählt eine Permutation π der Menge $\{0, 1, \ldots, p-1\}$ und das zufällige „Rauschen“ d mit $0 \leq d < q - 1$. Sie definiert

$$b_i = a_{\pi(i)} + d \text{ für } 0 \leq i < p.$$

Anna veröffentlicht $b_0, b_1, \ldots, b_{p-1}$ und q, für sich behält sie den geheimen Schlüssel bestehend aus g, d und π. Ob sie f veröffentlicht oder nicht, ist nicht von Belang, da die durch die möglichen f definierten Körper isomorph zueinander sind.

Nachrichten m werden in eine Folge von p–Tupeln $y = (y_0, y_1, y_2, \ldots, y_{p-1})$ mit $y_i \in \{0, 1\}$ für alle $0 \leq i < p$ und $\sum_{i=0}^{p-1} y_i = s$ unterteilt. Wie Beatrice eine beliebige Nachricht in diese Form bringt, ist aus Abschnitt 1.1 auf Seite 3 bekannt. Beatrice möchte nun einen solchen geheimen Block y an Anna senden.

Beatrice codiert y mit der Verschlüsselungsfunktion

$$E: \quad \{0,1\}^p \longrightarrow \mathbb{Z}_{q-1},\ E(y) \equiv \sum_{i=0}^{p-1} y_i b_i \mod q - 1.$$

Anna decodiert die verschlüsselte Nachricht $v = E(y)$, indem sie $v' = v - sd \mod q - 1$ und dann

$$g(x)^{v'} \equiv \prod_{i=0}^{p-1} (x - \pi(i))^{y_i} \mod f(x)$$

berechnet. $D(v')$ soll das Polynom kleinsten Grades in der Äquivalenzklasse von $g^{v'}$ modulo f sein. $\prod_{i=0}^{p-1}(x - \pi(i))^{y_i}$ ist normiert mit den Grad $s = \mathrm{Grad}(f)$, $D(v')$ hingegen hat einen Grad, der kleiner als s ist. Darum ist $D(v') + f(x)$ normiert mit Grad s, also faktorisiert Anna das Polynom $D(v') + f$.

Durch die Feststellung der Vielfachheiten (0 oder 1) der Nullstellen bei der Faktorisierung des Polynoms $D(v') \mod f$ und die Anwendung der zu π inversen Permutation erhält Anna den Block y.

Bemerkung 1.3
Fordert man nicht $\sum_{i=0}^{p-1} y_i = s$, sondern nur $\sum_{i=0}^{p-1} y_i \leq s$, so muß man ohne das Rauschen d arbeiten, da nicht klar ist, wie oft man es von der Nachricht v abziehen muß. Der Vorteil wäre aber, daß man $s = p$ wählen könnte und so auf das Rucksack–Problem zurückkäme, bei dem das Umwandeln der Bit–Folgen in Vektoren vom Gewicht s gespart wird.

Eine weitere Verallgemeinerung ist leicht möglich, indem man für die y_i beliebige natürliche Zahlen zulässt.

Bemerkung 1.4
Das zu faktorisierende Polynom $D(v')$ besteht nur aus linearen Faktoren, darum braucht man für die Faktorisierung nicht unbedingt echte Faktorisierungs–Algorithmen, wie sie in Abschnitt 5 beschrieben werden, sondern es reichen die Algorithmen aus Abschnitt 3 zum Finden der Nullstellen.

Bemerkung 1.5
Für eine Lauscherin Cathrin sind y, g, d und π nicht bekannt. Ist f nicht bekannt, so kann sie sich ein beliebiges $\mathbb{F}_q$ definierendes f wählen, da alle Körper der Ordnung q isomorph zueinander sind. Cathrin versucht nun, einen Erzeuger $g \in \mathbb{F}_q$ zu finden, so daß die Abstände der Logarithmen denen der b_i entsprechen. Wie Cathrin das System genau knackt, wird in [57] beschrieben.

Beispiel 1.3
Anna wählt $p = 7$, $s = 3$ und somit $q = 343$. Als $\mathbb{F}_{343}$ definierendes normiertes Polynom wählt Anna $f(x) = x^3 + 2$, primitives Element ist $g(x) = x^2 + 5$.

Anna berechnet die Logarithmen als $a_0 = 95$, $a_1 = 82$, $a_2 = 4$, $a_3 = 201$, $a_4 = 142$, $a_5 = 159$ und $a_6 = 153$. Anna entscheidet sich für die Permutation π, die die Menge A der Größe nach sortiert, und für

$d = 11$. *Sie veröffentlicht somit die Menge* $\{b_0 = 15, b_1 = 93, b_2 = 106, b_3 = 153, b_4 = 164, b_5 = 170, b_6 = 212\}$.

Beatrice möchte Anna den Block $m = 0001110$ *der Länge sieben vom Gewicht drei codiert schicken. Sie berechnet* $E(m) = 153 + 164 + 170 \equiv 145 \mod 342$ *und schickt diese Summe an Anna.*

Anna erhält die Nachricht $v = E(m)$ *und berechnet* $v' = v - sd = 145 - 3 \cdot 11 \equiv 112 \mod 342$. *Nun bildet sie*

$$g(x)^{v'} \equiv 6x^2 + 4x + 4 \mod f(x)$$

und faktorisiert

$$D(v') + f(x) = (6x^2 + 4x + 4) + (x^3 + 2) \equiv x^3 + 6x^2 + 4x + 6 \equiv (x-4) \cdot (x-6) \cdot (x-5) \mod 7.$$

Mit der Invertierung von π *erhält Anna den Block* 0001110.

1.3 Faktorisierung beim Entschlüsseln

Das vorangegangene Kapitel zeigt, daß man die Faktorisierung von Polynomen über endlichen Körpern zum Beispiel beim Entschlüsseln von Nachrichten brauchen kann, die mit dem Verfahren von Chor und Rivest verschlüsselt wurden. Bei Verschlüsselungsverfahren darf es zwar einen gewissen Aufwand kosten, die Schlüssel zu erstellen, das Ver– und Entschlüsseln im täglichen Gebrauch soll jedoch schnell vonstatten gehen. Dies nehme ich zum Anlaß, mich in den folgenden Abschnitten mit effizienten Faktorisierungsmethoden zu beschäftigen.

Grundsätzlich passiert jede Faktorisierung mit dem folgenden Gerüst eines Algorithmus:

Algorithmus 3 FAKTORISIERUNG $(f, \mathbb{F}_q)$
Eingabe: Das normierte Polynom f und der endliche Körper $\mathbb{F}_q$ mit $f \in \mathbb{F}_q[x]$.
Ausgabe: f als Produkt von Potenzen normierter irreduzibler Polynome.

1. Zerlege f in quadratfreie Teilpolynome g_i mit $f = g_1^1 \cdots g_k^k$.
2. Zerlege diese Teilpolynome g_i in Polynome $h_{i,d}$, bei denen der Grad d der irreduziblen Faktoren bekannt ist.
3. Zerlege die verschiedenen Polynome $h_{i,d}$ in irreduzible Faktoren vom Grad d.
4. Gib f in der gewünschten Form $f_1^{l_1} \cdots f_t^{l_t}$ aus.

Die einzelnen Schritte sind jeweils genauer zu beschreibende Algorithmen. Den ersten Schritt beschreibe ich in Abschnitt 2.1 auf Seite 9, den zweiten in Abschnitt 2.2 auf Seite 13. Für den dritten Schritt gibt es viele verschiedene Strategien, diese darzustellen ist das eigentliche Thema meiner Arbeit, dem ich mich in den Abschnitten 4 bis 6 widme. Der vierte Schritt, die schöne Ausgabe der Faktorisierung von f, ist reine Formsache.

Von gewissem Interesse ist noch die Frage, wie wohl die Zerlegung eines zufälligen Polynoms aussieht. Um diese Frage klären zu können, benötige ich die folgenden Aussagen.

Satz 1.6
Ist $\mathbb{F}_q$ *ein endlicher Körper und* $g \in \mathbb{F}_q[x]$ *irreduzibel mit* $\mathrm{Grad}(g) = m$, *so gilt für* $d \in \mathbb{N} \setminus \{0\}$

$$g(x) \mid x^{q^d} - x \qquad \Leftrightarrow \qquad m \mid d.$$

Beweis: „$\Rightarrow$“: $g(x)$ teile $x^{q^d} - x$ und α sei eine Nullstelle von g im Zerfällungskörper von g über $\mathbb{F}_q$. Dann ist $\alpha^{q^d} = \alpha$, also ist $\alpha \in \mathbb{F}_{q^d}$. Somit ist $\mathbb{F}_q(\alpha) \subset \mathbb{F}_{q^d}$. Da aber $[\mathbb{F}_q(\alpha) : \mathbb{F}_q] = m$ und $[\mathbb{F}_{q^d} : \mathbb{F}_q] = d$, muß gelten $[\mathbb{F}_{q^d} : \mathbb{F}_q(\alpha)] = d/m \in \mathbb{N}$, also muß m Teiler von d sein.

„$\Leftarrow$": Es werde d von m geteilt, dann ist $\mathbb{F}_{q^m}$ ein Unterkörper von $\mathbb{F}_{q^d}$. Ist α eine Nullstelle von g im Zerfällungskörper über $\mathbb{F}_q$, so ist $[\mathbb{F}_q(\alpha) : \mathbb{F}_q] = m$, also ist $\mathbb{F}_q(\alpha) = \mathbb{F}_{q^m}$. Darum ist $\alpha \in \mathbb{F}_{q^d}$, woraus $\alpha^{q^d} = \alpha$ folgt. Somit ist α eine Nullstelle von $x^{q^d} - x$. g zerfällt in lineare Faktoren $x - \alpha$, welche alle $x^{q^d} - x$ teilen, also wird $x^{q^d} - x$ von $g(x)$ geteilt. □

Die folgenden Aussagen über die Möbius–Funktion sind auch bei R. Lidl und H. Niederreiter in [31] nachzulesen:

Definition 1.7
Die Möbius–Funktion μ ist für $n \in \mathbb{N} \setminus \{0\}$ gegeben als

$$\mu(n) = \begin{cases} 1 & \text{für } n = 1, \\ (-1)^k & \text{falls } n \text{ das Produkt } k \text{ verschiedener Primzahlen ist und} \\ 0 & \text{falls } n \text{ durch das Quadrat einer Primzahl teilbar ist.} \end{cases}$$

Lemma 1.8
Für $n \in \mathbb{N} \setminus \{0\}$ gilt

$$\sum_{d|n} \mu(d) = \begin{cases} 1 & \text{für } n = 1, \\ 0 & \text{falls } n > 1. \end{cases}$$

Beweis: Dies gilt, da für $n > 1$ nur diejenigen d betrachtet werden müssen, für die $\mu(d) \neq 0$ ist, also diejenigen, die entweder 1 oder ein quadratfreies Produkt von Primzahlen sind. Der Fall $n = 1$ ist trivial. Seien nun $p_1, p_2, \ldots, p_k$ die verschiedenen Primteiler von $n > 1$. Dann gilt

$$\begin{aligned} \sum_{d|n} \mu(d) &= \mu(1) + \sum_{i=1}^{k} \mu(p_i) + \sum_{1 \le i_1 < i_2 \le k} \mu(p_{i_1} p_{i_2}) + \cdots + \mu(p_1 p_2 \cdots p_k) \\ &= 1 + \binom{k}{1}(-1) + \binom{k}{2}(-1)^2 + \cdots + \binom{k}{k}(-1)^k \\ &= (1 + (-1))^k = 0. \end{aligned}$$

□

Lemma 1.9
Seien h und H zwei Funktionen von $\mathbb{N} \setminus \{0\}$ in eine additive abelsche Gruppe G. Es ist $H(n) = \sum_{d|n} h(d)$ für alle $n \in \mathbb{N} \setminus \{0\}$ genau dann, wenn

$$h(n) = \sum_{d|n} \mu\left(\frac{n}{d}\right) H(d) = \sum_{d|n} \mu(d) H\left(\frac{n}{d}\right)$$

gilt für alle $n \in \mathbb{N} \setminus \{0\}$.

Beweis: Dies zeigt sich durch

$$\begin{aligned} \sum_{d|n} \mu\left(\frac{n}{d}\right) H(d) = \sum_{d|n} \mu(d) H\left(\frac{n}{d}\right) &= \sum_{d|n} \mu(d) \sum_{c|(n/d)} h(c) \\ = \sum_{c|n} \sum_{d|(n/c)} \mu(d) h(c) &= \sum_{c|n} h(c) \sum_{d|(n/c)} \mu(d) \\ = h(n), \end{aligned}$$

und

$$\sum_{d|n} h(d) = \sum_{d|n} \sum_{c|(n/d)} \mu\left(\frac{n}{cd}\right) H(c) \qquad = \sum_{c|n} \sum_{d|(n/c)} \mu\left(\frac{n}{cd}\right) H(c)$$
$$= \mu(1)H(n) + \sum_{\substack{c|n \\ c\neq n}} H(c) \underbrace{\sum_{d|(n/c)} \mu\left(\frac{n}{cd}\right)}_{0,\ \text{da}\ n/c\neq 1} = H(n).$$

□

Jetzt kann ich die Anzahl der irreduziblen Polynome vom Grad n bestimmen:

Satz 1.10
Die Anzahl $N_q(n)$ der normierten irreduziblen Polynome vom Grad n in $\mathbb{F}_q[x]$ ist

$$N_q(n) = \frac{1}{n}\sum_{d|n} \mu\left(\frac{n}{d}\right) q^d = \frac{1}{n}\sum_{d|n} \mu(d) q^{n/d}.$$

Beweis: $x^{q^n} - x$ ist nach Lemma 1.6 für alle $n \in \mathbb{N} \setminus \{0\}$ das Produkt aller normierten irreduziblen Polynome, deren Grad n teilt, darum ist $q^n = \sum_{d|n} d \cdot N_q(d)$. Sei $h(n) = nN_q(n)$ und $H(n) = q^n$ für alle $n \in \mathbb{N} \setminus \{0\}$, dann gilt $H(n) = \sum_{d|n} h(d)$ und also auch

$$\frac{h(n)}{n} = N_q(n) = \frac{1}{n}\sum_{d|n} \mu\left(\frac{n}{d}\right) H(d) = \frac{1}{n}\sum_{d|n} \mu\left(\frac{n}{d}\right) q^d$$

□

Damit ist es mir möglich, die Wahrscheinlichkeit für die Irreduzibilität eines Polynoms vom Grad n zu schätzen:

Satz 1.11
Ein zufällig gewähltes Polynom $f \in \mathbb{F}_q[x]$ vom Grad $n \in \mathbb{N} \setminus \{0\}$ ist mit der Wahrscheinlichkeit $P \sim 1/n$ irreduzibel.

Beweis: Sei f ohne Einschränkung normiert mit Grad n. Die Wahrscheinlichkeit P, daß es sich um ein irreduzibles Polynom handelt, ist $P = \frac{N_q(n)}{q^n}$. Diese Wahrscheinlichkeit ist nach abzuschätzen.

Es ist $q^n = \sum_{d|n} d \cdot N_q(d)$ mit $N_q(1) = q$ und $q^d \geq dN_q(d) \geq 0$ für alle $d \in \mathbb{N}$. Somit ist

$$q^n \geq n \cdot N_q(n) + q \quad \text{und} \quad q^n \leq n \cdot N_q(n) + \sum_{d=0}^{n/2} q^d,$$

mit $\sum_{d=0}^{n/2} q^d \leq q^{n/2+1}$ gilt

$$q^n \leq n \cdot N_q(n) + q^{n/2+1}.$$

Es ergibt sich die Abschätzung

$$\frac{1}{n} - \frac{1}{nq^{n/2-1}} \leq \frac{N_q(n)}{q^n} \leq \frac{1}{n} - \frac{q}{nq^n},$$

also ist $P \sim 1/n$. □

2 Vorbereitung der Faktorisierung

Im gesamten folgenden Text werde ich mich mit Polynomen über dem Körper $\mathbb{F}_q$ mit $q = p^s$, $p \in \mathbb{N}$ prim und $s \in \mathbb{N} \setminus \{0\}$ beschäftigen. Bei Rechnungen innerhalb dieses Körpers $\mathbb{F}_q$ werde ich nicht das Kongruenzsymbol zusammen mit einem $\mathbb{F}_q$ definierenden irreduziblen Polynom benutzen, sondern das Gleichheitszeichen. Die modulare Schreibweise werde ich nur dann verwenden, wenn ich modulo einem Polynom aus $\mathbb{F}_q[x]$ rechne.

Ein Polynom $f(x) = \sum_{i=0}^{n} a_i x^i$ vom Grad $n \geq 1$ ($a_n \neq 0$) mit $a_i \in \mathbb{F}_q$ für alle $0 \leq i \leq n$ besitzt eine eindeutige Faktorisierung der Form $f(x) = a \prod_{i=1}^{t} f_i(x)^{l_i}$ über $\mathbb{F}_q[x]$, wobei die $f_i \in \mathbb{F}_q[x]$ paarweise verschiedene normierte irreduzible Polynome und die $l_i \in \mathbb{N} \setminus \{0\}$ deren Vielfachheiten sind. $a \neq 0$ ist aus dem Körper $\mathbb{F}_q$ und hat das multiplikative Inverse a^{-1}. Das Polynom $a^{-1} \cdot f$ ist normiert, die normierten irreduziblen Faktoren f_i bleiben erhalten. Darum kann ich ohne Einschränkung f als normiert voraussetzen.

2.1 Quadratfreie Polynome

Die Aufgabe, das Polynom f zu faktorisieren, läßt sich reduzieren auf die Faktorisierung mehrerer Polynome f, in denen kein irreduzibler Faktor mehrfach vorkommt, die also quadratfrei sind.

Ich betrachte dazu ein Polynom $f(x) = f_1(x)^k \cdot f_2(x)$, wobei $k \in \mathbb{N} \setminus \{0\}$. f_1 und f_2 sind beliebige, nicht notwendig irreduzible, Polynome aus $\mathbb{F}_q[x]$ mit $\mathrm{ggT}(f_1(x), f_2(x)) = 1$, f_1 ist quadratfrei.

1. Fall, $p \nmid k$: Die Ableitung von f ist

$$\begin{aligned} f'(x) &= k \cdot f_1(x)^{k-1} \cdot f_1'(x) \cdot f_2(x) + f_1(x)^k \cdot f_2'(x) \\ &= f_1(x)^{k-1} \cdot (k \cdot f_1'(x) \cdot f_2(x) + f_1(x) \cdot f_2'(x)) . \end{aligned}$$

Da $k \cdot f_1' \cdot f_2$ nicht von f_1 geteilt wird, enthält der größte gemeinsame Teiler $g = \mathrm{ggT}(f, f')$ den Faktor f_1 genau $k-1$ mal:

$$f_1(x)^{k-1} \mid \mathrm{ggT}(f(x), f'(x)),$$
$$f_1(x)^{k} \nmid \mathrm{ggT}(f(x), f'(x)).$$

2. Fall, $p \mid k$: Die Ableitung von f ist

$$f'(x) = f_1(x)^k \cdot f_2'(x).$$

Der größte gemeinsame Teiler $g = \mathrm{ggT}(f, f')$ enthält den Faktor f_1 genau k mal:

$$f_1(x)^k \mid \mathrm{ggT}(f(x), f'(x)).$$

Aus diesen Überlegungen ergibt sich:

- Ist $g = 1$, so enthält f keinen Faktor mehrfach, ist also „quadratfrei“.
- Ist $g = f$, was gleichbedeutend ist mit $f' = 0$, so ist $f(x) = \sum_{i=0}^{n/p} a_{ip} x^{ip}$, also $f = h^p$ für ein passendes Polynom $h \in \mathbb{F}_q[x]$, nämlich $h(x) = \sum_{i=0}^{n/p} a_{ip}^{1/p} x^i$.
- Ist $g \neq 1$ und $g \neq f$, so ist g ein echter Faktor von f. f/g ist quadratfrei. Durch das Faktorisieren von f/g und g bekomme ich die gesamte Faktorisierung von f, wobei g eventuell noch quadratfrei gemacht werden muß.

Bemerkung 2.1
Ist $q = p$, so gilt $\alpha^{1/p} = \alpha$ für alle $\alpha \in \mathbb{F}_q$. Für $q = p^s$ mit $s \geq 2$ ist $\alpha^{1/p} = \alpha^{p^{s-1}}$.

Aus den drei Punkten kann man zwei Algorithmen ableiten, die ein Polynom f in mehrere quadratfreie Polynome zerlegen. Welcher besser geeignet ist, hängt dabei sehr von der Beschaffenheit des Polynoms ab:

1. Kommen wenige Faktoren mit hohen Vielfachheiten oder
2. viele Faktoren mit kleinen Vielfachheiten vor?

Ein zufällig gewähltes Polynom ist wahrscheinlich von der zweiten Art.

Wenige Faktoren mit hohen Vielfachheiten

Zerfällt f in wenige Faktoren mit hohen Vielfachheiten, etwa $l_i > p$ für einige $1 \leq i \leq t$, so bietet sich das hier beschriebene Verfahren an:

Man leitet so weit wie möglich ab: $g_0 = f$, $g_1 = \mathrm{ggT}(g_0, g_0')$, $g_2 = \mathrm{ggT}(g_1, g_1')$ usw. solange, bis $g_m = 1$ oder $g_{m-1}' = 0$. Ist $g_m = 1$, so war keine der Vielfachheiten l_i größer als p. Wenn jedoch die Vielfachheit l_i größer als p war, so kommt sie in g_{m-1} genau mit der größten durch p teilbaren Vielfachheit vor, die kleiner als l_i ist, also mit dem p–Anteil. An dieser Stelle wird g_{m-1} durch $g_{m-1}^{1/p}$ ersetzt und das Verfahren rekursiv aufgerufen. Die Elemente der Liste von Faktoren, die man auf diese Weise erhält, müssen noch paarweise teilerfremd gemacht werden.

Algorithmus 4 QUADRATFREI $(f, \mathbb{F}_q)$
Eingabe: Ein normiertes Polynom $f \in \mathbb{F}_q[x]$ und der Körper $\mathbb{F}_q$ mit $q = p^s$.
Ausgabe: Die Zerlegung von $f = b_1^{l_1} \cdots b_k^{l_k}$ in paarweise teilerfremde quadratfreie Teilpolynome $b_1, \ldots, b_k$.
Programm: Basis.mws: sqf1

1. Setze $g(x) \leftarrow f(x)$ und $out \leftarrow 1$.
2. Ist $g'(x) = 0$, so gehe zu Schritt 4. Andernfalls setze $f(x) \leftarrow g(x)$, $g(x) \leftarrow \mathrm{ggT}(g(x), g'(x))$ und $out \leftarrow out \cdot (f(x)/g(x))$.
3. Gehe zu Schritt 2.
4. Ist $g(x) \neq 1$, so setze $out \leftarrow out \cdot \left(\text{QUADRATFREI } (g^{1/p}, \mathbb{F}_q)\right)^p$.
5. out ist eine Liste von k Faktoren $b_1(x), \ldots, b_k(x)$ mit Vielfachheiten $l_1, \ldots, l_k$. Setze $i \leftarrow 1$.
6. Setze $j \leftarrow i + 1$.
7. Setze $g(x) \leftarrow \mathrm{ggT}(b_i(x), b_j(x))$. Ist $g(x) = 1$, so gehe zu Schritt 9, sonst zu Schritt 8.
8. Ist $g(x) = b_i(x)$, so setze $b_j(x) \leftarrow b_j(x)/b_i(x)$ und $l_i \leftarrow l_i + l_j$.
   Ist $g(x) = b_j(x)$, so setze $b_i(x) \leftarrow b_i(x)/b_j(x)$ und $l_j \leftarrow l_i + l_j$.
   Sonst setze $b_i(x) \leftarrow b_i(x)/g(x)$, $b_j(x) \leftarrow b_j(x)/g(x)$, $k \leftarrow k+1$, $b_k(x) \leftarrow g(x)$ und $l_k \leftarrow l_i + l_j$.
9. Ist $j \leq k - 1$, so setze $j \leftarrow j + 1$ und gehe zu Schritt 7.
10. Ist $i < k - 1$, so setze $i \leftarrow i + 1$ und gehe zu Schritt 6.
11. Liefere out ohne die Potenzen von 1 zurück.

Beispiel 2.1
Die interessanten Phänomene treten bei diesem Verfahren nur bei Polynomen mit sehr großem Grad auf, darum werde ich in diesem Beispiel nicht konkret rechnen, sondern die Funktionsweise an dem Konzept einer Faktorisierung verdeutlichen.

Sei $q = p = 7$, wir möchten das Polynom $f = g_0 = f_1^8 f_2^9 f_3^{50}$, das wir nur in der Form $f(x) = \sum_{i=0}^n a_i x^i$ vorliegen haben, in quadratfreie Faktoren zerlegen. Dabei sind die Polynome f_i für $i \in \{1, 2, 3\}$ paarweise teilerfremd und quadratfrei.

- *Es ist* $g_1 = \mathrm{ggT}(f, f') = f_1^7 f_2^8 f_3^{49}$, *Teiler ist* $t_1 = f/g_1 = f_1 f_2 f_3$.
- *Es ist* $g_2 = \mathrm{ggT}(g_1, g_1') = f_1^7 f_2^7 f_3^{49}$, *Teiler ist* $t_2 = g_1/g_2 = f_2$.
- *Es ist* $g_2' = 0$, *rechne darum weiter mit* $g_3 = g_2^{1/7} = f_1 f_2 f_3^7$.
- *Es ist* $g_4 = \mathrm{ggT}(g_3, g_3') = f_3^7$, *Teiler ist* $t_3 = g_3/g_4 = f_1 f_2$.
- *Es ist* $g_4' = 0$, *rechne darum weiter mit* $g_5 = g_4^{1/7} = f_3$.
- *Es ist* $g_6 = \mathrm{ggT}(g_5, g_5') = 1$, *das Verfahren bricht ab mit* $t_4 = f_3$.

Die Faktorisierung ist also $f(x) = f_1 f_2 f_3 \cdot f_2 \cdot \left(f_1 f_2 \cdot (f_3)^7\right)^7 = t_1 \cdot t_2 \cdot (t_3 \cdot t_4^7)^7$, *dabei sind Produkte ohne Multiplikationszeichen nur ausmultipliziert, nicht aber in ihrer Zerlegung bekannt. Die einzelnen Teiler können noch paarweise teilerfremd zueinander gemacht werden. Bilde dafür*

- $\mathrm{ggT}(t_4, t_3) = 1$, $\mathrm{ggT}(t_4, t_2) = 1$ *und* $\mathrm{ggT}(t_4, t_1) = t_4$ *mit* $t_1/t_4 = t_3$, *schreibe darum*
$$f(x) = t_3 \cdot t_4 \cdot t_2 \cdot \left(t_3 \cdot (t_4)^7\right)^7.$$
- $\mathrm{ggT}(t_3, t_2) = t_2$ *mit* $t_3/t_2 = f_1$, *schreibe darum*
$$f(x) = f_1 \cdot t_2 \cdot t_4 \cdot t_2 \cdot \left(f_1 \cdot t_2 \cdot t_4^7\right)^7.$$

Man liest ab, daß $f(x) = f_1^8 \cdot f_2^9 \cdot f_3^{50}$ *ist.*

Komplexität

Ich gebe die Komplexität der Laufzeit aller Algorithmen bezüglich der elementaren Operationen in einem endlichen Körper (Addition, Subtraktion, Multiplikation, Division in $\mathbb{F}_q$) an und gehe dabei davon aus, daß jedes Objekt sich Computer–intern in einem Wort darstellen läßt.

Um die Komplexität des Verfahrens zu berechnen, müssen wir zuerst einmal die Anzahl der elementaren Operationen für die Addition und Multiplikation zweier Polynome vom Grad n modulo f mit $\mathrm{Grad}(f) = n$ bestimmen, außerdem die Anzahl der Operationen für die Berechnung des größten gemeinsamen Teilers zweier Polynome vom Grad n. Dann kann man die Anzahl der Operationen in Abhängigkeit von der maximalen Vielfachheit l berechnen.

Starten wir mit der Addition, die dieselbe Anzahl von Operationen benötigt wie die Subtraktion. Für alle $i \in \{0, \dots, n\}$ werden die Koeffizienten von x^i addiert, es werden genau $n + 1$ elementare Additionen durchgeführt.

Die Ableitung eines Polynoms vom Grad n kostet n elementare Multiplikationen.

Bei der Multiplikation der Polynome $a(x) = \sum_{i=0}^n a_i x^i$ und $b(x) = \sum_{i=0}^n b_i x^i$ wird jedes $b_i x^i$ von b mit dem Polynom a, also mit jedem seiner Koeffizienten, multipliziert, wobei das x^i keine Auswirkungen auf die Multiplikation hat, man startet mit $i = n$ und vermindert i in jedem weiteren Schritt um 1. In jedem Schritt wird modulo $f \cdot x^i$ gerechnet, das heißt es wird ein Polynom $\alpha \cdot f \cdot x^i$ mit $\alpha \in \mathbb{F}_q$ von dem Ergebnis der Multiplikation $a(x) \cdot b_i x^i$ abgezogen, α zu bestimmen kostet eine elementare Operation. Damit ergibt sich für jedes $b_i x^i$ die Anzahl $(n + 1) + 1 + 2(n + 1) = 3n + 4$. Eine Multiplikation zweier Polynome modulo f kostet also $(3n + 4)(n + 1) = 3n^2 + 7n + 4$ elementare Operationen. Eine Division benötigt maximal n Operationen mehr. Multiplikation und Division haben also die Komplexität $\mathcal{O}(n^2)$.

Man möchte nun den größten gemeinsamen Teiler zweier Polynome a_0 und a_1 vom Grad n bestimmen, in jedem Schritt ist $a_i = r_{i+1} a_{i+1} + a_{i+2}$, es gilt $\mathrm{Grad}(a_i) \geq \mathrm{Grad}(a_{i+1})$ für alle $i \geq 1$. Ist $\mathrm{ggT}(a_0, a_1) = 1$, so braucht der Euklidische Algorithmus also im schlimmsten Fall n Schritte. Wenn sich der Grad immer nur um 1 reduziert, dann ist aber $\mathrm{Grad}(r_i) = 0$ für alle $i \geq 1$. Ist $\mathrm{Grad}(r_i) = m > 0$, so werden sozusagen m Schritte des Euklidischen Algorithmus in einem ausgeführt. Wieviel kostet nun ein Schritt des Euklidischen Algorithmus? In einer Operation wird r_{i+1} festgestellt, die $n+1$ Koeffizienten von a_{i+1} werden mit $r_{i+1} \in \mathbb{F}_q$ multipliziert, das Ergebnis wird von a_i abgezogen und wir erhalten a_{i+2}. Ein Schritt dauert also $1+(n+1)+(n+1) = 2n+3$ Operationen, der Euklidische Algorithmus kostet darum $2n^2+3n$ Operationen und hat die Komplexität $\mathcal{O}(n^2)$.

Das von mir beschriebene Verfahren, das ein Polynom f quadratfrei macht, hängt von allen Vielfachheiten l_i ab. Ist eine von ihnen derart, daß l_i+1 von p geteilt wird, so braucht der Algorithmus $p-1$ Schritte bis er das erste Mal versucht, ein Polynom durch seine p–te Wurzel zu ersetzen. Gilt $p^k \mid l_i$ und $p^{k+1} \nmid l_i$, so muß das linear von p abhängige Reduzieren des Polynoms k mal durchgeführt werden. Sei nun l die maximale Vielfachheit, dann müssen im schlimmsten Fall $(\log_p(l)+1)\cdot(p-1) = (p-1)\log_p(l)+p-1$ Schritte des Algorithmus ausgeführt werden. Jeder Schritt besteht aus einer Ableitung und einer Bestimmung des größten gemeinsamen Teilers zweier Polynome vom Grad n (oder kleiner), manchmal wird auch die p–te Wurzel gebildet. Ein Schritt besteht also maximal aus $n+(2n^2+3n)+n/p = 2n^2+(4+\frac{1}{p})n$ Operationen. Bevor die Teiler paarweise teilerfremd gemacht werden, werden also höchstens $((p-1)\log_p(l)+p-1)(p-1)(2n^2+(4+\frac{1}{p})n)$ elementare Operationen ausgeführt. Um die gefundenen Faktoren teilerfremd zu machen, müssen weniger als $(p\log_p(l))!$ größte gemeinsame Teiler berechnet werden. Der Aufwand ist also kleiner als $(p\log_p(l))!\cdot(2n^2+3n)$ Operationen. Insgesamt hat dieser Algorithmus also die Komplexität $\mathcal{O}((p\log(l))!\cdot n^2)$.

Da l zu Beginn des Algorithmus nicht bekannt ist, kann man l viel zu grob als $l \leq n$ abschätzen, dann ergibt sich eine Komplexität von $\mathcal{O}(n^2(p\log(n))!))$. Diese Abschätzung ist viel zu grob, da l und n nicht korreliert sind, das heißt, daß zum Beispiel n beliebig groß werden und l dennoch 1 sein kann.

Da die maximale Vielfachheit l meist nicht sehr groß ist, reduziert man allerdings Polynome besser mit dem im folgenden Abschnitt beschriebenen Verfahren für viele Faktoren mit kleinen Vielfachheiten.

Viele Faktoren mit kleinen Vielfachheiten

Dieses Verfahren wird zum Beispiel von K. O. Geddes, S. R. Czapor und G. Labahn in [14] beschrieben und empfiehlt sich, wenn für alle $1 \leq i \leq t$ gilt $l_i \leq p$. Es benutzt $g_1 = f/g$ mit $g = \mathrm{ggT}(f, f')$ als Faktor von f, in dem jeder irreduzible Anteil höchstens einmal auftaucht. Man initialisiert $f_1 = f$. Nun wird sukzessive der größte gemeinsame Teiler h von f_1 und g_1 gebildet, danach wird f_1 auf f_1/h und g_1 auf h gesetzt. Es wird so lange iteriert, bis $g_1 = 1$ ist. Die Abbruchbedingung tritt ein, da in g_1 nur Faktoren mit der Vielfachheit 1 vorkommen, die auch in g und also auch in f enthalten sind und da die Vielfachheit der Faktoren in f_1 sich in jedem Iterationsschritt um 1 reduziert. Ist in f eine Vielfachheit durch p teilbar, so taucht dieser Faktor in g, nicht aber in g_1 auf, somit ist dann nach Abbruch der Iteration $f_1 \neq 1$. In diesem Fall werden durch einen rekursiven Aufruf des Verfahrens die Vielfachheiten der Faktoren von $f_1^{1/p}$ festgestellt und mit p multipliziert.

Algorithmus 5 QUADRATFREI $(f, \mathbb{F}_q)$
Eingabe: Ein normiertes Polynom $f \in \mathbb{F}_q[x]$ und der Körper $\mathbb{F}_q$ mit $q = p^s$.
Ausgabe: Die Zerlegung in paarweise teilerfremde quadratfreie Teilpolynome g_i mit $f = g_1^1 \cdots g_k^k$.
Programm: Basis.mws: sqf

1. Setze $i \leftarrow 1$ und $out \leftarrow 1$.
2. Ist $f'(x) = 0$, so setze $out \leftarrow (\text{QUADRATFREI}\ (f^{1/p}, \mathbb{F}_q))^p$ und gehe zu Schritt 6. Ist $f'(x) \neq 0$, so setze $g(x) \leftarrow \mathrm{ggT}(f(x), f'(x))$, $out \leftarrow 1$ und $g1(x) \leftarrow f(x)/g(x)$.
3. Ist $g1(x) = 1$, so gehe zu Schritt 5, andernfalls setze $h(x) \leftarrow \mathrm{ggT}(g1(x), g(x))$, $z(x) \leftarrow g1(x)/h(x)$, $out \leftarrow out \cdot z(x)^i$, $i \leftarrow i+1$, $g1(x) \leftarrow h(x)$ und $g(x) \leftarrow g(x)/h(x)$.
4. Gehe zu Schritt 3.
5. Ist $g(x) \neq 1$, so setze $out \leftarrow out \cdot (\text{QUADRATFREI}\ (g^{1/p}, \mathbb{F}_q))^p$.
6. Liefere out zurück.

Beispiel 2.2
Sei wiederum $q = p = 7$, wir möchten noch einmal das Polynom $f = f_1^8 f_2^9 f_3^{50}$, das wir nur in der Form $f(x) = \sum_{i=0}^n a_i x^i$ vorliegen haben, in quadratfreie Faktoren zerlegen. Dabei sind die Polynome f_i für $i \in \{1, 2, 3\}$ paarweise teilerfremd.

Bilde zum Start $g_1 = \frac{f}{\mathrm{ggT}(f,f')} = f_1 f_2 f_3$.

In jedem weiteren Schritt wird f einmal durch g_1 geteilt. Nach acht Schritten ist dieses Teilen nicht mehr möglich, man erhält den Faktor f_1 mit der Vielfachheit acht und rechnet weiter mit $g_2 = g_1/f_1 = f_2 f_3$. Im nächsten Schritt erhält man f_2 mit der Vielfachheit neun und rechnet weiter mit $g_3 = g_2/f_2 = f_3$. Nach 41 weiteren Schritten erhält man f_3 mit der Vielfachheit 50. Wie gewünscht ist $f = f_1^8 \cdot f_2^9 \cdot f_3^{50}$.

Komplexität

Dieses Verfahren hängt linear von der maximalen Vielfachheit l ab, da eine Ersetzung durch die p–te Wurzel nur dann durchgeführt wird, wenn eine der Vielfachheiten l_i genau durch p teilbar ist. Ist diese Vielfachheit nicht l, so verlängert sich das Verfahren: Zu den l Schritten kommen noch $\log_p(l_i)$ hinzu.

Zu Beginn des Algorithmus wird f abgeleitet, das kostet n elementare Operationen, das Bilden des größten gemeinsamen Teilers kostet $2n^2 + 3n$ Operationen. Ein Schritt des Algorithmus besteht aus dem Bilden eines größten gemeinsamen Teilers, was $2n^2 + 3n$ Operationen dauert. Der gesamte Algorithmus läuft also in $n + 2n^2 + 3n + (l + \log_p(l) + \dots) \cdot (2n^2 + 3n)$ Operationen mit der Komplexität $\mathcal{O}(ln^2)$ ab.

Zur Not schätzt man l wieder durch n ab und erhält die Komplexität $\mathcal{O}(n^3)$.

2.2 Teilpolynome mit gleichgradigen irreduziblen Faktoren

Dieses Verfahren wird dazu benutzt, ein quadratfreies Polynom f in Faktoren der folgenden Art zu zerlegen: Ich bestimme zu jedem $d \in \mathbb{N}$ das Polynom h_d, in dem genau alle irreduziblen Faktoren vom Grad d enthalten sind, die f teilen. Ich benutze dabei den Satz 1.6 auf Seite 6.

Ich starte mit $g_1(x) = f(x)$. Indem ich nacheinander für $d = 1, 2, \dots, n/2$ das Polynom $x^{q^d} - x \mod f(x)$ bilde, den Faktor $h_d(x) = \mathrm{ggT}(g_d(x), x^{q^d} - x)$ bestimme und mit $g_{d+1} = g_d/h_d$ weiterrechne, erhalte ich sukzessive die gesuchten $h_d(x)$. Ist $g_{n/2+1}(x) \neq 1$, so ist $g_{n/2+1}$ irreduzibel, denn wäre es reduzibel, so würde es in zwei Faktoren zerfallen, deren Grad jeweils kleiner oder gleich $n/2$ wäre. Darüm wären diese beiden Faktoren bereits in einem der h_d mit $d \leq n/2$ enthalten gewesen und dürften somit nicht mehr in $g_{n/2+1}$ auftauchen.

Um die Faktorisierung des ursprünglichen f zu erhalten, brauche ich nur noch die h_d zu faktorisieren. Ein Vorteil dabei ist, daß ich von jedem h_d weiß, daß es in Faktoren vom Grad d zerfällt, so daß ich schon die Anzahl $t = \sum_{d=1}^n \mathrm{Grad}(h_d)/d$ der Faktoren weiß, die zu bestimmen sind. An dieser Stelle ist also schon klar, ob es sich bei f um ein irreduzibles Polynom handelt oder nicht.

Algorithmus 6 VERSCHIEDENE GRADE $(f, \mathbb{F}_q)$
Eingabe: Ein quadratfreies normiertes Polynom $f \in \mathbb{F}_q[x]$ und der Körper $\mathbb{F}_q$.
Ausgabe: Die Zerlegung von f in Polynome h_d mit $f = h_1 \cdots h_k$, wobei jedes h_d das Produkt aller irreduziblen Faktoren von f ist, welche den Grad d haben.
Programm: Basis.mws: dd

1. Setze $d \leftarrow 1$ und $k \leftarrow \mathrm{Grad}(f)/2$.
2. Ist $d > k$, so gehe zu Schritt 5. Sonst setze $h_d(x) \leftarrow \mathrm{ggT}(x^{q^d} - x, f(x))$.
3. Ist $h_d(x) \neq 1$, so setze $f(x) \leftarrow f(x)/h_d(x)$ und $k \leftarrow \mathrm{Grad}(f)/2$.
4. Setze $d \leftarrow d + 1$ und gehe zu Schritt 2.
5. Gib $f \cdot h_1 \cdots h_k$ zurück.

Beispiel 2.3
Es sei $q = p = 7$, wir wollen das quadratfreie Polynom

$$f(x) = x^{12} + 6x^8 + 6x^7 + 2x^6 + 5x^5 + 2x^4 + 6x^3 + 2x^2 + 5x + 4$$

in Teilpolynome mit gleichgradigen Faktoren zerlegen.

Es ist $\mathrm{Grad}(f) = 12$, *also werden wir mit den Polynomen* $x^{7^d} - x$ *mit* $d \in \{1, \dots, 6\}$ *arbeiten. Dadurch ergibt sich:*

- $d = 1$, $h_1(x) = \mathrm{ggT}(f(x), x^7 - x) = x^2 + x + 1$ *ist der Teil von* f, *der in Faktoren vom Grad 1 zerfällt, rechne weiter mit* $f_1(x) = f(x)/h_1(x) = x^{10} + 6x^9 + x^7 + 5x^6 + 4x^4 + x^3 + 4x^2 + x + 4$.
- $d = 2$, $x^{7^2} - x \equiv 6x^9 + 5x^8 + 3x^7 + 3x^6 + 4x^5 + x^4 + 3x^3 + 3x^2 + 5x + 3 \mod f_1(x)$. *Dann ist* $h_2(x) = \mathrm{ggT}(f_1(x), 6x^9 + 5x^8 + 3x^7 + 3x^6 + 4x^5 + x^4 + 3x^3 + 3x^2 + 5x + 3) = x^6 + 5x^5 + 2x^4 + x^2 + 5x + 2$ *der Teil von* f, *der in Faktoren vom Grad 2 zerfällt, rechne weiter mit* $f_2(x) = f_1(x)/h_2(x) = x^4 + x^3 + 6x + 2$.
- $d = 3$, $x^{7^3} - x \equiv 3x^3 + 4x^2 + x + 3 \mod f_2(x)$. *Dann ist* $h_3(x) = \mathrm{ggT}(f_2(x), 3x^3 + 4x^2 + x + 3) = 1$, *darum enthält* f *keine Faktoren vom Grad 3.*
- $d = 4$, $x^{7^4} - x \equiv 0 \mod f_2(x)$. *Dann ist* $h_4(x) = \mathrm{ggT}(f_2(x), 0) = f_2(x)$, *darum ist* f_2 *der Faktor von* f *vom Grad 4, das Weiterrechnen mit* $f_3(x) = f_2(x)/h_4(x) = 1$ *erübrigt sich.*

Das Ergebnis ist somit, daß $f(x) = (x^2 + x + 1)(x^6 + 5x^5 + 2x^4 + x^2 + 5x + 2)(x^4 + x^3 + 6x + 2)$ *gilt. Dabei zerfällt* $x^2 + x + 1$ *in zwei Faktoren vom Grad 1 und* $x^6 + 5x^5 + 2x^4 + x^2 + 5x + 2$ *in drei Faktoren vom Grad zwei, während* $x^4 + x^3 + 6x + 2$ *ein irreduzibles Polynom vom Grad vier ist.*

Komplexität

Beim Potenzieren eines x mit $k \in \mathbb{N}$ modulo einem Polynom f mit $\mathrm{Grad}(f) = n$ müssen $\log_2(k)$ Polynome multipliziert werden, um die Polynome x^{2^i} für $i \in \{1, \dots, \lfloor \log_2(k) \rfloor\}$ zu erhalten, dann wird mit der Binärdarstellung von k und maximal $\lfloor \log_2(k) \rfloor$ Multiplikationen der soeben bestimmten Potenzen von x das Polynom x^k berechnet. Der Weg zum x^k kostet also $2 \cdot \lfloor \log_2(k) \rfloor \cdot (3n^2 + 7n + 4)$ Operationen.

Zu Beginn des Algorithmus wird somit in $2 \log_2(q)(3n^2 + 7n + 4)$ elementaren Operationen x^q berechnet.

Weiterhin durchläuft der Algorithmus im schlimmsten Fall $\frac{n}{2}$ Schritte. In jedem dieser Schritte wird das Polynom x^{q^i} modulo f mit x^q multipliziert, das kostet $3n^2 + 7n + 4$ elementare Operationen. Danach wird der größte gemeinsame Teiler zweier Polynome vom Grad $n - 1$ gebildet, das kostet $2n^2 + 3n$ Operationen. Insgesamt werden also $2 \log_2(q)(3n^2 + 7n + 4) + \frac{n}{2}((3n^2 + 7n + 4) + (2n^2 + 3n))$ elementare Operationen durchgeführt, der Algorithmus hat die Komplexität $\mathcal{O}(n^3 + n^2 \log(q))$.

3 Nullstellen

Einige Faktorisierungsmethoden aus dem Abschnitt 5 auf Seite 45 benötigen Instrumente, die die Nullstellen eines Polynoms finden, das ausschließlich aus linearen Faktoren zusammengesetzt ist. Wenn man sich nur für die Nullstellen eines Polynoms interessiert, so kann man konzeptuell einfachere Algorithmen benutzen als zur totalen Faktorisierung.

Ist die Ordnung q des endlichen Körpers, über dem das zu faktorisierende f sich befindet, klein, so findet man die Nullstellen am besten durch Ausprobieren. Ist q hingegen nicht klein, so ist es vorzuziehen, einen der folgenden Algorithmen zu benutzen.

3.1 Der Algorithmus von Berlekamp und Rabin

Ich betrachte ein quadratfreies Polynom f vom Grad $n \geq 1$ über $\mathbb{F}_q$, das ich faktorisieren möchte. Die Charakteristik von $\mathbb{F}_q$ ist p und q ist gegeben als $q = p^s$.

In diesem Abschnitt begegnen uns zum ersten Mal die drei typischen Methoden, die bei der Faktorisierung von Polynomen benutzt werden. Sie unterscheiden den Fall, daß die Charakteristik des endlichen Körpers ungerade ist, von dem der geraden Charakteristik. Ist die Charakteristik p ungerade, so benutzt man die Abbildung in die multiplikative Untergruppe der Ordnung zwei; ist die Charakteristik $p = 2$, so gibt es zwei verschiedene Herangehensweisen, deren eine sich der Abbildung in die multiplikative Untergruppe der Ordnung drei bedient, die andere arbeitet mit der Spur.

Wie bereits im Satz 1.6 erwähnt, ist $x^q - x$ das Produkt aller linearen Faktoren $(x - \alpha)$ mit $\alpha \in \mathbb{F}_q$. Darum bilde ich zuerst den größten gemeinsamen Teiler von f und $x^q - x$, um nur noch mit dem Teil zu arbeiten, in dem genau alle linearen Faktoren von f enthalten sind. Sei also in diesem Abschnitt ohne Einschränkung f ein normiertes quadratfreies Polynom, das in lineare Faktoren zerfällt.

Theorie für ungerades q

Ist die Charakteristik p ungerade, so wenden wir die typische Methode an, die die Abbildung in die Untergruppe der Ordnung zwei benutzt:

Für jedes $\delta \in \mathbb{F}_q$ erhalte ich die Umformung

$$\begin{aligned} x^q - x &= x^q + \delta^q - x - \delta \\ &= (x+\delta)^q - (x+\delta) \\ &= (x+\delta) \cdot \left((x+\delta)^{q-1} - 1\right) \\ &= (x+\delta)((x+\delta)^{(q-1)/2} - 1)((x+\delta)^{(q-1)/2} + 1) \end{aligned}$$

mit der dritten binomischen Formel. Speziell im Fall $\delta = 0$ ergibt sich

$$x^q - x = x\left(x^{(q-1)/2} - 1\right)\left(x^{(q-1)/2} + 1\right).$$

Ich habe also $x^q - x$ aufgespalten in ein Produkt von Polynomen, und da f Teiler von $x^q - x$ ist, hoffe ich, daß das Bilden des größten gemeinsamen Teilers von f mit einem der Faktoren mir einen nicht trivialen Faktor von f liefert.

Ist $f(\delta) = 0$, so habe ich eine Nullstelle von f gefunden und teile darum f durch $x - \delta$. Danach rechne ich mit dem Ergebnis dieser Division weiter. Ist $f(\delta) \neq 0$, so ist

$$f(x) = \mathrm{ggT}\left(f(x), (x+\delta)^{(q-1)/2} - 1\right) \cdot \mathrm{ggT}\left(f(x), (x+\delta)^{(q-1)/2} + 1\right).$$

Die Nullstellen von $(x+\delta)^{(q-1)/2} - 1$ sind genau die Elemente der Menge $\beta_\delta = \{\beta \mid \beta + \delta \text{ Quadrat in } \mathbb{F}_q\}$. Somit ist $g(x) = \mathrm{ggT}\left(f(x), (x+\delta)^{(q-1)/2} - 1\right)$ genau dann ein nichttrivialer Faktor von f, wenn f mindestens eine Nullstelle hat, die in β_δ ist, und eine, die nicht in β_δ ist.

Lemma 3.1
Für alle $\alpha, \beta \in \mathbb{F}_q$ mit $\alpha \neq \beta$ hat die Menge der $\delta \in \mathbb{F}_q$, für die $(\alpha+\delta)^{(q-1)/2} = (\beta+\delta)^{(q-1)/2}$ ist, genau $\frac{q-1}{2}$ Elemente.

Beweis: Für ein konkretes $\delta \in \mathbb{F}_q$ sei ohne Einschränkung $\beta \neq -\delta$, sonst vertausche man α und β. Dann sind $(\alpha+\delta)^{(q-1)/2}, (\beta+\delta)^{(q-1)/2} \in \{\pm 1, 0\}$. Es ist $(\alpha+\delta)^{(q-1)/2} = (\beta+\delta)^{(q-1)/2}$ genau dann, wenn

$$\left(\frac{\alpha+\delta}{\beta+\delta}\right)^{(q-1)/2} = 1.$$

Die Gleichung $x^{(q-1)/2} = 1$ hat genau $\frac{q-1}{2}$ Lösungen in $\mathbb{F}_q$. Ich betrachte die Abbildung

$$\Phi(\delta) = (\alpha+\delta)/(\beta+\delta).$$

Der Definitionsbereich ist gegeben als die Menge $\{\delta \mid \delta \in \mathbb{F}_q \backslash \{-\beta\}\}$ und dort gilt

$$\frac{\alpha+\delta}{\beta+\delta} = \gamma \qquad \Leftrightarrow \qquad \delta = \frac{\gamma\beta - \alpha}{1-\gamma}$$

für $\gamma \neq 1$. Φ nimmt somit alle Werte aus $\mathbb{F}_q \backslash \{1\}$ an. Daraus ergibt sich wie gewünscht, daß $\Phi(\delta)^{(q-1)/2} \neq 1$ für genau $\frac{q+1}{2}$ Werte $\delta \in \mathbb{F}_q$. □

Satz 3.2
Für alle Paare $\alpha, \beta \in \mathbb{F}_q$ mit $\alpha \neq \beta$ gibt es mindestens ein $\delta \in \mathbb{F}_q$, so daß $\alpha + \delta$ Quadrat in $\mathbb{F}_q$ ist, $\beta + \delta$ hingegen nicht.

Es gibt also für alle $\alpha \neq \beta$ ein δ, das mit der Anwendung des Verfahrens die beiden Linearfaktoren $x - \alpha$ und $x - \beta$ voneinander trennt.

Um die Nullstellen eines quadratfreien Polynoms f, welches ausschließlich aus linearen Faktoren zusammengesetzt ist, zu bestimmen, wählt man ein zufälliges $\delta \in \mathbb{F}_q$. Für dieses δ bestimmt man $f(\delta)$ und

$$g(x) = \mathrm{ggT}\left(f(x), (x+\delta)^{(q-1)/2} - 1\right).$$

Ist $f(\delta) = 0$, so ist eine partielle Zerlegung von f erreicht, die Wahrscheinlichkeit dafür ist n/q. Die Zerlegung ist trivial, wenn $x - \delta$ kein Linearfaktor von f ist (Wahrscheinlichkeit $(q-n)/q$) und entweder alle Nullstellen von f in β_δ oder aber alle in $\mathbb{F}_q \backslash \beta_\delta$ sind.

Satz 3.3
Sei P die Wahrscheinlichkeit für eine nicht triviale Faktorisierung von f. Dann gilt $P \geq 1/2$.

Beweis: Ist $n \geq \frac{q+1}{2}$, so ist $P = 1$. Sonst gilt

$$P = 1 - \frac{q-n}{q} \cdot 2 \cdot \left(\frac{q-1}{2q}\right)^n.$$

Ich beweise $\frac{q-n}{q} \cdot 2 \cdot \left(\frac{q-1}{2q}\right)^n \leq \frac{1}{2}$ mit einer vollständigen Induktion nach n.

Induktionsanfang: Ist $n = 2$, so ist

$$\frac{q-n}{q} \cdot 2 \cdot \left(\frac{q-1}{2q}\right)^n = \frac{(q-2)\cdot(q-1)^2}{2q^3}.$$

Dann gilt

$$\begin{aligned} P \geq \frac{1}{2} \quad &\Leftrightarrow \quad (q-2)\cdot(q-1)^2 \leq q^3 \\ \Leftrightarrow \quad q^3 - 4q^2 + 5q - 2 \leq q^3 \quad &\Leftrightarrow \quad 4q^2 - 5q + 2 \geq 0 \\ \Leftrightarrow \quad \left(q - \frac{5}{8}\right)^2 &\geq \frac{-7}{64}. \end{aligned}$$

Die letzte Ungleichung ist für alle q erfüllt.

Induktionsvoraussetzung: Es gilt $\frac{q-n}{q} \cdot 2 \cdot \left(\frac{q-1}{2q}\right)^n \leq \frac{1}{2}$ für ein $n \in \{1, \ldots, \frac{q-1}{2}\}$.

Induktionsschluß: Es ist

$$\frac{q-(n+1)}{q} \cdot 2 \cdot \left(\frac{q-1}{2q}\right)^{n+1} = \left(\frac{q-n}{q} \cdot 2 \cdot \left(\frac{q-1}{2q}\right)^n\right) \cdot \frac{q}{q-n} \cdot \frac{q-n-1}{q} \cdot \frac{q-1}{2q},$$

um zu zeigen, daß $\frac{q-(n+1)}{q} \cdot 2 \cdot \left(\frac{q-1}{2q}\right)^{n+1} \leq 1/2$ ist, zeige ich, daß

$$\frac{q}{q-n} \cdot \frac{q-n-1}{q} \cdot \frac{q-1}{2q} \leq 1.$$

Es ist

$$\frac{q}{q-n} \cdot \frac{q-n-1}{q} \cdot \frac{q-1}{2q} \leq 1 \quad \Leftrightarrow \quad (q-n-1)(q-1) \leq 2q(q-n)$$

$$\Leftrightarrow \quad q^2 + q(2-n) - (n+1) \geq 0 \quad \Leftrightarrow \quad \left(q + \frac{2-n}{2}\right)^2 \geq \frac{n^2+8}{4}.$$

Ich schätze mit $0 \leq n$ den Term

$$\frac{n^2+8}{4} \leq \frac{n^2+8n+16}{4}$$

ab und erhalte

$$\left(q + \frac{2-n}{2}\right)^2 \geq \frac{n^2+8n+16}{4} \quad \Leftrightarrow \quad q + \frac{2-n}{2} \geq \frac{n+4}{2}$$

$$\Leftrightarrow \quad q \geq n+1.$$

Die letzte Ungleichung ist wegen $n \leq \frac{q-1}{2}$ und $q \geq 3$ erfüllt. Insgesamt ist also

$$\frac{q}{q-n} \cdot \frac{q-n-1}{q} \cdot \frac{q-1}{2q} \leq 1$$

und somit $P \geq 1/2$.

□

Nach diesem Satz 3.3 ist das Verfahren in der Komplexitätsklasse RP der Probleme, die in randomisierter polynomialer Zeit lösbar sind. Diese Sorte von Problemen wird von N. Koblitz in [22] erklärt.

Nach der partiellen Zerlegung wird das Verfahren mit einem anderen $\delta \in \mathbb{F}_q$ neu auf g und f/g beziehungsweise eventuell auf $f/(g \cdot (x-\delta))$ angewendet, so lange, bis alle gefundenen Faktoren linear sind — das ist der Fall, wenn $\mathrm{Grad}(f) = n$ Faktoren von f gefunden wurden. Wird durch g keine echte Zerlegung von f erreicht, so wählt man dennoch ein weiteres δ und wendet das Verfahren erneut auf f an.

Algorithmus 7 BERLEKAMP RABIN $(f, \mathbb{F}_q)$
Eingabe: Ein normiertes quadratfreies Polynom $f \in \mathbb{F}_q[x]$, welches in lineare Faktoren zerfällt, und der Körper $\mathbb{F}_q$ mit $q = p^s$ und ungeradem p.
Ausgabe: Die Zerlegung von f in lineare Faktoren.
Programm: BerRab.wms: berrab

1. Setze $n \leftarrow \mathrm{Grad}(f)$ und $h(x) \leftarrow 1$. Ist $n \leq 1$, so liefere $f(x)$ zurück. Sonst wähle ein zufälliges $\delta \in \mathbb{F}_q$.
2. Ist $f(\delta) = 0$, so setze $f(x) \leftarrow f(x)/(x-\delta)$ und $h(x) \leftarrow x - \delta$. Ist nun $\mathrm{Grad}(f) = 1$, so liefere $f(x)$ und $h(x)$ zurück, sonst setze $g(x) \leftarrow \mathrm{ggT}(f(x), (x+\delta)^{(q-1)/2} - 1)$.
3. Liefere $h(x) \cdot$ BERLEKAMP RABIN $(g, \mathbb{F}_q) \cdot$ BERLEKAMP RABIN $(f/g, \mathbb{F}_q)$ zurück.

Beispiel 3.1
Sei $q = p = 7$, wir möchten das quadratfreie Polynom

$$f(x) = x^4 + 6x^3 + x + 6,$$

welches ausschließlich in lineare Faktoren zerfällt, in diese zerlegen. Daß es ausschließlich in lineare Faktoren zerfällt, erkennt man an der Tatsache, daß $\mathrm{ggT}(x^7 - x, f(x)) = f(x)$.

- *Sei $\delta = 1$. Es ist $f(1) = 0$, also ist $x - 1$ ein Faktor von f, rechne weiter mit*

  $$f_1(x) = f(x)/(x + 6) = x^3 + 1.$$

  Weiterhin ist

  $$(x - \delta)^3 - 1 \equiv x^3 + 4x^2 + 3x + 5 \mod f(x) \textit{ und}$$
  $$\mathrm{ggT}(f_1(x), x^3 + 4x^2 + 3x + 5) = x^2 + 6x + 1,$$

  also ist $f_1(x)/(x^2 + 6x + 1) = x + 1$ ein linearer Faktor von f_1, finde noch die restlichen Faktoren von $f_2(x) = x^2 + 6x + 1$.

- *Sei $\delta = 2$. Es ist $f_2(2) = 3$, also ist $x - 2$ kein Faktor von f_2. Weiterhin ist*

  $$(x - \delta)^3 - 1 \equiv x^3 + x^2 + 5x + 5 \mod f(x) \textit{ und}$$
  $$\mathrm{ggT}(f(x), x^3 + x^2 + 5x + 5) = x + 4,$$

  somit sind $x + 4$ und $f_2(x)/(x + 4) = x + 2$ die beiden linearen Faktoren von f_2.

Insgesamt ist also $f(x) = (x + 6) \cdot (x + 1) \cdot f_2(x) = (x + 1) \cdot (x + 6) \cdot (x + 2) \cdot (x + 4)$.

Theorie mit der Abbildung in die Gruppe der Ordnung drei für gerade Charakteristik

In Körpern mit Charakteristik zwei sind alle Elemente Quadrate und es ist $\frac{q-1}{2} \notin \mathbb{N}$. Darum ist die soeben entwickelte Theorie für gerades q nicht anwendbar. Ich beschreibe hier das erste Standardverfahren, mit dieser Situation umzugehen. Wenn man nicht in die multiplikative Untergruppe der Ordnung zwei abbilden kann, was liegt da näher, als es mit der Untergruppe der Ordnung drei zu probieren! Dabei ergibt sich zuerst einmal das Problem, daß nicht jeder Körper der Ordnung $q = 2^s$ eine Untergruppe der Ordnung drei hat.

Satz 3.4
Genau alle Körper $\mathbb{F}_{2^s}$ mit geradem $s \in \mathbb{N} \setminus \{0\}$ haben eine multiplikative Untergruppe der Ordnung drei.

Beweis: Die multiplikative Untergruppe des $\mathbb{F}_{2^s}$ hat $2^s - 1$ Elemente. Für $\mathbb{F}_4 \subset \mathbb{F}_{2^s}$ muß gelten $3 \mid 2^s - 1$, das heißt $2^s \equiv 1 \mod 3$. Da $2 \equiv -1 \mod 3$, muß also s gerade sein. □

Ist s ungerade, so müssen wir $\mathbb{F}_q$ erweitern, um an eine Untergruppe der Ordnung drei heranzukommen. Ich setze

- $\tilde{q} = q$ für $q \equiv 1 \mod 3$ und
- $\tilde{q} = q^2$ für $q \equiv 2 \mod 3$

und betrachte f als ein Polynom über $\mathbb{F}_{\tilde{q}}$.

Ist s gerade, so muß man eines der zwei möglichen Elemente $\xi \in \mathbb{F}_{\tilde{q}}$ bestimmen, das in $\mathbb{F}_4 \setminus \{0, 1\}$ liegt. ξ hat die Eigenschaft, daß $\xi^3 = 1$ mit $\xi \neq 1$ ist. Wähle also zufällige Elemente $\alpha \in \mathbb{F}_{\tilde{q}} \setminus \{0\}$ und bilde $\alpha^{(\tilde{q}-1)/3}$. Ist $\alpha^{(\tilde{q}-1)/3} \neq 1$, so setze $\xi = \alpha^{(\tilde{q}-1)/3}$.

In jedem Fall ist nun $\mathbb{F}_q \subset \mathbb{F}_{\tilde{q}}$ und $\mathbb{F}_4 = \{0, 1, \xi, \xi + 1\} \subset \mathbb{F}_{\tilde{q}}$, wobei ξ der Gleichung $\xi^2 = \xi + 1$ genügt.

Im Körper $\mathbb{F}_{\tilde{q}}$ gilt

$$\begin{aligned}\left(x^{\frac{\tilde{q}-1}{3}}+1\right)\left(x^{\frac{\tilde{q}-1}{3}}+\xi\right)\left(x^{\frac{\tilde{q}-1}{3}}+\xi+1\right) &= \left(x^{\frac{\tilde{q}-1}{3}}+1\right)\left(x^{2\cdot\frac{\tilde{q}-1}{3}}+x^{\frac{\tilde{q}-1}{3}}+1\right)\\ &= x^{\tilde{q}-1}+1.\end{aligned}$$

Indem ich x durch $x+\delta$ mit $\delta \in \mathbb{F}_{\tilde{q}}$ substituiere und weiß, daß $x^{\tilde{q}} - x = (x+\delta)^{\tilde{q}} - (x+\delta)$ ist, erhalte ich

$$x^{\tilde{q}} - x = (x+\delta)\cdot\left((x+\delta)^{\frac{\tilde{q}-1}{3}}+1\right)\cdot\left((x+\delta)^{\frac{\tilde{q}-1}{3}}+\xi\right)\cdot\left((x+\delta)^{\frac{\tilde{q}-1}{3}}+\xi+1\right)$$

und habe wie gewünscht eine Zerlegung von $x^{\tilde{q}} - x$ in ein Produkt.

Es bleibt die Frage, ob man durch Probieren genügend vieler $\delta \in \mathbb{F}_{\tilde{q}}$ alle linearen Faktoren von f findet.

Lemma 3.5
Für alle $\alpha, \beta \in \mathbb{F}_q$ mit $\alpha \neq \beta$ hat die Menge der $\delta \in \mathbb{F}_{\tilde{q}}$, für die $(\alpha+\delta)^{(\tilde{q}-1)/3} = (\beta+\delta)^{(\tilde{q}-1)/3}$ ist, genau $\frac{\tilde{q}-1}{3}$ Elemente.

Beweis: Der Beweis verläuft analog zu dem Beweis des Lemmas 3.1: Für ein konkretes $\delta \in \mathbb{F}_{\tilde{q}}$ sei ohne Einschränkung $\beta \neq -\delta$, sonst vertausche man α und β. Dann sind $(\alpha+\delta)^{(\tilde{q}-1)/3}$ und $(\beta+\delta)^{(\tilde{q}-1)/3}$ Elemente des $\mathbb{F}_4$. Ich suche diejenigen $\delta \in \mathbb{F}_{\tilde{q}}$, für die

$$\left(\frac{\alpha+\delta}{\beta+\delta}\right)^{(\tilde{q}-1)/3} = 1$$

ist. Die Gleichung $x^{(\tilde{q}-1)/3} = 1$ hat genau $\frac{\tilde{q}-1}{3}$ Lösungen in $\mathbb{F}_{\tilde{q}}$. Ich betrachte die Abbildung

$$\Phi(\delta) = (\alpha+\delta)/(\beta+\delta).$$

Aus dem Beweis von Lemma 3.1 wissen wir bereits, daß der Definitionsbereich von Φ die Menge $\{\delta \mid \delta \in \mathbb{F}_{\tilde{q}}\backslash\{-\beta\}\}$ ist und daß Φ alle Werte aus $\mathbb{F}_{\tilde{q}}\backslash\{1\}$ annimmt. Daraus ergibt sich wie gewünscht, daß $\Phi(\delta)^{(\tilde{q}-1)/3} \neq 1$ für genau $2\cdot\frac{\tilde{q}-1}{3}+1 = \frac{2\tilde{q}+1}{3}$ Werte $\delta \in \mathbb{F}_{\tilde{q}}$. □

Ganz analog zu Satz 3.3 auf Seite 16 ist die Wahrscheinlichkeit P für eine nicht triviale Zerlegung von f größer oder gleich $1/2$, darum ist das Verfahren auch bei gerader Charakteristik in der Komplexitätsklasse RP.

Das Ärgerliche an dieser Herangehensweise ist, daß man für ungerades s den Körper $\mathbb{F}_{\tilde{q}}$, aus dem die Elemente δ zufällig ausgewählt werden, tatsächlich konstruieren muß. Diese Konstruktion wirkt sich negativ auf die Effizienz aus.

Algorithmus 8 BERLEKAMP RABIN $(f, \mathbb{F}_q)$
Eingabe: Ein normiertes quadratfreies Polynom $f \in \mathbb{F}_q[x]$, welches in lineare Faktoren zerfällt, und der Körper $\mathbb{F}_q$ mit $q = p^s$ und $p = 2$.
Ausgabe: Die Zerlegung von f in lineare Faktoren.
Programm: BerRab.mws: berrab

1. Setze $n \leftarrow \mathrm{Grad}(f)$ und $h(x) \leftarrow 1$. Ist $n \leq 1$, so liefere $f(x)$ zurück.
2. Ist s ungerade, so erweitere $\mathbb{F}_q$ auf $\mathbb{F}_{\tilde{q}}$ mit $\tilde{q} = q^2$. Ist s hingegen gerade, so setze $\tilde{q} \leftarrow q$ und bestimme ξ. Fasse f als Polynom aus $\mathbb{F}_{\tilde{q}}[x]$ auf.
3. Wähle ein zufälliges $\delta \in \mathbb{F}_{\tilde{q}}$ und identifiziere $\xi \in \mathbb{F}_{\tilde{q}}$ mit $\xi^2 = \xi + 1$.
4. Ist $f(\delta) = 0$, so setze $f(x) \leftarrow f(x)/(x-\delta)$ und $h(x) \leftarrow x - \delta$. Ist nun $\mathrm{Grad}(f) = 1$, so liefere $f(x)$ und $h(x)$ zurück, sonst setze $g_1(x) \leftarrow \mathrm{ggT}(f(x), (x+\delta)^{(\tilde{q}-1)/3} - 1)$ und $g_2(x) \leftarrow \mathrm{ggT}(f(x), (x+\delta)^{(\tilde{q}-1)/3} - \xi)$.
5. Liefere $h(x) \cdot$ BERLEKAMP RABIN $(g_1, \mathbb{F}_{\tilde{q}}) \cdot$ BERLEKAMP RABIN $(g_2, \mathbb{F}_{\tilde{q}}) \cdot$ BERLEKAMP RABIN $(f/(g_1 \cdot g_2), \mathbb{F}_{\tilde{q}})$ in der Darstellung in $\mathbb{F}_q$ zurück.

Beispiel 3.2
Sei $p = 2$, $q = 16$ *mit* $\mathbb{F}_{16} = \mathbb{F}_2[\xi]/(\xi^4 + \xi + 1)\mathbb{F}_2[\xi]$, *wir möchten das quadratfreie Polynom*

$$f(x) = x^3 + (\xi^3 + \xi + 1)x^2 + (\xi^3 + 1)x + \xi + 1,$$

welches ausschließlich in lineare Faktoren zerfällt, in diese zerlegen. Daß f *ausschließlich in lineare Faktoren zerfällt, erkennt man an der Tatsache, daß* $\mathrm{ggT}(x^{16} - x, f(x)) = f(x)$.

Die multiplikative Untergruppe des $\mathbb{F}_{16}$ *hat die Ordnung* 15, *um ein Element dieser Untergruppe in die multiplikative Untergruppe des* $\mathbb{F}_4 \subset \mathbb{F}_{16}$ *abzubilden, potenzieren wir das Element mit* $(q-1)/3 = 5$. *Nun können wir* $\mathbb{F}_4$ *mit* $\{0, 1, \xi^2 + \xi, \xi^2 + \xi + 1\}$ *identifizieren.*

- *Sei* $\delta = \xi^3 + 1$. *Es ist* $f(\xi^3 + 1) = \xi^3 + \xi + 1$, *also ist* $x + \xi^3 + 1$ *kein Faktor von* f. *Weiterhin ist*

  $$(x + \xi^3 + 1)^5 \equiv (\xi^3 + \xi^2 + \xi + 1)x^2 + (\xi^3 + \xi^2 + \xi)x \mod f(x),$$

  bilde

  $$\begin{aligned} h_1(x) = &\mathrm{ggT}\left(f(x), (x + \xi^3 + 1)^5 + 1\right) &&= x^2 + (\xi^3 + 1)x + \xi^3, \\ &\mathrm{ggT}\left(f(x), (x + \xi^3 + 1)^5 + \xi^2 + \xi\right) &&= x + \xi, \\ &\mathrm{ggT}\left(f(x), (x + \xi^3 + 1)^5 + \xi^2 + \xi + 1\right) &&= 1, \end{aligned}$$

  rechne weiter mit $f(x) = h_1(x) \cdot (x + \xi)$.

- *Sei* $\delta = \xi^3 + \xi^2 + 1$. *Es ist* $h_1(\xi^3 + \xi^2 + 1) = \xi^3 + 1$, *also ist* $x + \xi^3 + \xi^2 + 1$ *kein Faktor von* h_1. *Weiterhin ist*

  $$(x + \xi^3 + \xi^2 + 1)^5 \equiv (\xi^3 + \xi^2 + \xi + 1)x^2 + \xi x + \xi^3 + \xi^2 \mod f(x),$$

  bilde

  $$\begin{aligned} &\mathrm{ggT}\left(h_1(x), (x + \xi^3 + \xi^2 + 1)^5 + 1\right) &&= x + 1, \\ &\mathrm{ggT}\left(h_1(x), (x + \xi^3 + \xi^2 + 1)^5 + \xi^2 + \xi\right) &&= 1, \\ &\mathrm{ggT}\left(h_1(x), (x + \xi^3 + \xi^2 + 1)^5 + \xi^2 + \xi + 1\right) &&= x + \xi^3. \end{aligned}$$

Die gesuchte Faktorisierung ist $f(x) = (x + 1)(x + \xi^3)(x + \xi)$.

Theorie mit der Spur für gerade Charakteristik

Die zweite typische Herangehensweise für das Nullstellenproblem bei gerader Charakteristik wurde von M. O. Rabin in [45] vorgeschlagen. Statt der Abbildung in die multiplikative Untergruppe der Ordnung drei wird die Spur benutzt.

Definition 3.6
Für ein Element $\alpha \in \mathbb{F}_{q^k}$ *mit* $k \in \mathbb{N}$ *und den Körper* $\mathbb{F}_q$ *mit* $q = p^s$ *Elementen,* p *prim, ist die Spur von* α *über* $\mathbb{F}_q$ *definiert als*

$$\mathrm{Spur}_{\mathbb{F}_{q^k}/\mathbb{F}_q}(\alpha) = \alpha + \alpha^q + \cdots + \alpha^{q^{k-1}}.$$

Die Spur ist eine surjektive lineare Abbildung, da die Abbildung $\alpha \mapsto \alpha^q$ linear und surjektiv ist.

Ich wende mich jetzt wieder der Theorie für den angestrebten Algorithmus zum Finden von Nullstellen bei Charakteristik zwei zu. Der Übersichtlichkeit halber definiere ich für $q = 2^s$ das $\mathrm{Spur}_{\mathbb{F}_q/\mathbb{F}_2}$–Polynom $T(x) = \sum_{i=0}^{s-1} x^{2^i}$. Es gilt

$$\begin{aligned} T(x) \cdot (T(x) + 1) = T(x)^2 + T(x) &= \left(\sum_{i=0}^{s-1} x^{2^i}\right)^2 + \sum_{i=0}^{s-1} x^{2^i} = \sum_{i=1}^{s} x^{2^i} + \sum_{i=0}^{s-1} x^{2^i} \\ = x^{2^s} + x \qquad &= x^q + x, \end{aligned}$$

also wird $T(x) \cdot (T(x) + 1)$ von $f(x)$ geteilt.

Es gilt also für alle $\alpha \in \mathbb{F}_q$, daß $T(\alpha) \in \{0, 1\}$. Die Spur T ist linear. Außerdem gilt das folgende Lemma:

Lemma 3.7
Für alle $\alpha, \beta \in \mathbb{F}_q$ mit $\alpha \neq \beta$ und $q = 2^s$ hat die Menge der $\delta \in \mathbb{F}_q \setminus \{0\}$, für die $T(\delta \cdot \alpha) = T(\delta \cdot \beta)$ ist, genau 2^{s-1} Elemente.

Beweis: Die Spur ist linear, darum ist $T(\delta \cdot \alpha) = T(\delta \cdot \beta)$ genau dann, wenn $T(\delta \cdot (\alpha + \beta)) = 0$ ist. Es ist $\alpha \neq \beta$, also $\alpha + \beta \neq 0$, darum bildet die Abbildung $\Phi(\delta) = \delta(\alpha + \beta)$ mit Definitionsbereich $\delta \in \mathbb{F}_{2^s}$ surjektiv auf $\mathbb{F}_{2^s}$ ab. Da $\mathrm{Grad}(T) = 2^{s-1}$ ist, so werden für genau 2^{s-1} Werte δ die Nullstellen von T angenommen. □

Der Algorithmus, der sich aus Lemma 3.7 ergibt, ist ganz analog zu dem für ungerade q: Ich wähle ein zufälliges Element $\delta \in \mathbb{F}_q$, bilde $\mathrm{ggT}\,(f, T(\delta x))$ und hoffe, so einen nicht trivialen Faktor von f zu bekommen. Ist $n \geq q/2$, so ist die Wahrscheinlichkeit 0, daß durch die Spur alle Nullstellen zu 0 werden. Sonst ist die Wahrscheinlichkeit dafür, daß durch die Spur alle Nullstellen zu 0 werden, $\frac{1}{2^n}$, genau wie die Wahrscheinlichkeit dafür, daß alle Nullstellen zu 1 werden. Darum ist die Wahrscheinlichkeit P, eine nicht triviale Faktorisierung zu erhalten,

$$P = 1 - 2 \cdot \left(\frac{1}{2}\right)^n = 1 - \left(\frac{1}{2}\right)^{n-1} \geq \frac{1}{2},$$

und das Verfahren ist in der Komplexitätsklasse RP.

Algorithmus 9 BERLEKAMP RABIN $(f, \mathbb{F}_q)$
Eingabe: Ein normiertes quadratfreies Polynom $f \in \mathbb{F}_q[x]$, welches in lineare Faktoren zerfällt, und der Körper $\mathbb{F}_q$ mit $q = p^s$ und $p = 2$.
Ausgabe: Die Zerlegung von f in lineare Faktoren.
Programm: BerRab.mws: berrab

1. Setze $n \leftarrow \mathrm{Grad}(f)$ und $h(x) \leftarrow 1$. Ist $n \leq 1$, so liefere $f(x)$ zurück. Sonst wähle ein zufälliges $\delta \in \mathbb{F}_q$ und setze $T(x) \leftarrow \sum_{i=0}^{s-1} x^{2^i}$.
2. Ist $f(\delta) = 0$, so setze $f(x) \leftarrow f(x)/(x + \delta)$ und $h(x) \leftarrow x + \delta$. Ist nun $\mathrm{Grad}(f) = 1$, so liefere $f(x)$ und $h(x)$ zurück, sonst setze $g(x) \leftarrow \mathrm{ggT}(f(x), T(\delta x))$.
3. Liefere $h(x) \cdot$ BERLEKAMP RABIN $(g, \mathbb{F}_q) \cdot$ BERLEKAMP RABIN $(f/g, \mathbb{F}_q)$ zurück.

Beispiel 3.3
Sei $p = 2$, $q = 16$ mit $\mathbb{F}_{16} = \mathbb{F}_2[\xi]/(\xi^4 + \xi + 1)\mathbb{F}_2[\xi]$, wir möchten das quadratfreie Polynom

$$f(x) = x^3 + (\xi^3 + \xi + 1)x^2 + (\xi^3 + 1)x + \xi + 1,$$

welches ausschließlich in lineare Faktoren zerfällt, in diese zerlegen. $T(x) = x^8 + x^4 + x^2 + x$ ist die Spur von $\mathbb{F}_{16}$ nach $\mathbb{F}_2$.

- *Sei $\delta_1 = 1$. Dann ist*

$$T(x) \equiv (\xi^3 + \xi + 1)x^2 + (\xi^3 + \xi^2 + \xi)x + \xi^2 + 1 \mod f(x),$$

bilde

$$h_1(x) = \mathrm{ggT}(f(x), T(x)) = x^2 + (\xi + 1)x + \xi$$

und rechne weiter mit $f(x) = h_1(x) \cdot (x + \xi^3)$.

- *Sei* $\delta_2 = \xi^2$. *Dann ist*

$$T(\xi^2 x) \equiv (\xi^2)x^2 + \xi x + \xi^2 + \xi \mod f(x),$$

bilde

$$\mathrm{ggT}(h_1(x), T(\xi^2 x)) = x + 1.$$

Die Faktorisierung ist $f(x) = (x+1)(x+\xi)(x+\xi^3)$.

Praxis

Die Umsetzung des Algorithmus ist an sich klar. Da ich die Algorithmen jedoch nur programmiere für endliche Körper mit der Primzahlordnung $q = p$, also für $s = 1$, hake ich den Fall $q = 2$ einfacher als vorgeschlagen ab, indem ich den Grad des Polynoms f ansehe. Ist $\mathrm{Grad}(f) = 1$, so ist f linear und somit bereits die Zerlegung in Linearfaktoren. Andernfalls ist der Grad von f genau zwei, also kommen beide möglichen Nullstellen vor.

Komplexität

Ich führe die Komplexitätsberechnungen für die Laufzeit exemplarisch für den Fall q ungerade vor.

Bestimmen wir zuerst die Anzahl der elementaren Operationen, die in jedem Schritt des Algorithmus ausgeführt werden müssen.

Eingabe ist ein quadratfreies Polynom vom Grad $t = n$, das in n lineare Faktoren zerfällt.

Wir führen zwei Additionen und eine Potenzierung mit $(q-1)/2$ durch, dann wird ein größter gemeinsamer Teiler gebildet. Die Additionen kosten $2(n+1)$ Operationen, der größte gemeinsame Teiler $2n^2 + 3n$.

Das Bilden einer $(q-1)/2$–ten Potenz kostet $2 \cdot \log(q-1) - 1) \cdot (3n^2 + 7n + 4)$ elementare Operationen.

Es ist $q \geq 2$, also ist $\log_2(q-1) \geq 0$. Da außerdem $n \geq 1$ ist, kostet ein Schritt des Algorithmus insgesamt

$$\begin{aligned} &(2n+2) + (2n^2+3n) + 2 \cdot (\log_2(q-1) - 1) \cdot (3n^2 + 7n + 4) \\ &\quad = (6\log_2(q-1) - 4)n^2 + (14\log_2(q-1) - 9)n + 6\log_2(q-1) - 6 \\ &\quad < (6\log_2(q-1) - 4)n^2 + (14\log_2(q-1) - 9)n + 6\log_2(q-1) \cdot n \end{aligned}$$

Operationen.

Der Algorithmus ist probabilistisch und wird mit dem stochastischen Modell der Bälle und Tonnen modelliert, siehe dazu auch V. Shoup in [60]. Das Modell ist das folgende:

Wir haben k Tonnen und $t \geq 2$ Bälle. Wird ein Ball geworfen, so landet er mit der Wahrscheinlichkeit p_i in der i–ten Tonne, $1 \leq i \leq k$. Der Ball landet mit Sicherheit in einer Tonne und nie auf dem Boden, das heißt $\sum_{i=1}^{k} p_i = 1$. Sei $p_i \leq w < 1$ für alle $1 \leq i \leq k$.

Mit dem Spiel wird eine Kostenfunktion $\mathrm{kost} : \mathbb{R}_+ \to \mathbb{R}_+$ assoziiert. Das Spiel funktioniert so: Die Spielerin spielt höchstens $\mathrm{kost}(t)$ Einheiten, dann wirft sie die Bälle in die Tonnen. Für jede Tonne, die mehr als einen Ball enthält, spielt sie das Spiel rekursiv mit den enthaltenen Bällen noch einmal.

Gesucht wird $\mathrm{Kost}(t)$, die Anzahl aller gespielten Einheiten während des gesamten Spiels, inklusive der rekursiven Aufrufe. Diese Anzahl wird mit dem Erwartungswert $E(\mathrm{Kost}(t))$ geschätzt.

Für jedes Paar von Bällen ist die Wahrscheinlichkeit, daß sie in derselben Tonne landen, höchstens w. Darum ist die Wahrscheinlichkeit, daß dieses Paar nach j rekursiven Aufrufen noch nicht getrennt ist, höchstens w^j. Die Anzahl der Ballpaare ist $(t-1) + (t-2) + \cdots + 2 + 1 = \frac{1}{2}t^2 + \frac{t}{2} < t^2$, darum ist die

Wahrscheinlichkeit $P(D \geq j) < t^2 w^{j-1}$, wobei D die Rekursionstiefe ist. Der Erwartungswert $E(D)$ der Rekursionstiefe berechnet sich als

$$\begin{aligned} E(D) &= \sum_{j=1}^{\infty} P(D \geq j) \\ &= \sum_{1 \leq j \leq \log_{1/w}(t^2)+1} P(D \geq j) + \sum_{j > \log_{1/w}(t^2)+1} P(D \geq j) \\ &\leq (\log_{1/w}(t^2) + 1) + \frac{1}{1-w} \\ &= \frac{2\log(t)}{\log(1/w)} + 1 + \frac{1}{1-w}, \end{aligned}$$

wobei in der linken Summe der zweiten Zeile jeder Summand als kleiner oder gleich 1 abgeschätzt wird, da es sich um Wahrscheinlichkeiten handelt.

In der j–ten Schicht des Rekursionsbaums wird höchstens l^j mal die Kostenfunktion kost ausgeführt. Gilt aber $\mathrm{kost}(at) \geq a\,\mathrm{kost}(t)$ für jedes $a \in \mathbb{R}$ mit $a \geq 1$ und jedes $t \in \mathbb{N} \setminus \{0\}$, so summieren sich die Einheiten in jeder Schicht höchstens zu $\mathrm{kost}(t)$ auf.

In unserem probabilistischen Algorithmus entsprechen die $l = 3$ Tonnen den Resten $0, 1$ und -1 des zufälligen Polynoms $(x+\delta)^{(q-1)/2}$ modulo f_i mit $1 \leq i \leq t$, die Bälle sind die t linearen Faktoren f_i von f, w ist hier $w = \frac{q-1}{2q}$. Die Kostenfunktion ist $\mathrm{kost}(t) = 4t^2\log_2(q-1) - 4t^2 + 16t\log_2(q-1) - 5t$ ist die obere Schranke für die Anzahl der elementaren Operationen, die in jedem Schritt des Algorithmus durchgeführt werden müssen. Für die Kostenfunktion gilt $\mathrm{kost}(at) \geq a\,\mathrm{kost}(t)$ mit $a \geq 1$ und da $t \geq 1$ ist:

$$\begin{aligned} \mathrm{kost}(at) \geq a \cdot \mathrm{kost}(t) \qquad &\Leftrightarrow 2a^2t^2(2\log_2(q-1)-1) \geq 2at^2(2\log_2(q-1)-1) \\ \Leftrightarrow 2a(\log_2(q-1)-1) \geq 2\log_2(q-1) - 1 \Leftrightarrow \qquad & a \geq 1 \end{aligned}$$

Der Erwartungswert der Rekursionstiefe ist $E(D) = \frac{2\log_2(t)}{\log_2(2q/(q-1))} + 1 + \frac{2q}{q+1} < 2\log_2(t) + 1 + 2$. Ich schätze somit die Anzahl der im Algorithmus durchzuführenden Operationen auf weniger als

$$\mathrm{kost}(t) \cdot E(D) < \left(6n^2\log_2(q-1) - 4n^2 + 20n\log_2(q-1) - 9n\right)(2\log_2(n) + 3),$$

das entspricht der Komplexität $\mathcal{O}(n^2\log(q)\log(n))$.

3.2 Der Spur–Algorithmus von Berlekamp

Dieses Verfahren verallgemeinert das Verfahren von Berlekamp und Rabin mit der Spur für gerade Charakteristik auf eine beliebige Charakteristik.

Es sei f ein quadratfreies normiertes Polynom aus $\mathbb{F}_q[x]$, das in lineare Faktoren zerfällt. f sei gegeben durch $f(x) = \sum_{i=0}^{n} a_i x^i = \prod_{i=1}^{t} f_i(x)$.

Ich definiere wegen der kürzeren Schreibweise das Polynom $T(x) = \mathrm{Spur}_{\mathbb{F}_q/\mathbb{F}_p}(x) = \sum_{i=0}^{s-1} x^{p^i}$. Es gibt für jedes $\lambda \in \mathbb{F}_p$ genau p^{s-1} Elemente $\alpha \in \mathbb{F}_q$ mit der Eigenschaft $T(\alpha) = \lambda$. Darum gilt

$$\prod_{\lambda \in \mathbb{F}_p} (T(x) - \lambda) = x^q - x.$$

Sei $\beta \in \mathbb{F}_q$ ein Erzeuger der multiplikativen Untergruppe des $\mathbb{F}_q$, dann sind $1, \beta, \beta^2, \ldots, \beta^{s-1}$ linear unabhängig über $\mathbb{F}_p$. Wenn ich in der obigen Gleichung für $0 \leq i < s$ das x durch $\beta^i x$ ersetze, so erhalte ich

$$\begin{aligned} \prod\nolimits_{\lambda \in \mathbb{F}_p} \left(T(\beta^i x) - \lambda\right) &= (\beta^i)^q x^q - \beta^i x \\ \Leftrightarrow \quad \prod\nolimits_{\lambda \in \mathbb{F}_p} \left(T(\beta^i x) - \lambda\right) &= \beta^i\,(x^q - x)\,. \end{aligned}$$

Da f über $\mathbb{F}_q$ in lineare Faktoren zerfällt, wird $x^q - x$ von f geteilt, also wird für $0 \leq i < s$ auch $\prod_{\lambda \in \mathbb{F}_p} \left(T(\beta^i x) - \lambda\right)$ von f geteilt. Deshalb ist für $0 \leq i < s$

$$f(x) = \prod_{\lambda \in \mathbb{F}_p} \operatorname{ggT}\left(f(x), T(\beta^i x) - \lambda\right). \tag{3}$$

Will man die Nullstellen eines Polynoms f berechnen, so bestimmt man sukzessive für alle $\lambda \in \mathbb{F}_p$ und alle $0 \leq i < s$ die Faktoren $\operatorname{ggT}\left(f(x), T(\beta^i x) - \lambda\right)$ von f.

Satz 3.8
Ist $\operatorname{Grad}(f) \geq 2$, so ist für mindestens ein i mit $0 \leq i < s$ die Faktorisierung, die durch Gleichung (3) erreicht wird, nicht trivial.

Beweis: Angenommen, die Aussage wäre nicht wahr, so gäbe es für jedes i mit $0 \leq i < s$ mindestens ein $\lambda_i \in \mathbb{F}_p$ mit der Eigenschaft $T(\beta^i x) \equiv \lambda_i \mod f(x)$. Insbesondere wäre dann für zwei verschiedene Nullstellen $\alpha_1, \alpha_2 \in \mathbb{F}_q$ die Gleichung

$$T(\beta^i \alpha_1) = T(\beta^i \alpha_2) = \lambda_i$$

wahr für alle $0 \leq i < s$. Die Spur ist linear, darum gälte $T(\beta^i(\alpha_1 - \alpha_2)) = 0$ für alle $0 \leq i < s$ und somit, da die β^i eine Basis des $\mathbb{F}_q$ bilden, $T(\alpha(\alpha_1 - \alpha_2)) = 0$ für alle $\alpha \in \mathbb{F}_q$.

Von der Spur werden genau p^{s-1} Elemente auf 0 abgebildet, darum müßte hier gelten

$$\alpha_1 - \alpha_2 = 0.$$

Dies wäre ein Widerspruch zur Wahl der Nullstellen α_1 und α_2 als verschieden. □

Da für jeweils zwei Nullstellen α_1 und α_2 der soeben durchgeführte Beweis richtig ist, findet man mit Hilfe der durch Gleichung (3) gegebenen Faktorisierungen alle linearen Faktoren von f.

Für $s = 1$ ist die Spur gerade die Identität, das Verfahren läuft auf wahlloses Ausprobieren hinaus. Darum habe ich es nicht programmiert.

Bemerkung 3.9
Im Fall $p = 2$ ist dieses Verfahren dem Verfahren von Berlekamp und Rabin mit der Spur nicht unähnlich, jedoch werden hier die Elemente $\delta \in \mathbb{F}_{\tilde{q}}$ nicht zufällig gewählt, sondern systematisch die konjugierten Elemente β^i ausgetestet.

Algorithmus 10 BERLEKAMP–SPUR $(f, \mathbb{F}_q)$
Eingabe: Ein normiertes quadratfreies Polynom $f \in \mathbb{F}_q[x]$, welches in lineare Faktoren zerfällt, und der Körper $\mathbb{F}_q$ mit $q = p^s$.
Ausgabe: Die Zerlegung von f in lineare Faktoren.

1. Setze $n \leftarrow \operatorname{Grad}(f)$ und $h(x) \leftarrow 1$. Ist $n \leq 1$, so liefere $f(x)$ zurück. Sonst setze $T(x) \leftarrow \sum_{i=0}^{s-1} x^{p^i}$.
2. Identifiziere ein $\beta \in \mathbb{F}_q$, welches Nullstelle eines irreduziblen Polynoms vom Grad s über $\mathbb{F}_p$ ist, und setze $g \leftarrow f$.
3. Berechne für alle gefundenen, nicht linearen Faktoren g von f und für alle $0 \leq i < s$ die partiellen Faktorisierungen $g(x) = \prod_{\lambda \in \mathbb{F}_p} \operatorname{ggT}\left(g(x), T(\beta^i x) - \lambda\right)$.
4. Liefere die Faktorisierung von $f(x)$ zurück.

Beispiel 3.4
Das Beispiel verläuft genau wie Beispiel 3.3 auf Seite 21, da wir damals zufällig gerade $\delta_1 = 1 = \xi^0$ gewählt hatten, $T(\xi x) \equiv 0 \mod f(x)$ ist und $\delta_2 = \xi^2$ die gesamte Faktorisierung ergab.

Komplexität

Wir suchen eine Nullstelle des $\mathbb{F}_q$ definierenden Polynoms g. Ist $g \in \mathbb{F}_p[\xi]$, so ist die gesuchte Nullstelle ξ, um sie zu bestimmen ist kein Rechenaufwand nötig.

Der Algorithmus durchläuft höchstens s Schritte, in denen jeweils $T(\xi^i x)$ und dann $p-1$ mal ein größter gemeinsamer Teiler bestimmt wird.

$\xi^i \mod g(\xi)$ zu bestimmen, kostet jeweils $2\log_2(i)(3n^2+7n+4)$ elementare Operationen. Die Spur wird durch $s-1$ Potenzierungen mit p und $s-1$ Additionen berechnet, kostet also

$(s-1)\left(2\log_2(p)(3n^2+7n+4)+n\right)$ Operationen.

Einen größten gemeinsamen Teiler zu bestimmen kostet hier $2n^2+3n$ Operationen.

Insgesamt braucht der Spur–Algorithmus also höchstens

$(s-1)\left(2\cdot\lfloor\log_2(p)\rfloor\cdot(3n^2+7n+4)+n+(p-1)\cdot(2n^2+3n)\right)$ Operationen,

er hat also die Komplexität $\mathcal{O}(s\cdot p\cdot n^2)$.

3.3 Das Finden von Nullstellen mit linearisierten Polynomen

1967 führten E. R. Berlekamp, H. Rumsey und G. Solomon in dem Artikel [4] ein Verfahren zum Finden der Nullstellen ein, das sich linearisierter Polynome bedient. Um es zu beschreiben, muß ich zwei neue Begriffe einführen:

Definition 3.10
Ein Polynom

$$L(x)=\sum_{i=0}^{l}\alpha_i x^{q^i}$$

mit Koeffizienten $\alpha_i \in \mathbb{F}_{q^k}$ *für* $0\leq i\leq l$ *und* $k,l\in\mathbb{N}$ *heißt* q*–Polynom über* $\mathbb{F}_{q^k}$*. Für ein vorgegebenes festes* q *wird* L *auch als linearisiertes Polynom über* $\mathbb{F}_{q^k}$ *bezeichnet.*

Definition 3.11
Ein Polynom

$$A(x)=L(x)+\alpha$$

mit $\alpha\in\mathbb{F}_{q^k}$ *und einem linearisierten Polynom* $L(x)$ *über* $\mathbb{F}_{q^k}$ *heißt Affines Polynom über* $\mathbb{F}_{q^k}$*.*

Für das Polynom $f(x)\in\mathbb{F}_{q^k}$ ist das kleinste affine Vielfache von f definiert als das Affine Polynom $A\in\mathbb{F}_{q^k}[x]$ kleinsten Grades, das von f geteilt wird. Ein solches kleinstes affines Vielfaches existiert immer, da es immer ein $N\in\mathbb{N}$ gibt, so daß das linearisierte Polynom $x^{q^N}-x$ von $f(x)$ geteilt wird.

Theorie

Sei $f(x)=\sum_{i=0}^{n}a_i x^i\in\mathbb{F}_q[x]$ wie immer normiert und quadratfrei und zerfalle in lineare Faktoren $f(x)=\prod_{i=1}^{n}f_i(x)$. Wir interessieren uns für die p–Polynome über $\mathbb{F}_q$ mit $q=p^s$.

Lemma 3.12
Sei $A(x)$ *das kleinste affine Vielfache von* $f\in\mathbb{F}_q[x]$ *mit* $\mathrm{Grad}(f)=n$*. Dann gilt*

$$\mathrm{Grad}(A)\leq p^{n-1}.$$

Beweis: Der Wert $l \in \mathbb{N}$ und die Elemente $\alpha, \alpha_i \in \mathbb{F}_q$ für $0 \leq i \leq l$ werden für das kleinste affine Vielfache $A(x) = L(x) + \alpha = \sum_{i=0}^{l} \alpha_i x^{p^i} + \alpha$ von $f(x)$ gesucht. Ich berechne sukzessive für $k = 0, 1, \ldots$ die Ausdrücke

$$x^{p^i} \equiv \sum_{j=0}^{n-1} b_{j,i} x^j \mod f(x) \text{ für } 0 \leq i \leq k.$$

Wird A von f geteilt, so gilt

$$\begin{aligned} & \alpha + \textstyle\sum_{i=0}^{k} \alpha_i \sum_{j=0}^{n-1} b_{j,i} x^j && \equiv 0 \mod f(x) \\ \Leftrightarrow\quad & \alpha + \textstyle\sum_{i=0}^{k} \alpha_i \sum_{j=0}^{n-1} b_{j,i} x^j && = 0, \end{aligned}$$

da das Polynom auf der linken Seite höchstens den Grad $n-1$ haben kann. Ich stelle für $k = 0, 1, \ldots$ das Gleichungssystem

$$\begin{bmatrix} 1 & b_{0,0} & b_{0,1} & \ldots & b_{0,k} \\ 0 & b_{1,0} & b_{1,1} & \ldots & b_{1,k} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & b_{n-2,0} & b_{n-2,1} & \ldots & b_{n-2,k} \\ 0 & b_{n-1,0} & b_{n-1,1} & \ldots & b_{n-1,k} \end{bmatrix} \begin{pmatrix} \alpha \\ \alpha_0 \\ \vdots \\ \alpha_k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}$$

auf und löse es. Für das erste $k \in \mathbb{N}$, für das der Lösungsraum nicht trivial ist, gilt $l = k$. Der Vektor $(\alpha, \alpha_0, \ldots, \alpha_k)^\top$ bezeichnet dann das kleinste affine Vielfache A von f über $\mathbb{F}_q$. Ist $k \geq n$, so besteht die Matrix aus n Zeilen und mindestens $n+2$ Spalten. Der Rang der Matrix kann nun höchstens n sein, also ist die Dimension des Kerns der Matrix mindestens zwei. Wir suchen jedoch einen Kern der Dimension 1, darum gilt $l < n$. □

Durch diesen Beweis ist gleichzeitig das Verfahren gegeben, mit dem man das kleinste affine Vielfache eines Polynoms bestimmt.

Mit Hilfe des kleinsten affinen Vielfachen ergibt sich eine neue Möglichkeit, die Nullstellen von f zu bestimmen: Ich bestimme zuerst die Nullstellen des kleinsten affinen Vielfachen A von f über $\mathbb{F}_q$ und teste dann, ob diese Nullstellen auch welche des Polynoms f sind. Dieses Verfahren ist effizient, da die Nullstellen von A leicht zu bestimmen sind:

Sei $\{\gamma_1, \gamma_2, \ldots, \gamma_s\}$ eine Basis des $\mathbb{F}_q$ über $\mathbb{F}_p$ und habe ein $\gamma \in \mathbb{F}_q$ bezüglich dieser Basis die Darstellung $\gamma = \sum_{i=1}^{s} c_i \gamma_i$ mit $c_i \in \mathbb{F}_p$ für $1 \leq i \leq s$, so ist γ Nullstelle des Affinen Polynoms $A(x) = L(x) + \alpha$ genau dann, wenn $\sum_{i=1}^{s} c_i L(\gamma_i) = -\alpha$ ist.

Ich berechne $L(\gamma_i)$ für alle $1 \leq i \leq s$ und stelle die Ergebnisse bezüglich der Basis dar:

$$L(\gamma_i) = \sum_{j=1}^{s} l_{j,i} \gamma_j \text{ mit } l_{j,i} \in \mathbb{F}_p \text{ für alle } 1 \leq i, j \leq s.$$

Ist $-\alpha = \sum_{i=1}^{s} d_i \gamma_i$ mit $d_i \in \mathbb{F}_p$ für $1 \leq i \leq s$ die Darstellung von $-\alpha$ bezüglich der Basis $\{\gamma_1, \gamma_2, \ldots, \gamma_s\}$, so ergibt sich das Gleichungssystem

$$\begin{bmatrix} l_{1,1} & l_{1,2} & \ldots & l_{1,s} \\ l_{2,1} & l_{2,2} & \ldots & l_{2,s} \\ \vdots & \vdots & & \vdots \\ l_{n-1,1} & l_{n-1,2} & \ldots & l_{n-1,s} \end{bmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_s \end{pmatrix} = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_s \end{pmatrix}.$$

Jeder Lösungsvektor $(c_1, c_2, \ldots, c_s)^\top$ liefert eine Nullstelle $\gamma = \sum_{i=1}^{s} c_i \gamma_i$ von $A(x)$ in $\mathbb{F}_q$.

Praxis

Das Verfahren zur Bestimmung der Nullstellen eines Polynoms f ist aus der Theorie heraus völlig klar. Aber auch dieses Verfahren werde ich nicht programmieren, da ich ja nur über Körpern mit Primzahlordnung $q = p$ faktorisieren möchte. Würde man das Verfahren auf ein Polynom $f \in \mathbb{F}_p[x]$ anwenden, so wäre

- $x^{p^0} \equiv x \mod f(x)$ und
- $x^{p^1} \equiv x \mod f(x)$,

da f nur aus linearen Faktoren besteht. Somit wäre das kleinste affine Vielfache von f gegeben als

$$A(x) = x^p - x + 0.$$

Die Nullstellen von A sind genau alle $\gamma \in \mathbb{F}_p$. Das Finden der Nullstellen von f wäre darum einfaches Ausprobieren, was keine Verbesserung des herkömmlichen Verfahrens von Berlekamp und Rabin aus Abschnitt 3.1 auf Seite 15 ist.

Algorithmus 11 LINEARISIERT $(f, \mathbb{F}_q)$
Eingabe: Ein normiertes quadratfreies Polynom $f \in \mathbb{F}_q[x]$, welches in lineare Faktoren zerfällt, und der Körper $\mathbb{F}_q$ mit $q = p^s$.
Ausgabe: Die Zerlegung von f in lineare Faktoren.

1. Berechne das kleinste affine Vielfache $A(x)$ von $f(x)$. $A(x)$ ist ein p–Polynom.
2. Berechne die linearen Faktoren von $A(x)$ und teste, ob sie Teiler von $f(x)$ sind.
3. Liefere die linearen Faktoren von $f(x)$ zurück.

Beispiel 3.5
Sei $p = 2$, $q = 16$ mit $\mathbb{F}_{16} = \mathbb{F}_2[\xi]/(\xi^4 + \xi + 1)\mathbb{F}_2[\xi]$, wir möchten das quadratfreie Polynom

$$f(x) = x^3 + (\xi^3 + \xi + 1)x^2 + (\xi^3 + 1)x + \xi + 1,$$

welches ausschließlich in lineare Faktoren zerfällt, in diese zerlegen. Es ist

$$x^4 \equiv (\xi^3 + \xi^2 + \xi + 1)x + \xi^3 + \xi^2 + \xi \mod f(x),$$

finde also die Basis $B = \{(\xi^3 + \xi^2 + \xi, \xi^3 + \xi^2 + \xi + 1, 0, 1)^\top\}$ des Kerns der Matrix

$$\begin{bmatrix} 1 & 0 & 0 & \xi^3 + \xi^2 + \xi \\ 0 & 1 & 0 & \xi^3 + \xi^2 + \xi + 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Das kleinste affine Vielfache von f ist somit $A(x) = x^4 + (\xi^3 + \xi^2 + \xi + 1)x + \xi^3 + \xi^2 + \xi$, wir arbeiten mit dem linearisierten Polynom $L(x) = x^4 + (\xi^3 + \xi^2 + \xi + 1)x$.
Eine Basis des $\mathbb{F}_{16}$ über $\mathbb{F}_2$ ist die Menge $\{1, \xi, \xi^2, \xi^3\}$, jedes $\gamma \in \mathbb{F}_{16}$ hat eine eindeutige Darstellung als $\gamma = \sum_{i=0}^{3} c_i \xi^i$. Ich berechne $L(\xi^i)$ für jedes $i \in \{0, 1, 2, 3\}$:

$$\begin{aligned} L(1) &= \xi^3 + \xi^2 + \xi, & L(\xi) &= \xi^3 + \xi^2 + \xi, \\ L(\xi^2) &= \xi^3 + \xi^2, & L(\xi^3) &= \xi^3 + \xi^2 + \xi. \end{aligned}$$

Hier ist $\alpha = \xi^3 + \xi^2 + \xi$, ich löse also das lineare Gleichungssystem

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

durch die beiden Gleichungen $c_2 = 0$ und $c_0 + c_1 + c_3 = 1$. Die Nullstellen von A sind also die Elemente der Menge $\{1, \xi, \xi^3, \xi^3 + \xi + 1\}$, davon sind $1, \xi$ und ξ^3 auch Nullstellen von f.

Komplexität

Der Algorithmus benötigt maximal $n-1$ Schritte, deren jeder aus der Berechnung einer p–ten Potenz und einer Gaußelimination besteht. Diese Schritte liefern das kleinste affine Vielfache von f.

Eine Potenzierung mit p kostet $2 \cdot \lfloor \log_2(p) \rfloor \cdot (3n^2 + 7n + 4)$ elementare Operationen.

Bestimmen wir nun den Nullraum der Matrix mit Gaußelimination, so werden mit der ersten Zeile $n-1$ Zeilen addiert, mit der zweiten $n-2$ usw., dann wird dieselbe Prozedur mit Start bei der letzten Zeile wiederholt. Zwei Zeilen zu addieren kostet n Additionen, vor der Addition muß aber die eine Zeile und das entsprechende Element des Vektors auf der rechten Seite noch mit einem $\alpha \in \mathbb{F}_q$ multipliziert werden. Die Umwandlung der Matrix in „Diagonalform" kostet also $2\frac{n(n-1)}{2} \cdot 2(n+1) = 2n^3 - 2n$ Operationen, der Lösungsraum läßt sich ablesen.

Nachdem wir das kleinste affine Vielfache von f gefunden haben, müssen wir seine Nullstellen bestimmen, indem wir für die Basiselemente γ_i mit $1 \leq i \leq s$ des $\mathbb{F}_q$ über $\mathbb{F}_p$ jeweils $L(\gamma_i)$ bestimmen. Das kostet zusammen $s(n-1)(2\log_2(p) \cdot (3n^2 + 7n + 4) + n)$ Operationen.

Dann muß ein lineares Gleichungssystem in $2n^3 - 2n$ elementaren Operationen gelöst werden und schließlich jede gefundene Nullstelle γ, die in $2s$ elementaren Additionen und Multiplikationen bestimmt wird, darauf geprüft werden, ob sie Nullstelle von f ist. Dies kostet jeweils $2(n+1)$ elementare Operationen.

Insgesamt braucht der Algorithmus

$(n-1)(2\log_2(p) \cdot (3n^2+7n+4) + 2n^3 - 2n) + s(n-1)(2\log_2(p) \cdot (3n^2+7n+4) + n) + 2n^3 - 2n + 2s^2 + 2s(n+1)$ Operationen,

hat also die Komplexität $\mathcal{O}(n^4 + sn^3\log(p) + s^2)$.

4 Die Berlekamp–Unteralgebra

1970 führte Berlekamp in [2] die Berlekamp–Unteralgebra ein, auf die viele Faktorisierungsmethoden zumindest in der Theorie zugreifen.

Ich betrachte ein quadratfreies normiertes Polynom f vom Grad $n \geq 1$ aus $\mathbb{F}_q[x]$, welches ich faktorisieren möchte. f ist gegeben als $f(x) = \sum_{i=0}^{n} a_i x^i$ und hat eine eindeutige Faktorisierung $f(x) = \prod_{i=1}^{t} f_i(x)$, wobei die $f_i \in \mathbb{F}_q[x]$ paarweise verschiedene irreduzible normierte Polynome mit $\mathrm{Grad}(f_i) \geq 1$ sind.

Zum Faktorisieren erzeugt man oft eine aufspaltende Menge:

Definition 4.1
Eine Menge S von Polynomen über $\mathbb{F}_q$ heißt aufspaltende Menge für f, wenn es für jedes Paar (i, j) mit $1 \leq i < j \leq t$ ein $s \in S$ und ein $\alpha \in \mathbb{F}_q$ gibt, so daß

$$\begin{aligned} f_i(x) &\mid \mathrm{ggT}\left(f(x), s(x) - \alpha\right) \text{ und} \\ f_j(x) &\nmid \mathrm{ggT}\left(f(x), s(x) - \alpha\right). \end{aligned}$$

Es sei

$$\mathcal{R} = \mathbb{F}_q[x]/(f\mathbb{F}_q[x])$$

der Ring der Polynome über $\mathbb{F}_q$ modulo f. Ich repräsentiere die Polynome einer Äquivalenzklasse eindeutig durch das des kleinsten Grades. Sei $\mathcal{R}^{(i)} = \mathbb{F}_q[x]/(f_i\mathbb{F}_q[x])$, so gilt mit dem Chinesischen Restsatz

$$\mathcal{R} \cong \mathcal{R}^{(1)} \times \mathcal{R}^{(2)} \times \ldots \times \mathcal{R}^{(t)}, \tag{4}$$

dabei ist jedes $\mathcal{R}^{(i)}$ mit $1 \leq i \leq t$ ein Körper über $\mathbb{F}_q$. Laut Chinesischem Restsatz gibt es für jedes i ein eindeutig bestimmtes Polynom $e_i \in \mathbb{F}_q[x]$ mit

1. $\mathrm{Grad}(e_i) < n$,
2. $e_i \equiv 1 \mod f_i(x)$,
3. $e_i \equiv 0 \mod f_j(x)$ für alle $i \neq j$ mit $1 \leq j \leq t$.

Die e_i sind paarweise orthogonal zueinander, das heißt $e_i e_j \equiv 0 \mod f(x)$ für alle $i \neq j, i, j \in \{1, \ldots, t\}$. Außerdem sind die e_i idempotent, das bedeutet $e_i(x)^2 \equiv e_i(x) \mod f(x)$. Es gilt

$$\begin{aligned} \sum_{i=1}^{t} e_i(x) &\equiv 1 \qquad \mod f_j(x) \text{ für alle } 1 \leq j \leq t, \\ \Leftrightarrow \sum_{i=1}^{t} e_i(x) &\equiv 1 \qquad \mod f(x), \\ \Leftrightarrow \sum_{i=1}^{t} e_i(x) &= 1, \end{aligned}$$

da die e_i alle einen Grad echt kleiner als n haben. Ich weiß für jedes $g \in \mathcal{R}$, daß gilt

$$g(x) = g(x) \cdot 1 = g(x) \sum_{i=1}^{t} e_i(x) = \sum_{i=1}^{t} g(x) e_i(x).$$

Sei $g(x) \equiv g_i(x) \mod f_i(x)$ für $1 \leq i \leq t$, dann ist $g(x) \equiv \sum_{i=1}^{t} g_i(x) e_i(x) \mod f(x)$. Also kann ich jedes $g \in \mathcal{R}$ schreiben als

$$g(x) = \sum_{i=1}^{t} g_i(x) e_i(x) \text{ mit } \mathrm{Grad}(g_i) < \mathrm{Grad}(f_i).$$

Ich stelle ein Element $g \in \mathcal{R}$ eindeutig dar als den Vektor der Polynome $(g_1(x), g_2(x), ..., g_t(x))$ mit $\mathrm{Grad}(g_i) < \mathrm{Grad}(f_i)$. Diese Darstellung nenne ich die Darstellung in Komponentenschreibweise. Für die Skalare $\alpha \in \mathbb{F}_q$ gilt

$g(x) = \alpha$ entspricht in der Komponentenschreibweise dem Vektor $(\alpha, \alpha, ..., \alpha)$.

Es wird komponentenweise addiert und multipliziert, also wird auch komponentenweise potenziert. $g(x)$ ist darum idempotent genau dann, wenn $g_i(x) \in \{0, 1\}$ für alle $1 \le i \le t$. Es gilt $g(x)^q \equiv g(x) \mod f(x)$ genau dann, wenn $g_i(x) \in \mathbb{F}_q$ für alle $1 \le i \le t$. Ist $\mathbb{F}_r$ mit $r = p^u$ ein Unterkörper von $\mathbb{F}_q$, so ist $g(x)^r \equiv g(x) \mod f(x)$ genau dann, wenn $g_i(x) \in \mathbb{F}_r$ für alle $1 \le i \le t$.

Definition 4.2
$\mathcal{B}(f, r) = \left\{ b(x) \in \mathcal{R} \mid b(x) = \sum_{i=1}^{t} c_i e_i(x)\ \text{mit}\ c_i \in \mathbb{F}_r\ \text{für alle}\ 1 \le i \le t \right\}$ ist ein t–dimensionaler $\mathbb{F}_r$–linearer Untervektorraum von $\mathcal{R}$ und heißt Berlekamp–Unteralgebra.

Die beiden Fälle $r = p$ und $r = q$ sind mindestens möglich, die klassische Theorie der Berlekamp–Unteralgebra arbeitet ausschließlich mit $r = q$. $\mathcal{B}(f, p)$ wird von S. Gao und J. von zur Gathen in [13] auch als absolute Berlekamp–Unteralgebra bezeichnet. Ich kann $\mathcal{B}(f, r)$ schreiben als

$$\mathcal{B}(f, r) = \{b(x) \in \mathcal{R} \mid b(x)^r \equiv b(x) \mod f(x)\}.$$

$b(x) = \sum_{i=1}^{t} c_i e_i(x) \in \mathcal{B}(f, r)$ mit $c_i \in \mathbb{F}_r$ ist Einheit modulo f genau dann, wenn für alle i mit $1 \le i \le t$ gilt $c_i \ne 0$.

Satz 4.3
Ist $b(x) \in \mathcal{B}(f, r)$ nicht Einheit und ist $b \ne 0$, so liefert $\mathrm{ggT}\,(f, b)$ einen nicht trivialen Faktor von f.

Beweis: b hat mindestens eine Komponente, die 0 ist, und eine, die nicht 0 ist. Ist die i–te Komponente 0, so ist $b(x) \equiv 0 \mod f_i$ für $c_i = 0$, was gleichbedeutend ist mit $f_i \mid \mathrm{ggT}(f, b)$. Ist die i–te Komponente nicht 0, so gilt $f_i \nmid \mathrm{ggT}(f, b)$. □

Satz 4.4
Ist $b(x) \in \mathcal{B}(f, r)$, so gibt es für jedes i mit $1 \le i \le t$ ein $\alpha \in \mathbb{F}_r$ mit $f_i(x) \mid \mathrm{ggT}\,(f(x), b(x) - \alpha)$.

Beweis: b habe die i–te Komponente c_i, dann hat $b(x) - c_i$ die i–te Komponente 0. Somit teilt $f_i(x)$ sowohl $b(x) - c_i$ als auch $f(x)$. □

Es gilt also für jedes $b(x) \in \mathcal{B}(f, r)$:

$$f(x) = \prod_{\alpha \in \mathbb{F}_r} \mathrm{ggT}\,(f(x), b(x) - \alpha)\,, \tag{5}$$

denn die i–te Komponente von $b(x) - \alpha$ ist 0 für genau ein $\alpha \in \mathbb{F}_r$, weshalb im obigen Produkt jede Komponente durch genau ein α anulliert wird.

Sei I eine Indexmenge. Die Menge $S = \{s_i(x) \mid i \in I, s_i \in \mathcal{B}(f, r)\}$ ist genau dann eine aufspaltende Menge für f, wenn für jedes Paar (k, l) mit $1 \le k < l \le t$ ein $i \in I$ existiert, so daß $s_{ik} \ne s_{il}$. Dabei beschreibt s_{ik} die k–te Komponente von s_i. Jede Basis und somit auch jedes Erzeugendensystem der Berlekamp–Unteralgebra $\mathcal{B}(f, r)$ ist nun eine aufspaltende Menge. Eine aufspaltende Menge kann aber auch aus nur einem einzigen Element bestehen, das in den t Komponenten der Komponentendarstellung paarweise verschiedene Werte hat.

4.1 Das Verfahren von Berlekamp

Das Verfahren von Berlekamp ergibt sich ganz natürlich aus der Bestimmung der Berlekamp–Unteralgebra $\mathcal{B}(f, r)$ mit $r = p^u$ und $q = p^s$:

Ich suche eine Basis B des Lösungsraums der Gleichung

$$b(x)^r \equiv b(x) \mod f(x) \tag{6}$$

und bilde danach für alle $b \in B$ und alle $\alpha \in \mathbb{F}_r$

$$\mathrm{ggT}\,(f(x), b(x) - \alpha)\,,$$

um eine komplette Faktorisierung zu erhalten, da B Basis der Berlekamp–Unteralgebra und somit aufspaltende Menge für f ist — dabei rechne ich nach jeder Teilfaktorisierung mit den gefundenen Faktoren weiter.

Theorie

Die Lösungen der Gleichung (6) finde ich, indem ich sie auf ein lineares Gleichungssystem bringe: n ist der Grad von f, ich erzeuge die $(n \times n)$–Matrix $B_{r,1}(f) = ((\beta_{ij}))$ für $0 \leq i, j \leq n-1$. Dabei ist

$$x^{ir} \equiv \sum_{j=0}^{n-1} \beta_{ij} x^j \mod f(x) \text{ für } 0 \leq i \leq n-1.$$

Die Matrix $B_{r,1}(f)$ wird als Berlekamp–Matrix (bezüglich r) bezeichnet, warum ich den zweiten Index notiere, wird auf der nächsten Seite klar. Ein Polynom $g \in \mathbb{F}_q[x]$ mit $\mathrm{Grad}(g) < n$ hat die Form

$$g(x) = \sum_{i=0}^{n-1} \gamma_i x^i$$

und soll hier dem Vektor $(\gamma_0, \gamma_1, \ldots, \gamma_{n-1})$ entsprechen. Diese Vektordarstellung hat natürlich nichts mit der Darstellung in Komponentenschreibweise zu tun, sondern erleichtert nur das Rechnen mit dem Polynom g. Nun ist

$$g(x)^r = \sum_{i=0}^{n-1} \gamma_i^r x^{ir} \equiv \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \beta_{ij} \gamma_i^r x^j \mod f(x),$$

die Gleichung (6) wird somit genau dann gelöst, wenn

$$\sum_{i=0}^{n-1} \gamma_i x^i = \sum_{j=0}^{n-1} \left(\sum_{i=0}^{n-1} \beta_{ji} \gamma_i^r \right) x^j$$

gilt. Damit ergibt sich das Gleichungssystem

$$B_{r,1}(f)^\top \begin{pmatrix} \gamma_0^r \\ \gamma_1^r \\ \vdots \\ \gamma_{n-1}^r \end{pmatrix} = \begin{pmatrix} \gamma_0 \\ \gamma_1 \\ \vdots \\ \gamma_{n-1} \end{pmatrix}. \tag{7}$$

Im Fall $r = q$ ist $\gamma_i^r = \gamma_i$ für alle $0 \leq i < n$ und das Gleichungssystem ist linear. Ich löse dann

$$(B_{q,1}(f)^\top - I) \begin{pmatrix} \gamma_0 \\ \gamma_1 \\ \vdots \\ \gamma_{n-1} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

dabei ist I die $(n \times n)$–Einheitsmatrix. Der Kern der Matrix $B_{q,1}(f)^\top - I$ ist genau $\mathcal{B}(f,q)$ in Vektorschreibweise, darum ist die Dimension des Nullraums von $B_{q,1}(f)^\top - I$ genau t, die Anzahl der verschiedenen irreduziblen Faktoren von f. Der Rang der Matrix $B_{q,1}(f)^\top - I$ ist also $n-t$. Ich bestimme eine Basis dieses Nullraums, um eine aufspaltende Menge S zu erhalten.

Ist hingegen $r < q$, so ist (7) ein algebraisches Gleichungssystem. Um es zu lösen, bestimmt man eine Normalbasis und stellt das entsprechende lineare $(nv \times nv)$–Gleichungssystem $B_{r,v}(f)^\top - I = 0$ auf — nun erklärt sich die Indexschreibweise $B_{r,1}(f)$ auf der vorherigen Seite.

Bemerkung 4.5
Eine Normalbasis von $\mathbb{F}_q$ über $\mathbb{F}_r$ ist eine geordnete Basis der Form $B = \{\alpha, \alpha^r, \alpha^{r^2}, \ldots, \alpha^{r^{(s/u)-1}}\}$ mit einem $\alpha \in \mathbb{F}_q$. Eine solche Normalbasis existiert für jede Körpererweiterung $\mathbb{F}_q/\mathbb{F}_r$ und erleichtert die Behandlung der linken Seite der Gleichung.

Praxis

Das konstante Polynom $b_1(x) = 1$, welches dem Vektor $(1,0,\ldots,0)^\top$ entspricht, ist immer Lösung der Gleichung (6) und außerdem immer in der Basis enthalten, die ich durch Umformung der Matrix $B_{r,v}(f)^\top - I$ mit Spaltenoperationen bekomme. $b_1 = 1$ trägt jedoch nicht zur Faktorisierung bei. Darum nehme ich dieses Basiselement bei der Implementation des Algorithmus nicht in die aufspaltende Menge S auf.

Bemerkung 4.6
Statt 1 aus B herauszunehmen und für alle $\alpha \in \mathbb{F}_r$ und alle $b(x) \in B$ den $\mathrm{ggT}(f(x), b(x) - \alpha)$ zu bilden, könnte man auch ein beliebiges Element $\beta(x) \in B$ festhalten, es aus B entfernen und dann für alle $\alpha \in \mathbb{F}_r$ und alle $b(x) \in B$ den $\mathrm{ggT}(f(x), b(x) - \alpha\beta(x))$ bilden. So würde ebenfalls eine komplette Faktorisierung erreicht, allerdings müßte man dafür zuerst jeweils β mit α multiplizieren und dann zwei Polynome subtrahieren, um auf $b(x) - \alpha\beta(x)$ zu kommen, das würde somit $2n$ elementare Operationen kosten. Wählt man hingegen $\beta(x) = 1$, so entfällt die Multiplikation und die Subtraktion beschränkt sich auf das konstante Glied, darum kostet diese Konstruktion nur eine elementare Operation.

Bemerkung 4.7
Ein echter Vorteil der Berlekamp–Unteralgebra $\mathcal{B}(f,r)$ ist, daß für ihre Elemente b nur $r-1$ statt $q-1$ größte gemeinsame Teiler $\mathrm{ggT}(f(x), b(x) - \alpha)$ bestimmt werden müssen. Will man unbedingt mit Normalbasen arbeiten, so ist also die Bestimmung der Berlekamp–Unteralgebra $\mathcal{B}(f,p)$ anzuraten.

Das Hantieren mit Normalbasen erfordert einen zusätzlichen Aufwand, darum wählt man $r = q$.

Algorithmus 12 BERLEKAMP $(f, \mathbb{F}_q)$
Eingabe: Das normierte, quadratfreie Polynom $f \in \mathbb{F}_q[x]$ und der Körper $\mathbb{F}_q$.
Ausgabe: Die Faktorisierung von f.
Programm: Berlekamp.mws: brl

1. Setze $n \leftarrow \mathrm{Grad}(f)$ und berechne $B \leftarrow B_{q,1}(f)^\top$.
2. Berechne eine Basis $orth$ des Nullraums der Matrix $B - I$ und interpretiere die Basiselemente als Basispolynome der Berlekamp–Unteralgebra $\mathcal{B}(f,q)$. Die Basis ist eine Liste $b_1(x), \ldots, b_t(x)$ von Polynomen. Setze $i \leftarrow 1$.
3. Ist $b_i(x) \neq 1$, so berechne für alle bisher gefundenen Faktoren $g(x)$ von $f(x)$ die partielle Faktorisierung $g(x) = \prod_{\alpha \in \mathbb{F}_q} \mathrm{ggT}(g(x), b_i(x) - \alpha)$.
4. Ist die Anzahl der bisher gefundenen Faktoren von $f(x)$ kleiner als t, so setze $i \leftarrow i+1$ und gehe zu Schritt 3.
5. Liefere die Faktorisierung von $f(x)$ zurück.

Ich werde jede der von mir hier und in den folgenden Abschnitten beschriebenen Faktorisierungsmethoden an jeweils denselben beiden Beispielen illustrieren. Ich wähle immer dieselben Beispiele, damit die Unterschiede in den Faktorisierungsmethoden besser nachvollziehbar sind. Die drei Standardverfahren, die mit Abbildungen in multiplikative Untergruppen beziehungsweise mit der Spur arbeiten, kennen wir bereits aus Abschnitt 3.1 auf Seite 15. Weicht in den folgenden Verfahren die Behandlung der Charakteristik $p = 2$ von der Behandlung der ungeraden Charakteristik ab, so werde ich jede der Behandlungen illustrieren, den Fall der ungeraden Charakteristik in $\mathbb{F}_7$, den der geraden in $\mathbb{F}_4$. Ich wähle diese relativ kleinen Körper, damit die Beispiele handhabbar bleiben.

Unterscheidet das Verfahren nicht zwischen gerader und ungerader Charakteristik, so werde ich nur das Beispiel in $\mathbb{F}_7$ ausführen. Die Beispiele sind quadratfreie Polynome, bei denen der Grad d der irreduziblen Faktoren bekannt ist.

1. $f(x) = x^{12} + 4x^{11} + 2x^{10} + 2x^9 + 6x^6 + 3x^4 + 3x^3 + 4x^2 + 5$ ist quadratfrei über dem Körper $\mathbb{F}_7$ und zerfällt in vier Faktoren vom Grad drei.

2. $f(x) = x^9 + \xi x^8 + (\xi + 1)x^7 + \xi x^4 + (\xi + 1)x^3 + \xi x + 1$ ist quadratfrei über dem Körper $\mathbb{F}_4 = \mathbb{F}_2[\xi]/\left((\xi^2 + \xi + 1)\mathbb{F}_2[\xi]\right)$ und zerfällt in drei Faktoren vom Grad drei.

Beispiel 4.1
Betrachte

$$f(x) = x^{12} + 4x^{11} + 2x^{10} + 2x^9 + 6x^6 + 3x^4 + 3x^3 + 4x^2 + 5 \in \mathbb{F}_7[x].$$

Dann ist

$$\begin{aligned}
x^{2\cdot 7} &\equiv 6x^{11} + x^{10} + x^8 + 3x^7 + 4x^6 + 2x^5 + x^4 + 2x^3 + 2x^2 + 6x \mod f(x),\\
x^{3\cdot 7} &\equiv 4x^{11} + 6x^{10} + 6x^9 + x^7 + x^6 + 6x^5 + 3x^4 + 2x \mod f(x),\\
x^{4\cdot 7} &\equiv x^{11} + 5x^{10} + x^8 + x^7 + 2x^6 + 4x^5 + 2x^4 + 2x^3 + x^2 + 3x + 4 \mod f(x),\\
x^{5\cdot 7} &\equiv x^{10} + 6x^9 + 4x^8 + 2x^7 + 5x^6 + 3x^5 + 5x^4 + 2x^2 + 3x + 3 \mod f(x),\\
x^{6\cdot 7} &\equiv 3x^{11} + 4x^9 + 4x^6 + 6x^5 + 5x^4 + 3x^3 + 6x^2 + 6x + 3 \mod f(x),\\
x^{7\cdot 7} &\equiv 4x^{10} + 3x^9 + 5x^8 + 4x^6 + x^5 + 6x^4 + 4x^3 + x^2 + 4x + 4 \mod f(x),\\
x^{8\cdot 7} &\equiv 2x^{11} + 5x^{10} + 3x^9 + 6x^7 + 5x^6 + 3x^5 + x^2 \mod f(x),\\
x^{9\cdot 7} &\equiv x^{11} + 2x^{10} + 5x^9 + 2x^7 + 4x^5 + 5x^4 + 4x^3 + 3x^2 + 4x + 6 \mod f(x),\\
x^{10\cdot 7} &\equiv 6x^{11} + x^{10} + 2x^9 + 6x^8 + 2x^7 + x^6 + 2x^5 + 3x^4 + x^3 + 2x^2 + 4 \mod f(x),\\
x^{11\cdot 7} &\equiv 3x^{11} + 3x^{10} + 4x^9 + 2x^8 + 5x^6 + x^4 + 3x^3 + 2x^2 + 6 \mod f(x).
\end{aligned}$$

Darum ergibt sich die Matrix

$$B_{7,1}(f)^\top - I = \begin{bmatrix}
0 & 0 & 0 & 0 & 4 & 3 & 3 & 4 & 0 & 6 & 4 & 6\\
0 & 6 & 6 & 2 & 3 & 3 & 6 & 4 & 0 & 4 & 0 & 0\\
0 & 0 & 1 & 0 & 1 & 2 & 6 & 1 & 1 & 3 & 2 & 2\\
0 & 0 & 2 & 6 & 2 & 0 & 3 & 4 & 0 & 4 & 1 & 3\\
0 & 0 & 1 & 3 & 1 & 5 & 5 & 6 & 0 & 5 & 3 & 1\\
0 & 0 & 2 & 6 & 4 & 2 & 6 & 1 & 3 & 4 & 2 & 0\\
0 & 0 & 4 & 1 & 2 & 5 & 3 & 4 & 5 & 0 & 1 & 5\\
0 & 1 & 3 & 1 & 1 & 2 & 0 & 6 & 6 & 2 & 2 & 0\\
0 & 0 & 1 & 0 & 1 & 4 & 0 & 5 & 6 & 0 & 6 & 2\\
0 & 0 & 0 & 6 & 0 & 6 & 4 & 3 & 3 & 4 & 2 & 4\\
0 & 0 & 1 & 6 & 5 & 1 & 0 & 4 & 5 & 2 & 0 & 3\\
0 & 0 & 6 & 4 & 1 & 0 & 3 & 0 & 2 & 1 & 6 & 2
\end{bmatrix},$$

deren Nullraum von den Basisvektoren $(1,0,0,0,0,0,0,0,0,0,0,0)^\top$, $(0,6,1,2,0,6,3,4,5,0,0,1)^\top$, $(0,3,4,2,1,5,5,1,1,0,1,0)^\top$ *und* $(0,0,2,1,3,3,4,6,5,1,0,0)^\top$ *aufgespannt wird. Somit ist*

- $b_1(x) = 1$,
- $b_2(x) = 6x + x^2 + 2x^3 + 6x^5 + 3x^6 + 4x^7 + 5x^8 + x^{11}$,
- $b_3(x) = 3x + 4x^2 + 2x^3 + x^4 + 5x^5 + 5x^6 + x^7 + x^8 + x^{10}$ *und*
- $b_4(x) = 2x^2 + x^3 + 3x^4 + 3x^5 + 4x^6 + 6x^7 + 5x^8 + x^9$

eine Basis der Berlekamp–Unteralgebra $\mathcal{B}(f,7)$. *Das Basiselement* 1 *trägt nicht zur Faktorisierung bei. Teste* $\mathrm{ggT}\,(f(x), b_2(x) - \alpha)$ *für alle* $\alpha \in \mathbb{F}_7$*:*

- *Es ist* $\mathrm{ggT}\,(f(x), b_2(x) - 3) = x^3 + 6x^2 + 5x + 6$, *setze*

  $$g_1(x) = x^9 + 5x^8 + 2x^7 + x^6 + 3x^5 + 6x^3 + 2x^2 + 3x + 2$$

  und rechne weiter mit $f(x) = g_1(x) \cdot (x^3 + 6x^2 + 5x + 6)$.
- *Es ist* $\mathrm{ggT}\,(g_1(x), b_2(x) - 5) = x^6 + 6x^5 + 4x^4 + x^3 + 3x^2 + 2x + 2$, *setze*

  $$g_2(x) = x^6 + 6x^5 + 4x^4 + x^3 + 3x^2 + 2x + 2$$

  und rechne weiter mit $f(x) = g_2(x) \cdot (x^3 + 6x^2 + 4x + 1) \cdot (x^3 + 6x^2 + 5x + 6)$.

Teste $\mathrm{ggT}\,(g_2(x), b_3(x) - \alpha)$ *für alle* $\alpha \in \mathbb{F}_7$*: Es ist*

$$\mathrm{ggT}\,(g_2(x), b_3(x) - 2) = x^3 + 4.$$

Insgesamt ist also $f(x) = (x^3 + 4)(x^3 + 6x^2 + 4x + 4)(x^3 + 6x^2 + 4x + 1)(x^3 + 6x^2 + 5x + 6)$.

Beispiel 4.2
Bevor ich das Arbeiten mit Normalbasen total ablehne, möchte ich es in diesem Beispiel ein einziges Mal demonstrieren: Betrachte $f(x) = x^9 + \xi x^8 + (\xi+1)x^7 + \xi x^5 + (\xi+1)x^4 + (\xi+1)x + 1 \in \mathbb{F}_4[x]$, *wobei* $\xi^2 + \xi + 1 = 0$. *Ich berechne*

$$\begin{aligned}
x^{2\cdot 5} &\equiv x^7 + \xi x^6 + x^4 + (\xi+1)x^2 + \xi \mod f(x),\\
x^{2\cdot 6} &\equiv (\xi+1)x^7 + x^6 + \xi x^5 + \xi x^2 + (\xi+1)x + 1 \mod f(x),\\
x^{2\cdot 7} &\equiv x^5 + (\xi+1)x^3 + x^2 + \xi x + (\xi+1) \mod f(x) \text{ und}\\
x^{2\cdot 8} &\equiv x^7 + (\xi+1)x^5 + x^4 + \xi x^3 + (\xi+1)x^2 \mod f(x)
\end{aligned}$$

und erhalte die Matrix

$$B_{2,1}(f)^{\top} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & \xi & 1 & \xi+1 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & \xi+1 & \xi & 0\\
0 & 1 & 0 & 0 & 0 & \xi+1 & \xi & 1 & \xi+1\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \xi+1 & \xi\\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1\\
0 & 0 & 0 & 0 & 0 & 0 & \xi & 1 & \xi+1\\
0 & 0 & 0 & 1 & 0 & \xi & 1 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 1 & \xi+1 & 0 & 1\\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0
\end{bmatrix}.$$

Ich wähle die Normalbasis $N = \{\xi, \xi+1\}$. *Möchte ich die beiden Elemente* $a = a_1 + \xi a_2$ *und* $b = b_1 + \xi b_2$ *multiplizieren, so ist das Ergebnis*

$$a \cdot b = (a_1b_1 + a_2b_2) + \xi(a_1b_2 + a_2b_1 + a_2b_2) = \xi(a_1b_1 + a_1b_2 + a_2b_1) + (\xi+1)(a_1b_1 + a_2b_2),$$

ich suche nun eine a entsprechende Matrix a', die b in der Vektordarstellung $(b_1 + b_2, b_1)^\top$ bezüglich N auf den Vektor $(a_1b_1 + a_1b_2 + a_2b_1, a_1b_1 + a_2b_2)^\top$ abbildet:

$$a' = \begin{bmatrix} a_1 & a_2 \\ a_2 & a_1 + a_2 \end{bmatrix}$$

tut dies. Schreibe also $B_{2,1}(f)^\top$ um zu

$$B_2 = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}.$$

Weiterhin ist zu bemerken, daß $b^2 = \xi^2(b_1 + b_2) + (\xi + 1)^2 b_1 = \xi b_1 + (\xi + 1)(b_1 + b_2)$ ist. Das Quadrieren von b entspricht also der Multiplikation

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{pmatrix} b_1 + b_2 \\ b_1 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_1 + b_2 \end{pmatrix},$$

an dieser Stelle erkennt man, warum die gewählte Basis eine Normalbasis sein sollte: Das Potenzieren wird zu einem Shift des Vektors.

Stelle also zum Quadrieren diese Matrix auf:

$$E1 = \begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{bmatrix}.$$

Nun ist $B_{2,2}(f)^\top = B_2 \cdot E1$*, finde die Basis des Nullraums der Matrix*

$$B_{2,2}(f)^\top - I = \begin{bmatrix} 1&1&0&0&0&0&0&0&0&0&1&0&0&1&1&1&0&0\\ 1&1&0&0&0&0&0&0&0&0&1&1&1&0&0&1&0&0\\ 0&0&1&0&0&0&0&0&0&0&0&0&1&1&1&0&0&0\\ 0&0&0&1&0&0&0&0&0&0&0&0&0&1&1&1&0&0\\ 0&0&0&1&1&0&0&0&0&0&1&1&1&0&0&1&1&1\\ 0&0&1&0&0&1&0&0&0&0&0&1&1&1&1&0&0&1\\ 0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&1&1&0\\ 0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&1&1\\ 0&0&0&0&0&1&0&0&1&0&0&1&0&0&0&0&0&1\\ 0&0&0&0&1&0&0&0&0&1&1&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&0&0&0&1&0&1&0&0&1&1&1\\ 0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&0&0&1\\ 0&0&0&0&0&0&0&1&0&0&1&0&1&1&0&0&0&0\\ 0&0&0&0&0&0&1&0&0&0&1&1&1&1&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&0&0&1\\ 0&0&0&0&0&0&0&0&0&0&1&0&0&1&0&1&1&0\\ 0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&1 \end{bmatrix},$$

die aus den Vektoren

$$\begin{aligned} &(1,0,1,1,0,1,1,0,0,0,0,1,0,0,1,0,0,0)^\top,\\ &(1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)^\top \text{ und}\\ &(0,0,0,1,1,0,1,1,0,0,1,0,0,0,0,1,0,0)^\top \end{aligned}$$

besteht, und interpretiere sie als Basis $B = \{b_1(x), b_2(x), b_3(x)\}$ *des Lösungsraums der Gleichung* $b(x)^2 \equiv b(x) \mod f(x)$ *mit*

$$\begin{aligned} b_1(x) &= \xi x^7 + (\xi+1)x^5 + \xi x^3 + (\xi+1)x^2 + x + \xi,\\ b_2(x) &= 1,\\ b_3(x) &= (\xi+1)x^7 + \xi x^5 + x^3 + \xi x^2 + (\xi+1)x. \end{aligned}$$

Ich finde mit den Elementen der aufspaltenden Menge B *für* f *die komplette Faktorisierung von* f*:*

$$\begin{aligned} & & \operatorname{ggT}(f(x), b_1(x)) &= x^3 + \xi x + 1,\\ h(x) &= & \operatorname{ggT}(f(x), b_1(x)+1) &= x^6 + \xi x^5 + x^4 + \xi x^3 + \xi x^2 + x + 1,\\ & & \operatorname{ggT}(h(x), b_3(x)) &= x^3 + x + 1, \end{aligned}$$

es ist also $f(x) = (x^3 + \xi x + 1)(x^3 + x + 1)(x^3 + \xi x^2 + 1)$.

Beispiel 4.3
Zum Vergleich bestimme ich die Faktorisierung des Polynoms

$$f(x) = x^9 + \xi x^8 + (\xi+1)x^7 + \xi x^5 + (\xi+1)x^4 + (\xi+1)x + 1 \in \mathbb{F}_4[x]$$

mit der Berlekamp–Matrix $B_{4,1}(f)$*. Ich berechne*

$$\begin{aligned} x^{4\cdot 3} &\equiv (\xi+1)x^7 + x^6 + \xi x^5 + \xi x^2 + (\xi+1)x + 1 \mod f(x),\\ x^{4\cdot 4} &\equiv x^7 + (\xi+1)x^5 + x^4 + \xi x^3 + (\xi+1)x^2 \mod f(x),\\ x^{4\cdot 5} &\equiv x^8 + \xi x^7 + (\xi+1)x^6 + \xi x^4 + (\xi+1)x^3 + (\xi+1) \mod f(x),\\ x^{4\cdot 6} &\equiv x^3 \mod f(x),\\ x^{4\cdot 7} &\equiv x^7 \mod f(x) \text{ und}\\ x^{4\cdot 8} &\equiv x^8 + \xi x^7 + x^5 + (\xi+1)x^3 + \xi x \mod f(x) \end{aligned}$$

und erhalte die Matrix

$$B_{4,1}(f)^\top = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & \xi+1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \xi+1 & 0 & 0 & 0 & 0 & \xi \\ 0 & 0 & 0 & \xi & \xi+1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \xi & \xi+1 & 1 & 0 & \xi+1 \\ 0 & 1 & 0 & 0 & 1 & \xi & 0 & 0 & 0 \\ 0 & 0 & 0 & \xi & \xi+1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & \xi+1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \xi+1 & 1 & \xi & 0 & 1 & \xi \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

Eine Basis des Nullraums der Matrix $B_{4,1}(f) - I$ besteht aus den Vektoren

$$\begin{aligned} &[1,0,0,0,0,0,0,0,0]^\top, \\ &[0,\xi,1,\xi+1,0,1,0,0,0]^\top \text{ und} \\ &[0,0,0,0,0,0,0,1,0]^\top \end{aligned}$$

die ich als Basis $B = \{b_1(x), b_2(x), b_3(x)\}$ des Lösungsraums der Gleichung $b(x)^4 \equiv b(x) \mod f(x)$ mit

$$\begin{aligned} b_1(x) &= 1, \\ b_2(x) &= x^5 + (\xi+1)x^3 + x^2 + \xi x \text{ und} \\ b_3(x) &= x^7 \end{aligned}$$

interpretiere. Die aufspaltende Menge B ermöglicht mir die Faktorisierung von f:

$$\begin{aligned} \operatorname{ggT}(f(x), b_2(x)) &= x^3 + \xi x^2 + 1, \\ \operatorname{ggT}(f(x), b_2(x)+1) &= x^3 + \xi x + 1, \\ \operatorname{ggT}(f(x), b_2(x)+\xi) &= x^3 + x + 1. \end{aligned}$$

Insgesamt ist also $f(x) = (x^3 + \xi x^2 + 1)(x^3 + \xi x + 1)(x^3 + x + 1)$.

Komplexität

Ich berechne die Komplexität für die Arbeit mit $\mathcal{B}(f, q)$.

Bei der Generierung der Matrix $B_{q,1}(f)$ wird zuerst in $2\log_2(q)(3n^2 + 7n + 4)$ Operationen das Polynom x^q erzeugt. Nun wird für jede der n Spalten der Matrix $B_{q,1}(f)$ eine Multiplikation zweier Polynome durchgeführt, welche $3n^2 + 7n + 4$ Operationen kostet. Die Generierung der Matrix kostet somit höchstens $(2 \cdot \lfloor \log_2(q) \rfloor + n) \cdot (3n^2 + 7n + 4)$ Operationen.

Bestimmen wir nun den Nullraum der Matrix $B_{q,1}(f) - I$ mit Gaußelimination, so werden mit der ersten Zeile $n-1$ Zeilen addiert, mit der zweiten $n-2$ usw., dann wird dieselbe Prozedur mit Start bei der letzten Zeile wiederholt. Zwei Zeilen zu addieren kostet n Additionen, vor der Addition muß aber die eine Zeile noch mit einem $\alpha \in \mathbb{F}_q$ multipliziert werden. Die Umwandlung der Matrix in „Diagonalform“ kostet also $2\frac{n(n-1)}{2} \cdot 2n = 2n^3 - 2n^2$ Operationen, der Nullraum läßt sich ablesen.

Eine Basis von $\mathcal{B}(f, q)$ zu bestimmen kostet also $(2\log_2(q) + n) \cdot (3n^2 + 7n + 4) + 2n^3 - 2n^2$ Operationen, hat also die Komplexität $\mathcal{O}(n^3 + n^2 \log(q))$.

Für jeden der t Basisvektoren müssen $q - 1$ größte gemeinsame Teiler bestimmt werden, Kostenpunkt sind zusammen $t(q-1)(2n^2 + 3n)$ Operationen.

Zusammen ist die beim Algorithmus von Berlekamp durchzuführende Anzahl an Operationen darum

$$(2\log_2(q) + n) \cdot (3n^2 + 7n + 4) + (2n^3 - 2n^2) + t(q-1)(2n^2 + 3n),$$

die Komplexität ist $\mathcal{O}(tqn^2 + n^3)$.

Wie man die Basis des Lösungsraums der Gleichung (6) mit Hilfe der Automatentheorie bestimmt, wird von E. Lange in [25] beschrieben.

Im 3. Schritt sind pro Basiselement $r - 1$ Berechnungen des größten gemeinsamen Teilers notwendig. Ist $r \in \{2, 3, 4\}$, so bedeutet Schritt 3 also keinen großen Aufwand. Ist aber z. B. $r \geq 25$, so wendet man besser eine der Modifizierungen des Berlekamp–Algorithmus an. Einige dieser Modifizierungen beschreibe ich in den Abschnitten 5.1 bis 5.6. Zuvor möchte ich jedoch im Abschnitt 4.2 noch eine zweite Methode vorstellen, mit der man ein Erzeugendensystem der Berlekamp–Unteralgebra $\mathcal{B}(f, r)$ finden kann.

4.2 Das Verfahren von McEliece

1969 stellte R. J. McEliece in dem Artikel [32] seine Idee vor, wie man effektiv ein Erzeugendensystem der Berlekamp–Unteralgebra berechnen und mit dessen Hilfe ein Polynom f über dem endlichen Körper $\mathbb{F}_q$ komplett faktorisieren kann.

Theorie

Sei $f \in \mathbb{F}_q[x]$ mit $\mathrm{Grad}(f) = n$ gegeben als $f(x) = \sum_{j=0}^{n} a_j x^j = \prod_{i=1}^{t} f_i(x)$ mit für $1 \leq i \leq t$ paarweise verschiedenen irreduziblen Polynomen f_i mit $\mathrm{Grad}(f_i) = n_i$. Es sei $N \in \mathbb{N} \setminus \{0\}$ die kleinste natürliche Zahl, für die $x^{q^N} \equiv x \mod f(x)$ ist, dann ist $N = \mathrm{kgV}(n_1, n_2, \ldots, n_t)$ der Grad des Zerfällungskörpers $\mathbb{F}_{q^N}$ von f über $\mathbb{F}_q$. Im Satz 4.8 auf der nächsten Seite, der die Grundlage für die von McEliece entwickelte Theorie bildet, benötige ich zwei weitere Aussagen über Spuren, die man bei Lidl und Niederreiter in [31] nachschlagen kann:

- Es ist $\mathrm{Spur}_{\mathbb{F}_{q^k}/\mathbb{F}_q}(\alpha) = k \cdot \alpha$ für alle $\alpha \in \mathbb{F}_q$ und alle $k \in \mathbb{N}$.
- Ist $\mathbb{F}_q$ der endliche Körper der Ordnung q mit den Erweiterungskörpern $\mathbb{F}_{q^u}$ und $\mathbb{F}_{q^s}$ derart, daß $\mathbb{F}_q \subset \mathbb{F}_{q^u} \subset \mathbb{F}_{q^s}$, so ist
$$\mathrm{Spur}_{\mathbb{F}_{q^u}/\mathbb{F}_q}\left(\mathrm{Spur}_{\mathbb{F}_{q^s}/\mathbb{F}_{q^u}}(x)\right) = \mathrm{Spur}_{\mathbb{F}_{q^s}/\mathbb{F}_q}(x),$$
die Spur ist also transitiv.

Die $\mathrm{Spur}_{\mathbb{F}_{q^N}/\mathbb{F}_q}$ ist gegeben durch das Polynom

$$T_{q^N/q}(x) = \sum_{j=0}^{N-1} x^{q^j},$$

die $\mathrm{Spur}_{\mathbb{F}_{q^{n_i}}/\mathbb{F}_q}$ ist für alle $1 \leq i \leq t$ gegeben durch

$$T_i(x) = \sum_{j=0}^{n_i-1} x^{q^j}.$$

Für jedes Polynom $g \in \mathbb{F}_q[x]$ und alle $1 \leq i \leq t$ gilt dann

$$\begin{aligned}
T_{q^N/q}(g) = \sum_{i=0}^{N-1} g^{q^i} &= \sum_{j=0}^{n_i-1}\left(\sum_{k=0}^{N/n_i-1} g^{q^{k\cdot n_i}}\right)^{q^j} = \sum_{k=0}^{N/n_i-1}\sum_{j=0}^{n_i-1} g^{q^{j+n_i\cdot k}} = \sum_{k=0}^{N/n_i-1}\left(\sum_{j=0}^{n_i-1} g^{q^j}\right)^{q^{n_i\cdot k}} \\
&\equiv \sum_{k=0}^{N/n_i-1} (T_i(g))^{q^{n_i\cdot k}} \mod f_i \quad \equiv \sum_{k=0}^{N/n_i-1} (T_i(g)) \mod f_i \\
&\equiv \frac{N}{n_i} T_i(g) \mod f_i(x),
\end{aligned}$$

da $T_i(g) \in \mathbb{F}_q$ ist für alle $1 \leq i \leq t$. Es gilt der folgende Satz von McEliece:

Satz 4.8
Ist f reduzibel, so befindet sich unter den Polynomen $T_{q^N/q}(x^i)$ mit $1 \leq i < n$ mindestens eines, das durch Bilden von

$$\mathrm{ggT}(f(x), T_{q^N/q}(x^i) - \alpha) \text{ für alle } \alpha \in \mathbb{F}_q$$

eine nicht triviale partielle Faktorisierung von f liefert.

Beweis: Ich betrachte ein Polynom $g \in \mathbb{F}_q[x]$, das die Komponentendarstellung $(g_1(x), g_2(x), \ldots, g_t(x))$ hat. g wird durch die Spur T abgebildet auf $(\frac{N}{n_1}T_1(g_1), \frac{N}{n_2}T_2(g_2), \ldots, \frac{N}{n_t}T_t(g_t))$. Ist $N/n_i \equiv 0 \mod p$ für ein $1 \leq i \leq t$, so wird die i–te Komponente durch die Spur T immer auf 0 geworfen, alle anderen Komponenten durchlaufen $\mathbb{F}_q$. g ist nur modulo f interessant und wird somit von der Basis $\{1, x, x^2, \ldots, x^{n-1}\}$ erzeugt. Da die Spur linear ist, ist also die Dimension des Erzeugnisses der Elemente $T_{q^N/q}(1), T_{q^N/q}(x), T_{q^N/q}(x^2), \ldots, T_{q^N/q}(x^{n-1})$ genau die Anzahl der Faktoren f_i von f, für die N/n_i nicht von der Charakteristik p geteilt wird. Es gibt mindestens ein solches f_i: Wird N von p^k geteilt, nicht aber von p^{k+1}, so gibt es mindestens ein i mit $1 \leq i \leq t$ und $p^k \mid n_i$, da $N = \mathrm{kgV}(n_1, \ldots, n_t)$.

$x^0 = 1$ beziehungsweise $T_{q^N/q}(1)$ trägt als konstantes Polynom nicht zur Faktorisierung von f bei, darum kann es aus der Menge der Elemente, die nicht triviale partielle Faktorisierungen liefern, herausgenommen werden. □

Die Faktorisierung eines Polynoms f mit $\mathrm{Grad}(f) = n$ ist mit Hilfe von Satz 4.8 wie folgt zu bewerkstelligen: Man bestimmt zuerst N durch die Eigenschaft $x^{q^N} \equiv x \mod f(x)$ und berechnet dann nacheinander die $T_{q^N/q}(x^i)$ für $1 \leq i \leq n$. Für jedes dieser $T_{q^N/q}(x^i)$ berechnet man für alle $\alpha \in \mathbb{F}_q$ den dazugehörigen Faktor $g(x) = \mathrm{ggT}(f(x), T_{q^N/q}(x^i) - \alpha)$. Sobald dieser Faktor nicht trivial ist, wendet man das Verfahren rekursiv auf f/g und g an. Diese rekursive Vorgehensweise ist notwendig, da die $T_{q^N/q}(x^i)$ zwar garantiert Faktoren von f liefern, aber nicht unbedingt eine aufspaltende Menge für f bilden, da Faktoren f_i und f_j mit

- $1 \leq i, j \leq t$, $i \neq j$, $\mathrm{Grad}(f_i) = n_i$, $\mathrm{Grad}(f_j) = n_j$,
- $p \mid \frac{N}{n_i}$ und $p \mid \frac{N}{n_j}$

nicht getrennt werden. Die Abbruchbedingung ist erfüllt, wenn alle $T_{q^N/q}(x^i) - \alpha$ triviale Faktoren von f liefern, denn dann ist f irreduzibel.

Praxis

Die Bestimmung des N ist zwar nicht sehr mühsam, hat man jedoch eine partielle Faktorisierung von f wie aus Abschnitt 2.2 auf Seite 13 in Teilpolynome mit d–gradigen Faktoren vorliegen, so ist $N = d$ von vornherein bekannt. Ein weiterer Vorteil der Zerlegung in Teilpolynome gleichen Grades ist, daß dadurch N nicht zu groß wird, und schließlich gilt dann der folgende Satz:

Satz 4.9
Sei $f \in \mathbb{F}_q[x]$ ein quadratfreies Polynom, das in t Faktoren vom Grad d zerfällt, sei außerdem

$$T_{q^d/q}(x) = \sum_{j=0}^{d-1} x^{q^j}$$

das $\mathrm{Spur}_{\mathbb{F}_{q^d}/\mathbb{F}_q}$–Polynom. Bei den Polynomen $T_{q^d/q}(x^i)$ für $0 \leq i < n$ handelt es sich um ein Erzeugendensystem der Berlekamp–Unteralgebra $\mathcal{B}(f, q)$.

Beweis: Haben alle Faktoren f_i von f den gleichen Grad d, so ist $N/d = 1$ und also ist keine Komponente der Elemente $T_{q^d/q}(x^i)$ permanent 0 für alle $1 \leq i \leq t$. Somit ist die Dimension des Erzeugnisses der Elemente $T_{q^d/q}(1), T_{q^d/q}(x), T_{q^d/q}(x^2), \ldots, T_{q^d/q}(x^{n-1})$ genau t. Da das Erzeugnis Teilmenge der Berlekamp–Unteralgebra $\mathcal{B}(f, q)$ ist, welche die Dimension t hat, folgt die Behauptung. □

Im Fall $N = d$ ist also das rekursive Vorgehen überflüssig, da die Polynome $T_{q^d/q}(x^i)$ mit $0 \leq i < n$ ein Erzeugendensystem der Berlekamp–Unteralgebra $\mathcal{B}(f, q)$ sind. Ist man nicht an einem Erzeugendensystem, sondern nur an einer aufspaltenden Menge für f interessiert, so kann man die folgende Verbesserung vornehmen:

Satz 4.10
Sei $f \in \mathbb{F}_q[x]$ ein quadratfreies Polynom, das in t Faktoren vom Grad d zerfällt. Bei den Polynomen $T_{q^d/q}(x^i)$ für $0 < i < 2d$ handelt es sich um eine aufspaltende Menge für f.

Beweis: Die Polynome $T_{q^d/q}(x^i)$ sind für $0 \leq i < n$ ein Erzeugendensystem der Berlekamp–Unteralgebra, also eine aufspaltende Menge. Ich nehme nun an, daß die partielle Faktorisierung, die durch das Bilden von

$$\mathrm{ggT}(f(x), T_{q^d/q}(x^i) - \alpha) \text{ für alle } \alpha \in \mathbb{F}_q$$

erreicht wird, für alle $0 \leq i < 2d$ trivial ist. Dann gäbe es Elemente $c_0, c_1, \ldots, c_{2d-1} \in \mathbb{F}_q$ derart, daß für $0 \leq i < 2d$ gälte $T_{q^d/q}(x^i) \equiv c_i \mod f(x)$.

Für jedes $g \in \mathbb{F}_q[x]$ mit $g(x) = \sum_{j=0}^{2d-1} b_j x^j$, also mit $\mathrm{Grad}(g) < 2d$, gälte

$$T_{q^d/q}(g(x)) = T_{q^d/q}\left(\sum_{j=0}^{2d-1} b_j x^j\right) = \sum_{j=0}^{2d-1} b_j T_{q^d/q}(x^j) \equiv \sum_{j=0}^{2d-1} b_j c_j \mod f(x).$$

Ich setze $c(g) = \sum_{j=0}^{2d-1} b_j c_j \in \mathbb{F}_q$ und erhalte aus $T_{q^d/q}(g(x)) \equiv c(g) \mod f(x)$ die Kongruenz

$$T_{q^d/q}(g(x)) \equiv c(g) \mod f_i(x) \text{ für } 0 \leq i \leq t.$$

Betrachte nun zwei Faktoren f_1 und f_2 von f, dann ist $\mathrm{Grad}(f_1) = \mathrm{Grad}(f_2) = d$. Sei θ eine Nullstelle von f_1 in $\mathbb{F}_{q^d}$. Dann existiert ein $h \in \mathbb{F}_q[x]$ mit $\mathrm{Grad}(h) < n$, so daß

$$T_{q^d/q}(h(\theta)) = 1.$$

Ich wende den Chinesischen Restsatz an, um mir ein Polynom $g \in \mathbb{F}_q[x]$ mit $\mathrm{Grad}(g) < n$ derart vorzustellen, daß

$$g(x) \equiv h(x) \mod f_1(x) \quad \text{und} \quad g(x) \equiv 0 \mod f_2(x).$$

g ist nur modulo $f_1 \cdot f_2$ interessant und dort eindeutig, darum kann man $\mathrm{Grad}(g) < 2d$ wählen. Dann ist

$$T_{q^d/q}(g(x)) \equiv 1 \mod f_1(x) \quad \text{und} \quad T_{q^d/q}(g(x)) \equiv 0 \mod f_2(x),$$

was ein Widerspruch zu der Forderung $T_{q^d/q}(g(x)) \equiv c(g) \mod f_i(x)$ für alle $1 \leq i \leq t$ wäre.

Nun ist also mit den Polynomen aus $\{T_{q^d/q}(x^i) \mid 0 \leq i < 2d\}$ eine nicht triviale partielle Faktorisierung von f möglich. $T_{q^d/q}$ und g sind ausschließlich von q und d abhängig, das heißt, daß für zwei beliebige irreduzible Faktoren $f_1, f_2 \in \mathbb{F}_q[x]$ in der Menge $\{T_{q^d/q}(x^i) \mid 0 \leq i < 2d\}$ ein Polynom ist, das die beiden Faktoren voneinander trennt, weshalb es sich um eine aufspaltende Menge für f handelt. □

Wir haben gerade einen Weg entdeckt, eine aufspaltende Menge zu finden, bei der der größte gemeinsame Teiler mit Elementen aus $\mathbb{F}_q$ gebildet werden muß. Im vorigen Abschnitt 4.1 waren wir jedoch in der Lage, auch für die Berlekamp–Unteralgebren $\mathcal{B}(f, r)$ mit $q = r^v$ Basen zu bestimmen. Diese Verallgemeinerung liegt bei dem Verfahren von McEliece ganz nahe, indem wir T umdefinieren zur $\mathrm{Spur}_{\mathbb{F}_{q^d}/\mathbb{F}_r}$:

Satz 4.11
Sei $f \in \mathbb{F}_q[x]$ ein quadratfreies Polynom, das in t Faktoren vom Grad d zerfällt, sei außerdem

$$T_{q^d/r}(x) = \sum_{j=0}^{dv-1} x^{r^j}$$

das $\mathrm{Spur}_{\mathbb{F}_{q^d}/\mathbb{F}_r}$–Polynom. Die Menge $\{T_{q^d/r}(x^i) \mid 0 \leq i < n\}$ ist ein Erzeugendensystem der Berlekamp–Unteralgebra $\mathcal{B}(f, r)$.

Beweis: Es gibt für alle $1 \leq i \leq t$ und alle $\alpha \in \mathbb{F}_q$ ein Polynom $g \in \mathbb{F}_q[x]$ mit der Komponentendarstellung $(g_1(x), g_2(x), \ldots, g_t(x))$, wobei $g_i = \alpha$ und $g_j = 0$ ist für alle $1 \leq j \leq t$ mit $i \neq j$. Wähle nun α so, daß $T_{q^d/r}(\alpha) = 1$ ist, dann gilt

$$\begin{aligned} T_{q^d/r}(g) &\equiv 1 \mod f_i, \\ T_{q^d/r}(g) &\equiv 0 \mod f_j \text{ für alle } 1 \leq j \leq t \text{ mit } i \neq j. \end{aligned}$$

Somit ist wie behauptet die Menge $1, T_{q^d/r}(x), \ldots, T_{q^d/r}(x^{2d-1})\}$ ein Erzeugendensystem der Berlekamp–Unteralgebra $\mathcal{B}(f, r)$. □

Camion stellte in [6] fest, daß man die Menge $\{T_{q^d/r}(x^i)\}$ im Fall $r = 2$ noch ausdünnen kann:

Lemma 4.12
Ist $r = 2$, also $u = 1$, so gilt für alle $g \in \mathbb{F}_q[x]$

$$T_{q^d/r}(g(x)^2) = T_{q^d/r}(g(x)).$$

Dabei ist die Spur

$$T_{q^d/r}(x) = \sum_{j=0}^{dv-1} x^{r^j} = T_{q^d/2}(x) = \sum_{j=0}^{ds-1} x^{2^j}.$$

Beweis: Für alle $g \in \mathbb{F}_q[x]$ gilt

$$\begin{aligned} T_{q^d/2}(g(x)^2) = \left(\sum_{i=0}^{sd-1} g(x)^{2^i}\right)^2 &= \sum_{i=0}^{sd-1} g(x)^{2^{i+1}} \\ = g(x)^{q^d} + \sum_{i=1}^{sd-1} g(x)^{2^i} &\equiv g(x) + \sum_{i=1}^{sd-1} g(x)^{2^i} \mod f(x) \\ = \sum_{i=0}^{sd-1} g(x)^{2^i} &= T_{q^d/2}(g(x)). \end{aligned}$$

□

Ist $r = 2$, so arbeitet man also mit dem Erzeugendensystem $\{1\} \cup \{T_{q^d/2}(x^{2i+1}) \mid 0 \leq i \leq n/2\}$ beziehungsweise mit der aufspaltenden Menge $\{T_{q^d/2}(x^{2i+1}) \mid 0 \leq i < d\}$.

Algorithmus 13 MCELIECE $(f, \mathbb{F}_q)$
Eingabe: Das normierte, quadratfreie Polynom $f \in \mathbb{F}_q[x]$, welches in t Faktoren vom Grad d zerfällt, und der Körper $\mathbb{F}_q$ mit $q = p^s$.
Ausgabe: Die Faktorisierung von f.
Programm: McEliece.mws: mc

1. Setze $n \leftarrow \mathrm{Grad}(f)$. Ist $n = d$, so liefere f zurück.
2. Setze $i \leftarrow 0$ und $T(x) \leftarrow \sum_{j=0}^{s\,d-1} x^{p^j}$. Ist $p = 2$, so setze $\xi \leftarrow x^2$, sonst setze $\xi \leftarrow x$.
3. Ist $\mathrm{Grad}\,(T(x \cdot \xi^i)) > 0$, so bilde für alle bisher gefundenen Faktoren $g(x)$ von $f(x)$ die partielle Faktorisierung $g(x) = \prod_{\alpha \in \mathbb{F}_p} \mathrm{ggT}\,\left(g(x), T(x \cdot \xi^i) - \alpha\right)$.
4. Ist die Anzahl der bisher gefundenen Faktoren kleiner als t, so setze $i \leftarrow i + 1$ und gehe zu Schritt 3.
5. Gib die Faktorisierung von $f(x)$ aus.

Beispiel 4.4
Betrachte

$$f(x) = x^{12} + 4x^{11} + 2x^{10} + 2x^9 + 6x^6 + 3x^4 + 3x^3 + 4x^2 + 5 \in \mathbb{F}_7[x].$$

Ich möchte eine aufspaltende Menge mit Hilfe der Spur $T(x) = x^{49} + x^7 + x$ von $\mathbb{F}_{7^3}$ nach $\mathbb{F}_7$ bestimmen. Die Charakteristik ist ungerade und $d = 3$, also muß ich die Spur der Elemente x^i mit $i \in \{0, \ldots, 5\}$ berechnen:

- *Es ist $g_1(x) = T(x)$, also*

  $$g_1(x) \equiv 4x^{10} + 3x^9 + 5x^8 + x^7 + 4x^6 + x^5 + 6x^4 + 4x^3 + x^2 + 5x + 4 \mod f(x).$$

  Bilde

  $$h_1(x) = \mathrm{ggT}(f(x), g_1(x) - 1) = x^9 + 4x^8 + 2x^7 + 5x^6 + 5x^5 + 6x^4 + x^2 + 3,$$

  rechne weiter mit $f(x) = h_1(x) \cdot (x^3 + 4)$. Durch die restlichen $\mathrm{ggT}(f(x), g_1(x) - \alpha)$ werden nur noch triviale Teiler von h_1 gefunden.

- *Es ist $g_2(x) = T(x^2) = x^{98} + x^{14} + x^2$, also ist*
  $g_2(x) \equiv 2x^{11} + 2x^{10} + x^9 + 3x^8 + 2x^7 + 6x^6 + 4x^5 + 5x^4 + 2x^3 + 5x^2 + 4x + 4 \mod f(x).$
  Bilde

  $$h_2(x) = \mathrm{ggT}(h1(x), g_2(x) - 5) = x^3 + 6x^2 + 5x + 6,$$

  rechne weiter mit $f(x) = h_{2'}(x) \cdot (x^3 + 6x^2 + 5x + 6)(x^3 + 4)$, wobei

  $$h_{2'}(x) = h_1(x)/h_2(x) = x^6 + 5x^5 + 2x^4 + 4x^3 + 4x^2 + 6x + 4$$

  ist. Durch die anderen $\mathrm{ggT}(h_1(x), g_2(x) - \alpha)$ werden nur triviale Teiler von $h_{2'}$ gefunden.

- *Es ist $g_3(x) = T(x^3) = x^{147} + x^{21} + x^3$, also*
  $g_3(x) \equiv x^{11} + 5x^{10} + 2x^9 + 6x^8 + x^6 + 2x^5 + 4x^4 + 4x^2 + 2 \mod f(x).$
  Bilde

  $$\mathrm{ggT}(h_{2'}(x), g_3(x) - 5) = x^3 + 6x^2 + 4x + 4,$$

  die gesuchte Zerlegung von f in irreduzible Faktoren ist somit

  $$f(x) = (x^3 + 6x^2 + 4x + 4)(x^3 + 6x^2 + 4x + 1)(x^3 + 6x^2 + 5x + 6)(x^3 + 4).$$

Die Menge $\{1, g_1, g_2, g_3\}$ ist eine aufspaltende Menge für f. Sie ist nicht notwendig ein Erzeugendensystem der Berlekamp–Unteralgebra $\mathcal{B}(f,7)$, hier handelt es sich aber zufälligerweise tatsächlich um eine Basis, da die Menge vier Elemente enthält, die voneinander linear unabhängig sind.

Beispiel 4.5
Betrachte

$$f(x) = x^9 + \xi x^8 + (\xi+1)x^7 + \xi x^5 + (\xi+1)x^4 + (\xi+1)x + 1 \in \mathbb{F}_4[x],$$

wobei $\xi^2 + \xi + 1 = 0$ ist. Ich möchte die Berlekamp–Unteralgebra $\mathcal{B}(f,4)$ mit Hilfe der Spur

$$T(x) = x^{16} + x^4 + x$$

von $\mathbb{F}_{4^3}$ nach $\mathbb{F}_4$ bestimmen. Da die Charakteristik $p = 2$ ist, muß ich die Spur nur für die Elemente x^{2i+1} mit $i \in \{0,1,2\}$ berechnen:

- *Es ist $g_1(x) = T(x) \equiv x^7 + (\xi+1)x^5 + \xi x^3 + (\xi+1)x^2 + x \mod f(x)$. Bilde*

$$\begin{aligned} h_1(x) = \operatorname{ggT}(f(x), g_1(x)) &= x^6 + (\xi+1)x^4 + \xi x^2 + (\xi+1)x + 1, \\ \operatorname{ggT}(f(x), g_1(x)+1) &= 1 \text{ und} \\ \operatorname{ggT}(f(x), g_1(x)+\xi) &= x^3 + \xi x^2 + 1, \end{aligned}$$

  dann ist $h_1(x)(x^3 + \xi x^2 + 1)$ eine partielle Faktorisierung von f.

- *Es ist $g_3(x) = T(x^3) = x^{48} + x^{12} + x^3 \equiv (\xi+1)x^7 + \xi x^5 + x^3 + \xi x^2 + (\xi+1)x + 1 \mod f(x)$. Bilde*

$$\begin{aligned} \operatorname{ggT}(h_1(x), g_3(x)) &= 1, \\ \operatorname{ggT}(h_1(x), g_3(x)+1) &= h_1(x) \text{ und} \\ \operatorname{ggT}(h_1(x), g_3(x)+\xi) &= 1, \end{aligned}$$

  es ergibt sich keine neue partielle Faktorisierung von f.

- *Es ist $g_5(x) = T(x^5) = x^{80} + x^{20} + x^5 \equiv \xi x^7 + (\xi+1) \mod f(x)$. Bilde*

$$\begin{aligned} \operatorname{ggT}(h_1(x), g_5(x)) &= 1, \\ \operatorname{ggT}(h_1(x), g_5(x)+1) &= x^3 + x + 1 \text{ und} \\ \operatorname{ggT}(h_1(x), g_5(x)+\xi) &= x^3 + \xi x + 1. \end{aligned}$$

f zerfällt somit in $f(x) = (x^3+\xi x^2+1)(x^3+x+1)(x^3+\xi x+1)$. Ein Erzeugendensystem der Berlekamp–Unteralgebra $\mathcal{B}(f,4)$ ist die Menge $\{1, g_1, g_3, g_5\}$, welches zu der Basis $\{1, g_1, g_5\}$ ausgedünnt werden kann.

Beispiel 4.6
Mit dem Verfahren von McEliece ist es nicht wesentlich schwieriger, ein Erzeugendensystem der Berlekamp–Unteralgebra $\mathcal{B}(f,2)$ zu finden als eines von $\mathcal{B}(f,4)$, darum möchte ich es hier vorführen.

Betrachte wiederum

$$f(x) = x^9 + \xi x^8 + (\xi+1)x^7 + \xi x^5 + (\xi+1)x^4 + (\xi+1)x + 1 \in \mathbb{F}_4[x].$$

Ich arbeite mit der Spur $T(x) = x^{32} + x^{16} + x^8 + x^4 + x^2 + x$ von $\mathbb{F}_{4^3}$ nach $\mathbb{F}_2$. Da die Charakteristik $p = 2$ ist, muß ich die Spur nur für die Elemente x^{2i+1} mit $i \in \{0,1,2\}$ bestimmen:

- *Es ist $g_1(x) = T(x) \equiv (\xi+1)x^7 + \xi x^5 + x^3 + \xi x^2 + (\xi+1)x \mod f(x)$. Bilde*

$$h_1(x) = \operatorname{ggT}(f(x), g_1(x)) = x^6 + x^4 + \xi x^4 + \xi x^2 + (\xi+1)x + 1,$$

  dann ist $h_1(x) \cdot (x^3 + \xi x^2 + 1)$ eine partielle Faktorisierung von f.

- *Es ist* $g_3(x) = T(x^3) = x^{96} + x^{48} + x^{24} + x^{12} + x^6 + x^3 \equiv 0 \mod f(x)$.
- *Es ist* $g_5(x) = T(x^5) \equiv \xi x^7 + (\xi+1)x^5 + \xi x^3 + (\xi+1)x^2 + x + (\xi+1) \mod f(x)$. *Bilde*

$$\mathrm{ggT}(h_1(x), g_5(x)) = x^3 + x + 1.$$

f zerfällt in $f(x) = (x^3 + \xi x + 1)(x^3 + x + 1)(x^3 + \xi x^2 + 1)$. Ein Erzeugendensystem der Berlekamp–Unteralgebra $\mathcal{B}(f, 4)$ ist die Menge $\{1, g_1, g_5\}$. Da das Erzeugendensystem nur drei Elemente enthält, handelt es sich sogar schon um eine Basis.

Komplexität

Ist p gerade, so müssen nur halb so viele Polynome $T(x^i)$ bestimmt werden wie im Fall p ungerade. Die Komplexität ist somit aber dieselbe.

Ist p ungerade, so werden die Polynome $T(x^i)$ für $0 < i < 2d$ bestimmt, also $2d - 2$ Stück. Zur Berechnung der Polynome $T(x^i)$ muß zu Beginn jeweils das Polynom x^{i-1} modulo f mit x multipliziert werden, das passiert in $2(n+1)$ Operationen: Der Shift von x^{i-1} um 1 zählt nicht mit, von ihm wird ein $\mathbb{F}_q$–Vielfaches von f abgezogen.

Das Potenzieren von x^i mit r kostet $(2\log_2(r)) \cdot (3n^2 + 7n + 4)$ Operationen.

Die Berechnung der Spur bedarf $dv - 1$ solcher Berechnungen und $dv - 2$ Additionen, benötigt also insgesamt $2(n+1) + (dv-1)\log_2(r)(3n^2 + 7n + 4) + (dv-2)(n+1)$ elementare Operationen.

Es werden $2d - 2$ Spuren berechnet, dann werden für jede dieser Spuren $r - 1$ größte gemeinsame Teiler berechnet. Der Algorithmus von McEliece braucht also

$(2d-2)(2(n+1) + (dv-1)\log_2(r)(3n^2+7n+4) + (dv-2)(n+1) + (r-1)(2n^2+3n))$ Operationen,

hat also eine Komplexität von $\mathcal{O}(vd^2n^2\log(r) + rn^2) = \mathcal{O}(d^2n^2\log(q) + rn^2)$.

5 Weitere Faktorisierungsmethoden, die auf der Berlekamp–Unteralgebra aufbauen

Betrachtet man die Berlekamp–Unteralgebra $\mathcal{B}(f, r)$, so muß man $\mathrm{ggT}(f(x), b(x) - \alpha)$ für alle r Elemente $\alpha \in \mathbb{F}_r$ bestimmen. Man möchte die Anzahl der Berechnungen von größten gemeinsamen Teilern aber möglichst klein halten, darum empfiehlt es sich, $r = p$ zu wählen. Ist bereits p relativ groß, so versucht man mit anderen Methoden, die Berechnung der r größten gemeinsamen Teiler für jedes Element des Erzeugendensystems zu umgehen.

In den hier vorgestellten Faktorisierungsmethoden wird oftmals ein Erzeugendensystem B der Berlekamp–Unteralgebra gebraucht. Man kann sich dann aussuchen, ob man es mit dem Algorithmus von Berlekamp aus Abschnitt 4.1 auf Seite 31 oder mit dem von McEliece aus Abschnitt 4.2 auf Seite 38 bestimmen möchte.

Es gibt deterministische und probabilistische Faktorisierungsmethoden, die sich die Eigenschaften der Berlekamp–Unteralgebra zunutze machen. Der erste Ansatz, der sich auch gar nicht auf die Berlekamp–Unteralgebra bezieht, ist natürlich, den Körper, über dem sich das Polynom f befindet, zu erweitern, so daß f in Linearfaktoren zerfällt. In diesem Erweiterungskörper kann man nun mit den Methoden aus Abschnitt 3 die Nullstellen von f bestimmen. Unter den Nullstellen sucht man diejenigen, die konjugiert zueinander sind, multipliziert die entsprechenden Linearfaktoren und erhält so die Faktorisierung von f über dem ursprünglichen Körper. Das wäre jedoch mit Kanonen auf Spatzen geschossen, da der Aufwand für eine konkret durchgeführte Körpererweiterung immens ist. Aus diesem Grund wurden die folgenden Algorithmen entwickelt.

5.1 Der Algorithmus von Camion

1983 benutzte Camion in [6] die Tatsache, daß $1 = \sum_{i=1}^{t} e_i(x)$ ist, um ein probabilistisches Verfahren zu entwerfen. 1 entspricht in der Komponentenschreibweise $(1, \dots, 1)$. Das Ziel ist, die e_i zu bestimmen, da $\mathrm{ggT}(f(x), 1 - e_i(x)) = f_i(x)$ ist für alle $1 \le i \le t$ und also nur t statt $t(r-1)$ größte gemeinsame Teiler berechnet werden müssen.

Theorie für ungerades q

Die e_i sind gerade die primitiven idempotenten Elemente aus $\mathbb{F}_q[x]/(f\mathbb{F}_q[x])$, also diejenigen idempotenten Elemente, die sich nicht mehr in eine Summe aus idempotenten Elementen ungleich 0 zerlegen lassen. Die primitiven idempotenten Elemente sind in genau einer Komponente 1, in allen anderen 0, außerdem sind sie orthogonal zueinander.

Die Idee ist die: Ich habe ein idempotentes $k(x)$ gegeben, k hat also in der Komponentendarstellung nur Nullen und Einsen. Ich zerlege es in die Summe zweier zueinander orthogonaler idempotenter Elemente $k = k' + (k - k')$: In k' werden einige Komponenten, die in k 1 sind, durch 0 ersetzt. Dann sind in $k - k'$ genau die Komponenten 1, die in k 1 und in k' 0 sind, k' und $k - k'$ sind orthogonal zueinander.

Ich wende dieses Verfahren rekursiv wieder auf k' und $k - k'$ an und breche ab, wenn sich das gegebene k nicht mehr zerlegen läßt. Wenn ich mit $k(x) = 1$ starte (1 ist in allen Komponenten 1), so erhalte ich durch dieses Verfahren gerade die Darstellung $1 = \sum_{i=1}^{t} e_i(x)$.

Praxis

Ich bestimme ein Erzeugendensystem $B = \{b_1(x), \dots, b_\tau(x)\}$ der Berlekamp–Unteralgebra $\mathcal{B}(f, r)$ und habe ein idempotentes $k \in \mathcal{B}(f, r)$ gegeben — Startwert ist $k(x) = 1$. Dann wähle ich eine $\mathbb{F}_r$–Linearkombination $w(x)$ der Menge $\{kb_1, \dots, kb_t\}$ derart, daß $w \neq \alpha k$ für alle $\alpha \in \mathbb{F}_r$.

Da w in der Berlekamp–Unteralgebra $\mathcal{B}(f, r)$ ist, ist $w(x) \equiv \sum_{i=1}^{t} w_i e_i(x) \mod f(x)$ mit $w_i \in \mathbb{F}_r$ für alle $1 \le i \le t$. Dabei ist für $k_i = 0$ auch $w_i = 0$.

Ich setze

$$m(x) \equiv w(x)^{(r-1)/2} \mod f(x),$$

dann ist

$$m(x) = \sum_{i=1}^{t} m_i e_i(x) = \sum_{i=1}^{t} w_i^{(r-1)/2} e_i(x) \text{ mit } m_i \in \mathbb{F}_r \text{ für } 1 \le i \le t.$$

Das Potenzieren der w_i mit $\frac{r-1}{2}$ entspricht dem Abbilden in die multiplikative Untergruppe $\{1, -1\}$ der Ordnung zwei von $\mathbb{F}_r$. Darum nehmen die $m_i = w_i^{(r-1)/2}$ die folgenden Werte an:

- $m_i = 0$, wenn $w_i = 0$,
- $m_i = 1$, wenn w_i Quadrat in $\mathbb{F}_r$ und
- $m_i = -1$, wenn w_i kein Quadrat in $\mathbb{F}_r$ ist.

m ist nicht 0, da sonst bereits $w \equiv 0 \mod f$ gewesen sein müßte. Ist $m(x) \not\equiv \pm k(x) \mod f(x)$, so gibt es ein Paar (i, j) mit den Eigenschaften

1. $1 \le i < j \le t$,
2. $k_i \ne 0$ und $k_j \ne 0$ und
3. $m_i \ne m_j$,

da ja gerade $\sum_{i=1}^{t} k_i \cdot e_i(x) \equiv k(x) \mod f(x)$ mit $k_i \in \{0, 1\}$ für alle $1 \le i \le t$ ist.

Um die Komponenten -1 von m zu annullieren, setze ich$k'(x) \equiv m(x)\,(m(x) + 1)\,/2 \mod f(x)$ und erhalte $k'(x) \equiv \sum_{i=1}^{t} k'_i e_i(x) \mod f(x)$ mit $k'_i \in \{0, 1\}$ für alle $1 \le i \le t$, denn

- $m_i = 0 \Rightarrow k'_i = 0 \cdot 1/2 = 0,$
- $m_i = 1 \Rightarrow k'_i = 1 \cdot 2/2 = 1,$
- $m_i = -1 \Rightarrow k'_i = -1 \cdot 0/2 = 0.$

Es kann sein, daß $m_i \in \{0, -1\}$ für alle $1 \le i \le t$, dann ist $k'(x) \equiv 0 \mod f(x)$ und ich muß eine andere Linearkombination w der erzeugenden Elemente aus B wählen. Die Wahrscheinlichkeit dafür, daß gilt $k'(x) \equiv 0 \mod f(x)$, ist jedoch nur $\left(\frac{r+1}{2}\right)^{\sum_{i=1}^{t} k_i}$, da die Wahrscheinlichkeit, nicht ein Quadrat $\alpha \in \mathbb{F}_r$ zu erwischen, genau $\frac{r-1}{2}$ ist. Der Fall $k' \equiv k \mod f$ kann nicht eintreten, da dann bereits $m \equiv k \mod f$ gewesen sein müßte.

Ist aber $k'(x) \not\equiv 0 \mod f(x)$, so ist $k'_i = 1$ für mindestens ein $1 \le i \le t$, nicht aber für alle. k' und $k - k'$ sind dann zueinander orthogonale idempotente Elemente. Ihre i–te Komponente ist jeweils 0, falls die i–te Komponente von k schon 0 war. Darum ist $k = k' + (k - k')$ eine Zerlegung von k in die Summe der orthogonalen idempotenten Elemente k' und $k - k'$.

Das soeben beschriebene Verfahren wende ich auf k' und $k - k'$ erneut an, falls die Dimension des Erzeugnisses des mit k' bzw. $k - k'$ multiplizierten Erzeugendensystems nicht schon 1 ist. Ist die Dimension 1, so ist das Ziel, ein primitives idempotentes Element zu finden, erreicht worden und die Rekursion muß nur noch für das andere idempotente Element fortgeführt werden.

Die Wahrscheinlichkeit P für eine nicht triviale Zerlegung läßt sich hier ganz analog zum Satz 3.3 auf Seite 16 abschätzen mit $P \ge 1/2$, das Verfahren ist in der Komplexitätsklasse RP.

Algorithmus 14 CAMION $(k, f, \mathbb{F}_q, B, r)$
Eingabe: Das idempotente Element k der Berlekamp–Unteralgebra $\mathcal{B}(f, r)$ (Startwert $k(x) = 1$), das normierte, quadratfreie Polynom $f \in \mathbb{F}_q[x]$, der Körper $\mathbb{F}_q$, ein Teil $B = \{b_1, \ldots, b_t\}$ einer Basis der Berlekamp–Unteralgebra (Startwert ist eine Basis der Berlekamp–Unteralgebra), r.
Ausgabe: Die Menge der primitiven idempotenten Elemente $e_i(x)$ für $1 \leq i \leq t$.
Programm: Camion.mws: camion

1. Wähle eine zufällige Linearkombination $w(x) \leftarrow \sum_{i=1}^{t} \alpha_i \cdot b_i(x)$ der Elemente von B mit $a_i \in \mathbb{F}_r$, bilde $m(x) \leftarrow w(x)^{(q-1)/2}$ und $k'(x) \leftarrow m(x)\,(m(x)+1)\,/2$.
2. Ist $k'(x) = k(x)$ oder $k'(x) = 0$, so gehe zu Schritt 1.
3. Bilde $C_1 \leftarrow \{k'(x)b_1(x), \ldots, k'(x)b_t(x)\}$, $C_2 \leftarrow \{(k(x) - k'(x))\,b_1(x), \ldots, (k(x) - k'(x))\,b_t(x)\}$.
4. Dünne mit Gaußelimination C_1 zu einer Basis B_1' und C_2 zu einer Basis B_2' aus.
5. Haben B_1' und B_2' jeweils nur ein Element, so liefere $k'(x)$ und $k(x) - k'(x)$ zurück.
   Hat B_1' nur ein Element, so liefere $k'(x)$ und CAMION $(k - k', f, \mathbb{F}_q, B_2', r)$ zurück.
   Hat B_2' nur ein Element, so liefere $k(x) - k'(x)$ und CAMION $(k', f, \mathbb{F}_q, B_1', r)$ zurück.
   Sonst liefere CAMION $(k - k', f, \mathbb{F}_q, B_2', r)$ und CAMION $(k', f, \mathbb{F}_q, B_1', r)$ zurück.

Beispiel 5.1
Betrachte

$$f(x) = x^{12} + 4x^{11} + 2x^{10} + 2x^9 + 6x^6 + 3x^4 + 3x^3 + 4x^2 + 5 \in \mathbb{F}_7[x],$$

die Menge $\{b_1, b_2, b_3, b_4\}$ *mit*

$$\begin{aligned} b_1(x) &= 1, \\ b_2(x) &= 6x + x^2 + 2x^3 + 6x^5 + 3x^6 + 4x^7 + 5x^8 + x^{11}, \\ b_3(x) &= 3x + 4x^2 + 2x^3 + x^4 + 5x^5 + 5x^6 + x^7 + x^8 + x^{10} \text{ und} \\ b_4(x) &= 2x^2 + x^3 + 3x^4 + 3x^5 + 4x^6 + 6x^7 + 5x^8 + x^9 \end{aligned}$$

ist uns als Basis der Berlekamp–Unteralgebra bereits aus Beispiel 4.1 auf Seite 33 bekannt. Ich wähle die Linearkombination $w = 2b_1 + b_2 + 6b_3 + 2b_4$ *der Basiselemente, dann ist* $m \equiv w^{(r-1)/2} \mod f$ *und*

$$\begin{aligned} k'(x) &= 6x^{11} + 2x^{10} + 4x^8 + 5x^7 + 4x^5 + 2x^4 + 2x^3 + 2 \\ &\equiv m(x)\,(m(x)+1)\,/2 \mod f(x). \end{aligned}$$

Es ist k' *weder 1 noch 0, fahre also fort mit*

$$k_1(x) = k(x) - k'(x) = x^{11} + 5x^{10} + 3x^8 + 2x^7 + 3x^5 + 5x^4 + 5x^3 + 6$$

und $k_2(x) = k'(x)$.

($\boldsymbol{k_1}$) *Multipliziert man alle Basiselemente mit* k_1 *und dünnt diese Menge zu einer Basis aus, so erhält man die neue Basis* $\{b_{11}, b_{12}\}$ *mit*

$$\begin{aligned} b_{11}(x) &= 4x^{11} + 2x^{10} + 6x^9 + 3x^8 + 5x^7 + 4x^6 + 3x^5 + 6x^4 + 4x^3 + 3x^2 + 2x \text{ und} \\ b_{12}(x) &= 6x^{10} + 5x^9 + 3x^8 + x^7 + x^6 + 3x^5 + 3x^3 + 6x^2 + 4x + 1. \end{aligned}$$

Ich wähle die Linearkombination $w = 5b_{12}$, *dann ist* $m \equiv w^{(r-1)/2} \mod f$ *und*

$$\begin{aligned} k_1'(x) &= 3x^{11} + 6x^{10} + 3x^9 + x^8 + x^7 + 2x^6 + x^5 + x^4 + 5x^2 + x + 6 \\ &\equiv m(x)\,(m(x)+1)\,/2 \mod f(x). \end{aligned}$$

Es ist k_1' weder k_1 noch 0, fahre also fort mit

$$\begin{aligned} k_3(x) &= k_1(x) - k_1'(x) \\ &= 5x^{11} + 6x^{10} + 2x^8 + x^7 + 2x^5 + 4x^4 + 5x^3 + 4x^9 + 5x^6 + 2x^2 + 6 \end{aligned}$$

und $k_4(x) = k_1'(x)$.

($\boldsymbol{k_3}$) *Das Erzeugnis der mit k_3 multiplizierten neuen Basiselemente hat die Dimension 1, darum ist k_3 ein primitives idempotentes Element. Der dazugehörige Faktor von f ist*

$$\mathrm{ggT}(f(x), 1 - k_3(x)) = x^3 + 6x^2 + 4x + 4.$$

($\boldsymbol{k_4}$) *Betrachte nun $k_4 = k_1'$. Das Erzeugnis der mit k_4 multiplizierten neuen Basiselemente hat die Dimension 1, darum ist k_4 ein primitives idempotentes Element. Der dazugehörige Faktor von f ist*

$$\mathrm{ggT}(f(x), 1 - k_4(x)) = x^3 + 6x^2 + 4x + 1.$$

($\boldsymbol{k_2}$) *Betrachte nun* $k_2 = k' = 6x^{11} + 2x^{10} + 4x^8 + 5x^7 + 4x^5 + 2x^4 + 2x^3 + 2$. *Multipliziert man alle Basiselemente der Berlekamp–Unteralgebra mit k_2 und dünnt diese Menge zu einer Basis aus, so erhält man die neue Basis $\{b_{41}, b_{42}\}$ mit*

$$\begin{aligned} b_{41}(x) &= 4x^{11} + 5x^{10} + x^9 + 2x^8 + 6x^7 + 6x^6 + 3x^5 + x^4 + 5x^3 + 5x^2 + 4x \text{ und} \\ b_{42}(x) &= x^{10} + 6x^9 + 3x^8 + 2x^7 + x^6 + 2x^5 + 5x^4 + x^3 + 2x^2 + 3x + 6. \end{aligned}$$

Ich wähle die Linearkombination $w = 6b_{41} + 2b_{42}$, *dann ist* $m \equiv w^{(r-1)/2} \mod f$ *und*

$$\begin{aligned} k_2'(x) &= 6x^{11} + 6x^{10} + 3x^9 + 2x^8 + 6x^7 + 4x^6 + 5x^5 + x^4 + 6x^3 + x^2 + 5x + 5 \\ &\equiv m(x)\,(m(x) + 1)\,/2 \mod f(x). \end{aligned}$$

Es ist k_2' weder k_2 noch 0, fahre also fort mit

$$\begin{aligned} k_5(x) &= k_2(x) - k_2'(x) \\ &= 3x^{10} + 2x^8 + 6x^7 + 6x^5 + x^4 + 3x^3 + 4x^9 + 3x^6 + 6x^2 + 2x + 4 \end{aligned}$$

und $k_6(x) = k_2'(x)$.

($\boldsymbol{k_5}$) *Das Erzeugnis der mit k_5 multiplizierten neuen Basiselemente hat die Dimension 1, darum ist k_5 ein primitives idempotentes Element. Der dazugehörige Faktor von f ist*

$$\mathrm{ggT}(f(x), 1 - k_5(x)) = x^3 + 4.$$

($\boldsymbol{k_6}$) *Betrachte nun $k_6 = k_2'$. Das Erzeugnis der mit k_6 multiplizierten neuen Basiselemente hat die Dimension 1, darum ist k_6 ein primitives idempotentes Element. Der dazugehörige Faktor von f ist*

$$\mathrm{ggT}(f(x), 1 - k_6(x)) = x^3 + 6x^2 + 5x + 6.$$

Insgesamt ist $f(x) = (x^3 + 6x^2 + 4x + 4)(x^3 + 6x^2 + 4x + 1)(x^3 + 4)(x^3 + 6x^2 + 5x + 6)$.

Theorie mit der Abbildung in die Gruppe der Ordnung drei für gerade Charakteristik

Im Fall $q = 2^s$ benutze ich zur Erzeugung eines Polynoms mit einer Komponentendarstellung, in der nur Nullen und Einsen vorkommen, nicht den Homomorphismus in die multiplikative Untergruppe der Ordnung zwei, sondern den in die der Ordnung drei — die Idee dieses Standardverfahrens ist uns schon aus der Suche nach Nullstellen in Abschnitt 3.1 bekannt.

r ist hier gegeben durch $r = 2^u$. Ist u gerade, so ist die Ordnung der multiplikativen Gruppe von $\mathbb{F}_r$ durch drei teilbar und das Verfahren verläuft ganz ähnlich wie in Abschnitt 5.1. Ist jedoch u ungerade, so erweitere ich den Körper $\mathbb{F}_r$ auf $\mathbb{F}_{r^2}$, indem ich $\mathbb{F}_{r^2} = \mathbb{F}_r[\xi]/((\xi^2 + \xi + 1) \cdot \mathbb{F}_r[\xi])$ setze. Hat $\mathbb{F}_q$ eine multiplikative Untergruppe der Ordnung drei, so wähle ξ aus dieser Untergruppe. Ich setze also

$$\tilde{r} = r \quad \text{für } r \equiv 1 \mod 3 \text{ und}$$
$$\tilde{r} = r^2 \text{ für } r \equiv 2 \mod 3.$$

$\mathbb{F}_{\tilde{r}}$ muß Unterkörper von $\mathbb{F}_q$ sein. Das ist aber nur dann der Fall, wenn s gerade ist — dann kann ich $\xi \in \mathbb{F}_q$ berechnen und muß nicht wirklich eine Körpererweiterung durchführen. Ist s nicht gerade, so muß ich mir auch $\mathbb{F}_q$ erweitert vorstellen. Ich setze $\mathbb{F}_{\tilde{r}} \subset \mathbb{F}_{\tilde{q}}$.

Praxis

Ist s gerade, u hingegen ungerade, so muß man eines der zwei Elemente $\xi \in \mathbb{F}_q$ bestimmen, das in $\mathbb{F}_4 \backslash \{0, 1\}$ liegt, das also der Gleichung $\xi^2 = \xi + 1$ genügt. Meine Empfehlung, mit dem Fall s gerade umzugehen, ist jedoch, $r = q$ zu setzen.

Bei der Durchführung des Algorithmus wähle ich wiederum eine $\mathbb{F}_{\tilde{r}}$–Linearkombination $w(x)$ der Menge $k(x) \cdot B$ derart, daß $w(x) \neq \alpha k(x)$ für alle $\alpha \in \mathbb{F}_{\tilde{q}}$. Sei $w(x) = \sum_{i=1}^t w_i e_i(x)$. Ich setze

$$m(x) \equiv w(x)^{(\tilde{r}-1)/3} \mod f(x),$$

dann ist

$$m(x) = \sum_{i=1}^{t} w_i^{(\tilde{r}-1)/3} e_i(x) = \sum_{i=1}^{t} m_i e_i(x) \text{ mit } m_i \in \mathbb{F}_4 \text{ für } 1 \leq i \leq t.$$

Es ist $m_i = w_i^{(\tilde{r}-1)/3} = 1$, falls w_i dritte Potenz eines Elements von $\mathbb{F}_{\tilde{r}}$ ist.

Ich bilde $k'^{[1]}(x) \equiv m(x) \cdot (m(x) + 1) \mod f(x)$. Enthält $\mathbb{F}_4 \subset \mathbb{F}_{\tilde{r}}$ die Elemente $\{0, 1, \xi, \xi^2\}$, so gilt wegen $\xi^2 = \xi + 1$:

- $m_i = 0 \Rightarrow k'^{[1]}_i = 0 \cdot 1 = 0,$
- $m_i = 1 \Rightarrow k'^{[1]}_i = 1 \cdot 0 = 0,$
- $m_i = \xi \Rightarrow k'^{[1]}_i = \xi \cdot (\xi + 1) = 1,$
- $m_i = \xi^2 \Rightarrow k'^{[1]}_i = (\xi + 1) \cdot \xi = 1.$

Ist also $m_i \in \{0, 1\}$ für alle $1 \leq i \leq t$, so ist $k'^{[1]}(x) = 0$ und es wird nur eine triviale partielle Faktorisierung erreicht. Dasselbe passiert, wenn $m_i \in \{\xi, \xi^2\}$ für alle $i \in \{j \mid 1 \leq j \leq t \text{ und } k_j \neq 0\}$, denn dann ist $k'^{[1]}(x) = k(x)$. In jedem anderen Fall erhalte ich durch $k(x) = k'^{[1]}(x) + (k(x) - k'^{[1]}(x))$ eine Zerlegung von k in eine nicht triviale Summe zweier orthogonaler idempotenter Elemente, auf die ich das Verfahren rekursiv anwenden kann.

Ich kann statt des obigen $k'^{[1]}$ auch $k'^{[\xi]}(x) \equiv \xi \cdot m(x) \cdot (m(x) + \xi) \mod f(x)$ bilden, dann gilt

- $m_i = 0 \Rightarrow k'^{[\xi]}_i = \xi(0 \cdot \xi) = 0,$

- $m_i = 1 \Rightarrow {k'}_i^{[\xi]} = \xi(1 \cdot (\xi + 1)) = 1,$
- $m_i = \xi \Rightarrow {k'}_i^{[\xi]} = \xi(\xi \cdot 0) = 0,$
- $m_i = \xi^2 \Rightarrow {k'}_i^{[\xi]} = \xi((\xi + 1) \cdot 1) = 1$

und erhalte eine andere Zerlegung von k, die nur in den Fällen $m_i \in \{0, \xi\}$ für alle $1 \leq i \leq t$ oder alle $m_i \in \{1, \xi + 1\}$ nicht gelingt.

Dasselbe Spiel kann ich mit ${k'}^{[\xi+1]}(x) \equiv (\xi + 1) \cdot m(x) \cdot (m(x) + \xi + 1) \mod f(x)$ treiben:

- $m_i = 0 \Rightarrow {k'}_i^{[\xi+1]} = (\xi + 1)(0 \cdot (\xi + 1)) = 0,$
- $m_i = 1 \Rightarrow {k'}_i^{[\xi+1]} = (\xi + 1)(1 \cdot \xi) = 1,$
- $m_i = \xi \Rightarrow {k'}_i^{[\xi+1]} = (\xi + 1)(\xi \cdot 1) = 1,$
- $m_i = \xi^2 \Rightarrow {k'}_i^{[\xi+1]} = (\xi + 1)((\xi + 1) \cdot 0) = 0.$

Ich kann nun jede dieser Zerlegungen für sich wie im Fall ungerader Charakteristik nutzen oder aber die drei Zerlegungen kombiniert anwenden.

Bemerkung 5.1
Sind $i, j \in \{1, \xi, \xi + 1\}$, so ist

$$ {k'}^{[i]} + {k'}^{[j]} = {k'}^{[i+j]}. $$

Satz 5.2
Die Zerlegung

$$ k = k' + (k - k') $$

mit $k' = {k'}^{[1]} \cdot {k'}^{[\xi]} + {k'}^{[1]} \cdot {k'}^{[\xi+1]} + {k'}^{[\xi]} \cdot {k'}^{[\xi+1]}$ ist eine Zerlegung von k in eine Summe vier orthogonaler idempotenter Elemente.

Beweis: Daß für alle $1 \leq i \leq t$ mit $k_i = 0$ auch die i–te Komponente eines jeden Summanden der Zerlegung 0 ist, ist unmittelbar einsichtig. Es bleibt zu zeigen, daß für $k_i = 1$ in genau einem der Summanden die i–te Komponente 1 ist. Sei in den folgenden Punkten $j \in \{1, \xi, \xi + 1\}$:

- Ist $m_i = 0$, so ist ${k'}_i^{[j]} = (j)(0 \cdot (j)) = 0$ für alle $j \in \{1, \xi, \xi + 1\}$, darum ist $(k_i - k'_i) = 1$.
- Ist $m_i = 1$, so ist ${k'}_i^{[1]} = 0$, ${k'}_i^{[\xi]} = 1$ und ${k'}_i^{[\xi+1]} = 1$, darum ist ${k'}_i^{[\xi]} \cdot {k'}_i^{[\xi+1]} = 1$, in den anderen Summanden ist die i–te Komponente 0.
- Ist $m_i = \xi$, so ist ${k'}_i^{[1]} = 1$, ${k'}_i^{[\xi]} = 0$ und ${k'}_i^{[\xi+1]} = 1$, darum ist ${k'}_i^{[1]} \cdot {k'}_i^{[\xi+1]} = 1$, in den anderen Summanden ist die i–te Komponente 0.
- Ist $m_i = \xi + 1$, so ist ${k'}_i^{[1]} = 1$, ${k'}_i^{[\xi]} = 1$ und ${k'}_i^{[\xi+1]} = 0$, darum ist ${k'}_i^{[1]} \cdot {k'}_i^{[\xi]} = 1$, in den anderen Summanden ist die i–te Komponente 0.

□

Vorteil dieser Methode ist, daß man in einem Schritt des Algorithmus im Glücksfall vier statt nur zwei orthogonale idempotente Elemente erhält, auf die man das Verfahren rekursiv anwenden kann. Es wird also die Rekursionstiefe verringert.

Die Wahrscheinlichkeit P für eine nicht triviale Faktorisierung läßt sich auch hier ganz ähnlich wie im Satz 3.3 auf Seite 16 abschätzen durch $P \leq 1/2$, das Verfahren ist in der Komplexitätsklasse RP.

Algorithmus 15 CAMION (k, f, $\mathbb{F}_{\tilde{q}}$, B, $\tilde{r}$)
Eingabe: Das idempotente Element k der Berlekamp–Unteralgebra $\mathcal{B}(f, \tilde{r})$ (Startwert $k(x) = 1$), das normierte, quadratfreie Polynom $f \in \mathbb{F}_{\tilde{q}}[x]$, der Körper $\mathbb{F}_{\tilde{q}}$, ein Teil $B = \{b_1, \dots, b_t\}$ einer Basis der Berlekamp–Unteralgebra (Startwert ist eine Basis der Berlekamp–Unteralgebra), $\tilde{r}$.
Ausgabe: Die Menge der primitiven idempotenten Elemente $e_i(x)$ für $1 \leq i \leq t$.

1. Wähle eine zufällige Linearkombination $w(x) \leftarrow \sum_{i=1}^{t} \alpha_i \cdot b_i(x)$ der Elemente von B mit $a_i \in \mathbb{F}_{\tilde{r}}$, bilde $m(x) \leftarrow w(x)^{(\tilde{q}-1)/3}$, $k'^{[1]}(x) \leftarrow m(x) \cdot (m(x) + 1)$, $k'^{[\xi]}(x) \leftarrow \xi m(x) \cdot (m(x) + \xi)$ und $k'^{[\xi+1]}(x) \leftarrow k'^{[1]}(x) + k'^{[\xi]}(x)$.
2. Sind alle $k'^{[i]}(x)$ mit $i \in \{1, \xi, \xi + 1\}$ entweder 0 oder $k(x)$, so gehe zu Schritt 1.
3. Bilde die Mengen $C_{\xi+1} \leftarrow k'^{[1]} k'^{[\xi]} B$, $C_1 \leftarrow k'^{[\xi]} k'^{[\xi+1]} B$, $C_\xi \leftarrow k'^{[1]} k'^{[\xi+1]} B$ und $C_0 \leftarrow k'^{[0]} B$ mit $k'^{[0]} = k + k'^{[1]} k'^{[\xi]} + k'^{[\xi]} k'^{[\xi+1]} + k'^{[1]} k'^{[\xi+1]}$.
4. Dünne mit Gaußelimination die C_i mit $i \in \{0, 1, \xi, \xi + 1\}$ zu Basen B'_i aus.
5. Ordne jedem B'_i mit $i \in \{1, \xi, \xi + 1\}$ eine Ausgabe out_i zu: Enthält B'_i nur 0, so setze $out_i \leftarrow 0$, hat B'_i nur ein Element, so setze $out_i \leftarrow k'^{[i+1]} k'^{[i+\xi]}$, sonst setze $out \leftarrow$ CAMION $(k'^{[i+1]} k'^{[i+\xi]}, f, \mathbb{F}_{\tilde{q}}, B'_i, \tilde{r})$.
6. Hat B'_0 nur ein Element b'_i, so setze $out_i \leftarrow k'^{[0]}$, sonst setze $out \leftarrow$ CAMION $(k'^{[0]}, f, \mathbb{F}_{\tilde{q}}, B'_i, \tilde{r})$.
7. Liefere out_0, out_1, out_ξ und $out_{\xi+1}$ zurück.

Beispiel 5.2
Dieses Beispiel wird ausnahmsweise nicht über $\mathbb{F}_4$ durchgeführt, da es sich dort nicht von dem Verfahren von Berlekamp unterscheidet. Stattdessen betrachte ich ein Polynome $f \in \mathbb{F}_{16}[x]$ mit

$$f(x) = x^6 + (\xi^3 + \xi^2 + \xi)x^5 + (\xi^3 + \xi)x^4 + \xi^2 x^3 + (\xi^3 + \xi^2 + \xi)x^2 + \xi^2 x + \xi^3 + \xi^2 + \xi + 1$$

und $\mathbb{F}_{16} = \mathbb{F}_2[\xi]/((\xi^4 + \xi + 1)\mathbb{F}_2[\xi])$. f zerfällt in drei irreduzible Faktoren vom Grad zwei.

Die Polynome

$$\begin{aligned} b_1(x) &= 1, \\ b_2(x) &= x^4 + (\xi^3 + \xi)x^3 + (\xi^3 + \xi)x^2 + (\xi^2 + 1)x \text{ und} \\ b_3(x) &= x^5 + (\xi^3 + 1)x^3 + (\xi^3 + \xi + 1)x^2 + (\xi^3 + \xi)x \end{aligned}$$

bilden eine Basis B der Berlekamp–Unteralgebra $\mathcal{B}(f, 16)$, die Menge $\{0, 1, \xi^2 + \xi, \xi^2 + \xi + 1\}$ entspricht $\mathbb{F}_4 \subset \mathbb{F}_{16}$.

Wir möchten mit $(q-1)/3 = 5$ potenzieren, um die Komponenten der Elemente der Berlekamp–Unteralgebra auf $\mathbb{F}_4$ abzubilden.

Ich wähle die zufällige Linearkombination $w = b_2$, dann ist

$$\begin{aligned} m(x) &= (\xi^2 + \xi + 1)x^5 + (\xi + 1)x^4 + (\xi^2 + \xi + 1)x^3 + (\xi^3 + 1)x^2 + (\xi^3 + \xi^2)x + \xi^3 + \xi^2 \\ &\equiv w(x)^5 \mod f(x). \end{aligned}$$

Bilde nun

$$\begin{aligned}
k'^{[1]} &= (\xi^3+\xi^2)x^5 + (\xi^2+\xi)x^3 + (\xi^3+\xi^2+1)x^2 + x \\
&\equiv m(x)\cdot(m(x)+1) \mod f(x), \\
k'^{[\xi]} &= (\xi^3+1)x^5 + (\xi+1)x^4 + x^2 + (\xi^3+\xi)x + \xi^3 + \xi^2 \\
&\equiv \xi\cdot m(x)\cdot(m(x)+\xi) \mod f(x) \text{ und} \\
k'^{[\xi+1]} &= (\xi^2+1)x^5 + (\xi+1)x^4 + (\xi^2+\xi)x^3 + (\xi^3+\xi^2)x^2 + (\xi^3+\xi+1)x + \xi^3 + \xi^2 \\
&= k'^{[1]} + k'^{[\xi]},
\end{aligned}$$

dann sind

$$\begin{aligned}
k'^{[1]}\cdot k'^{[\xi]} &\equiv (\xi^2+1)x^5 + (\xi+1)x^4 + (\xi^2+\xi)x^3 + (\xi^3+\xi^2)x^2 + (\xi^3+\xi+1)x + \xi^3 + \xi^2 + 1, \\
k'^{[\xi]}\cdot k'^{[\xi+1]} &\equiv (\xi^3+\xi^2)x^5 + (\xi^2+\xi)x^3 + (\xi^3+\xi^2+1)x^2 + x + 1 \text{ und} \\
k'^{[1]}\cdot k'^{[\xi+1]} &\equiv (\xi^3+1)x^5 + (\xi+1)x^4 + x^2 + (\xi^3+\xi)x + \xi^3 + \xi^2 + 1
\end{aligned}$$

modulo $f(x)$ drei idempotente Elemente, von denen niemals zwei in derselben Komponente eine 1 haben. Da f in drei irreduzible Faktoren zerfällt, muß es sich bereits um die primitiven idempotenten Elemente handeln. Finde die entsprechenden Faktoren von f:

$$\begin{aligned}
\mathrm{ggT}(f(x), k'^{[1]}\cdot k'^{[\xi]} + 1) &= x^2 + \xi x + \xi^2 + 1, \\
\mathrm{ggT}(f(x), k'^{[\xi]}\cdot k'^{[\xi+1]} + 1) &= x^2 + (\xi^2+\xi+1)x + \xi^2 \text{ und} \\
\mathrm{ggT}(f(x), k'^{[1]}\cdot k'^{[\xi+1]} + 1) &= x^2 + (\xi^3+\xi+1)x + \xi^2.
\end{aligned}$$

Es ist somit $f(x) = (x^2+\xi x+\xi^2+1)(x^2+(\xi^2+\xi+1)x+\xi^2)(x^2+(\xi^3+\xi+1)x+\xi^2)$.

Theorie mit der Spur für gerade Charakteristik

Das zweite Standardverfahren bei Charakteristik $p = 2$ benutzt die Spur. Wir wählen wie gehabt eine $\mathbb{F}_r$–Linearkombination $w(x) \equiv \sum_{i=1}^t w_i e_i(x) \mod f(x)$ in der Menge $k \cdot B$ derart, daß $w(x) \neq \alpha k(x)$ für alle $\alpha \in \mathbb{F}_r$. Für $1 \leq i \leq t$ ist $w_i \in \mathbb{F}_r$ und soll auf ein Element von $\mathbb{F}_2$ abgebildet werden. Dafür bietet sich die Spur an, ich setze

$$k'(x) \equiv T(w(x)) \mod f(x)$$

mit $T(x) = \mathrm{Spur}_{\mathbb{F}_r/\mathbb{F}_2}(x) = x^{2^{u-1}} + x^{2^{u-2}} + \cdots + x^2 + x$. Dann ist

$$k'(x) = \sum_{i=1}^t T(w_i)e_i(x) = \sum_{i=1}^t k'_i e_i(x) \text{ mit } k'_i \in \mathbb{F}_2$$

und ich erhalte wie gewünscht eine Zerlegung $k = k' + (k - k')$, auf deren Summanden ich das Verfahren rekursiv anwenden kann.

Die Wahrscheinlichkeit P für eine nicht triviale Zerlegung ist

$$P = 1 - 2\cdot\left(\frac{1}{2}\right)^t = 1 - \left(\frac{1}{2}\right)^{t-1},$$

also ist das Verfahren in der Komplexitätsklasse RP.

Praxis

Die praktische Umsetzung ist ganz analog zur Umsetzung bei ungerader Charakteristik. Im Fall $r = 2$ ist T die Identität und das Verfahren sieht wie folgt aus:

Ich wähle ein beliebiges $w \in k \cdot \mathcal{B}(f, r)$ mit $w(x) \notin \{0, k\}$ und setze $k' = w$. Dann ist $k = k' + (k - k')$ eine Zerlegung von k in eine nicht triviale Summe zweier orthogonaler idempotenter Elemente.

Algorithmus 16 CAMION $(k, f, \mathbb{F}_q, B, r)$
Eingabe: Das idempotente Element k der Berlekamp–Unteralgebra $\mathcal{B}(f, r)$ (Startwert $k(x) = 1$), das normierte, quadratfreie Polynom $f \in \mathbb{F}_q[x]$, der Körper $\mathbb{F}_q$, ein Teil $B = \{b_1, \ldots, b_t\}$ einer Basis der Berlekamp–Unteralgebra (Startwert ist eine Basis der Berlekamp–Unteralgebra), $r = 2^u$.
Ausgabe: Die Menge der primitiven idempotenten Elemente $e_i(x)$ für $1 \leq i \leq t$.
Programm: Camion.mws: camion

1. Setze $T(x) \leftarrow \sum_{i=0}^{u} x^{2^i}$.
2. Wähle eine zufällige Linearkombination $w(x) \leftarrow \sum_{i=1}^{t} \alpha_i \cdot b_i(x)$ der Elemente von B mit $a_i \in \mathbb{F}_r$ und bilde $k'(x) \leftarrow T(w(x))$.
3. Ist $k'(x) = k(x)$ oder $k'(x) = 0$, so gehe zu Schritt 2.
4. Bilde $C_1 \leftarrow \{k'(x)b_1(x), \ldots, k'(x)b_t(x)\}$, $C_2 \leftarrow \{(k(x) + k'(x))\, b_1(x), \ldots, (k(x) + k'(x))\, b_t(x)\}$.
5. Dünne mit Gaußelimination C_1 zu einer Basis B_1' und C_2 zu einer Basis B_2' aus.
6. Haben B_1' und B_2' jeweils nur ein Element, so liefere $k'(x)$ und $k(x) + k'(x)$ zurück.
   Hat B_1' nur ein Element, so liefere $k'(x)$ und CAMION $(k + k', f, \mathbb{F}_q, B_2', r)$ zurück.
   Hat B_2' nur ein Element, so liefere $k(x) + k'(x)$ und CAMION $(k', f, \mathbb{F}_q, B_1', r)$ zurück.
   Sonst liefere CAMION $(k + k', f, \mathbb{F}_q, B_2', r)$ und CAMION $(k', f, \mathbb{F}_q, B_1', r)$ zurück.

Beispiel 5.3
Sei $p = 2$, $q = 4$ mit $\mathbb{F}_4 = \mathbb{F}_2[\xi] / \left((\xi^2 + \xi + 1)\mathbb{F}_2[\xi]\right)$, wir möchten das quadratfreie Polynom

$$f(x) = x^9 + \xi x^8 + (\xi + 1)x^7 + \xi x^5 + (\xi + 1)x^4 + (\xi + 1)x + 1 \in \mathbb{F}_4[x]$$

zerlegen, welches in drei Faktoren vom Grad drei zerfällt. $T(x) = x^2 + x$ ist die Spur von $\mathbb{F}_4$ nach $\mathbb{F}_2$, die Menge $\{b_1, b_2, b_3\}$ mit

$$\begin{aligned} b_1(x) &= 1, \\ b_2(x) &= x^5 + (\xi + 1)x^3 + x^2 + \xi x \text{ und} \\ b_3(x) &= x^7 \end{aligned}$$

ist uns als Basis der Berlekamp–Unteralgebra bereits aus Beispiel 4.3 auf Seite 36 bekannt. Ich wähle die Linearkombination $w = b_1 + b_2$ der Basiselemente, dann ist

$$\begin{aligned} k'(x) &= x^7 + x^5 + (\xi + 1)x^3 + x^2 + \xi x + \xi \\ &\equiv T(w(x)) \mod f(x). \end{aligned}$$

Es ist k' weder 1 noch 0, fahre also fort mit

$$k_1(x) = k(x) - k'(x) = x^7 + x^5 + (\xi + 1)x^3 + x^2 + \xi x + \xi + 1$$

und $k_2(x) = k'(x)$.

($\boldsymbol{k_1}$) *Multipliziert man alle Basiselemente mit k_1 und dünnt diese Menge zu einer Basis aus, so erhält man die neue Basis $\{b_{11}, b_{12}\}$ mit*

$$\begin{aligned} b_{11}(x) &= x^7 + x^5 + (\xi+1)x^3 + x^2 + \xi x + \xi + 1 \text{ und} \\ b_{12}(x) &= \xi x^7 + (\xi+1)x^5 + \xi x^3 + (\xi+1)x^2 + x + \xi + 1. \end{aligned}$$

Ich wähle die Linearkombination $w = b_{11} + \xi b_{12}$, dann ist

$$\begin{aligned} k_1'(x) &= \xi x^7 + (\xi+1)x^5 + \xi x^3 + (\xi+1)x^2 + x + \xi + 1 \\ &\equiv T(w(x)) \mod f(x). \end{aligned}$$

Es ist k_1' weder k_1 noch 0, fahre also fort mit

$$k_3(x) = k_1(x) - k_1'(x) = (\xi+1)x^7 + \xi x^5 + x^3 + \xi x^2 + (\xi+1)x$$

und $k_4(x) = k_1'(x)$.

($\boldsymbol{k_3}$) *Das Erzeugnis der mit k_3 multiplizierten neuen Basiselemente hat die Dimension 1, darum ist k_3 ein primitives idempotentes Element. Der dazugehörige Faktor von f ist*

$$\mathrm{ggT}(f(x), 1 - k_3(x)) = x^3 + \xi x^2 + 1.$$

($\boldsymbol{k_4}$) *Betrachte nun $k_4 = k_1'$. Das Erzeugnis der mit k_4 multiplizierten neuen Basiselemente hat die Dimension 1, darum ist k_4 ein primitives idempotentes Element. Der dazugehörige Faktor von f ist*

$$\mathrm{ggT}(f(x), 1 - k_4(x)) = x^3 + \xi x + 1.$$

($\boldsymbol{k_2}$) *Das Erzeugnis der mit k_2 multiplizierten Basiselemente hat die Dimension 1, darum ist k_2 ein primitives idempotentes Element. Der dazugehörige Faktor von f ist*

$$\mathrm{ggT}(f(x), 1 - k_2(x)) = x^3 + x + 1.$$

Insgesamt ist $f(x) = (x^3 + \xi x^2 + 1)(x^3 + \xi x + 1)(x^3 + x + 1)$.

Komplexität

Ich bestimme die Komplexität examplarisch für ungerade Charakteristik p.

Zu Beginn des Verfahrens wird in $A = (2\log_2(q) + n) \cdot (3n^2 + 7n + 4) + 2n^3 - 2n^2$ Operationen die Berlekamp–Unteralgebra $\mathcal{B}(f, r)$ bestimmt.

In jedem weiteren Schritt des Algorithmus werden höchstens t Polynome mit einem Polynom k multipliziert, das dauert $t(3n^2 + 7n + 4)$ Operationen. Dann wird der Rang einer $(t \times n)$–Matrix durch Gaußelimination bestimmt, dies geschieht innerhalb von $t^2 n - tn$ elementaren Operationen. Die Zeilen der Matrix werden als Polynome aufgefaßt, aus denen eine zufällige Linearkombination erstellt wird: Jedes Polynom wird in n Operationen mit einem Element des endlichen Körpers multipliziert, die Ergebnisse werden addiert. Die Linearkombination bedarf also höchstens $tn + (t-1)(n+1) = 2tn - (t + n + 1)$ Operationen. Das zufällige Polynom wird mit $\frac{r-1}{2}$ potenziert, dazu werden $2 \cdot (\lfloor \log_2(r) \rfloor - 1) \cdot (3n^2 + 7n + 4)$ Operationen getätigt. Zu dem Ergebnis wird in einer Operation 1 hinzugezählt, dann wird in $3n^2 + 7n + 4$ Operationen eine Multiplikation zweier Polynom durchgeführt. Deren Ergebnis wird in $n + 1$ Operationen von k abgezogen. Schließlich wird ein größter gemeinsamer Teiler in $2n^2 + 3n$ Operationen bestimmt. Ein Schritt des Algorithmus benötigt darum insgesamt

$t(3n^2 + 7n + 4) + (t^2 n - tn) + (2tn - (t + n + 1)) + (2(\log_2(r) - 1)(3n^2 + 7n + 4)) + 1 + (3n^2 + 7n + 4) + n + 1 + (2n^2 + 3n)$ Operationen.

Dann berechnet sich eine obere Schranke für die Anzahl der Operationen in einem Schritt wegen $t \geq 1$ und $r \geq 2 \Leftrightarrow \log_2(r) \geq 0$ als

$$\begin{aligned}
&nt^2 + t(3n^2 + 8n + 3) + n^2(6\log_2(r) - 1) + n(14\log_2(r) - 3) + 8\log_2(r) - 2 \\
&\quad < nt^2 + t(3n^2 + 8n + 3) + \left(6n^2\log_2(r) + 14n\log_2(r) + 8\log_2(r)\right)t \\
&\quad = nt^2 + \left(n^2(6\log_2(r) + 3) + n(14\log_2(r) + 8) + 8\log_2(r) + 3\right)t.
\end{aligned}$$

Die Rekursionstiefe wird mit dem Bälle und Tonnen–Modell auf $E(D) = \frac{2\log_2(t)}{\log_2(2r/(r+1))} + 1 + \frac{2r}{r-1}$ geschätzt, wobei die irreduziblen Faktoren von f den t Bällen entsprechen, die zwei Tonnen sind die Reste 0 oder 1 des zufälligen Polynoms k' modulo f_i für $1 \leq i \leq t$, w ist $\left(\frac{r-1}{2} + 1\right) \cdot \frac{1}{r} = \frac{r+1}{2r}$. Ich schätze außerdem ab:

$$\begin{aligned}
\frac{2r}{r+1} \geq \frac{4}{3} \quad &\Leftrightarrow \quad \log_2(2r/(r+1)) \geq \log_2(4/3) \\
\Leftrightarrow \quad \frac{1}{\log_2(2r/(r+1))} \leq \frac{1}{\log_2(4/3)} \quad &\Leftrightarrow \quad \frac{1}{\log_2(2r/(r+1))} < 3
\end{aligned}$$

und $\frac{2r}{r-1} < 4$. Somit ist $E(D) < 6\log_2(t) + 5$. Die Kostenfunktion $\mathrm{kost}(t)$ ist die obere Schranke der Operationen in einem Schritt. Ich schätze somit die Anzahl der im Algorithmus durchzuführenden Operationen auf weniger als

$$\begin{aligned}
&A + \mathrm{kost}(t) \cdot E(D) = (2\log_2(q) + n)(3n^2 + 7n + 4) + \\
&+(2n^3 - 2n^2) + \left(nt^2 + t\left(n^2(6\log_2(r) + 2) + n(14\log_2(r) + 5) + 8\log_2(r) + 1\right)\right) \cdot (6\log_2(t) + 5),
\end{aligned}$$

das entspricht der Komplexität $\mathcal{O}(n^3 + nt^2\log(t) + n^2t\log(r)\log(t))$.

5.2 Weiterentwicklung des Algorithmus von Camion

Sei die Anzahl t der Faktoren, in die das Polynom f zerfällt, schon bekannt, z. B. nach einer partiellen Faktorisierung in Teilpolynome mit gleichgradigen Faktoren wie in Abschnitt 2.2 auf Seite 13 oder nach der Bestimmung einer Basis der Berlekamp–Unteralgebra $\mathcal{B}(f, r)$. In diesem Fall wandelt Camion in [7] auf Anregung von Lazard in [26] den Algorithmus ab, so daß er schneller wird:

Ich habe eine Menge idempotenter Elemente gegeben — Startwert ist $\{1\}$. Ich bestimme wie im vorherigen Abschnitt 5.1 ein zufälliges idempotentes Element k'. Dann multipliziere ich jedes idempotente Element aus der Menge sowohl mit k' als auch mit $1 - k'$. Aus der neuen Menge nehme ich 0 heraus und wende dasselbe Verfahren auf die neue Menge idempotenter Elemente an. Sobald in der Menge t Elemente enthalten sind, handelt es sich um die primitiven idempotenten Elemente, da alle Elemente der Menge orthogonal zueinander und idempotent sind. Mit Hilfe der primitiven idempotenten Elemente aus der Menge kann ich leicht alle irreduziblen Faktoren von f berechnen.

Der Vorteil dieses Verfahrens ist, im Vergleich zum vorigen in Abschnitt 5.1 auf Seite 45, daß nicht in jedem Rekursionsschritt die Dimension des Erzeugnisses eines Erzeugendensystems bestimmt werden muß, dadurch ist es schneller.

Algorithmus 17 CAMION (f, $\mathbb{F}_q$, B, r, t)
Eingabe: Das normierte, quadratfreie Polynom $f \in \mathbb{F}_q[x]$, der Körper $\mathbb{F}_q$, die Basis $B = \{b_1, \ldots, b_t\}$ der Berlekamp–Unteralgebra $\mathcal{B}(f, r)$, r und die Anzahl t der Faktoren von f.
Ausgabe: Die Menge der primitiven idempotenten Elemente $e_i(x)$ für $1 \le i \le t$.
Programm: CamionWeiter.mws: camion

1. Setze $L \leftarrow \{1\}$.
2. Erzeuge ein zufälliges idempotentes Element $k'(x)$.
3. Ersetze jedes Element $e(x)$ der Liste L durch die beiden idempotenten Elemente $k'(x) \cdot e(x)$ und $(1 - k'(x)) \cdot e(x)$.
4. Streiche aus der Menge L das Element 0 heraus.
5. Hat L nun t Elemente, so liefere sie zurück, sonst gehe zu Schritt 2.

Beispiel 5.4
Betrachte

$$f(x) = x^{12} + 4x^{11} + 2x^{10} + 2x^9 + 6x^6 + 3x^4 + 3x^3 + 4x^2 + 5 \in \mathbb{F}_7[x],$$

wir wissen, daß f in vier Faktoren vom Grad drei zerfällt. Des weiteren ist uns aus Beispiel 4.1 auf Seite 33 die Menge $\{b_1, b_2, b_3, b_4\}$ mit

$$\begin{aligned} b_1(x) &= 1, \\ b_2(x) &= 6x + x^2 + 2x^3 + 6x^5 + 3x^6 + 4x^7 + 5x^8 + x^{11}, \\ b_3(x) &= 3x + 4x^2 + 2x^3 + x^4 + 5x^5 + 5x^6 + x^7 + x^8 + x^{10} \text{ und} \\ b_4(x) &= 2x^2 + x^3 + 3x^4 + 3x^5 + 4x^6 + 6x^7 + 5x^8 + x^9 \end{aligned}$$

als Basis der Berlekamp–Unteralgebra bekannt. Ich wähle nun die Linearkombination $w = 2b_1 + 3b_2 + 3b_3$ der Basiselemente, dann ist $m \equiv w^{(r-1)/2} \mod f$ und

$$\begin{aligned} k'(x) &= 5x^{11} + 2x^{10} + x^9 + 4x^8 + x^6 + x^5 + 5x^4 + x^3 + x^2 + x + 4 \\ &\equiv m(x)\,(m(x) + 1)\,/2 \mod f(x), \\ 1 - k'(x) &= 2x^{11} + 5x^{10} + 6x^9 + 3x^8 + 6x^6 + 6x^5 + 2x^4 + 6x^3 + 6x^2 + 6x + 4. \end{aligned}$$

Es ist k' weder 1 noch 0, beginne also eine Liste $L = \{L_1, L_2\}$ mit $L_1 = k'$ und $L_2 = 1 - k'$.

Ich wähle eine neue Linearkombination $w = 4b_1 + 5b_2 + 4b_3 + 3b_4$ der Basiselemente, dann ist $m \equiv w^{(r-1)/2} \mod f$ und

$$\begin{aligned} k'(x) &= 3x^{10} + 4x^9 + 2x^8 + 6x^7 + 3x^6 + 6x^5 + x^4 + 3x^3 + 6x^2 + 2x + 4 \\ &\equiv m(x)\,(m(x) + 1)\,/2 \mod f(x), \\ 1 - k'(x) &= 4x^{10} + 3x^9 + 5x^8 + x^7 + 4x^6 + x^5 + 6x^4 + 4x^3 + x^2 + 5x + 4. \end{aligned}$$

Es ist k' weder 1 noch 0, k' ist auch noch nicht in der Liste L enthalten. Darum multipliziere ich jedes Element der Liste sowohl mit k' als auch mit $1 - k'$:

$$\begin{aligned} L_1(x) \cdot k'(x) &\equiv 3x^{10} + 4x^9 + 2x^8 + 6x^7 + 3x^6 + 6x^5 + x^4 + 3x^3 + 6x^2 + 2x + 4, \\ L_1(x) \cdot (1 - k'(x)) &\equiv 5x^{11} + 6x^{10} + 4x^9 + 2x^8 + x^7 + 5x^6 + 2x^5 + 4x^4 + 5x^3 + 2x^2 + 6x, \\ L_2(x) \cdot k'(x) &\equiv 0, \\ L_2(x) \cdot (1 - k'(x)) &\equiv L_2(x), \end{aligned}$$

alle Äquivalenzen sind modulo $f(x)$. *Ich erhalte die neue Liste* $L = \{L_{11}, L_{12}, L_{13}\}$ *mit*

$$\begin{aligned}
L_{11}(x) &= 3x^{10} + 4x^9 + 2x^8 + 6x^7 + 3x^6 + 6x^5 + x^4 + 3x^3 + 6x^2 + 2x + 4,\\
L_{12}(x) &= 5x^{11} + 6x^{10} + 4x^9 + 2x^8 + x^7 + 5x^6 + 2x^5 + 4x^4 + 5x^3 + 2x^2 + 6x \text{ und}\\
L_{13}(x) &= 2x^{11} + 5x^{10} + 6x^9 + 3x^8 + 6x^6 + 6x^5 + 2x^4 + 6x^3 + 6x^2 + 6x + 4,
\end{aligned}$$

in die ich nur idempotente Elemente aufnehme, welche nicht 0 sind.

Wähle ich die Linearkombination $w = 2b_1 + b_2 + 6b_3 + 2b_4$ *der Basiselemente, dann ist* $m \equiv w^{(r-1)/2} \mod f$ *und*

$$\begin{aligned}
k'(x) &= 3x^{11} + 6x^{10} + 3x^9 + x^8 + x^7 + 2x^6 + x^5 + x^4 + 5x^2 + x + 6\\
&\equiv m(x)\,(m(x) + 1)\,/2 \mod f(x),\\
1 - k'(x) &= 4x^{11} + x^{10} + 4x^9 + 6x^8 + 6x^7 + 5x^6 + 6x^5 + 6x^4 + 2x^2 + 6x + 2.
\end{aligned}$$

Es ist k' *weder 1 noch 0,* k' *ist auch noch nicht in der Liste* L *enthalten. Darum multipliziere ich jedes Element der Liste* L *sowohl mit* k' *als auch mit* $1 - k'$*:*

$$\begin{aligned}
L_{11}(x) \cdot k'(x) &\equiv 0,\\
L_{11}(x) \cdot (1 - k'(x)) &\equiv L_1(x),\\
L_{12}(x) \cdot k'(x) &\equiv 0,\\
L_{12}(x) \cdot (1 - k'(x)) &\equiv L_2(x),\\
L_{13}(x) \cdot k'(x) &\equiv 3x^{11} + 6x^{10} + 3x^9 + x^8 + x^7 + 2x^6 + x^5 + x^4 + 5x^2 + x + 6,\\
L_{13}(x) \cdot (1 - k'(x)) &\equiv 6x^{11} + 6x^{10} + 3x^9 + 2x^8 + 6x^7 + 4x^6 + 5x^5 + x^4 + 6x^3 + x^2 + 5x + 5,
\end{aligned}$$

alle Äquivalenzen sind modulo $f(x)$. *Ich erhalte die neue Liste* $L = \{L_{21}, L_{22}, L_{23}, L_{24}\}$ *mit*

$$\begin{aligned}
L_{21}(x) &= 3x^{10} + 4x^9 + 2x^8 + 6x^7 + 3x^6 + 6x^5 + x^4 + 3x^3 + 6x^2 + 2x + 4,\\
L_{22}(x) &= 5x^{11} + 6x^{10} + 4x^9 + 2x^8 + x^7 + 5x^6 + 2x^5 + 4x^4 + 5x^3 + 2x^2 + 6x,\\
L_{23}(x) &= 3x^{11} + 6x^{10} + 3x^9 + x^8 + x^7 + 2x^6 + x^5 + x^4 + 5x^2 + x + 6 \text{ und}\\
L_{24}(x) &= 6x^{11} + 6x^{10} + 3x^9 + 2x^8 + 6x^7 + 4x^6 + 5x^5 + x^4 + 6x^3 + x^2 + 5x + 5.
\end{aligned}$$

Da L *inzwischen vier Elemente hat, haben wir also die Liste mit den vier primitiven idempotenten Elementen gefunden. Die zu diesen Elementen gehörenden Faktoren von* f *sind:*

- $\operatorname{ggT}(f(x), 1 - L_{21}(x)) = x^3 + 4$,
- $\operatorname{ggT}(f(x), 1 - L_{22}(x)) = x^3 + 6x^2 + 4x + 4$,
- $\operatorname{ggT}(f(x), 1 - L_{23}(x)) = x^3 + 6x^2 + 4x + 1$ *und*
- $\operatorname{ggT}(f(x), 1 - L_{24}(x)) = x^3 + 6x^2 + 5x + 6$.

Insgesamt ist somit $f(x) = (x^3 + 4)(x^3 + 6x^2 + 4x + 4)(x^3 + 6x^2 + 4x + 1)(x^3 + 6x^2 + 5x + 6)$.

Komplexität

Zu Beginn des Verfahrens wird in $A = (2\log_2(q) + n) \cdot (3n^2 + 7n + 4) + (2n^3 - 2n^2)$ elementaren Operationen die Berlekamp–Unteralgebra $\mathcal{B}(f, r)$ bestimmt.

In jedem weiteren Schritt des Algorithmus wird in $2tn - (t + n + 1)$ Operationen eine zufällige Linearkombination der Basispolynome erstellt.

Das zufällige Polynom wird mit $\frac{r-1}{2}$ potenziert, dazu werden $2 \cdot (\log_2(r - 1) - 1) \cdot (3n^2 + 7n + 4)$ Operationen getätigt.

Zu dem Ergebnis wird in einer Operation 1 hinzugezählt, dann wird in $3n^2 + 7n + 4$ Operationen eine Multiplikation zweier Polynome durchgeführt. Deren Ergebnis wird mit den höchstens $t - 1$ primitiven Elementen auf den größten gemeinsamen Teiler überprüft, das dauert $(t - 1)(2n^2 + 3n)$ Operationen.

Ein Schritt des Algorithmus benötigt darum insgesamt

$$\begin{aligned} 2tn - (t+n) + (2\log_2(r-1) - 2) \cdot (3n^2 + 7n + 4) + (3n^2 + 7n + 2) + (t-1)(2n^2 + 3n) \\ < t(2n^2 + 4n - 1) + \left(6n^2 \log_2(r) + 14n \log_2(r) + 8 \log_2(r)\right) \\ \leq t(2n^2 + 4n - 1) + \left(6n^2 \log_2(r) + 14n \log_2(r) + 8 \log_2(r)\right) t \\ = t\left(n^2(6\log_2(r) + 2) + n(14\log_2(r) + 4) + 8\log_2(r) - 1\right) \end{aligned}$$

Operationen, ich benutze die letzte Zeile als obere Schranke.

Die geschätzte Rekursionstiefe beträgt wie zuvor $E(D) < 6\log_2(t) + 5$, wobei die irreduziblen Faktoren von f den t Bällen entsprechen, die $l = 2$ Tonnen sind die Reste 0 oder 1 des zufälligen Polynoms k' modulo f_i für $1 \leq i \leq t$, w ist $\frac{r+1}{2r}$, alles wie bei der ersten Beschreibung des Algorithmus von Camion. Die Kostenfunktion $\operatorname{kost}(t)$ ist auch hier die obere Schranke der in einem Schritt durchzuführenden Operationen, es gilt $\operatorname{kost}(at) = a \operatorname{kost}(t)$, also schätzt man die Laufzeit auf weniger als

$A + \operatorname{kost}(t) E(D) = (2\log_2(q) + n)(3n^2 + 7n + 4) +$
$+ 2n^3 - 2n^2 + t\left(n^2(6\log_2(r) + 2) + n(14\log_2(r) + 4) + 8\log_2(r) - 1\right) \cdot (6\log_2(t) + 5)$ Operationen.

Dies entspricht der Komplexität $\mathcal{O}\left(n^3 + n^2 t \log(t) \log(r)\right)$.

5.3 Der Algorithmus von Cantor und Zassenhaus

Dies ist eine weitere probabilistische Methode zur Faktorisierung von Polynomen über endlichen Körpern, die 1981 von Cantor und Zassenhaus in [8] entwickelt wurde. Sie funktioniert ähnlich wie der Algorithmus von Camion im vorigen Abschnitt 5.1, greift aber nicht auf die Berlekamp–Unteralgebra zu, welche darum nicht bestimmt werden muß. Das macht diese Methode besonders schnell.

f sei normiert, quadratfrei und es sei bekannt, daß alle irreduziblen Faktoren von f den Grad d haben. Des weiteren sei f gegeben durch $f(x) = \sum_{i=0}^{n} a_i x^i = \prod_{i=1}^{t} f_i(x)$ — dies ist immer nach einer partiellen Faktorisierung in Teilpolynome mit gleichgradigen Faktoren wie in Abschnitt 2.2 auf Seite 13 gegeben.

Theorie für ungerades q

Ist $g \in \mathbb{F}_q[x]$ ein zufällig gewähltes Polynom mit $\operatorname{Grad}(g) < n$, so hat g die Darstellung

$$g(x) = \sum_{i=1}^{t} g_i(x) e_i(x)$$

mit passenden $g_i \in \mathbb{F}_q[x]$ mit $\operatorname{Grad}(g_i) < d$. g entspricht also in der Komponentendarstellung dem Vektor $(g_1(x), g_2(x), \ldots, g_t(x))$. Da $\operatorname{Grad}(f_i) = d$ für alle $1 \leq i \leq t$, kann man sich die Komponenten für jedes i vorstellen als Elemente des Körpers $\mathbb{F}_{q^d}$. Auf diese Elemente kann ich den Homomorphismus in die multiplikative Untergruppe der Ordnung zwei anwenden: $g(x)^{(q^d-1)/2} \mod f(x)$ entspricht in der Komponentenschreibweise $(g_1(x)^{(q^d-1)/2}, g_2(x)^{(q^d-1)/2}, \ldots, g_t(x)^{(q^d-1)/2}) = (\gamma_1, \gamma_2, \ldots, \gamma_t)$, wobei $\gamma_i \in \{0, 1, -1\}$ für $1 \leq i \leq t$.

Gibt es nun ein Paar (i, j) mit $1 \leq i, j \leq t$ mit $i \neq j$ derart, daß $\gamma_i \neq \gamma_j$ und $\gamma_i = 1$, so ist $g^{(q^d-1)/2}(x) - 1$ ein Polynom, das durch f_i, nicht aber durch f_j teilbar ist. Darum liefert die Berechnung von

$$\operatorname{ggT}(f(x), g(x)^{(q^d-1)/2} - 1)$$

einen nicht trivialen Faktor von f.

Die Wahrscheinlichkeit P, daß ein solcher nicht trivialer Faktor gefunden wird, wird ganz analog zu den Berechnungen in Satz 3.3 auf Seite 16 abgeschätzt durch $P \geq 1/2$, das Verfahren ist in der Komplexitätsklasse RP.

Algorithmus 18 CANTOR UND ZASSENHAUS $(f, \mathbb{F}_q, d, t)$
Eingabe: Das normierte, quadratfreie Polynom $f \in \mathbb{F}_q[x]$ vom Grad n, welches in t Faktoren vom Grad d zerfällt, und der Körper $\mathbb{F}_q$.
Ausgabe: Die Faktorisierung von f.
Programm: CanZass.mws: canzass

1. Wähle ein zufälliges Polynom $g(x)$ mit $\mathrm{Grad}(g) < n$ und Koeffizienten in $\mathbb{F}_q$. Bestimme $g_1(x) \leftarrow g(x)^{(q^d-1)/2}$.
2. Setze $h_1(x) \leftarrow \mathrm{ggT}(f(x), g_1(x)-1)$, $h_2(x) \leftarrow \mathrm{ggT}(f(x), g_1(x))$, $h_3(x) \leftarrow \mathrm{ggT}(f(x), g_1(x)+1)$.
3. Bestimme für jedes $i \in \{1,2,3\}$ eine Ausgabe out_i: Ist $\mathrm{Grad}(h_i) = d$, so setze $out_i \leftarrow h_i(x)$, ist $\mathrm{Grad}(h_i) < d$, so setze $out_i \leftarrow 1$, sonst setze $out_i \leftarrow$ CANTOR UND ZASSENHAUS $(h_i, \mathbb{F}_q, d, n/d)$.
4. Liefere out_1, out_2 und out_3 zurück.

Beispiel 5.5
Betrachte

$$f(x) = x^{12} + 4x^{11} + 2x^{10} + 2x^9 + 6x^6 + 3x^4 + 3x^3 + 4x^2 + 5 \in \mathbb{F}_7[x].$$

f zerfällt in vier Faktoren vom Grad drei. Ich wähle ein zufälliges Polynom g mit einem Grad kleiner als zwölf:

$$g(x) = 2x^{11} + 3x^{10} + 4x^8 + 2x^6 + 2x^4 + x^3 + 2.$$

Dieses Polynom soll mit $(q^d - 1)/2 = 171$ potenziert werden. Berechne

$$g(x)^{171} \equiv 6x^{11} + 4x^{10} + 6x^8 + 3x^6 + 4x^4 + 6x^3 + x^2 + 6x + 5 \mod f(x)$$

als Polynom, das nur die Komponenten $0, 1$ und -1 hat. Dann ist

$$\begin{aligned} h_1(x) &= \mathrm{ggT}\left(f(x), g(x)^{171} - 1\right) = x^6 + 6x^5 + 4x^4 + 5x^3 + 3x^2 + 2x + 4 \text{ und} \\ h_2(x) &= f(x)/h_1(x) \qquad\qquad\qquad = x^6 + 5x^5 + 3x^4 + x^3 + 3x^2 + 2x + 3, \end{aligned}$$

rechne also mit $f = h_1 \cdot h_2$ weiter. Das nächste zufällig gewählte Polynom sei

$$g(x) = x^{11} + 2x^{10} + 5x^9 + 2x^7 + 2x^6 + 2x^5 + 2x^4 + 3x + 3.$$

Berechne

$$g(x)^{171} \equiv 2x^{11} + 3x^{10} + 6x^8 + 4x^7 + 6x^5 + 3x^4 + 3x^3 + 4 \mod f(x)$$

als Polynom, das nur die Komponenten $0, 1$ und -1 hat. Dann ist

$$\begin{aligned} h_3(x) &= \mathrm{ggT}\left(h_1(x), g(x)^{171} - 1\right) = x^3 + 6x^2 + 4x + 1, \\ h_4(x) &= h_1(x)/h_3(x) \qquad\qquad\quad = x^3 + 4, \\ h_5(x) &= \mathrm{ggT}\left(h_2(x), g(x)^{171} - 1\right) = x^3 + 6x^2 + 4x + 4 \text{ und} \\ h_6(x) &= h_2(x)/h_5(x) \qquad\qquad\quad = x^3 + 6x^2 + 5x + 6. \end{aligned}$$

Wir haben vier Faktoren gefunden, also ist

$$f(x) = (x^3 + 6x^2 + 4x + 1)(x^3 + 4)(x^3 + 6x^2 + 4x + 4)(x^3 + 6x^2 + 5x + 6).$$

Theorie mit der Abbildung in die Gruppe der Ordnung drei für gerade Charakteristik

Im Fall $q = 2^s$ benutzen Cantor und Zassenhaus zur Erzeugung eines Polynoms, das in der Komponentenschreibweise nur Nullen und Einsen enthält, nicht den Homomorphismus in die multiplikative Untergruppe der Ordnung zwei, sondern den in die multiplikative Untergruppe der Ordnung drei, dieses Standardverfahren kennen wir bereits von der Suche nach Nullstellen in Abschnitt 3.1.

Ist s gerade, so ist die Ordnung der multiplikativen Gruppe von $\mathbb{F}_{2^s}$ durch drei teilbar und das Verfahren verläuft wie zuvor. Ist jedoch $s \equiv 1 \mod 2$, so erweitern Cantor und Zassenhaus den Körper $\mathbb{F}_{2^s} = \mathbb{F}_q$ auf $\mathbb{F}_{q^2}$: Betrachte f als ein Polynom über $\mathbb{F}_{\tilde{q}}$ mit $\mathbb{F}_4 \subset \mathbb{F}_{\tilde{q}}$, also auch $\mathbb{F}_4 \subset \mathbb{F}_{\tilde{q}^d}$. Bestimme $\xi \in \mathbb{F}_4$.

Ist $g \in \mathbb{F}_{\tilde{q}}[x]$ ein zufällig gewähltes Polynom mit $\mathrm{Grad}(g) < n$, so sei $(g_1(x), g_2(x), \ldots, g_t(x))$ mit passenden $g_i \in \mathbb{F}_{\tilde{q}}[x]$ und $\mathrm{Grad}(g_i) < d$ der Vektor, dem g in der Komponentendarstellung entspricht. Man stellt sich die Komponente für jedes i vor als Element des Körpers $\mathbb{F}_{\tilde{q}^d}$ und kann auf dieses Element den Homomorphismus anwenden, der die Elemente der multiplikativen Gruppe von $\mathbb{F}_{\tilde{q}^d}$ auf die Elemente der multiplikativen Untergruppe der Ordnung drei abbildet: $g(x)^{(\tilde{q}^d-1)/3} \mod f(x)$ entspricht in der Komponentendarstellung dem Vektor

$$(g_1(x)^{(\tilde{q}^d-1)/3}, g_2(x)^{(\tilde{q}^d-1)/3}, \ldots, g_t(x)^{(\tilde{q}^d-1)/3}) = (\gamma_1, \gamma_2, \ldots, \gamma_t),$$

wobei $\gamma_i \in \{0, 1, \xi, \xi + 1\}$ für $1 \leq i \leq t$.

Gibt es nun ein Paar (i, j) mit $1 \leq i, j \leq t$ und $i \neq j$ derart, daß $\gamma_i \neq \gamma_j$ und $\gamma_i = 1$, so ist $g(x)^{(\tilde{q}^d-1)/3} + 1$ ein Polynom, das durch f_i, nicht aber durch f_j teilbar ist. Darum liefert die Berechnung von

$$\mathrm{ggT}(f(x), g(x)^{(\tilde{q}^d-1)/3} + 1)$$

einen nicht trivialen Faktor von $f \in \mathbb{F}_{\tilde{q}}[x]$. Genauso lassen sich durch die Berechnung von

$$\mathrm{ggT}(f(x), g(x)^{(\tilde{q}^d-1)/3} + \xi)$$

weitere Faktoren finden. Diejenigen, die durch die Berechnung von

$$\mathrm{ggT}(f(x), g(x)^{(\tilde{q}^d-1)/3} + \xi + 1)$$

erhalten werden, sind diejenigen, die nach den beiden vorhergehenden Versuchen noch im Restpolynom enthalten sind, zusätzlich zu denjenigen, deren $\gamma_i = 0$ ist. Da jedoch die Wahrscheinlichkeit, daß ein $\gamma_i = 0$ ist, nur t/q^d ist, spart man sich die Bildung des $\mathrm{ggT}(f, g^{(\tilde{q}^d-1)/3} + \xi + 1)$.

Die Wahrscheinlichkeit P, so einen nicht trivialen Faktor von f zu finden, läßt sich ähnlich wie in Satz 3.3 auf Seite 16 abschätzen durch $P \geq 1/2$, das Verfahren ist in der Komplexitätsklasse RP. Wenn sich keine neuen Faktoren mehr finden, werden alle bezüglich $\mathbb{F}_4$ zueinander konjugierten Faktoren von f miteinander multipliziert, womit eine Faktorisierung von f über $\mathbb{F}_q$ erreicht wird.

Praxis

Ich behaupte, daß Cantor und Zassenhaus ihren Ansatz nicht konsequent bis zu Ende ausgeführt haben. Entscheidend dafür, ob $\mathbb{F}_{q^d}$ eine multiplikative Untergruppe der Ordnung drei hat, ist nämlich nicht etwa die Frage, ob $s \equiv 0 \mod 2$, sondern ob $sd \equiv 0 \mod 2$ ist. Ich setze

- $\tilde{q} = q$ für $2^{sd} \equiv 1 \mod 3$ und
- $\tilde{q} = q^2$ für $2^{sd} \equiv 2 \mod 3$.

Ich muß den Körper $\mathbb{F}_q$ also nur erweitern, wenn sowohl s als auch d ungerade sind und spare mir darum die Körpererweiterung in der Hälfte der Fälle, in denen Cantor und Zassenhaus erweitern. Aber ich spare auch in der anderen Hälfte der Fälle: Ist d ungerade, so zerfällt ein Faktor f_i auch über dem Körper $\mathbb{F}_{q^2}$

nicht in zwei Faktoren vom Grad $d/2$, da $d/2 \notin \mathbb{N}$. Es ist also in keinem Fall notwendig, am Ende die Liste der Faktoren von f durchzugehen, konjugierte herauszufinden und diese dann zu multiplizieren.

Was hier allerdings schwieriger ist, das ist im Fall s ungerade und d gerade die Bestimmung eines Polynoms h, das in jeder Komponente entweder ξ oder $\xi+1$ ist. Man könnte zufällige Polynome $g(x) \in \mathbb{F}_q[x]$ wählen, diese mit $(2^{sd}-1)/3$ potenzieren und die Bedingung

$$\left(g^{(q^d-1)/3}\right)^2 \equiv g^{(q^d-1)/3}+1 \mod f$$

überprüfen. Ist sie erfüllt, so setze $h \equiv g^{(q^d-1)/3}$. Die Wahrscheinlichkeit, daß die Bedingung erfüllt ist, ist jedoch nur $\frac{2t(q^d-1)}{3}$, darum schlage ich vor, nur mit der Zerlegung von f in

$$f(x) = \mathrm{ggT}\left(f(x), g(x)^{(\tilde{q}^d-1)/3}+1\right) \cdot \frac{f(x)}{\mathrm{ggT}\left(f(x), g(x)^{(\tilde{q}^d-1)/3}+1\right)}$$

zu arbeiten. Dadurch verringert sich die Wahrscheinlichkeit P, in einem Schritt des Algorithmus eine partielle Faktorisierung von f zu finden, aber indem man zwei Schritte als einen ansieht, gelangt man wieder zu dem Ergebnis, daß das Verfahren in der Komplexitätsklasse RP ist.

Algorithmus 19 CANTOR UND ZASSENHAUS $(f, \mathbb{F}_{\tilde{q}}, d, t)$
Eingabe: Das normierte, quadratfreie Polynom $f \in \mathbb{F}_{\tilde{q}}[x]$ vom Grad n, welches in t Faktoren vom Grad d zerfällt, und der Körper $\mathbb{F}_{\tilde{q}}$.
Ausgabe: Die Faktorisierung von f.
Programm: CanZass.mws: canzass

1. Wähle ein zufälliges Polynom $g(x)$ mit $\mathrm{Grad}(g) < n$ und Koeffizienten in $\mathbb{F}_{\tilde{q}}$. Bestimme $g_1(x) \leftarrow g(x)^{(\tilde{q}^d-1)/3}$.
2. Ist $g_1(x)=0$, $g_1(x)=1$ oder $g_1(x)^2 = g_1(x)+1$, so gehe zu Schritt 1.
3. Setze $h_1(x) \leftarrow \mathrm{ggT}(f(x), g_1(x)+1)$, $h_2(x) \leftarrow \mathrm{ggT}(f(x), g_1(x))$, $h_3(x) \leftarrow f(x)/(h_1(x)h_2(x))$.
4. Bestimme für jedes $i \in \{1,2,3\}$ eine Ausgabe out_i: Ist $\mathrm{Grad}(h_i) = d$, so setze $out_i \leftarrow h_i(x)$, ist $\mathrm{Grad}(h_i) < d$, so setze $out_i \leftarrow 1$, sonst setze $out_i \leftarrow$ CANTOR UND ZASSENHAUS $(h_i, \mathbb{F}_{\tilde{q}}, d, n/d)$.
5. Liefere out_1, out_2 und out_3 zurück.

Beispiel 5.6
Dieses Beispiel versucht nicht, das sonst bei gerader Charakteristik von mir gewählte Polynom aus $\mathbb{F}_4$ zu faktorisieren, da ich an dieser Stelle natürlich den Ansatz vorführen möchte, den Cantor und Zassenhaus nicht bis zu Ende gedacht haben. Um den Ansatz zu illustrieren, muß $q \equiv 2 \mod 3$ sein, ich wähle $q=2$. Weiterhin soll das Polynom in irreduzible Faktoren von geradem Grad d zerfallen. Es gibt nur drei irreduzible Polynome in $\mathbb{F}_2[x]$ vom Grad vier, darum wähle ich stattdessen $d=6$. Ein Polynom, das den soeben geforderten Eigenschaften genügt, ist

$$f(x) = x^{18}+x^{17}+x^{15}+x^{11}+x^6+x^5+1 \in \mathbb{F}_2[x],$$

welches in drei Faktoren vom Grad sechs zerfällt.

Nun suche ich ein Polynom $y(x)$, welches der Eigenschaft

$$y(x)^2 \equiv y(x)+1 \mod f(x) \tag{8}$$

genügt. Um dieses Polynom zu finden, wähle ich zufällige Polynome $g \in \mathbb{F}_2[x]$, bilde $y = g^{(2^d-1)/3} = g^{21}$ und prüfe die geforderte Eigenschaft (8) nach. Ein mögliches y ist

$$y(x) = x^{17}+x^{13}+x^{10}+x^9+x^5+x^4+x^3+x^2+x+1.$$

Für jeden irreduziblen Faktor f_i von f sei in diesem Beispiel $\xi \equiv y(x) \mod f_i(x)$.

Um Faktoren von f zu finden, wähle ich ein zufälliges Polynom g mit einem Grad kleiner als 18:

$$g(x) = x^5 + x^2 + 1.$$

Berechne

$$\begin{aligned} g(x)^{21} &\equiv x^{17} + x^{13} + x^{10} + x^9 + x^5 + x^4 + x^3 + x^2 + x + 1 \mod f(x) \\ &\equiv y(x) \mod f(x) \end{aligned}$$

als Polynom, das nur die Komponenten $0, 1, \xi$ und $\xi + 1$ hat. Da $g^{21} = y$, ergibt sich so keine partielle Faktorisierung von f.

Das nächste zufällig gewählte Polynom sei

$$g(x) = x^5 + x^4 + x^3 + x,$$

berechne

$$g(x)^{21} \equiv x^{12} + x^{11} + x^8 + x^6 + x^5 + x^3 \mod f(x)$$

als Polynom, das nur die Komponenten $0, 1, \xi$ und $\xi + 1$ hat. Dann ist

$$\begin{aligned} \operatorname{ggT}\left(f(x), g(x)^{21}\right) &= 1, \\ h_1(x) = \operatorname{ggT}\left(f(x), g(x)^{21} + 1\right) &= x^{12} + x^{11} + x^8 + x^6 + x^5 + x^3 + 1, \\ \operatorname{ggT}\left(f(x), g(x)^{21} + y(x)\right) &= x^6 + x^3 + 1 \text{ und} \\ \operatorname{ggT}\left(f(x), g(x)^{21} + y(x) + 1\right) &= 1, \end{aligned}$$

rechne weiter mit $f(x) = h_1(x) \cdot (x^6 + x^3 + 1)$.

Wähle nun das zufällige Polynom

$$g(x) = x^5 + x^4 + x,$$

berechne

$$g(x)^{21} \equiv x^{17} + x^{14} + x^{13} + x^{11} + x^9 + x^8 + x^5 + x^4 + x^2 \mod f(x)$$

als Polynom, das nur die Komponenten $0, 1, \xi$ und $\xi + 1$ hat. Dann ist

$$\begin{aligned} \operatorname{ggT}\left(h_1(x), g(x)^{21}\right) &= 1, \\ \operatorname{ggT}\left(h_1(x), g(x)^{21} + 1\right) &= x^6 + x^5 + x^2 + x + 1, \\ \operatorname{ggT}\left(h_1(x), g(x)^{21} + y(x)\right) &= x^6 + x + 1 \text{ und} \\ \operatorname{ggT}\left(h_1(x), g(x)^{21} + y(x) + 1\right) &= 1, \end{aligned}$$

insgesamt ist also $f(x) = (x^6 + x^5 + x^2 + x + 1)(x^6 + x + 1)(x^6 + x^3 + 1)$.

Theorie mit der Spur für gerade Charakteristik

Das zweite Standardverfahren im Fall $p = 2$ ist bekanntlich, nicht etwa die Abbildung in die multiplikative Untergruppe der Ordnung drei zu benutzen, sondern die Spur.

Lemma 5.3
Zerfalle $f \in \mathbb{F}_q[x]$ für $q = 2^s$ in paarweise verschiedene irreduzible Faktoren vom Grad d und sei das Polynom $T \in \mathbb{F}_q[x]$ gegeben als

$$T(x) = \sum_{i=0}^{sd-1} x^{2^i} = \mathrm{Spur}_{\mathbb{F}_{q^d}/\mathbb{F}_2}(x).$$

Dann zerfällt f für jedes Polynom $g \in \mathbb{F}_q[x]$ in das Produkt

$$f(x) = \mathrm{ggT}\,(f, T(g)) \cdot \mathrm{ggT}\,(f, T(g) + 1)\,.$$

Beweis: Jede Komponente von $T(g)$ ist in der Menge $\{0, 1\}$ enthalten. Ist die i–te Komponente von $T(g)$ nun 0 für ein $1 \leq i \leq t$, so ist die i–te Komponente von $T(g) + 1$ gerade 1 und umgekehrt. Darum gilt für jedes $1 \leq i \leq t$ entweder $f_i \mid T(g)$ oder $f_i \mid (T(g) + 1)$. □

Die Wahrscheinlichkeit P für eine nicht triviale Faktorisierung von f ist

$$P = 1 - 2 \cdot \left(\frac{1}{2}\right)^t = 1 - \left(\frac{1}{2}\right)^{t-1},$$

das Verfahren ist in der Komplexitätsklasse RP.

Praxis

Der sich aus dieser Theorie ableitende Algorithmus wählt also zufällige Polynome $g \in \mathbb{F}_q[x]$ und versucht, f durch Bilden von $\mathrm{ggT}\,(f, T(g))$ in all seine irreduziblen Faktoren vom Grad d aufzuspalten. Diese Vorgehensweise kann man verbessern, indem man bemerkt, daß hier mit dem Verfahren von McEliece aus Abschnitt 4.2 auf Seite 38 Elemente der Berlekamp–Unteralgebra $\mathcal{B}(f, 2)$ erzeugt werden. Dann ist nämlich klar, daß man die Polynome $g \in \mathbb{F}_q[x]$ nicht zufällig wählen sollte, sondern daß die Menge $\{T(x^{2i+1}) \mid 0 \leq i < d\}$ eine aufspaltende Menge für f ist und wahrscheinlich schneller zu einer totalen Faktorisierung führt als zufällige Polynome. Im Zusammenhang mit dem Verfahren von Cantor und Zassenhaus ist auf diese Idee als erster H. Cohen in [10] gekommen.

Algorithmus 20 CANTOR UND ZASSENHAUS $(f, \mathbb{F}_q, d, t)$
Eingabe: Das normierte, quadratfreie Polynom $f \in \mathbb{F}_q[x]$ vom Grad n, welches in t Faktoren vom Grad d zerfällt, und der Körper $\mathbb{F}_q$ mit $q = 2^s$.
Ausgabe: Die Faktorisierung von f.
Programm: CanZass.mws: canzass

1. Setze $T(x) \leftarrow \sum_{i=0}^{sd-1} x^{2^i}$.
2. Wähle ein zufälliges Polynom $g(x)$ mit $\mathrm{Grad}(g) < n$ und Koeffizienten in $\mathbb{F}_q$. Bestimme $g_1(x) \leftarrow T(g(x))$.
3. Ist $\mathrm{Grad}(g_1) < 1$, so gehe zu Schritt 2.
4. Setze $h_1(x) \leftarrow \mathrm{ggT}(f(x), g_1(x) + 1)$ und $h_2(x) \leftarrow \mathrm{ggT}(f(x), g_1(x))$.
5. Bestimme für jedes $i \in \{1, 2\}$ eine Ausgabe out_i: Ist $\mathrm{Grad}(h_i) = d$, so setze $out_i \leftarrow h_i(x)$, sonst setze $out_i \leftarrow$ CANTOR UND ZASSENHAUS $(h_i, \mathbb{F}_q, d, n/d)$.
6. Liefere out_1 und out_2 zurück.

Beispiel 5.7
Das Beispiel, das an dieser Stelle gerechnet werden müßte, ist das Beispiel 4.6 auf Seite 43.

Komplexität

Ich bestimme die Komplexität exemplarisch für ungerade Charakteristik p.

Der Algorithmus von Cantor und Zassenhaus bestimmt in jedem Schritt ein zufälliges Polynom. Dieses wird mit $(q^d-1)/2$ potenziert, wozu $(\log_2(q^d-1)-1)\cdot(3n^2+7n+4)$ elementare Operationen gebraucht werden. Dann werden höchstens $t-1$ größte gemeinsame Teiler bestimmt, was in $(t-1)(2n^2+3n)$ Operationen abgehandelt wird. Ein Schritt kostet also

$$\begin{aligned}
&(\log_2(q^d-1)-1)\cdot(3n^2+7n+4)+(t-1)(2n^2+3n)\\
&\quad < t(2n^2+3n)+\big((3d\log_2(q))n^2+(7d\log_2(q))n+4d\log_2(q)\big)\\
&\quad \leq t(2n^2+3n)+\big((3d\log_2(q))n^2+(7d\log_2(q))n+4d\log_2(q)\big)\,t\\
&\quad = t\,\big((3d\log_2(q)+2)n^2+(7d\log_2(q)+3)n+4d\log_2(q)\big)
\end{aligned}$$

Operationen, ich schätze darum die obere Schranke für die in einem Schritt auszuführenden Operationen mit der letzten Zeile. Die erwartete Rekursionstiefe D des Algorithmus wird mit dem Modell der Bälle und Tonnen auf $E(D)=\frac{2\log_2(t)}{\log_2(1/w)}+1+\frac{1}{1-w}$ geschätzt, dabei werden die t irreduziblen Faktoren von f mit den Bällen identifiziert, die Reste $0, 1$ und -1 mit den drei Tonnen. w ist $\frac{q^d-1}{2q^d}$, der Erwartungswert läßt sich also umschreiben auf $E(D)=\frac{2\log_2(t)}{\log_2(2q^d/(q^d-1))}+1+\frac{2q^d}{q^d+1}$. Die Abschätzungen $\frac{1}{\log_2(2q^d/(q^d-1))}<1$ und $\frac{2q^d}{q^d+1}<2$ sind bereits bekannt, nun ist $E(D)<2\log_2(t)+3$. Die Kostenfunktion $\operatorname{kost}(t)$ ist die obere Schranke der in einem Schritt durchzuführenden Operationen, es gilt $\operatorname{kost}(at)=a\operatorname{kost}(t)$ für alle $a\geq 0$, $t\geq 1$, darum summieren sich die in einer Rekursionsschicht durchzuführenden Operationen höchstens zu $\operatorname{kost}(t)$ auf. Die geschätzte Anzahl an Operationen des Algorithmus von Cantor und Zassenhaus wird nun auf weniger als $\operatorname{kost}(t)\cdot E(D)=t\,\big((3d\log_2(q)+2)n^2+(7d\log_2(q)+3)n+4d\log_2(q)\big)\,(2\log_2(t)+3)$ geschätzt, somit liegt die Komplexität wegen $n=d\cdot t$ bei $\mathcal{O}\left(n^2dt\log(t)\log(q)\right)=\mathcal{O}\left(n^3\log(t)\log(q)\right)$.

5.4 Der Algorithmus von Moenck

Dieses deterministische Verfahren, das zu den Elementen b einer aufspaltenden Menge B zuerst die Menge C derjenigen $\alpha\in\mathbb{F}_q$ bestimmt, für die $\operatorname{ggT}(f(x), b(x)-\alpha)$ nicht trivial ist, wurde 1977 von R. T. Moenck in [37] beschrieben. Das Verfahren funktioniert ganz ähnlich wie das von H. Zassenhaus, auf welches ich in Abschnitt 5.5 auf Seite 66 näher eingehe.

Um das Verfahren entwickeln zu können, muß ich zuerst den Begriff der Resultante definieren.

Definition 5.4
Seien $g(x)=g_0+g_1x+\cdots+g_{n_1}x^{n_1}\in\mathbb{F}_q[x]$ und $h(x)=h_0+h_1x+\cdots+h_{n_2}x^{n_2}\in\mathbb{F}_q[x]$ zwei Polynome mit $n_1, n_2\in\mathbb{N}$ und $g_{n_1}=0$ und $h_{n_2}=0$ zugelassen. Die Matrix A sei definiert als

$$A=\begin{array}{l}\left[\begin{array}{cccccccc}
g_0 & g_1 & \dots & g_{n_1} & 0 & & \dots & 0\\
0 & g_0 & g_1 & \dots & g_{n_1} & 0 & \dots & 0\\
\vdots & & \ddots & \ddots & & & & \vdots\\
0 & \dots & 0 & g_0 & g_1 & & \dots & g_{n_1}\\
h_0 & h_1 & \dots & h_{n_2} & 0 & & \dots & 0\\
0 & h_0 & h_1 & \dots & h_{n_2} & 0 & \dots & 0\\
\vdots & & \ddots & \ddots & & & & \vdots\\
0 & \dots & 0 & h_0 & h_1 & & \dots & h_{n_2}
\end{array}\right]\end{array}
\begin{array}{l}\left.\vphantom{\begin{array}{c}1\\1\\1\\1\end{array}}\right\}n_2\text{ Zeilen}\\ \left.\vphantom{\begin{array}{c}1\\1\\1\\1\end{array}}\right\}n_1\text{ Zeilen}\end{array}.$$

Dann ist die Resultante $R(g,h)$ der beiden Polynome definiert als die Determinante von A:

$$R(g,h)=\det(A).$$

Ist $g_{n_1} \neq 0$ und ist $g(x) = g_{n_1}(x-\alpha_1)\cdots(x-\alpha_{n_1})$ in dem Zerfällungskörper von g über $\mathbb{F}_q$, so ist die Resultante $R(g,h)$ auch gegeben durch die bei Lidl und Niederreiter in [31] nachzuschlagende Formel

$$R(g,h) = (g_{n_1})^{n_2} \prod_{i=1}^{n_1} h(\alpha_i).$$

Für $g_{n_1} \neq 0$ gilt darum $R(g,h) = 0$ genau dann, wenn $\mathrm{ggT}(g,h) \neq 1$ ist.

Nun kann ich das eigentliche Faktorisierungsverfahren von Moenck entwickeln. f sei quadratfrei, ein Erzeugendensystem B der Berlekamp–Unteralgebra $\mathcal{B}(f,r)$ sei bekannt. Ich suche für ein $b \in B$, die Menge C der Elemente $\alpha \in \mathbb{F}_r$, für die $\mathrm{ggT}(f, b(x)-\alpha)$ keinen trivialen Faktor liefert, für die also $\mathrm{ggT}(f, b(x)-\alpha) \neq 1$ ist. Mit der Theorie über Resultanten ist C genau die Menge der Nullstellen des Polynoms $F(y) \in \mathbb{F}_q[y]$, das durch

$$F(y) = R(f(x), b(x) - y)$$

mit $\mathrm{Grad}(F) \leq n$ gegeben ist.

Da Konstanten $b(x) \in \mathbb{F}_r$ nichts zur Faktorisierung von f beitragen, nehme ich sie aus dem Erzeugendensystem B heraus. Die Menge C ist mit einem der Verfahren zum Finden der Nullstellen aus Abschnitt 3 auf Seite 15 leicht zu finden.

Algorithmus 21 MOENCK $(f, \mathbb{F}_q)$
Eingabe: Das normierte, quadratfreie Polynom $f \in \mathbb{F}_q[x]$ und der Körper $\mathbb{F}_q$.
Ausgabe: Die Faktorisierung von f.
Programm: Moenck.mws: moenck

1. Bestimme eine Basis $B = \{b_1(x), \ldots, b_t(x)\}$ der Berlekamp–Unteralgebra $\mathcal{B}(f,r)$, streiche daraus das Basiselement 1. Setze $i \leftarrow 1$.
2. Bestimme die Nullstellen $\alpha_1, \ldots, \alpha_k$ des Polynoms $R(f(x), b_i(x) - y)$, setze $j \leftarrow 1$.
3. Bestimme für alle gefundenen Faktoren $g(x)$ das Polynom $h(x) = \mathrm{ggT}(g(x), b_i(x) - \alpha_j)$ und rechne weiter mit den Faktoren $g(x) = h(x) \cdot g(x)/h(x)$.
4. Ist die Anzahl der bisher gefundenen Faktoren von f kleiner als t und ist $j < k-1$, so setze $j \leftarrow j+1$ und gehe zu Schritt 3.
5. Ist die Anzahl der bisher gefundenen Faktoren von f kleiner als t, so setze $i \leftarrow i+1$ und gehe zu Schritt 2.
6. Gib die Faktorisierung von $f(x)$ zurück.

Beispiel 5.8
Betrachte

$$f(x) = x^{12} + 4x^{11} + 2x^{10} + 2x^9 + 6x^6 + 3x^4 + 3x^3 + 4x^2 + 5 \in \mathbb{F}_7[x],$$

wir wissen nach der Aufspaltung in gleichgradige Faktoren, daß f in vier Faktoren vom Grad drei zerfällt. Des weiteren ist uns aus Beispiel 4.1 auf Seite 33 die Menge $\{b_1, b_2, b_3, b_4\}$ mit

$$\begin{aligned}
b_1(x) &= 1, \\
b_2(x) &= 2x^2 + x^3 + 3x^4 + 3x^5 + 4x^6 + 6x^7 + 5x^8 + x^9, \\
b_3(x) &= 3x + 4x^2 + 2x^3 + x^4 + 5x^5 + 5x^6 + x^7 + x^8 + x^{10}, \\
b_4(x) &= 6x + x^2 + 2x^3 + 6x^5 + 3x^6 + 4x^7 + 5x^8 + x^{11}
\end{aligned}$$

als Basis der Berlekamp–Unteralgebra bekannt. 1 trägt nicht zur Faktorisierung von f bei, bestimme darum die Menge C der $\alpha \in \mathbb{F}_7$, für die $\mathrm{ggT}(f, b_2(x) - \alpha)$ nicht einen trivialen Faktor liefert. Zu diesem Zweck *berechne ich*

$$\begin{aligned} R(f(x), b_2(x) - y) &= y^{12} + 6y^{11} + 6y^{10} + 3y^8 + 3y^7 + 3y^6 + 6y^5 + 2y^4 + 6y^3 \\ &= (y-4)^3(y-5)^3y^3(y-3)^3 \end{aligned}$$

mit dem Verfahren von Berlekamp und Rabin zum Finden von Nullstellen. Somit lassen sich durch die Berechnung von $\mathrm{ggT}(f, b_2(x) - \alpha)$ *mit* $\alpha \in \{4, 5, 0, 3\}$ *Faktoren von* f *finden:*

- *Es ist*

  $$\begin{aligned} h_1(x) = \mathrm{ggT}(f, b_2(x) - 5) &= x^3 + 6x^2 + 4x + 4, \\ h_2(x) = f(x)/h_1(x) &= x^9 + 5x^8 + 3x^7 + 2x^6 + 5x^5 + 6x^4 + 5x^3 + 3x^2 + 4x + 3, \end{aligned}$$

  rechne weiter mit $f(x) = (x^3 + 6x^2 + 4x + 4) \cdot h_2(x)$.

- *Es ist*

  $$\begin{aligned} h_3(x) = \mathrm{ggT}(h_2(x), b_2(x) - 4) &= x^3 + 6x^2 + 4x + 1, \\ h_4(x) = h_2(x)/h_3(x) &= x^6 + 6x^5 + 5x^4 + 3x^3 + 3x^2 + 6x + 3, \end{aligned}$$

  rechne weiter mit $f(x) = (x^3 + 6x^2 + 4x + 4)(x^3 + 6x^2 + 4x + 1) \cdot h_4(x)$.

- *Es ist* $\mathrm{ggT}(h_4(x), b_2(x)) = x^3 + 6x^2 + 5x + 6$.

Insgesamt ist also $f(x) = (x^3 + 6x^2 + 4x + 4)(x^3 + 6x^2 + 4x + 1)(x^3 + 6x^2 + 5x + 6)(x^3 + 4)$.

Komplexität

Ist x die Anzahl der Operationen zur Berechnung der Determinante einer $(k-1 \times k-1)$–Matrix, so ist die Anzahl bei einer $(k \times k)$–Matrix gerade $k \cdot (x+2) - 1$, da k mal eine $(k-1 \times k-1)$–Determinante berechnet werden muß, die Ergebnisse mit einem Körperelement multipliziert und dann zusammengezählt werden müssen. Eine mutige Abschätzung liefert für die Determinantenberechnung die Komplexität $\mathcal{O}(n^n)$, da die Anzahl der Operationen schneller als $n!$ wächst. Durch diese alles überschattende Größe n^n braucht man sich um Nichtigkeiten wie polynomiales Wachstum nicht mehr zu bemühen, die Komplexität des Algorithmus von Moenck steht fest als $\mathcal{O}(n^n)$, das Verfahren ist also denkbar schlecht. Ich habe es in meine Arbeit aufgenommen, da man es sehr gut auf das Verfahren von Niederreiter aus Abschnitt 6 auf Seite 74 übertragen kann und da es eine Vorstufe zu dem folgenden Verfahren von Zassenhaus ist.

5.5 Der Algorithmus von Zassenhaus

H. Zassenhaus hatte 1969 in dem Artikel [64] als erster die Idee, die Anzahl der zu berechnenden größten gemeinsamen Teiler aus Gleichung (5) zu reduzieren, indem wie in Abschnitt 5.4 zu jedem nicht konstanten b aus einer aufspaltenden Menge B zuerst die Menge C der $\alpha \in \mathbb{F}_q$ bestimmt wird, für die der größte gemeinsame Teiler $\mathrm{ggT}(f(x), b(x) - \alpha)$ nicht trivial ist. Das Verfahren ist deterministisch.

Theorie

Ich betrachte ein $b(x) \in \mathcal{B}(f, r)$ mit der Komponentendarstellung $(c_1, \ldots, c_t)$. C sei die Menge der $\alpha \in \mathbb{F}_q$, für die $\mathrm{ggT}(f(x), b(x) - \alpha) \neq 1$ ist, also ist $C = \{c_i \mid 1 \leq i \leq t\}$, wobei $c_i = c_j$ für $1 \leq i, j \leq t$ möglich ist.

Nach Gleichung (5) ist

$$f(x) = \prod_{\alpha \in C} \mathrm{ggT}\left(f(x), b(x) - \alpha\right),$$

also wird $\prod_{\alpha \in C} (b(x) - \alpha)$ von $f(x)$ geteilt. Ich definiere das Polynom

$$G(y) = \prod_{\alpha \in C} (y - \alpha),$$

dann wird also $G(b(x))$ von $f(x)$ geteilt.

Lemma 5.5
$G(y)$ ist das normierte Polynom kleinsten Grades in $\mathbb{F}_q[y]$ derart, daß gilt $f(x) \mid G(b(x))$.

Beweis: Die Menge der Polynome $h(y) \in \mathbb{F}_q[y]$ derart, daß $h(b(x))$ von $f(x)$ geteilt wird, ist ein Hauptideal von $\mathbb{F}_q[y]$. Es werde von dem normierten Polynom $g(y)$ erzeugt, dann wird $G(y)$ von $g(y)$ geteilt. Ist g gegeben durch $g(y) = \sum_{j=0}^{k} g_j y^j$ und entspricht $b(x)$ in der Komponentendarstellung dem Vektor $(c_1, \ldots, c_t)$, so entspricht $g(b(x))$ in der Komponentendarstellung dem Vektor $(g(c_1), \ldots, g(c_t))$. Da $f(x) \mid g(b(x))$, so gilt auch $f_i(x) \mid g(b(x))$ für alle $1 \leq i \leq t$. Darum ist $g(c_i) = 0$ für alle $1 \leq i \leq t$. Da $g(y)$ und $G(y)$ dieselben Nullstellen haben, ist also $g(y) = G(y)$. □

G hat höchstens t Nullstellen. Ist $\mathrm{Grad}(G) = k$, so ist

$$G(y) = \prod_{\alpha \in C} (y - \alpha) = \sum_{j=0}^{k} g_j y^j,$$

wobei $g_j \in \mathbb{F}_q$ für alle $1 \leq j \leq k$.

Ich bestimme also G mit den folgenden Eigenschaften:

1. k ist der kleinste Wert, für den $1, b(x), b(x)^2, \ldots, b(x)^k$ über $\mathbb{F}_q$ linear abhängig sind modulo $f(x)$, das heißt $g_0 + g_1 b(x) + g_2 b(x)^2 + \cdots + g_k b(x)^k = 0$ mit Koeffizienten g_i, die nicht alle 0 sind. 0 soll durch $1, b(x), b(x)^2, \ldots, b(x)^{k-1}$ nicht darstellbar sein.

2. Setze dann $G(y) = \sum_{j=0}^{k} g_j y^j$.

Ich berechne mit den Methoden aus Abschnitt 3 auf Seite 15 nacheinander zu jedem $b(x)$ aus dem Erzeugendensystem B der Berlekamp–Unteralgebra $\mathcal{B}(f, r)$ die Nullstellen α von $G(y)$ und brauche nur noch mit ihnen den größten gemeinsamen Teiler $\mathrm{ggT}(f(x), b(x) - \alpha)$ zu bestimmen. Ich wende dieses Verfahren so lange an, bis ich f in alle seine t Faktoren zerlegt habe — mit sehr viel Glück klappt das gleich mit dem ersten $b(x)$.

Praxis

Die konstanten Polynome $b(x) \in \mathbb{F}_r[x]$ streiche ich aus B, da sie nichts zur Faktorisierung von f beitragen.

Um für ein $b(x)$ aus dem Erzeugendensystem B der Berlekamp–Unteralgebra $\mathcal{B}(f, r)$ das k und die g_i mit $1 \leq i \leq k$ zu bestimmen, bilde ich für $0 \leq i \leq t$ sukzessive die Polynome

$$b(x)^i \equiv \sum_{j=0}^{n-1} \beta_{j,i} x^j \mod f(x)$$

und schreibe die Koeffizienten in die $(n \times (i+1))$–Matrix A:

$$A = \begin{bmatrix} \beta_{0,0} & \cdots & \beta_{0,i} \\ \beta_{1,0} & \cdots & \beta_{1,i} \\ \vdots & & \vdots \\ \beta_{n-1,0} & \cdots & \beta_{n-1,i} \end{bmatrix}.$$

Dann bestimme ich den Nullraum von A. Ist dieser $\{0\}$, so fahre ich mit dem Verfahren fort, bis ich einen Vektor g erhalte, der durch Anwendung von A zu 0 wird. Nun enthält g die Koeffizienten des Polynoms $G(y)$ und es gilt $k = i$.

Mit den Methoden aus Abschnitt 3 bestimme ich die Nullstellen von G, welche die Menge C bilden.

Algorithmus 22 ZASSENHAUS $(f, \mathbb{F}_q)$
Eingabe: Das normierte, quadratfreie Polynom $f \in \mathbb{F}_q[x]$ und der Körper $\mathbb{F}_q$.
Ausgabe: Die Faktorisierung von f.
Programm: Zassenhaus.mws: zass

1. Bestimme eine Basis $B = \{b_1(x), \ldots, b_t(x)\}$ der Berlekamp–Unteralgebra $\mathcal{B}(f, r)$, streiche daraus das Basiselement 1. Setze $i \leftarrow 1$.
2. Setze $k \leftarrow 1$, $b(x) \leftarrow 1$ und A eine Matrix mit n Zeilen und einer Spalte. Trage in die Spalte von A die Koeffizienten von $b(x)$ ein.
3. Setze $b(x) \leftarrow b(x) \cdot b_i(x)$ und klebe eine Spalte mit den Koeffizienten von $b(x)$ rechts an die Matrix A an.
4. Bestimme den Nullraum N von A. Ist $N = \{0\}$, so setze $k \leftarrow k + 1$ und gehe zu Schritt 3, sonst identifiziere mit dem Basisvektor von N das Polynom $G(y)$.
5. Bestimme die Nullstellen $\alpha_1, \ldots, \alpha_k$ des Polynoms $G(y)$, setze $j \leftarrow 1$.
6. Bestimme für alle gefundenen Faktoren $g(x)$ das Polynom $h(x) = \mathrm{ggT}(g(x), b_i(x) - \alpha_j)$ und rechne weiter mit den Faktoren $g(x) = h(x) \cdot g(x)/h(x)$.
7. Ist die Anzahl der bisher gefundenen Faktoren von $f(x)$ kleiner als t und ist $j < k - 1$, so setze $j \leftarrow j + 1$ und gehe zu Schritt 6.
8. Ist die Anzahl der bisher gefundenen Faktoren von $f(x)$ kleiner als t, so setze $i \leftarrow i+1$ und gehe zu Schritt 2.
9. Gib die Faktorisierung von $f(x)$ zurück.

Beispiel 5.9
Betrachte das Polynom

$$f(x) = x^{12} + 4x^{11} + 2x^{10} + 2x^9 + 6x^6 + 3x^4 + 3x^3 + 4x^2 + 5 \in \mathbb{F}_7[x].$$

Uns ist aus Beispiel 4.1 auf Seite 33 die Menge $\{b_1, b_2, b_3, b_4\}$ *mit*

$$\begin{aligned}
b_1(x) &= 1,\\
b_2(x) &= 6x + x^2 + 2x^3 + 6x^5 + 3x^6 + 4x^7 + 5x^8 + x^{11},\\
b_3(x) &= 2x^2 + x^3 + 3x^4 + 3x^5 + 4x^6 + 6x^7 + 5x^8 + x^9 \text{ und}\\
b_4(x) &= 3x + 4x^2 + 2x^3 + x^4 + 5x^5 + 5x^6 + x^7 + x^8 + x^{10}
\end{aligned}$$

als Basis der Berlekamp–Unteralgebra bekannt, f zerfällt also in vier irreduzible Faktoren. 1 trägt nicht zur Faktorisierung von f bei, bestimme darum die Menge C der $\alpha \in \mathbb{F}_7$, für die $\mathrm{ggT}(f, b_2(x) - \alpha)$ nicht einen trivialen Faktor liefert. Zu diesem Zweck berechne ich nacheinander

- $b_2^0 = 1$, *die Matrix A ist also vorerst*

  $$A = [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]^\mathsf{T}$$

  und hat den Nullraum $\{0\}$.

- $b_2^1 = b_2$, *die Matrix A ist also*

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6 & 1 & 2 & 0 & 6 & 3 & 4 & 5 & 0 & 0 & 1 \end{bmatrix}^\top$$

  mit dem Nullraum $\{0\}$.

- $b_2(x)^2 = 5 + 6x^2 + 2x^3 + x^4 + 5x^6 + 5x^7 + 6x^8 + 3x^9 + 6x^{10} + 4x^{11}$, *die Matrix A ist also*

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6 & 1 & 2 & 0 & 6 & 3 & 4 & 5 & 0 & 0 & 1 \\ 5 & 0 & 6 & 2 & 1 & 0 & 5 & 5 & 6 & 3 & 6 & 4 \end{bmatrix}^\top$$

  mit dem Nullraum $\{0\}$.

- $b_2(x)^3 = 5 + x + 5x^2 + x^4 + x^5 + 2x^6 + x^7 + x^8 + 3x^9 + 6x^{10} + 3x^{11}$, *die Matrix A ist also*

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6 & 1 & 2 & 0 & 6 & 3 & 4 & 5 & 0 & 0 & 1 \\ 5 & 0 & 6 & 2 & 1 & 0 & 5 & 5 & 6 & 3 & 6 & 4 \\ 5 & 1 & 5 & 0 & 1 & 1 & 2 & 1 & 1 & 3 & 6 & 3 \end{bmatrix}^\top$$

  mit dem Nullraum, der von dem Vektor $(0, 1, 6, 1)^\top$ *aufgespannt wird.*

Das Polynom $G(y)$ ist nun gegeben als $y^3 + 6y^2 + y = (x - 5)(x)(x - 3)$, hat also die Nullstellen 5, 0 und 3. Benutze zwei der Nullstellen zur Zerlegung von f:

- *Es ist* $h_1(x) = \mathrm{ggT}(f, b_2(x) - 5) = x^6 + 6x^5 + 4x^4 + x^3 + 3x^2 + 2x + 2$, *rechne weiter mit* $f(x) = h_1(x) \cdot (x^6 + 5x^5 + 3x^4 + 5x^3 + 6x^2 + x + 6)$.
- $\mathrm{ggT}(x^6 + 5x^5 + 3x^4 + 5x^3 + 6x^2 + x + 6, b_2(x)) = x^3 + 6x^2 + 4x + 1$, *rechne weiter mit* $f(x) = h_1(x) \cdot (x^3 + 6x^2 + 4x + 1)(x^3 + 6x^2 + 5x + 6)$.

Wir haben erst zwei Faktoren vom Grad drei gefunden, darum bestimme ich nun die Menge C der $\alpha \in \mathbb{F}_7$, für die $\mathrm{ggT}(f, b_3(x) - \alpha)$ nicht einen trivialen Faktor liefert. Zu diesem Zweck berechne ich nacheinander

- $b_3^0 = 1$, *die Matrix A ist also vorerst* $A = [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]^\top$ *und hat den Nullraum* $\{0\}$.
- $b_3^1 = b_3$, *die Matrix A ist also*

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 & 3 & 3 & 4 & 6 & 5 & 1 & 0 & 0 \end{bmatrix}^\top$$

  mit dem Nullraum $\{0\}$.

- $b_3(x)^2 = 4 + 6x + x^2 + 2x^3 + 6x^5 + 3x^6 + 4x^7 + 5x^8 + 1x^{11}$, *die Matrix A ist also*

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 & 3 & 3 & 4 & 6 & 5 & 1 & 0 & 0 \\ 4 & 6 & 1 & 2 & 0 & 6 & 3 & 4 & 5 & 0 & 0 & 1 \end{bmatrix}^\top$$

  mit dem Nullraum $\{0\}$.

- $b_3(x)^3 = 4 + 3x + 5x^2 + 2x^3 + 2x^4 + 2x^5 + x^7 + 6x^8 + 4x^9 + 4x^{10} + 2x^{11}$, *die Matrix A ist also*

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 & 3 & 3 & 4 & 6 & 5 & 1 & 0 & 0 \\ 4 & 6 & 1 & 2 & 0 & 6 & 3 & 4 & 5 & 0 & 0 & 1 \\ 4 & 3 & 5 & 2 & 2 & 2 & 0 & 1 & 6 & 4 & 4 & 2 \end{bmatrix}^\top$$

  mit dem Nullraum $\{0\}$.

- $b_3(x)^4 = 6x + 4x^3 + x^4 + 6x^5 + x^6 + 2x^7 + 4x^8 + 3x^9 + 6x^{10} + 5x^{11}$, *die Matrix A ist also*

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 & 3 & 3 & 4 & 6 & 5 & 1 & 0 & 0 \\ 4 & 6 & 1 & 2 & 0 & 6 & 3 & 4 & 5 & 0 & 0 & 1 \\ 4 & 3 & 5 & 2 & 2 & 2 & 0 & 1 & 6 & 4 & 4 & 2 \\ 0 & 6 & 0 & 4 & 1 & 6 & 1 & 2 & 4 & 3 & 6 & 5 \end{bmatrix}^{\top}$$

  mit dem Nullraum, der von dem Vektor $(0, 3, 5, 2, 1)^{\top}$ aufgespannt wird.

Das Polynom $G(y)$ ist nun gegeben als $y^4 + 2y^3 + 5y^2 + 3y = (x-4)(x-5)(x)(x-3)$, hat also die Nullstellen $4, 5, 0$ und 3. Benutze höchstens drei der Nullstellen zur Zerlegung von $h_1(x)$:

$$\mathrm{ggT}(h_1(x), b_3(x) - 4) = x^3 + 6x^2 + 4x + 4$$

Somit ist $f(x) = (x^3 + 6x^2 + 4x + 4)(x^3 + 4)(x^3 + 6x^2 + 4x + 1)(x^3 + 6x^2 + 5x + 6)$.

Komplexität

Man bestimmt die Basis der Berlekamp–Unteralgebra in $(2\log_2(q) + n)(3n^2 + 7n + 4) + 2n^3 - 2n^2$ Operationen. Dann wird für die t Basispolynome die Menge C bestimmt.

In der Menge C können höchstens t Elemente sein, darum kann G höchstens den Grad t haben. Dann ist A jeweis höchstens eine $(n \times (t+1))$–Matrix.

Um für ein Basiselement $b \neq 1$ die Menge C zu bestimmen, werden also höchstens $t + 1$ Schrittte zur Bestimmung von G durchgeführt. Jeder dieser Schritte besteht aus der Multiplikation zweier Polynome und einer Gaußelimination.

Für die Multiplikation wird in $(n+1)^2 = n^2 + 2n + 1$ elementaren Operationen jeder Koeffizient des einen Polynoms mit jedem Koeffizienten des anderen multipliziert, danach werden $n^2 + n$ Additionen durchgeführt, Kostenpunkt pro Multiplikation zweier Polynome ist also $2n^2 + 3n + 1$ Operationen.

Die Gaußelimination addiert die erste Zeile mit gewissen Vielfachen der $n - 1$ folgenden Zeilen, dann die zweite Zeile mit den $n - 2$ folgenden und so weiter. Dann passiert dieselbe Prozedur noch einmal mit Startpunkt in der letzten Zeile, dann der vorletzten und so weiter. Es werden also $n^2 - n$ Additionen zweier Zeilen gemacht, jede davon kostet $2t$ elementare Operationen. Für jede Gaußelimination sind also $2t(n^2 + n)$ Operationen zu veranschlagen.

Pro Basiselement b braucht man also höchstens $2tn^2 + 3tn + t + 2t^2(n^2 + n)$ elementare Operationen, um G zu bestimmen.

Um G zum Beispiel mit Hilfe des Verfahrens von Berlekamp und Rabin in seine linearen Faktoren zu zerlegen, wird die Komplexität $\mathcal{O}(n^2 \log(n) \log(q))$ erreicht. Sind die linearen Faktoren bekannt, müssen noch höchstens t größte gemeinsame Teiler berechnet werden, es werden also höchstens noch $t(2n^2 + 3n)$ Operationen durchgeführt.

Somit hat der Algorithmus von Zassenhaus die Komplexität $\mathcal{O}(n^3 + t^3 n^2 + tn^2 \log(n) \log(q))$.

5.6 Der Algorithmus von Shoup

Dies ist eine weitere deterministische Methode zur Faktorisierung von Polynomen über endlichen Körpern, die 1994 von V. Shoup in [52] entwickelt wurde. Sie greift nur bei der Entwicklung der Theorie, nicht in der Implementation auf die Berlekamp–Unteralgebra zu und ist darum schnell.

Auch hier sei f normiert und quadratfrei und es sei bekannt, daß alle irreduziblen Faktoren von f den Grad d haben. f sei gegeben durch $f(x) = \sum_{i=0}^{n} a_i x^i = \prod_{i=1}^{t} f_i(x)$.

Theorie

Ich bestimme das Polynom $H(y) = \prod_{i=0}^{d-1}(y - x^{q^i})$ modulo f und definiere für $0 \leq i < d$ die Polynome $g_i(x)$ durch die Gleichung

$$H(y) \equiv \left(\sum_{i=0}^{d-1} g_i(x) y^i\right) + y^d \mod f(x).$$

Lemma 5.6
Die Polynome g_i sind für $0 \leq i < d$ in der Berlekamp–Unteralgebra $\mathcal{B}(f, q)$.

Beweis: Es ist

$$\begin{aligned}
\prod_{i=0}^{d-1}(y - x^{q^i q}) &= \prod_{i=1}^{d}(y - x^{q^i}) \\
&= (y - x^{q^d}) \cdot \prod_{i=1}^{d-1}(y - x^{q^i}) \\
&\equiv (y - x) \cdot \prod_{i=1}^{d-1}(y - x^{q^i}) \mod f(x) \\
&= \prod_{i=0}^{d-1}(y - x^{q^i}) \\
&= H(y),
\end{aligned}$$

da x^{q^i} als Element von $\mathbb{F}_{q^d}$ betrachtet werden kann für alle $0 \leq i < d$ und also $x^{q^d} \equiv x \mod f_i(x)$. Somit ist

$$H(y) \equiv \left(\sum_{i=0}^{d-1} g_i(x)^q y^i\right) + y^d \mod f(x),$$

womit klar ist, daß $g_i(x)^q \equiv g_i(x) \mod f(x)$ für alle $0 \leq i < d$ gilt. □

Lemma 5.7
Die Menge $\{g_0, g_1, \ldots, g_{d-1}\}$ ist eine aufspaltende Menge für f.

Beweis: Sei für $0 \leq i < d$ die Komponentendarstellung von g_i gegeben als $(\alpha_1^{(i)}, \alpha_2^{(i)}, \ldots, \alpha_t^{(i)})$. Dann ist zu zeigen, daß es für alle $1 \leq k < l \leq t$ ein $i \in \{0, 1, \ldots, d-1\}$ gibt, so daß $\alpha_k^{(i)} \neq \alpha_l^{(i)}$.
Sei θ_i der natürliche Homomorphismus von $\mathcal{R}$ nach $\mathcal{R}^{(i)}$ (siehe dazu auf Seite 29), der durch

$$\theta_i : b(x) \mapsto b(x) \mod f_i(x)$$

definiert ist. Dann ist für alle $0 \leq j < d$

$$f_i\left(\theta_i(x)^{q^j}\right) = \theta_i\left(f_i(x)^{q^j}\right) = 0,$$

die $\theta_i(x)^{q^j}$ sind für $0 \leq j < d$ die konjugierten Nullstellen von f_i. Darum ist

$$\theta_i\left(H(y)\right) = \prod_{j=0}^{d-1}\left(y - \theta_i(x)^{q^j}\right) = f_i(y).$$

Da also $H(y) \equiv f_i(y) \mod f_i(x)$ für alle $1 \leq i \leq t$ und da $f_k(x) \neq f_l(x)$ für $k \neq l$, so sind für mindestens ein i mit $1 \leq i \leq t$ die Koeffizienten von y^i in $H(y) \mod f_k$ und in $H(y) \mod f_l$ verschieden. Darum ist $\alpha_k^{(i)} \neq \alpha_l^{(i)}$ in der Komponentendarstellung von $g_i(x)$. Also ist $\{g_0, g_1, \ldots, g_{d-1}\}$ eine aufspaltende Menge. □

Das Verfahren findet also die aufspaltende Menge $\{g_0, g_1, \ldots, g_{d-1}\}$ und berechnet dann für alle $0 \le i < d$ und alle $\alpha \in \mathbb{F}_q$ die Teilfaktorisierungen, die sich durch

$$\mathrm{ggT}\,(f(x), g_i(x) - \alpha)$$

ergeben, um f total zu faktorisieren.

Man kann, wenn dies sinnvoll erscheint, zu den g_i mit den Methoden von Moenck oder Zassenhaus die Menge C der $\alpha \in \mathbb{F}_q$ bestimmen, für die $\mathrm{ggT}(f(x), b(x) - \alpha)$ einen nicht trivialen Faktor liefert.

Algorithmus 23 SHOUP $(f, \mathbb{F}_q, d)$
Eingabe: Das normierte, quadratfreie Polynom $f \in \mathbb{F}_q[x]$, der Körper $\mathbb{F}_q$ und der Grad d, den alle irreduziblen Faktoren von f haben.
Ausgabe: Die Faktorisierung von f.
Programm: Shoup.mws: shoup

1. Setze $t \leftarrow n/d$, $i \leftarrow 0$ und $H(y) \leftarrow 1$.
2. Setze $H(y) \leftarrow H(y) \cdot (y - x^{q^i})$.
3. Ist $i < d-1$, so setze $i \leftarrow i+1$ und gehe zu Schritt 2.
4. Identifiziere mit den d Koeffizienten von $H(y)$ die Polynome $g_0(x), \ldots, g_d(x)$ und vergesse $g_d(x) = 1$ sofort wieder.
5. Setze $i \leftarrow 0$.
6. Ist $g_i(x) \neq 1$, so berechne für alle bisher gefundenen Faktoren $g(x)$ von $f(x)$ die partielle Faktorisierung $g(x) = \prod_{\alpha \in \mathbb{F}_q} \mathrm{ggT}(g(x), g_i(x) - \alpha)$.
7. Ist die Anzahl der bisher gefundenen Faktoren von $f(x)$ kleiner als t, so setze $i \leftarrow i+1$ und gehe zu Schritt 6.
8. Liefere die Faktorisierung von $f(x)$ zurück.

Beispiel 5.10
Betrachte das Polynom

$$f(x) = x^{12} + 4x^{11} + 2x^{10} + 2x^9 + 6x^6 + 3x^4 + 3x^3 + 4x^2 + 5 \in \mathbb{F}_7[x],$$

welches in vier Faktoren vom Grad $d = 3$ zerfällt. Es ist

$$H(y) = \prod_{i=0}^{2} (y - x^{7^i}) \equiv g_0(x) + g_1(x)y + g_2(x)y^2 + y^3$$

mit

$$\begin{aligned}
g_0(x) &= 3x^{11} + x^{10} + 4x^9 + x^8 + 2x^7 + 2x^6 + 6x^4 + 5x^3 + x^2 + 3,\\
g_1(x) &= 6x^{11} + x^{10} + x^9 + x^8 + 3x^7 + 6x^6 + 2x^5 + 4x^4 + x^3 + 5x^2 + 4x \text{ und}\\
g_2(x) &= 3x^{10} + 4x^9 + 2x^8 + 6x^7 + 3x^6 + 6x^5 + x^4 + 3x^3 + 6x^2 + 2x + 3.
\end{aligned}$$

Um f zu zerlegen, bilde ich $\mathrm{ggT}(f(x), g_i(x) - \alpha)$ für alle $0 \le i < d$ und alle $\alpha \in \mathbb{F}_7$, ich starte mit g_0. Es ist

$$\begin{aligned}
h_1(x) &= \mathrm{ggT}(f(x), g_0(x) - 1) = x^3 + 6x^2 + 4x + 1,\\
h_2(x) &= f(x)/h_1(x) = x^9 + 5x^8 + 3x^7 + 5x^6 + 2x^5 + 5x^2 + x + 5,
\end{aligned}$$

rechne weiter mit $f(x) = (x^3 + 6x^2 + 4x + 1) \cdot h_2(x)$*. Es ist*

$$\begin{aligned} h_3(x) &= \mathrm{ggT}(h_2(x), g_0(x) - 4) &&= x^6 + 6x^5 + 4x^4 + x^3 + 3x^2 + 2x + 2, \\ h_4(x) &= h_2(x)/h_3(x) &&= x^3 + 6x^2 + 5x + 6, \end{aligned}$$

rechne weiter mit $f(x) = (x^3 + 6x^2 + 4x + 1) \cdot h_3(x) \cdot (x^3 + 6x^2 + 5x + 6)$*. Mit* g_0 *werden keine weiteren Faktoren von* h_3 *gefunden, untersuche darum* g_1*:*

$$\mathrm{ggT}(h_3(x), g_1(x)) = x^3 + 4,$$

insgesamt ist also $f(x) = (x^3 + 6x^2 + 4x + 1)(x^3 + 4)(x^3 + 6x^2 + 4x + 4)(x^3 + 6x^2 + 5x + 6)$.

Komplexität

Für die Faktoren $y - x^{q^i}$ muß zuerst x^q in $(2\log_2(q) - 1) \cdot (3n^2 + 7n + 4)$ elementaren Operationen bestimmt werden. Dann werden $d-1$ Polynome miteinander multipliziert, das dauert jeweils $3n^2 + 7n + 4$ Operationen.

Abschließend werden für $d - 1$ Polynome $q - 1$ größte gemeinsame Teiler bestimmt, das dauert jeweils $(2n^2 + 3n)$ Operationen.

Insgesamt braucht der Algorithmus von Shoup also

$(2\log_2(q) - 1)(3n^2 + 7n + 4) + (d - 1)(3n^2 + 7n + 4) + (d - 1)(q - 1)(2n^2 + 3n)$ Operationen

und hat darum die Komplexität $\mathcal{O}(dn^2q)$.

6 Der Algorithmus von Niederreiter

In [38], [39] und [41] entwickelte H. Niederreiter ab 1992 seinen Algorithmus zur Faktorisierung von Polynomen über endlichen Körpern und faßte ihn in [42] noch einmal zusammen. In Abschnitt 6.4 auf Seite 95 werde ich auf die Verbindung zum Algorithmus von Berlekamp eingehen, welche es ermöglicht, die Theorie von Niederreiter als Weiterentwicklung der Theorie von Berlekamp zu betrachten. Der Zugang als Fortführung der Theorie von Berlekamp wird von J. von zur Gathen und S. Gao in [13] beschrieben.

Niederreiter benutzt in seiner Theorie Hasse–Teichmüller–Ableitungen:

Definition 6.1
Die Hasse–Teichmüller–Ableitung $H^{(k)}$ der Ordnung $k \in \mathbb{N}$ für formale Laurent–Reihen σ in x^{-1}, das heißt $\sigma = \sum_{i=w}^{\infty} s_i x^{-i}$ mit $w \in \mathbb{Z}$, ist definiert als

$$H^{(k)}\left(\sum_{i=w}^{\infty} s_i x^{-i}\right) = \sum_{i=w}^{\infty} \binom{-i}{k} s_i x^{-i-k},$$

wobei bekanntlich für alle $\mu \in \mathbb{Z}$ und alle $k \in \mathbb{N}$ der Binomialkoeffizient $\binom{\mu}{k}$ definiert ist als

$$\binom{\mu}{k} = \begin{cases} \frac{\mu\cdot(\mu-1)\cdots(\mu-k+1)}{k!} & \text{für } k > 0, \\ 1 & \text{für } k = 0. \end{cases}$$

Ist $\mu \in \mathbb{N}$, so ist also

$$\begin{aligned} \binom{-\mu}{k} &= \frac{-\mu\cdot(-\mu-1)\cdots(-\mu-k+1)}{k!} = (-1)^k \frac{\mu\cdot(\mu+1)\cdots(\mu+k-1)}{k!} \\ &= (-1)^k \binom{\mu+k-1}{k}. \end{aligned} \tag{9}$$

Der rationale Funktionenkörper $\mathbb{F}_q(x)$ ist in $\mathbb{F}_q((x^{-1}))$ enthalten, darum ist die Hasse–Teichmüller–Ableitung auch auf $\mathbb{F}_q(x)$ definiert.

Lemma 6.2
Für jede formale Laurent–Reihe $\sigma = \sum_{i=w}^{\infty} s_i x^{-i}$ gilt für die gewöhnliche Ableitung

$$\sigma^{(k)} = k! \cdot H^{(k)}(\sigma).$$

Beweis:

$$\begin{aligned} \sigma^{(k)} &= \sum_{i=w}^{\infty} (-i)(-i-1)\cdots(-i-k+1)\cdot s_i x^{-i-k} \\ &= k! \cdot \sum_{i=w}^{\infty} \frac{(-i)(-i-1)\cdots(-i-k+1)}{k!} \cdot s_i x^{-i-k} \\ &= k! \cdot \sum_{i=w}^{\infty} \binom{-i}{k} s_i x^{-i-k} \end{aligned}$$

□

Also läßt sich $H^{(k)}$ nicht durch k–faches Iterieren von $H^{(1)}$ bestimmen.

Satz 6.3
$H^{(k)}$ ist eine $\mathbb{F}_q$–lineare Abbildung auf den formalen Laurent–Reihen und damit insbesondere auf dem rationalen Funktionenkörper $\mathbb{F}_q(x)$.

Beweis: Es ist bekannt, daß die gewöhnliche Ableitung eine lineare Abbildung auf den Laurentreihen ist, die insbesondere Elemente des $\mathbb{F}_q(x)$ auf Elemente des $\mathbb{F}_q(x)$ abbildet. Dasselbe gilt für die Hasse–Teichmüller–Ableitungen, da diese sich von den gewöhnlichen Ableitungen nur durch den Faktor $k!$ unterscheiden. □

Ich betrachte auch hier das Polynom $f \in \mathbb{F}_q[x]$ mit $\mathrm{Grad}(f) = n \geq 1$, $q = p^s$. f ist normiert und von der Form $f = f_1^{l_1} \cdot f_2^{l_2} \cdots f_t^{l_t}$ mit $l_i \in \mathbb{N} \setminus \{0\}$ und $f_i \in \mathbb{F}_q[x]$, wobei die $f_i \in \mathbb{F}_q[x]$ paarweise verschiedene normierte irreduzible Polynome mit $\mathrm{Grad}(f_i) \geq 1$ sind. f ist also nicht notwendig quadratfrei.

Niederreiters Idee war, sich für die Differentialgleichung

$$H^{(r-1)}(y) = y^r \tag{10}$$

zu interessieren, wobei $r = p^u$ eine Potenz der Charakteristik p von $\mathbb{F}_q$ ist. u teilt s, also ist $\mathbb{F}_r$ ein Unterkörper von $\mathbb{F}_q$.

Bemerkung 6.4
Niederreiter entwickelte seine Theorie zu Anfang nur für den Spezialfall $r = p$, in dem sich die Differentialgleichung

$$y^{(p-1)} = -y^p$$

ergibt.

Beweis: Wegen $(p-1)! \equiv -1 \mod p$ gilt

$$y^p = H^{(p-1)}(y) \Leftrightarrow y^p = (p-1)! y^{(p-1)} \Leftrightarrow -y^p = y^{(p-1)}.$$

□

Lemma 6.5
Die rationalen Lösungen $y = h/f$ der Gleichung (10) bilden einen $\mathbb{F}_r$–Unterraum von $\mathbb{F}_q(x)$, in welchem für alle Lösungen $y = h/f$ gilt $\mathrm{Grad}(h) < \mathrm{Grad}(f)$.

Beweis: Ist $y = h/f$, so ist der Grad der Laurententwicklung von y in x gerade $\mathrm{Grad}(h) - \mathrm{Grad}(f) = -w$, der Untergrad der Laurententwicklung von y in x^{-1} ist gerade $\mathrm{Grad}(f) - \mathrm{Grad}(h) = w$.

Der Unterraum ist linear über $\mathbb{F}_r$, da sowohl $H^{(r-1)}$ als auch y^r linear über $\mathbb{F}_r$ sind.

Betrachte die Gleichung (10). Für eine Lösung $y = \sum_{i=w}^{\infty} s_i x^{-i}$ mit $s_w \neq 0$ gilt

$$\begin{aligned} \mathrm{Grad}\left(H^{(r-1)}(y)\right) &= \mathrm{Grad}\left(H^{(r-1)}\left(\sum_{i=w}^{\infty} s_i x^{-i}\right)\right) \\ &= \mathrm{Grad}\left(\sum_{i=w}^{\infty} \binom{-i}{r-1} s_i x^{-i-(r-1)}\right) \\ &\leq -w - r + 1 \end{aligned}$$

in x, da $\binom{-w}{r-1} = 0$ möglich ist. Es ist $-w = \mathrm{Grad}(y)$ in x und $\mathrm{Grad}(y^r) = r \cdot \mathrm{Grad}(y)$. Wegen Gleichung (10) muß $\mathrm{Grad}(y^r) = \mathrm{Grad}\left(H^{(r-1)}(y)\right)$ gelten, darum ist

$$\begin{aligned} r \cdot \mathrm{Grad}(y) \leq \mathrm{Grad}(y) - r + 1 &\Leftrightarrow (r-1)\,\mathrm{Grad}(y) \leq -(r-1) \\ &\Leftrightarrow \mathrm{Grad}\,(y) \leq -1 \end{aligned}$$

Für Lösungen der Form $y = h/f$ gilt somit

$$\begin{aligned} \mathrm{Grad}\left(\frac{h}{f}\right) \leq -1 &\Leftrightarrow \mathrm{Grad}(h) < \mathrm{Grad}(f) \\ &\Leftrightarrow \mathrm{Grad}(h) < n. \end{aligned} \tag{11}$$

Ich kann also h schreiben als

$$h(x) = \sum_{i=0}^{n-1} h_i x^i,$$

wobei die $h_i \in \mathbb{F}_q$ für $0 \leq i < n$. $\square$

Die Menge der Polynome h, die Zähler einer rationalen Lösung $y = h/f$ von (10) sind, fasse ich in der Menge $\mathcal{N}(f,r) = \left\{h \mid H^{(r-1)}\left(\frac{h}{f}\right) = \left(\frac{h}{f}\right)^r\right\}$ zusammen, $\mathcal{N}(f,r)$ ist ein $\mathbb{F}_r$–linearer Untervektorraum von $\mathcal{R}$, aber keine Algebra.

Für Lösungen $y = h/f$ mit $h \in \mathbb{F}_q[x]$ kann die Differentialgleichung (10) geschrieben werden als

$$f^r H^{(r-1)}\left(\frac{h}{f}\right) = h^r. \tag{12}$$

Die Lösungen der Differentialgleichung sind durch den Satz 6.7 gegeben, zu dessen Beweis ich jedoch das folgende Lemma brauche.

Lemma 6.6
Für jedes $\alpha \in \mathbb{F}_q$ und alle $k, \mu \in \mathbb{N}$ gilt

$$H^{(k)}\left((x-\alpha)^{-\mu}\right) = \binom{-\mu}{k}(x-\alpha)^{-\mu-k}.$$

Beweis: Zuerst einmal ist

$$(x-\alpha)^{-1} = \frac{1}{x} \cdot \frac{1}{1-\frac{\alpha}{x}} = \frac{1}{x}\sum_{i=0}^{\infty}\left(\frac{\alpha}{x}\right)^i = \sum_{i=0}^{\infty} \alpha^i x^{-i-1},$$

indem ich in eine geometrische Reihe entwickle. Leite ich diese Gleichung $(\mu-1)$–mal ab, so erhalte ich

$$(-1)(-2)\cdots(-\mu+1)\cdot(x-\alpha)^{-\mu} = \sum_{i=0}^{\infty} \alpha^i \cdot (-i-1)(-i-2)\cdots(-i-\mu+1)\cdot x^{-i-\mu},$$

was äquivalent zu

$$(x-\alpha)^{-\mu} = \sum_{i=0}^{\infty} \alpha^i \frac{(i+\mu-1)(i+\mu-2)\cdots(i+1)}{(\mu-1)\cdots 2\cdot 1} x^{-i-\mu}$$

ist. Damit kann ich die Gleichungen

$$\begin{aligned} (x-\alpha)^{-\mu} &= \sum_{i=0}^{\infty}\binom{\mu+i-1}{\mu-1}\alpha^i x^{-i-\mu} \\ \Leftrightarrow \quad (x-\alpha)^{-\mu} &= \sum_{i=\mu}^{\infty}\binom{i-1}{\mu-1}\alpha^{i-\mu}x^{-i} \end{aligned}$$

aufstellen. Nun kann $H^{(k)}$ auf $(x-\alpha)^{-\mu}$ angewendet werden:

$$\begin{aligned} H^{(k)}\left((x-\alpha)^{-\mu}\right) &= H^{(k)}\left(\sum_{i=\mu}^{\infty}\binom{i-1}{\mu-1}\alpha^{i-\mu}x^{-i}\right) \\ &= \sum_{i=\mu}^{\infty}\binom{-i}{k}\binom{i-1}{\mu-1}\alpha^{i-\mu}x^{-i-k} \\ &= \sum_{i=\mu+k}^{\infty}\binom{k-i}{k}\binom{i-k-1}{\mu-1}\alpha^{i-k-\mu}x^{-i}. \end{aligned}$$

Es ist

$$\begin{aligned}
\binom{k-i}{k}\binom{i-k-1}{\mu-1} &= \frac{(k-i)(k-i-1)\cdots(-i+1)}{k!}\cdot \\
&\quad \frac{(i-k-1)(i-k-2)\cdots(i-k-\mu+1)}{(\mu-1)!} \\
&= \frac{(-1)^k\cdot(-\mu)\cdots(-\mu-k+1)}{(-1)^k(-\mu)\cdots(-\mu-k+1)}\cdot\frac{(k-i)\cdots(-i+1)}{k!}\cdot \\
&\quad \frac{(i-k-1)\cdots(i-k-\mu+1)}{(\mu-1)!} \\
&= \frac{(-\mu)\cdots(-\mu-k+1)}{k!}\cdot \\
&\quad \frac{(i-1)\cdots(i-k)(i-k-1)\cdots(i-k-\mu+1)}{(\mu-1)!(\mu+k-1)\cdots(\mu)} \\
&= \binom{-\mu}{k}\binom{i-1}{\mu+k-1},
\end{aligned}$$

darum ist

$$\begin{aligned}
H^{(k)}\left((x-\alpha)^{-\mu}\right) &= \binom{-\mu}{k}\sum_{i=(\mu+k)}^{\infty}\binom{i-1}{(\mu+k)-1}\alpha^{i-(\mu+k)}x^{-i} \\
&= \binom{-\mu}{k}(x-\alpha)^{-\mu-k}.
\end{aligned}$$

□

6.1 Die Faktorisierungsidee von Niederreiter

Der Satz, der die Lösungen $y = h/f$ der Differentialgleichung (10) liefert und so zu einem Faktorisierungs–Algorithmus führt, ist dieser:

Satz 6.7
Sei $f \in \mathbb{F}_q[x]$ ein normiertes, nicht konstantes Polynom. Für die Differentialgleichung $H^{(r-1)}(y) = y^r$ sind die Lösungen der Form $y = h/f$ mit $h \in \mathbb{F}_q[x]$ gegeben durch die Menge

$$\left\{ y = \sum_{i=1}^{t}\gamma_i\frac{f_i'}{f_i} \mid \gamma_1,\ldots,\gamma_t \in \mathbb{F}_r \right\},$$

wobei $f_1,\ldots,f_t \in \mathbb{F}_q[x]$ die paarweise verschiedenen normierten irreduziblen Faktoren von f sind.

Beweis: Durch Gleichung (11) weiß ich bereits, daß $\mathrm{Grad}(h) < \mathrm{Grad}(f)$ gelten muß. Sei $f^* = f_1\cdot f_2\cdots f_t$ der quadratfreie Teil von f, in dem jeder irreduzible Faktor f_i von f genau einmal vorkommt. Sei R die Menge der Nullstellen von f^* im Zerfällungskörper E von f^* über $\mathbb{F}_q$. Die Nullstellen von f sind genau die Elemente $\beta \in R$, jede dieser Nullstellen hat eine Vielfachheit $l(\beta)$. Sind zwei Nullstellen zueinander konjugiert, haben sie dieselbe Vielfachheit.

Ich wende auf $y = h/f$ die Partialbruchzerlegung über E (siehe dazu auch [24] und [49]) an:

$$y = \sum_{\beta\in R}\sum_{i=1}^{l(\beta)}\frac{\gamma(\beta,i)}{(x-\beta)^i}, \tag{13}$$

wobei alle $\gamma(\beta, i) \in E$, da f über E in lineare Faktoren zerfällt.

Durch Lemma 6.6 weiß ich, daß dann gilt

$$H^{(r-1)}(y) = \sum_{\beta \in R} \sum_{i=1}^{l(\beta)} \binom{-i}{r-1} \frac{\gamma(\beta, i)}{(x-\beta)^{i+r-1}}. \tag{14}$$

Angenommen, für ein $\beta_0 \in R$ und ein $i \geq 2$ gälte $\gamma(\beta_0, i) \neq 0$, so wählte ich das maximale i mit dieser Eigenschaft. y genügt der Differentialgleichung (10); da der Term $\gamma(\beta_0, i)^r (x-\beta_0)^{-ir}$ in y^r auftauchen müßte, so folgte aus der Eindeutigkeit der Partialbruchzerlegung, daß dieser Term auch in (14) auftauchen würde. Es müßte also gelten

$$\begin{aligned} & (x-\beta_0)^{-ir} = (x-\beta_0)^{-(i+r-1)} & \Leftrightarrow \quad & ir = i+r-1 \\ \Leftrightarrow \quad & r(i-1) = i-1 & \Leftrightarrow \quad & (i-1) = 0 \\ \Leftrightarrow \quad & i = 1. & & \end{aligned}$$

Dies ist ein Widerspruch zu der Annahme, daß $i \geq 2$.

Dieser Widerspruch liefert mir handlichere Darstellungen von (13) und (14):

$$\begin{aligned} y &= \sum_{\beta \in R} \frac{\gamma(\beta)}{x-\beta}, \\ H^{(r-1)}(y) &= \sum_{\beta \in R} \frac{\gamma(\beta)}{(x-\beta)^r}, \end{aligned}$$

wobei $\gamma(\beta) = \gamma(\beta, 1)$ aus (13) und (14).

Die Differentialgleichung (10) hat also die Form

$$\sum_{\beta \in R} \frac{\gamma(\beta)}{(x-\beta)^r} = \sum_{\beta \in R} \frac{\gamma(\beta)^r}{(x-\beta)^r}$$

und gilt somit genau dann, wenn $\gamma(\beta)^r = \gamma(\beta)$ für alle $\beta \in R$. Dies ist genau dann der Fall, wenn $\gamma(\beta) \in \mathbb{F}_r$ für alle $\beta \in R$.

$\beta_1, \beta_2 \in R$ seien über $\mathbb{F}_q$ konjugiert zueinander, was bedeutet, daß sie Nullstellen desselben irreduziblen Faktors f_i sind. Dann gibt es einen Körperautomorphismus θ, der β_1 auf β_2 abbildet. Für diesen Automorphismus gilt

$$\theta \left(\sum_{\substack{\beta \in R, \\ f_i(\beta)=0}} \frac{\gamma(\beta)}{x-\beta} \right) = \sum_{\substack{\beta \in R, \\ f_i(\beta)=0}} \frac{\gamma(\beta)}{x-\beta},$$

darum sind auch $\gamma(\beta_1)$ und $\gamma(\beta_2)$ zueinander konjugiert. Da aber $\gamma(\beta_1), \gamma(\beta_2) \in \mathbb{F}_r$ sind, muß gelten $\gamma(\beta_1) = \gamma(\beta_2)$. Somit erhalte ich für passende $\gamma_i \in \mathbb{F}_r$

$$\begin{aligned} y &= \sum_{i=1}^{t} \gamma_i \sum_{\substack{\beta \in R, \\ f_i(\beta)=0}} \frac{1}{x-\beta} = \sum_{i=1}^{t} \frac{\gamma_i}{f_i} \left(\sum_{\substack{\beta \in R, \\ f_i(\beta)=0}} \prod_{\substack{\alpha \in R, \alpha \neq \beta, \\ f_i(\alpha)=0}} (x-\alpha) \right) \\ &= \sum_{i=1}^{t} \gamma_i \frac{f_i'}{f_i}. \end{aligned}$$

Andererseits kann man an den soeben ausgeführten Schritten sehen, daß jedes solche y eine Lösung der Gleichung (10) ist. □

Löst $y = h/f$ die Differentialgleichung (10), so gilt

$$\mathrm{ggT}(f,h) = \prod_{\substack{i=1,\\ \gamma_i=0}}^{t} f_i \quad \Leftrightarrow \quad \frac{f}{\mathrm{ggT}(f,h)} = \prod_{\substack{i=1,\\ \gamma_i\neq 0}}^{t} f_i.$$

Die Lösungen bilden einen t–dimensionalen $\mathbb{F}_r$–Unterraum des rationalen Funktionenkörpers $\mathbb{F}_q(x)$, von dem ich eine Basis mit t Elementen bestimme. Die Elemente der Basis kombiniere ich so lange linear miteinander, bis ich f total faktorisiert habe. Ist f quadratfrei, so habe ich f genau dann total faktorisiert, wenn ich t Faktoren gefunden habe. Der größte gemeinsame Teiler von f und h gibt einen nicht trivialen Faktor von f, wenn für mindestens ein, nicht aber für alle i mit $0 < i \leq t$ gilt $\gamma_i = 0$.

Eine effizient Suche nach Faktoren von f ist mit dem folgenden Satz möglich:

Satz 6.8
Sei f ein quadratfreies Polynom aus $\mathbb{F}_q[x]$ mit $\mathrm{Grad}(f) = n$. Dann gilt

1. *Die Menge $\{\frac{f}{f_i} f_i' \mid 1 \leq i \leq t\}$ ist eine Basis von $\mathcal{N}(f,r)$ über $\mathbb{F}_r$.*
2. *Ist $\gamma \in \mathbb{F}_r$ und $h = \sum_{i=1}^{t} \gamma_i \frac{f}{f_i} f_i' \in \mathcal{N}(f,r)$ mit $\gamma_i \in \mathbb{F}_r$, so ist*
$$\mathrm{ggT}(f, h - \gamma f') = \prod_{\substack{i=1,\\ \gamma_i\neq\gamma}}^{t} f_i.$$
3. *Für jedes $h \in \mathcal{N}(f,r)$ gilt*
$$f = \prod_{\gamma\in\mathbb{F}_r} \mathrm{ggT}(f, h - \gamma f').$$

Beweis:
1. Die erste Aussage ist aus Satz 6.7 auf Seite 77 sofort klar.
2. Die zweite Aussage folgt aus der Tatsache, daß gelten
$$f' = \sum_{i=1}^{t} \frac{f}{f_i} f_i' \text{ und}$$
$$f_i \mid h - \gamma f' \Leftrightarrow \gamma_i = \gamma.$$
3. Die dritte Aussage ist nach dem Beweis der zweiten Aussage klar. □

Ist f quadratfrei, so bestimmt man eine Basis des Raums der Elemente h, für die

$$H^{(r-1)}\left(\frac{h}{f}\right) = \left(\frac{h}{f}\right)^r$$

ist. Dann bestimmt man für alle h aus der Basis und alle $\gamma \in \mathbb{F}_r$ die partiellen Faktorisierungen, die sich durch

$$f = \mathrm{ggT}\left(f, h - \gamma f'\right) \frac{f}{\mathrm{ggT}(f, h - \gamma f')}$$

ergeben. Da wir mit einer Basis arbeiten, wird f dadurch total faktorisiert, die Ähnlichkeit zum Algorithmus ergibt sich durch die beiden Methoden zum Finden partieller Faktorisierungen:

$$f(x) = \prod_{\alpha\in\mathbb{F}_r} \mathrm{ggT}(f(x), b(x) - \alpha)$$

bei Berlekamp und

$$f(x) = \prod_{\gamma\in\mathbb{F}_r} \mathrm{ggT}(f(x), h(x) - \gamma f'(x))$$

bei Niederreiter.

6.2 Die Lösung der Differentialgleichung

Das verbleibende Problem ist die Lösung der Differentialgleichung (10). Um die Lösungen $y = h/f$ zu finden, linearisiert Niederreiter das Problem.

Gleichung (12) lautet $f^r H^{(r-1)}\left(\frac{h}{f}\right) = h^r$, sie besteht aus Polynomen in x^r, der Grad beider Seiten in x ist kleiner oder gleich $(n-1)r$. Ich kann einen Koeffizientenvergleich der x^{jr} für $0 \le j < n$ anstellen und erhalte so das zu (12) äquivalente algebraische Gleichungssystem

$$N_r(f)\mathbf{h} = \mathbf{h}^r, \tag{15}$$

wobei $\mathbf{h} = (h_0, \ldots, h_{n-1})^\top \in \mathbb{F}_q^n$ der Koeffizientenvektor von h ist, $\mathbf{h}^r = (h_0^r, \ldots, h_{n-1}^r)^\top$.

Zur Lösung algebraischer Gleichungssysteme gibt es bekannte Algorithmen, darum interessiere ich mich jetzt nur noch dafür, wie ich effizient die Matrix $N_r(f)$ berechnen kann.

Im weiteren Verlauf meiner Ausführungen benötige ich einige Begriffe aus der Theorie der linearen Rekursionsfolgen, siehe dazu auch Lidl und Niederreiter in [31] unter dem Stichwort Shift Register Sequences:

Definition 6.9

- $\mathbb{F}_q^\infty$ *ist der Vektorraum der Folgen in* $\mathbb{F}_q$.
- *Für ein* $k \in \mathbb{N} \setminus \{0\}$ *und* $\alpha_0, \alpha_1, \ldots, \alpha_{k-1} \in \mathbb{F}_q$ *wird eine Relation der Form*

$$s_{i+k} = -\alpha_{k-1}s_{i+k-1} - \alpha_{k-2}s_{i+k-2} - \cdots - \alpha_0 s_i \tag{16}$$

  homogene lineare Rekursionsrelation über $\mathbb{F}_q$ *in* k *Termen genannt.*
- *Sei* $(s_i) = s_0, s_1, \ldots$ *aus* $\mathbb{F}_q^\infty$ *eine Folge von Elementen aus* $\mathbb{F}_q$. *Genügt diese Folge der homogenen linearen Rekursionsrelation (16) für alle* $i \in \mathbb{N}$, *so wird sie homogene lineare Rekursionsfolge* k*–ter Ordnung über* $\mathbb{F}_q$ *genannt.*
- *Mit einer homogenen linearen Rekursionsfolge, die durch die homogene lineare Rekursionsrelation (16) gegeben ist, assoziiert man die* $(k \times k)$*–Matrix* A *über* $\mathbb{F}_q$, *die wie folgt definiert ist:*

$$A = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \\ -\alpha_0 & -\alpha_1 & -\alpha_2 & \ldots & -\alpha_{k-1} \end{bmatrix}. \tag{17}$$

  Für $k = 0$ *ist* $A = [-\alpha_0]$.
- *Das charakteristische Polynom* $f(x) = x^k + \alpha_{k-1}x^{k-1} + \cdots + \alpha_0$ *der Matrix* A *wird charakteristisches Polynom der homogenen linearen Rekursionsrelation (16) genannt.*
- *Der Spaltenvektor* $\mathbf{s}_i = (s_i, s_{i+1}, \ldots, s_{i+k-1})^\top$ *wird* i*–ter Wertevektor der homogenen linearen Rekursionsfolge genannt. Der Wertevektor* $\mathbf{s}_0 = (s_0, s_1, \ldots, s_{k-1})^\top$ *heißt Anfangswertvektor, durch ihn ist die gesamte homogene lineare Rekursionsfolge* (s_i) *bestimmt.*

Ist (s_i) eine homogene lineare Rekursionsfolge in $\mathbb{F}_q$, die der Gleichung (16) genügt, und ist A die assoziierte Matrix, so gilt für die Wertevektoren der Folge

$$\mathbf{s}_i = A^i \mathbf{s}_0.$$

Definition 6.10

- *Die zur homogenen linearen Rekursionsrelation (16) gehörende Impuls–Antwort–Folge* (d_i) *ist diejenige homogene lineare Rekursionsfolge* (s_i), *die den Anfangswertvektor* $d_0 = d_1 = \cdots = d_{k-2} = 0$, $d_{k-1} = 1$ *hat. Für* $k = 1$ *ist dieser Anfangswertvektor gegeben durch* $d_0 = 1$.

- *Für eine Folge (s_i) von Elementen aus $\mathbb{F}_q$ hat man die sogenannte erzeugende Funktion*

$$g(x) = s_0 x^{-1} + s_1 x^{-2} + \cdots + s_i x^{-i-1} + \cdots = \sum_{i=0}^{\infty} s_i x^{-i-1}.$$

  Die erzeugende Funktion wird als alternative Schreibweise der Folge (s_i) benutzt.

- *Sei $(s_i) \in \mathbb{F}_q^{\infty}$ eine Folge in $\mathbb{F}_q$. Man definiert die folgenden zwei Abbildungen:*

$$\begin{aligned} \Delta_k(s_i) &= (s_{ik}) \text{ ist die Dezimierung um } k, \\ T(s_i) &= (s_{i+1}) \text{ ist der Shift um } 1. \end{aligned}$$

Beispiel 6.1

$$\begin{aligned} (s_i) &= s_0, s_1, \ldots, s_i, \ldots \\ \Delta_0(s_i) &= s_0, s_0, \ldots, s_0, \ldots \\ \Delta_1(s_i) &= s_0, s_1, \ldots, s_i, \ldots \qquad = (s_i) \\ \Delta_2(s_i) &= s_0, s_2, \ldots, s_{2i}, \ldots \\ \Delta_k(s_i) &= s_{0k}, s_{1k}, \ldots, s_{ik}, \ldots \\ T(s_i) &= s_1, s_2, \ldots, s_{i+1}, \ldots \\ T^k(s_i) &= s_{0+k}, s_{1+k}, \ldots, s_{i+k}, \ldots \end{aligned}$$

Die Abbildungen Δ_k und T^k sind für alle $k \in \mathbb{N}$ linear über $\mathbb{F}_q$.

Definition 6.11

- *Für ein Polynom $g(x) = \sum_{i=0}^{m} \alpha_i x^i$ mit $\alpha_i \in \mathbb{F}_q$ und $r = p^u$ mit $r^v = q$ sei*

$$g_r(x) = \sum_{i=0}^{m} \alpha_i^r x^i.$$

- *Für ein Polynom f sei $S(f)$ der Kern der linearen Abbildung $f(T)$.*

Lemma 6.12
$S(f)$ ist die Menge aller Folgen, die der homogenen linearen Rekursionsrelation mit charakteristischem Polynom f genügen.

Beweis: Die Abbildung $f(T)$ anzuwenden, bedeutet

$$\begin{aligned} f(T)(s_i) &= T^n(s_i) + \sum_{j=0}^{n-1} a_j T^j(s_i) \\ &= (s_{i+n}) + \sum_{j=0}^{n} a_j(s_{i+j}) \end{aligned}$$

zu bestimmen. Die Menge $S(f)$ ist also die Menge aller Folgen (s_i), bei denen

$$(s_{i+n}) + \sum_{j=0}^{n} a_j(s_{i+j}) = 0$$

ist für alle $i \in \mathbb{N}$. Dies entspricht der homogenen linearen Rekursionsrelation

$$s_{i+n} = -a_{n-1}s_{i+n-1} - a_{n-2}s_{i+n-2} - \cdots - a_0 s_i$$

mit dem charakteristischen Polynom

$$f(x) = x^n + a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_0.$$

□

Satz 6.13
Die Dezimierung Δ_r ist eine lineare Abbildung von $S(f)$ nach $S(f_r)$.

Beweis: Δ_r ist linear. Es bleibt die Behauptung zu zeigen, daß $\Delta_r(\sigma) \in S(f_r)$ für alle $\sigma \in S(f)$ gilt:

$f(x)$ teilt $f(x)^r = f_r(x^r)$. Daraus folgt, daß eine Folge σ im Kern von $f_r(T^r)$ ist, wenn sie im Kern von $f(T)$ ist. So bekomme ich für jedes $\sigma \in S(f)$ die Gleichungskette

$$\begin{aligned}
f_r(T)\left(\Delta_r(\sigma)\right) &= \sum_{j=0}^{n} \left(a_j^r T^j \left(\Delta_r(\sigma)\right)\right) && = \sum_{j=0}^{n} \left(a_j^r T^j \left((s_{ir})\right)\right) \\
&= \sum_{j=0}^{n} \left(a_j^r (s_{ir+jr})\right) && = \sum_{j=0}^{n} \left(a_j^r \Delta_r \left((s_{i+jr})\right)\right) \\
&= \Delta_r \left(\sum_{j=0}^{n} \left(a_j^r (s_{i+jr})\right)\right) && = \Delta_r \left(\sum_{j=0}^{n} \left(a_j^r T^{rj} \left((s_i)\right)\right)\right) \\
&= \Delta_r \left(f_r\left(T^r\right)(\sigma)\right) && = \Delta_r(0) \\
&= 0 \ \in \mathbb{F}_q^\infty.
\end{aligned}$$

Das bedeutet, daß $\Delta_r(\sigma) \in S(f_r)$, da $\Delta_r(\sigma)$ im Kern von $f_r(T)$ liegt. □

Die Folgen in $S(f)$ sind durch ihre n–dimensionalen Anfangswertvektoren bestimmt. Darum kann man $S(f)$ als n–dimensionalen Vektorraum ansehen.

Nun komme ich zu den Notationen, die ich in der Linearen Algebra verwende: Sind B_1 und B_2 zwei Basen des n–dimensionalen Vektorraums $S(f)$ und ist $s \in S(f)$, so bezeichnet $[s]_{B_1}$ den Vektor bzw. die Folge s in der Darstellung bezüglich der Basis B_1, $[s]_{B_1}$ ist ein Spaltenvektor. Ist A eine Abbildung auf $S(f)$, so ist $[A]_{B_1,B_2}$ die Abbildung A, die auf Elemente in der Darstellung bezüglich B_1 angewendet wird und Elemente in der Darstellung bezüglich B_2 liefert: Es ist

$$[A]_{B_1,B_2}[s]_{B_1} = [As]_{B_2}$$

für alle $s \in S(f)$.

Definition 6.14

- *Ich schreibe auch $[A]_{B_1,B_2} = B_2^{-1}AB_1$, wobei ich mit der Matrix B_1 die Matrix der Anfangswertvektoren der Basiselemente der Basis B_1 und mit B_2 die Matrix der Anfangswertvektoren der Basiselemente der Basis B_2 meine. (Ich fasse A auf als Matrix bezüglich der Standardbasis, B_1 liefert die B_1–Koordinaten für Elemente in Standardkoordinaten, B_2 liefert B_2–Koordinaten.)*
- *Die kanonische Basis E_1 des Vektorraums $S(f)$ ist die geordnete Menge*

  $$\{\omega_0 = T^0(\omega), \omega_1 = T^1(\omega), \ldots, \omega_{n-1} = T^{n-1}(\omega)\}$$

  für die Impuls–Antwort–Folge $\omega = (u_i)$, also der Folge aus $S(f)$ mit Anfangswertvektor $(0, \ldots, 0, 1)^\top$.
- *Die Standardbasis E_2 ist die übliche mit den Einheitsvektoren ε_i für $0 \leq i < n$ als Anfangswertvektoren $\left(\text{z. B. } \varepsilon_0 = (1, 0, \ldots, 0)^\top, \varepsilon_{n-1} = (0, \ldots, 0, 1)^\top\right)$.*

Bemerkung 6.15
E_1 ist tatsächlich eine Basis, da die Anfangswertvektoren eine gespiegelte Dreiecksmatrix mit den Einträgen 1 auf der Diagonalen bilden, da also E_1 vollen Rang hat.

Lemma 6.16
Die erzeugende Funktion y_k *von* ω_k *für* $0 \leq k < n$ *ist* $y_k(x) = \frac{x^k}{f(x)}$.

Beweis: Das charakteristische Polynom $f(x) = \sum_{i=0}^{n} a_i x^i$ mit $a_n = 1$ führt zu der homogenen linearen Rekursionsrelation

$$
\begin{aligned}
s_{n+i} &= -a_{n-1}s_{n+i-1} - a_{n-2}s_{n+i-2} - \cdots - a_0 s_i \\
\Leftrightarrow \quad 0 &= s_{n+i} + a_{n-1}s_{n+i-1} + a_{n-2}s_{n+i-2} + \cdots + a_0 s_i \\
\Leftrightarrow \quad 0 &= \sum_{j=0}^{n} a_j s_{i+j}.
\end{aligned}
$$

Für die erzeugende Funktion y_0 von ω ist

$$
\begin{aligned}
f(x) \cdot y_0(x) &= \sum_{i=0}^{n} a_i x^i \left(\sum_{j=0}^{\infty} u_j x^{-j-1} \right) = \sum_{i=0}^{n} a_i \left(\sum_{j=0}^{\infty} u_j x^{i-j-1} \right) \\
&= \sum_{i=0}^{n} a_i \left(\sum_{j=0}^{i-1} u_j x^{i-j-1} + \sum_{j=0}^{\infty} u_{j+i} x^{-j-1} \right) \\
&= \sum_{i=0}^{n} a_i \left(\sum_{j=0}^{i-1} u_j x^{i-j-1} \right) + \underbrace{\sum_{j=0}^{\infty} x^{-j-1} \underbrace{\left(\sum_{i=0}^{n} a_i u_{j+i} \right)}_{0}}_{0} \\
&= a_n u_{n-1} x^{n-(n-1)-1} = a_n,
\end{aligned}
$$

da ω die Impuls–Antwort–Folge ist und somit $u_i = 0$ für $0 \leq i < n-1$ und $u_{n-1} = 1$. Da $a_n = 1$ ist, ergibt sich

$$
\begin{aligned}
f(x) \cdot y_0(x) &= 1 \\
\Leftrightarrow \quad y_0(x) &= \frac{1}{f(x)}.
\end{aligned}
$$

Da $\omega_k = T^k(\omega)$ ist, so gilt für $0 \leq k < n$

$$y_k(x) = x^k \cdot y_0(x).$$

darum ist zusammengefaßt

$$y_k(x) = \frac{x^k}{f(x)}.$$

□

Lemma 6.17
Der Koordinatenvektor von $\sigma = (s_i)$ *bezüglich der Basis* E_1

$$[\sigma]_{E_1} = (h_0, h_1, \ldots, h_{n-1})^\top,$$

ist der Koeffizientenvektor des Zählerpolynoms der erzeugenden Funktion von σ*. Außerdem ist der Koordinatenvektor von* σ *bezüglich der Basis* E_2

$$[\sigma]_{E_2} = (s_0, s_1, \ldots, s_{n-1})^\top,$$

gerade der Anfangswertvektor von σ*.*

Beweis: $[\sigma]_{E_2} = (s_0, s_1, \ldots, s_{n-1})^\top$ ist klar.

E_1 ist eine Basis des Vektorraums $S(f)$. Also ist σ genau dann in $S(f)$, wenn σ eine erzeugende Funktion $g(x)$ der Form $h(x)/f(x)$ mit $h(x) = \sum_{k=0}^{n-1} h_k x^k$ und allen $h_k \in \mathbb{F}_q$ hat. Es ist

$$\frac{h(x)}{f(x)} = \sum_{k=0}^{n-1} h_k \frac{x^k}{f(x)},$$

womit sich für die Folgen die Relation

$$\sigma = \sum_{k=0}^{n-1} h_k \omega_k$$

ergibt. □

Im Beweis des Satzes 6.19 werde ich den Satz 6.18 von Lucas benutzen, den ich darum hier zitiere und beweise (siehe dazu auch [34]):

Satz 6.18
Für natürliche Zahlen m, k mit $m \geq k$ und jede Primzahl $p \in \mathbb{N}$ gilt

$$\binom{m}{k} = \binom{m_0}{k_0}\binom{m_1}{k_1}\cdots\binom{m_\tau}{k_\tau} \mod p,$$

wobei $m = m_0 + m_1 p + m_2 p^2 + \cdots + m_\tau p^\tau$ und $k = k_0 + k_1 p + k_2 p^2 + \cdots + k_\tau p^\tau$ mit $0 \leq m_i, k_i < p$ für $0 \leq i \leq \tau$.

Beweis: Die behauptete Eigenschaft ist gleichbedeutend mit dieser:

Für jede Primzahl $p \in \mathbb{N}$ ist

$$\binom{m}{k} \equiv \binom{m_0}{k_0}\binom{m'}{k'} \mod p,$$

wobei $m = m_0 + m'p$, $k = k_0 + k'p$ mit $0 \leq m_0, k_0 < p$.

Es ist

$$\begin{aligned}\sum_{i=0}^{m} \binom{m}{i} x^i &= (1+x)^m \equiv (1+x)^{m_0}(1+x^p)^{m'} \mod p \\ &= \left(\sum_{i_0=0}^{m_0} \binom{m_0}{i_0} x^{i_0}\right)\left(\sum_{i'=0}^{m'} \binom{m'}{i'} x^{p\cdot i'}\right).\end{aligned}$$

Da in der ersten Summe nur Exponenten i_0 von x auftauchen mit $0 \leq i_0 \leq m_0 < p$, in der zweiten Summe Exponenten $i'p$, die Vielfache von p sind, und da jede natürliche Zahl $\alpha \in \mathbb{N}$ eine eindeutige Darstellung als $\alpha = \alpha_0 + \alpha' p$ hat, ist der Koeffizient von x^k gerade

$$\binom{m}{k} \equiv \binom{m_0}{k_0}\binom{m'}{k'} \mod p.$$

□

Satz 6.19
Ist $y = \sum_{i=0}^{\infty} s_i x^{-i-1}$ die erzeugende Funktion einer beliebigen homogenen linearen Rekursionsfolge $\sigma = (s_i) \in \mathbb{F}_q^\infty$ und ist z die erzeugende Funktion von $\Delta_r(\sigma) = (s_{ri})$ für eine Potenz $r = p^u$ von p mit $u \in \mathbb{N} \setminus \{0\}$, so gilt

$$z(x^r) = H^{(r-1)}(y).$$

Beweis: Es ist $z(x) = \sum_{i=0}^{\infty} s_{ri}\, x^{-i-1}$, darum ist

$$z(x^r) = \sum_{i=0}^{\infty} s_{ri}\,(x^r)^{-i-1} = \sum_{i=0}^{\infty} s_{ri}\, x^{-ri-r}.$$

Außerdem ist

$$H^{(r-1)}(y) = H^{(r-1)}\left(\sum_{i=0}^{\infty} s_i\, x^{-i-1}\right) = \sum_{i=0}^{\infty} \binom{-i-1}{r-1} s_i\, x^{-i-r}.$$

Nun ist $(-1)^{r-1} = 1$ für alle in Frage kommenden r, denn wenn p ungerade ist, so ist $r-1$ gerade, ist aber p gerade, so ist $p = 2$ und somit $-1 = 1$. Darum ist nach Gleichung (9)

$$\binom{-i-1}{r-1} = \binom{i+r-1}{r-1},$$

also gilt

$$H^{(r-1)}(y) = \sum_{i=0}^{\infty} \binom{i+r-1}{r-1} s_i\, x^{-i-r}.$$

Weiterhin ist

$$\frac{r-1}{p-1} = \frac{p^u - 1}{p-1} = \sum_{j=0}^{u-1} p^j,$$

dann ist

$$r - 1 = \sum_{j=0}^{u-1} (p-1) \cdot p^j.$$

Sei nun $i + r - 1 \equiv \sum_{j=0}^{u-1} \alpha_j p^j \mod r$ für ein $i \geq 0$ und mit $0 \leq \alpha_j < p$ für alle $0 \leq j < u$. Dann ist laut Satz 6.18 von Lucas

$$\binom{i+r-1}{r-1} \equiv \prod_{j=0}^{u-1} \binom{\alpha_j}{p-1} \mod p.$$

Ist nun $i \not\equiv 0 \mod r$, so existiert ein j mit der Eigenschaft $\alpha_j < p - 1$. In dem Fall ist $\binom{\alpha_j}{p-1} = 0$ und somit $\binom{i+r-1}{r-1} \equiv 0 \mod p$.

Ist hingegen $i \equiv 0 \mod r$, so ist $\alpha_j = p - 1$ für alle $0 \leq j < u$. Also ist $\binom{\alpha_j}{p-1} = 1$ für alle $0 \leq j < u$ und somit $\binom{i+r-1}{r-1} \equiv 1 \mod p$.

Daraus ergibt sich

$$H^{(r-1)}(y) = \sum_{i=0}^{\infty} s_{ir}\, x^{-ir-r}.$$

□

Diese Begrifflichkeiten und Sätze reichen aus, um in meinen Bemühungen um die Gewinnung der Matrix $N_r(f)$ fortfahren zu können:

Wir waren ausgegangen von der Gleichung (10), welche $H^{(r-1)}(y) = y^r$ lautete und sich auch schreiben läßt als $f^r\, H^{(r-1)}\left(\frac{h}{f}\right) = h^r$. Diese Gleichung wollen wir zu dem algebraischen Gleichungssystem (15), $N_r(f)\mathbf{h} = \mathbf{h}^r$, umformen.

Satz 6.20
Es ist $N_r(f) = [\Delta_r]_{E_1, E_1'}$ *mit* $E_1 = E_1(f) = \{\omega_1, \dots, \omega_{n-1}\}$ *und* $E_1' = E_1(f_r)$.

Beweis: Sei $N_r(f) = (\nu_{jk})_{0 \leq j,k < n}$. Die zu E_1 gehörige Matrix ist $E_1 = [\omega_0, \omega_1, \dots, \omega_{n-1}]$. Die Behauptung ist nun, daß

$$[\Delta_r \omega_k]_{E_1'} = (\nu_{0,k}, \nu_{1,k}, \dots, \nu_{n-1,k})^\top$$

gilt. $h(x) = \sum_{k=0}^{n-1} h_k x^k$ mit $h_k \in \mathbb{F}_q$ liefert mir

$$f(x)^r H^{(r-1)}\left(\frac{h(x)}{f(x)}\right) = \sum_{k=0}^{n-1} f(x)^r h_k H^{(r-1)}\left(\frac{x^k}{f(x)}\right).$$

Darum ist also ν_{jk} der Koeffizient von x^{jr} in $f(x)^r H^{(r-1)}\left(\frac{x^k}{f(x)}\right)$. Wegen $f^r H^{(r-1)}\left(\frac{h}{f}\right) = h^r$ ist

$$f(x)^r H^{(r-1)}\left(\frac{x^k}{f(x)}\right)$$

ein Polynom in x^r mit einem $\mathrm{Grad} \leq (n-1)r$, deshalb gilt

$$f(x)^r H^{(r-1)}\left(\frac{x^k}{f(x)}\right) = \sum_{j=0}^{n-1} n_{j,k} x^{jr} \text{ für } 0 \leq k < n.$$

Ist y die erzeugende Funktion einer Folge $(s_i) \in S(f)$ und z die erzeugende Funktion von $\Delta_r(s_i) = (s_{ir})$, so ist nach Satz 6.19

$$z(x^r) = H^{(r-1)}(y(x)).$$

Sei nun $y_k(x) = \frac{x^k}{f(x)}$ für $0 \leq k < n$ jeweils die erzeugende Funktion von ω_k, $z_k(x)$ sei die erzeugende Funktion von $\Delta_r(\omega_k)$. Die $\omega_k' = \frac{x^k}{f_r(x)}$ bilden für $0 \leq k < n$ die kanonische Basis bezüglich f_r. Dann gilt

$$z_k(x^r) = H^{(r-1)}(y_k(x)) = H^{(r-1)}\left(\frac{x^k}{f(x)}\right) = \frac{1}{f(x)^r}\sum_{j=0}^{n-1} \nu_{j,k} x^{jr} = \frac{1}{f_r(x^r)}\sum_{j=0}^{n-1} \nu_{j,k} x^{jr},$$

also ist

$$z_k(x) = \frac{1}{f_r(x)}\sum_{j=0}^{n-1} \nu_{j,k} x^j$$

für $0 \leq k < n$. Wegen Gleichung $\frac{h(x)}{f(x)} = \sum_{k=0}^{n-1} h_k \frac{x^k}{f(x)}$ und $\sigma = \sum_{k=0}^{n-1} h_k \omega_k$ (siehe dazu auf Seite 84) angewendet auf f_r hat die Folge $\Delta_r(\omega_k)$ bezüglich der Basis E_1' die Darstellung

$$\Delta_r(\omega_k) = \sum_{j=0}^{n-1} \nu_{j,k} \omega_j'.$$

$\square$

Die in Satz 6.20 gemachte Aussage läßt sich auch expliziter ausdrücken. Dafür definiere ich drei weitere $(n \times n)$–Matrizen über $\mathbb{F}_q$:

Definition 6.21

- *Die Matrix* $G(f_r)$ *erhalte ich aus den Koeffizienten des Polynoms* $f_r(x) = \sum_{i=0}^{n} a_i^r x^i$*, indem ich*

$$G(f_r) = \begin{bmatrix} a_1^r & a_2^r & a_3^r & \dots & a_{n-1}^r & a_n^r \\ a_2^r & a_3^r & a_4^r & \dots & a_n^r & 0 \\ a_3^r & a_4^r & a_5^r & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ a_n^r & 0 & 0 & \dots & 0 & 0 \end{bmatrix}$$

bilde.

- *Die Berlekamp–Matrix* $B_{r,v}(f) = ((q_{jk}))_{0\le j,k<n}$*, die ich bereits in Abschnitt 4.1 mit* $r = q$ *vorgestellt habe, ist durch die folgenden Kongruenzen gegeben:*

$$x^{jr} \equiv \sum_{k=0}^{n-1} q_{jk} x^k \mod f(x) \text{ für } 0 \le j < n.$$

- *Die Hankel–Matrix* $U(f) = ((u_{j+k}))_{0\le j,k<n}$*, wobei* $\omega = (u_i)$ *die Impuls–Antwort–Folge für das charakteristische Polynom* f *ist.*

Jetzt kann ich die explizite Formel für $N_r(f)$ aufstellen:

Satz 6.22
Es ist $N_r(f) = G(f_r)B_{r,v}(f)U(f)$.

Beweis: Seien E_1 und E_1' die kanonischen Basen von $S(f)$ und $S(f_r)$, E_2 und E_2' die Standardbasen von $S(f)$ und $S(f_r)$, I und I' die Identitäten auf $S(f)$ und $S(f_r)$. Dann weiß ich aus Satz 6.20, daß $N_r(f) = [\Delta_r]_{E_1,E_1'}$. Dann ist also

$$N_r(f) = [I']_{E_2',E_1'}[\Delta_r]_{E_2,E_2'}[I]_{E_1,E_2}.$$

Ich werde zeigen, daß die Gleichungen

1. $[I]_{E_1,E_2} = U(f)$,
2. $[\Delta_r]_{E_2,E_2'} = B_{r,v}(f)$ und
3. $[I']_{E_2',E_1'} = G(f_r)$,

gelten.

zu 1. Ich rechne nach:

$$[I]_{E_1,E_2} = E_2^{-1} I E_1 = E_1 = \begin{bmatrix} u_0 & u_1 & \dots & u_{n-1} \\ u_1 & u_2 & \dots & u_n \\ \vdots & \vdots & & \vdots \\ u_{n-1} & u_n & \dots & u_{2n-2} \end{bmatrix} = U(f).$$

zu 2. Ich betrachte ε_k aus der geordneten Basis E_2 für $0 \le k < n$, ε_k sei die Folge (t_i). Für alle $0 \le j < n$ ist

$$\chi_j(x) = x^{jr} - \sum_{l=0}^{n-1} q_{jl} x^l \equiv 0 \mod f(x)$$

ein assoziiertes charakteristisches Polynom von ε_k, darum gilt

$$t_{i+jr} = \sum_{l=0}^{n-1} q_{jl} t_{i+l}$$

für alle $i \geq 0$ und $0 \leq j < n$.
ε_k hat den k–ten Einheitsvektor als Anfangswertvektor, also 1 als k–ten Eintrag und sonst 0. Darum ist für $0 \leq j < n$

$$t_{jr} = \sum_{l=0}^{n-1} q_{jl} t_l = q_{jk}.$$

Wegen Lemma 6.17 ist $[\Delta_r(\sigma)]_{E_2'} = (s_0, s_r, s_{2r}, \ldots, s_{(n-1)r})^\top$. Darum gilt für alle $\varepsilon_k \in E_2$ mit $0 \leq k < n$

$$[\Delta_r(\varepsilon_k)]_{E_2'} = (q_{0,k}, q_{1,k}, \ldots, q_{n-1,k})^\top,$$

also ist

$$[\Delta_r]_{E_2,E_2'} = \begin{bmatrix} q_{0,0} & q_{0,1} & \cdots & q_{0,n-1} \\ q_{1,0} & q_{1,1} & \cdots & q_{1,n-1} \\ \vdots & \vdots & & \vdots \\ q_{n-1,0} & q_{n-1,1} & \cdots & q_{n-1,n-1} \end{bmatrix} = B_{r,v}(f).$$

zu 3. Es gilt $[I']_{E_2',E_1'} = E_1'^{-1} I' E_2' = E_1' - 1 E_2'$. Sei $E_2' = \{\delta_0, \delta_1, \ldots, \delta_{n-1}\}$, so ist jeweils die erzeugende Funktion ν_k von δ_k gegeben durch

$$\nu_k = x^{-k-1} + \sum_{i=n}^{\infty} s_i x^{-i-1},$$

wobei $s_i \in \mathbb{F}_q$ für alle $i \geq n$. Da wegen $\frac{h(x)}{f(x)} = \sum_{k=0}^{n-1} h_k \frac{x^k}{f(x)}$ der Ausdruck $f_r \cdot \nu_k$ ein Polynom mit einem $\mathrm{Grad} \leq n$ sein muß, gilt

$$\begin{aligned} f_r \cdot \nu_k &= \left(\sum_{j=0}^{n} a_j^r x^j\right)\left(x^{-k-1} + \sum_{i=n}^{\infty} s_i x^{-i-1}\right) = \sum_{j=0}^{n} a_j^r x^{j-k-1} \\ &= \sum_{j=k+1}^{n} a_j^r x^{j-k-1} = \sum_{j=0}^{n-k-1} a_{j+k+1}^r x^j. \end{aligned}$$

Darum ist $[\delta_k]_{E_1'} = (a_{k+1}^r, a_{k+2}^r, \ldots, a_n^r, 0, \ldots, 0)^\top$ für $0 \leq k < n$ und somit ist

$$[I']_{E_2',E_1'} = \begin{bmatrix} a_1^r & a_2^r & \ldots & a_n^r \\ a_2^r & a_3^r & \ldots & 0 \\ \vdots & \vdots & & \vdots \\ a_n^r & 0 & \ldots & 0 \end{bmatrix}.$$

Insgesamt ergibt sich also gerade die Behauptung, daß $N_r(f) = G(f_r) B_{r,v}(f) U(f)$ ist. □

Ich kann also das zu (10) äquivalente Gleichungssystem

$$N_r(f)\mathbf{h} = \mathbf{h}^r$$

aufstellen, wobei $\mathbf{h} = (h_0, \ldots, h_{n-1})^\top \in \mathbb{F}_q^n$ der Koeffizientenvektor von h und $\mathbf{h}^r = (h_0^r, \ldots, h_{n-1}^r)^\top$ ist. Ich kann das Gleichungssystem aber noch einfacher gewinnen:

Lemma 6.23
Für $r = q$ gilt $G(f_q)^{-1} = U(f)$.

Beweis: Da f normiert und darum $a_n = 1$ ist, muß die Matrix $G(f_q)$ als Dreiecksmatrix mit a_n auf der Gegendiagonalen invertierbar sein. Es ist $U(f) = [I]_{E_1,E_2}$ und $G(f_q) = [I]_{E_2,E_1}$, da $f_q = f$. □

Im Fall $r = q$ kann also $G(f_q)$ als Basiswechselmatrix aufgefaßt werden und es ergibt sich

$$N_q(f) = G(f_q)B_{q,1}(f)G(f_q)^{-1}.$$

Ist aber $r < q$, so muß das algebraische Gleichungssystem mit Hilfe von Normalbasen in die Form eines linearen Gleichungssystems gebracht und dann gelöst werden: Es sei $r = p^u$ mit $u \in \mathbb{N} \setminus \{0\}$, $u \mid s$. Ich betrachte eine feste Basis $B = \{\alpha_0, \alpha_1, \alpha_2, \ldots, \alpha_{s/u-1}\}$ von $\mathbb{F}_q$ über $\mathbb{F}_r$, man wählt meist eine Normalbasis, da man denkt, daß die Behandlung des Systems dadurch erleichtert wird. Dann schreibe ich die Einträge der Matrix $N_r(f)$ sowie die Unbekannten h_i für $0 \leq i < n$ als Linearkombinationen der Elemente von B mit Koeffizienten aus $\mathbb{F}_r$. Für $0 \leq i < n$ seien $h_i^{(j)} \in \mathbb{F}_q$ für $0 \leq j < s/u$ die Koeffizienten aus der Darstellung der h_i bezüglich der Basis B:

$$h_i = \sum_{j=0}^{\frac{s}{u}-1} h_i^{(j)} \alpha_j.$$

Ich benutze diese Linearkombinationen in der Gleichung (15), $N_r(f)\mathbf{h} = \mathbf{h}^r$, und drücke jedes Produkt von Elementen von B auf der linken Seite der Gleichung aus als Linearkombination der Elemente von B. Schließlich nehme ich einen Vergleich der Koeffizienten der Elemente von B auf den beiden Seiten der sich ergebenden Gleichung vor. So gelange ich zu einem System homogener linearer Gleichungen

$$K_{q,r}(f, B)\mathbf{H} = 0 \in \mathbb{F}_q^{ns/u},$$

wobei $K_{q,r}(f, B)$ eine $(\frac{ns}{u} \times \frac{ns}{u})$–Matrix über $\mathbb{F}_r$ und $\mathbf{H} \in \mathbb{F}_r^{ns/u}$ der Spaltenvektor ist, der die unbekannten $h_i^{(j)}$ enthält. Dies ist die gewünschte Linearisierung, die nach der Wahl der Basis B eindeutig ist. Die Matrix $K_{q,r}(f, B)$ hat den Rang $(ns/u) - t$.

Algorithmus 24 NIEDERREITER $(f, \mathbb{F}_q)$
Eingabe: Das normierte, quadratfreie Polynom $f \in \mathbb{F}_q[x]$ und der Körper $\mathbb{F}_q$.
Ausgabe: Die Faktorisierung von f.
Programm: Nieder.mws: nieder

1. Setze $n \leftarrow \mathrm{Grad}(f)$ und berechne $G \leftarrow G(f_r)$, $B \leftarrow B_{r,v}(f)$ und $U \leftarrow U(f)$. Setze $N \leftarrow G(f_r) \cdot B_{r,v} \cdot U(f)$.
2. Berechne eine Basis $orth$ des Lösungsraums der Gleichung $N\mathbf{h} = \mathbf{h}^r$ und interpretiere die Basiselemente als Basispolynome des Raumes $\mathcal{N}_r(f)$. Die Basis ist eine Liste $h_1(x), \ldots, h_t(x)$ von Polynomen.
3. Setze $i \leftarrow 1$.
4. Berechne für alle bisher gefundenen Faktoren $g(x)$ von $f(x)$ die partielle Faktorisierung $g(x) = \prod_{\gamma \in \mathbb{F}_q} \mathrm{ggT}(g(x), h_i(x) - \gamma f'(x))$.
5. Ist die Anzahl der bisher gefundenen Faktoren von $f(x)$ kleiner als t, so setze $i \leftarrow i+1$ und gehe zu Schritt 4.
6. Liefere die Faktorisierung von $f(x)$ zurück.

Beispiel 6.2
Betrachte

$$f(x) = x^{12} + 4x^{11} + 2x^{10} + 2x^9 + 6x^6 + 3x^4 + 3x^3 + 4x^2 + 5 \in \mathbb{F}_7[x].$$

Man arbeitet mit der Differentialgleichung $H^{(6)}(h/f) = h^7/f^7$ *und den Matrizen*

$$G(f_7) = \begin{bmatrix} 0 & 4 & 3 & 3 & 0 & 6 & 0 & 0 & 2 & 2 & 4 & 1 \\ 4 & 3 & 3 & 0 & 6 & 0 & 0 & 2 & 2 & 4 & 1 & 0 \\ 3 & 3 & 0 & 6 & 0 & 0 & 2 & 2 & 4 & 1 & 0 & 0 \\ 3 & 0 & 6 & 0 & 0 & 2 & 2 & 4 & 1 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 & 2 & 2 & 4 & 1 & 0 & 0 & 0 & 0 \\ 6 & 0 & 0 & 2 & 2 & 4 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 2 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 2 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 2 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$B_{7,1}(f) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 6 & 2 & 2 & 1 & 2 & 4 & 3 & 1 & 0 & 1 & 6 \\ 0 & 2 & 0 & 0 & 3 & 6 & 1 & 1 & 0 & 6 & 6 & 4 \\ 4 & 3 & 1 & 2 & 2 & 4 & 2 & 1 & 1 & 0 & 5 & 1 \\ 3 & 3 & 2 & 0 & 5 & 3 & 5 & 2 & 4 & 6 & 1 & 0 \\ 3 & 6 & 6 & 3 & 5 & 6 & 4 & 0 & 0 & 4 & 0 & 3 \\ 4 & 4 & 1 & 4 & 6 & 1 & 4 & 0 & 5 & 3 & 4 & 0 \\ 0 & 0 & 1 & 0 & 0 & 3 & 5 & 6 & 0 & 3 & 5 & 2 \\ 6 & 4 & 3 & 4 & 5 & 4 & 0 & 2 & 0 & 5 & 2 & 1 \\ 4 & 0 & 2 & 1 & 3 & 2 & 1 & 2 & 6 & 2 & 1 & 6 \\ 6 & 0 & 2 & 3 & 1 & 0 & 5 & 0 & 2 & 4 & 3 & 3 \end{bmatrix},$$

$$U(f) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 3 & 0 & 6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 3 & 0 & 6 & 5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 3 & 0 & 6 & 5 & 3 \\ 0 & 0 & 0 & 0 & 0 & 1 & 3 & 0 & 6 & 5 & 3 & 2 \\ 0 & 0 & 0 & 0 & 1 & 3 & 0 & 6 & 5 & 3 & 2 & 0 \\ 0 & 0 & 0 & 1 & 3 & 0 & 6 & 5 & 3 & 2 & 0 & 1 \\ 0 & 0 & 1 & 3 & 0 & 6 & 5 & 3 & 2 & 0 & 1 & 0 \\ 0 & 1 & 3 & 0 & 6 & 5 & 3 & 2 & 0 & 1 & 0 & 4 \\ 1 & 3 & 0 & 6 & 5 & 3 & 2 & 0 & 1 & 0 & 4 & 4 \end{bmatrix} \text{ und}$$

$$N_7(f) = G(f_7) \cdot B_{7,1}(f) \cdot U(f) = \begin{bmatrix} 0 & 6 & 2 & 5 & 2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 3 & 6 & 4 & 5 & 0 & 4 & 5 & 0 & 6 & 2 & 5 & 2 \\ 4 & 1 & 6 & 4 & 5 & 2 & 2 & 3 & 6 & 4 & 5 & 0 \\ 2 & 0 & 3 & 4 & 6 & 4 & 3 & 4 & 1 & 6 & 4 & 5 \\ 0 & 2 & 2 & 3 & 6 & 0 & 0 & 2 & 0 & 3 & 4 & 6 \\ 6 & 2 & 6 & 6 & 0 & 6 & 1 & 0 & 2 & 2 & 3 & 6 \\ 3 & 2 & 4 & 5 & 3 & 0 & 1 & 6 & 2 & 6 & 6 & 0 \\ 1 & 6 & 5 & 4 & 3 & 5 & 6 & 3 & 2 & 4 & 5 & 3 \\ 0 & 3 & 1 & 1 & 3 & 1 & 2 & 1 & 6 & 5 & 4 & 3 \\ 6 & 5 & 3 & 2 & 4 & 6 & 0 & 0 & 3 & 1 & 1 & 3 \\ 0 & 0 & 0 & 0 & 1 & 3 & 0 & 6 & 5 & 3 & 2 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Der Nullraum der Matrix $N_7(f) - E$ ist durch die Basiselemente

$$(4,6,3,1,3,3,0,3,1,0,0,0)^\top,$$
$$(5,1,4,2,1,2,0,4,0,0,0,1)^\top,$$
$$(2,1,4,1,3,1,5,2,0,1,0,0)^\top \textit{ und}$$
$$(5,4,1,3,0,4,6,6,0,0,1,0)^\top$$

bestimmt, rechne also mit den Basispolynomen

$$\begin{aligned}
h_1(x) &= x^8+3x^7+3x^5+3x^4+x^3+3x^2+6x+4,\\
h_2(x) &= x^{11}+4x^7+2x^5+x^4+2x^3+4x^2+x+5,\\
h_3(x) &= x^{10}+6x^7+6x^6+4x^5+3x^3+x^2+4x+5 \textit{ und}\\
h_4(x) &= x^9+2x^7+5x^6+x^5+3x^4+x^3+4x^2+x+2
\end{aligned}$$

des Lösungsraums der Differentialgleichung weiter. Bestimme die Ableitung von f als

$$f'(x) = 5x^{11}+2x^{10}+6x^9+4x^8+x^5+5x^3+2x^2+x.$$

Starte mit h_1 und versuche, durch das Bilden von $\mathrm{ggT}(f(x), h_1(x)-\gamma f'(x))$ für alle $\gamma \in \mathbb{F}_7$ eine partielle Faktorisierung von f zu erreichen:

- *Es ist*

  $$\begin{aligned}
  b_1(x) &= \mathrm{ggT}(f(x), h_1(x)-3f'(x)) &&= x^3+6x^2+4x+4,\\
  b_2(x) &= f(x)/b_1(x) &&= x^9+5x^8+3x^7+2x^6+5x^5+6x^4+5x^3+3x^2+4x+3,
  \end{aligned}$$

  rechne weiter mit $f(x) = (x^3+6x^2+4x+4)\cdot b_2(x)$.

- *Es ist*

  $$\begin{aligned}
  b_3(x) &= \mathrm{ggT}(b_2(x), h_1(x)-4f'(x)) = x^3+6x^2+4x+1,\\
  b_4(x) &= b_2(x)/b_3(x) \qquad\qquad\quad = x^6+6x^5+5x^4+3x^3+3x^2+6x+3,
  \end{aligned}$$

  rechne weiter mit $f(x) = (x^3+6x^2+4x+4)(x^3+6x^2+4x+1)\cdot b_4(x)$.

Alle weiteren γ liefern keine neuen Faktoren von b_4, fahre darum mit h_2 fort:

- *Es ist*

  $$\begin{aligned}
  b_5(x) &= \mathrm{ggT}(b_4(x), h_2(x)-2f'(x)) = x^3+4,\\
  b_6(x) &= b_4(x)/b_5(x) \qquad\qquad\quad = x^3+6x^2+5x+6.
  \end{aligned}$$

Insgesamt ist darum $f(x) = (x^3+6x^2+4x+4)(x^3+6x^2+4x+1)(x^3+4)(x^3+6x^2+5x+6)$.

Komplexität

Ich bestimme die Komplexität für $r = q$.

Um den Lösungsraum zu finden, müssen die Matrizen $G(f_q)$, $B_{q,1}(f)$ und $U(f)$ bestimmt und miteinander multipliziert werden, danach wird eine Gaußelimination durchgeführt.

Für die Matrix $G(f_q)$ müssen n Elemente des $\mathbb{F}_q$ mit q potenziert werden, dies passiert jeweils in $\log(q)$ elementaren Multiplikationen. $G(f_q)$ kostet also $n\log(q)$ elementare Operationen.

Die Matrix $B_{q,1}(f)$ zu bestimmen kostet bekanntlich $(2\log_2(q)+n)\cdot(3n^2+7n+4)$ Operationen.

Für $U(f)$ müssen $n-1$ Werte der Impuls–Antwort–Folge bestimmt werden. Einen solchen Wert zu bestimmen kostet n Multiplikationen und n Additionen, also kostet die Matrix $U(f)$ höchstens

$2n(n-1) = 2n^2 - 2n$ elementare Operationen.

Bei $G(f_q)$ und $U(f)$ handelt es sich um Dreiecksmatrizen, darum geht das Multiplizieren zweier Matrizen hier schneller als im Normalfall. Ich beschreibe das Beispiel $G(f_q) \cdot B_{q,1}(f)$, die anschließende Multiplikation mit $U(f)$ läuft analog. Die erste Zeile von $G(f_q)$ hat n Einträge, die Multiplikation mit der ersten Spalte von $B_{q,1}(f)$ kostet somit n elementare Multiplikationen und $n-1$ elementare Additionen. Die zweite Zeile von $G(f_q)$ hat nur $n-1$ Einträge, es werden also nur $n-1+n-2$ elementare Operationen benötigt und so weiter. Um $G(f_q)$ mit der ersten Spalte von $B_{q,1}(f)$ zu multiplizieren, braucht man also $\frac{1}{2}(n(n+1)+n(n-1)) = n^2$ elementare Operationen. Da $B_{q,1}(f)$ insgesamt n Spalten hat, dauert hier also eine Multiplikation zweier Matrizen n^3 elementare Operationen — die Multiplikation zweier beliebiger $(n \times n)$–Matrizen kostet hingegen $2n^3 - 2n^2$ Operationen.

Den Lösungsraum der Matrix mit Gaußelimination zu bestimmen kostet wie gehabt $2n^3 - 2n^2$ Operationen.

Bestimme nun in n elementaren Multiplikationen die gewöhnliche Ableitung von f.

Die Vorbereitungen für die Suche nach Faktoren von f zu treffen kostet insgesamt also

$n\log(q) + (2\log(q) + n)(3n^2 + 7n + 4) + 2n^2 - 2n + 2n^3 + 2n^3 - 2n^2 + n$ Operationen.

Jedes Basiselement des Lösungsraums entspricht einem Polynom h, für welches $q-1$ Polynome auf das Liefern eines Faktors von f geprüft werden. Dies passiert wie folgt:

Die Ableitung f' von f wird in n elementaren Additionen von $h - \gamma f'$ abgezogen, das Ergebnis ist $h - (\gamma+1)f'$. Dann wird in $2n^2 + 3n$ Operationen der größte gemeinsame Teiler gebildet.

Die Suche nach Faktoren von f kostet also $t(q-1)(n + 2n^2 + 3n)$ Operationen. Darum braucht der Algorithmus von Niederreiter insgesamt

$n\log(q) + (2\log(q) + n)(3n^2 + 7n + 4) + 2n^2 - 2n + 2n^3 + 2n^3 - 2n^2 + n + t(q-1)(n + 2n^2 + 3n)$ elementare Operationen, hat also die Komplexität $\mathcal{O}(n^3 + tn^2q)$.

6.3 Niederreiters Algorithmus bei gerader Charakteristik

Satz 6.24
Ist $f(x) = \sum_{i=0}^{n} a_i x^i \in \mathbb{F}_{2^s}[x]$, so ist

$$N_2(f) = \underbrace{\begin{bmatrix} a_1 & a_0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ a_3 & a_2 & a_1 & a_0 & 0 & 0 & 0 & \dots & 0 & 0 \\ a_5 & a_4 & a_3 & a_2 & a_1 & a_0 & 0 & \dots & 0 & 0 \\ \vdots & & & & & & \ddots & & & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & a_n & a_{n-1} & a_{n-2} \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & a_n \end{bmatrix}}_{n \text{ Spalten}} \left.\vphantom{\begin{bmatrix} a \\ a \\ a \\ \vdots \\ a \\ a \end{bmatrix}}\right\} n \text{ Zeilen}.$$

Beweis: Ist $r = 2$, so hat die Differentialgleichung (10) die spezielle Form

$$\begin{aligned} & \left(\frac{h(x)}{f(x)}\right)' = -\left(\frac{h(x)}{f(x)}\right)^2 \\ \Leftrightarrow \quad & (f(x) \cdot h(x)))' = h(x)^2. \end{aligned}$$

Man stellt einen Koeffizientenvergleich für die Koeffizienten von x^{2j} mit $0 \leq j < n$ an. Der Koeffizient von x^{2j} in $(fh)'$ ist gerade der Koeffizient von x^{2j+1} in fh. Ist $f(x) = \sum_{i=0}^{n} a_i x^i$ und

$h(x) = \sum_{i=0}^{n-1} h_i x^i$, so ist $f(x)h(x) = \sum_{j=0}^{2n-1} x^j \left(\sum_{k=\max(j-n,0)}^{\min(j,n-1)} a_{j-k} h_k\right)$, also hat die Gleichung (15) hier die Form

$$\sum_{k=\max(2j+1-n,0)}^{\min(2j+1,n-1)} a_{2j+1-k} h_k = h_j \text{ für } 0 \leq j < n.$$

□

Durch diesen Satz ist klar, daß Niederreiters Algorithmus für den Fall $r = 2$ weniger komplex gestaltet werden kann als im allgemeinen, da sich die Matrix $N_2(f)$ ohne Probleme aus den Koeffizienten von f direkt hinschreiben läßt — Matrixmultiplikationen und die Aufstellung der zu multiplizierenden Matrizen werden eingespart. Im Fall $p = 2$ wähle man also auch $r = 2$.

Algorithmus 25 NIEDERREITER $(f, \mathbb{F}_q)$
Eingabe: Das normierte, quadratfreie Polynom $f \in \mathbb{F}_{2^s}[x]$ und der Körper $\mathbb{F}_q$ mit $q = 2^s$.
Ausgabe: Die Faktorisierung von f.
Programm: Nieder.mws: nieder

1. Setze $n \leftarrow \mathrm{Grad}(f)$ und berechne $N \leftarrow N_2(f)$ durch einsetzen der Koeffizienten von $f(x)$.
2. Berechne eine Basis $orth$ des Lösungsraums der Gleichung $N\mathbf{h} = \mathbf{h}^2$ und interpretiere die Basiselemente als Basispolynome des Raumes $\mathcal{N}_2(f)$. Die Basis ist eine Liste $h_1(x), \ldots, h_t(x)$ von Polynomen.
3. Setze $i \leftarrow 1$.
4. Berechne für alle bisher gefundenen Faktoren $g(x)$ von $f(x)$ die partielle Faktorisierung $g(x) = \prod_{\gamma \in \mathbb{F}_q} \mathrm{ggT}(g(x), h_i(x) - \gamma f'(x))$.
5. Ist die Anzahl der bisher gefundenen Faktoren von $f(x)$ kleiner als t, so setze $i \leftarrow i+1$ und gehe zu Schritt 4.
6. Liefere die Faktorisierung von $f(x)$ zurück.

Beispiel 6.3
Sei $\mathbb{F}_4 = \mathbb{F}_2[\xi]/\left((\xi^2+\xi+1)\cdot\mathbb{F}_2[\xi]\right)$ *und betrachte*

$$f(x) = x^9 + \xi x^8 + (\xi+1)x^7 + \xi x^5 + (\xi+1)x^4 + (\xi+1)x + 1 \in \mathbb{F}_4[x].$$

Man arbeitet mit der Differentialgleichung $H^{(1)}(h/f) = h^2/f^2$ *und der Matrix*

$$N_2(f) = \begin{bmatrix} \xi+1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \xi+1 & 1 & 0 & 0 & 0 & 0 & 0 \\ \xi & \xi+1 & 0 & 0 & \xi+1 & 1 & 0 & 0 & 0 \\ \xi+1 & 0 & \xi & \xi+1 & 0 & 0 & \xi+1 & 1 & 0 \\ 1 & \xi & \xi+1 & 0 & \xi & \xi+1 & 0 & 0 & \xi+1 \\ 0 & 0 & 1 & \xi & \xi+1 & 0 & \xi & \xi+1 & 0 \\ 0 & 0 & 0 & 0 & 1 & \xi & \xi+1 & 0 & \xi \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \xi & \xi+1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

die sich direkt von f *abschreiben läßt und ein algebraisches Gleichungssystem beschreibt. Um dieses zu lösen, erzeugt man sich wie in Beispiel 4.2 auf Seite 34 ein lineares Gleichungssystem, indem man mit der*

Normalbasis $B = \{\xi, \xi + 1\}$ *arbeitet. Sei also*

$$K_{4,2}(f, B) = \begin{bmatrix} 1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&1&1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&1&1&1&0&0&0&0&1&1&1&0&0&0&0&0&0&0 \\ 1&1&1&0&0&0&0&0&1&0&0&1&0&0&0&0&0&0 \\ 1&1&0&0&0&1&1&1&0&0&0&0&1&1&1&0&0&0 \\ 1&0&0&0&1&1&1&0&0&0&0&0&1&0&0&1&0&0 \\ 1&0&0&1&1&1&0&0&0&1&1&1&0&0&0&0&1&1 \\ 0&1&1&1&1&0&0&0&1&1&1&0&0&0&0&0&1&0 \\ 0&0&0&0&1&0&0&1&1&1&0&0&0&1&1&1&0&0 \\ 0&0&0&0&0&1&1&1&1&0&0&0&1&1&1&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0&1&1&1&0&0&0&1 \\ 0&0&0&0&0&0&0&0&0&1&1&1&1&0&0&0&1&1 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&1&1 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1 \end{bmatrix}$$

die Matrix $N_2(f)$ *bezüglich der Normalbasis,* A *sei die* (18×18)*–Matrix mit den Blöcken* $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ *auf der Diaginalen.* A *führt bezüglich der Normalbasis das Quadrieren aus, das lineare Gleichungssystem* $K_{4,2}(f, B) \cdot h = A \cdot h$ *ist zu lösen. Es sind also die Elemente des Kerns der Matrix*

$$K_{4,2}(f, B) - A = \begin{bmatrix} 1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&1&1&1&0&1&0&0&1&1&1&0&0&0&0&0&0&0 \\ 1&1&1&0&1&0&0&0&1&0&0&1&0&0&0&0&0&0 \\ 1&1&0&0&0&1&1&0&0&0&0&0&1&1&1&0&0&0 \\ 1&0&0&0&1&1&0&0&0&0&0&0&1&0&0&1&0&0 \\ 1&0&0&1&1&1&0&0&0&0&1&1&0&0&0&0&1&1 \\ 0&1&1&1&1&0&0&0&0&1&1&0&0&0&0&0&1&0 \\ 0&0&0&0&1&0&0&1&1&1&0&1&0&1&1&1&0&0 \\ 0&0&0&0&0&1&1&1&1&0&1&0&1&1&1&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0&1&1&0&0&0&0&1 \\ 0&0&0&0&0&0&0&0&0&1&1&1&0&0&0&0&1&1 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&1&1&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1 \end{bmatrix}$$

zu bestimmen. Eine Basis des Nullraums sind die Vektoren

$(1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0)^\top$,
$(0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)^\top$ *und*
$(0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1)^\top$,

eine Basis des Lösungsraums der Differentialgleichung ist also die Menge $H = \{h_1, h_2, h_3\}$ *mit*

$$\begin{aligned} h_1(x) &= \xi x^7 + x^5 + \xi x^4 + \xi x^3 + \xi x^2 + \xi x + 1, \\ h_2(x) &= (\xi + 1)x^3 + x^2 + \xi + 1 \text{ und} \\ h_3(x) &= x^8 + (\xi + 1)x^6 + \xi x^4 + \xi + 1. \end{aligned}$$

Die gewöhnliche Ableitung von f ist

$$f'(x) = x^8 + (\xi+1)x^6 + \xi x^4 + \xi + 1,$$

berechne nun nacheinander $\mathrm{ggT}(f, h_i - \gamma f')$ *für alle* $i \in \{1,2,3\}$ *und* $\gamma \in \mathbb{F}_2$*:*

- *Es ist*

$$\begin{aligned} & & \mathrm{ggT}\,(f(x), h_1(x)) &= x^3 + \xi x + 1, \\ b(x) &= & \mathrm{ggT}\,(f(x), h_1(x) - f'(x)) &= x^6 + \xi x^5 + x^4 + \xi x^3 + \xi x^2 + x + 1, \end{aligned}$$

  rechne weiter mit $f(x) = (x^3 + \xi x + 1)b(x)$.

- *Es ist*

$$\mathrm{ggT}\,(b(x), h_2(x)) = x^3 + \xi x^2 + 1,$$

insgesamt ist somit $f(x) = (x^3 + \xi x + 1) \cdot (x^3 + \xi x^2 + 1) \cdot (x^3 + x + 1)$.

Komplexität

Ich behandle den Fall $r = q = 2$. Er verläuft genau wie bei ungerader Charakteristik, bis auf die Tatsache, daß das Aufstellen der Matrix $N_2(f)$ keinen Rechenaufwand verursacht.

Das Durchführen der Gaußelimination kostet $2n^3 - 2n^2$ Operationen, f muß nicht abgeleitet werden und die Suche nach Faktoren kostet $t(2n^2 + 3n)$ elementare Operationen.

Bei $q = 2$ kostet der Algorithmus also $2n^3 - 2n^2 + t(2n^2 + 3n)$ Operationen, hat also die Komplexität $\mathcal{O}(n^3 + tn^2)$.

6.4 Vergleich der Algorithmen von Berlekamp und Niederreiter

Ein Vergleich der beiden Verfahren macht nur Sinn, wenn wir sie für dieselbe Art von Polynomen betrachten. Beim Algorithmus von Berlekamp muß das zu faktorisierende Polynom quadratfrei sein, bei Niederreiter nicht, wir vergleichen die Effizienz also nur bei quadratfreien Polynomen und halten als erstes Ergebnis fest, daß der Algorithmus von Niederreiter weniger Voraussetzungen benötigt als der von Berlekamp.

Seien c und b Polynome, die der Gleichung $c(x)f'(x) + b(x)f(x) = 1$ genügen, dann sei ψ der Isomorphismus, der durch $\psi\,(h(x)) \equiv c(x)h(x) \mod f(x)$ gegeben ist.

In Satz 6.26 auf Seite 97 werden wir sehen, daß es mit ψ möglich ist, eine Basis N des Raumes $\mathcal{N}(f,r)$ in eine Basis der Berlekamp–Unteralgebra $\mathcal{B}(f,r)$ umzuwandeln. Die Frage, die sich jetzt noch stellt ist, ob es Verfahren gibt, die zu den für den Algorithmus von Berlekamp entwickelten Verbesserungen analog sind und direkt auf N arbeiten.

Lemma 6.25
Seien f und h Polynome aus $\mathbb{F}_q[x]$ mit $\mathrm{Grad}(h) < \mathrm{Grad}(f) = n$. Dann gelten

1. $H^{(r-1)}\left(\frac{f'(x)}{f(x)}\right) - \left(\frac{f'(x)}{f(x)}\right)^r = 0,$
2. $H^{(k)}\left(\frac{h(x)}{f(x)}\right) \cdot f(x)^{k+1} \equiv (-1)^k \cdot h(x) \cdot (f'(x))^k \mod f(x)$ *und*
3. $\left(\frac{h(x)}{f(x)}\right)^{(k)} \cdot f(x)^{k+1} \equiv (-1)^k k! \cdot h(x) \cdot (f'(x))^k \mod f(x)$

Beweis: 1. Diese Aussage ist ein Spezialfall des Satzes 6.7 auf Seite 77 mit $\gamma_i = 1$ für alle $1 \leq i \leq t$.

2. Aus Lemma 6.6 auf Seite 76 ist bekannt, daß für alle $\mu, k \in \mathbb{N}$ gilt

$$H^{(k)}\left((x-\alpha)^{-\mu}\right) = \binom{-\mu}{k}(x-\alpha)^{-\mu-k}.$$

Sei $f^* = f_1 \cdot f_2 \cdots f_t$ der quadratfreie Teil von f, in dem jeder irreduzible Faktor f_i von f genau einmal vorkommt. Sei R die Menge der Nullstellen von f^* im Zerfällungskörper E von f^* über $\mathbb{F}_q$. Die Nullstellen von f sind genau die Elemente $\beta \in R$, jede dieser Nullstellen hat eine Vielfachheit $l(\beta)$. Sind zwei Nullstellen zueinander konjugiert, haben sie dieselbe Vielfachheit, und die Faktorisierung von f ist gegeben als $f(x) = \prod_{\beta \in R}(x-\beta)^{l(\beta)}$ in dem Zerfällungskörper E. Analog zu Satz 6.7 auf Seite 77 hat die Partialbruchzerlegung von h/f die Form

$$\frac{h}{f} = \sum_{\beta \in R} \sum_{i=1}^{l(\beta)} \frac{\gamma(\beta, i)}{(x-\beta)^i}$$

mit $\gamma(\beta, i)$ aus dem Zerfällungskörper von f^*. Darum ist

$$h = \sum_{\beta \in R} \sum_{i=1}^{l(\beta)} \gamma(\beta, i)(x-\beta)^{l(\beta)-i} \cdot \prod_{\alpha \in R \setminus \{\beta\}} (x-\alpha)^{l(\alpha)}$$

und es ergibt sich

$$\begin{aligned}
H^{(k)}\left(\frac{h(x)}{f(x)}\right) \cdot f(x)^{k+1} &= \sum_{\beta \in R} \sum_{i=1}^{l(\beta)} \binom{-i}{k} \frac{\gamma(\beta, i)}{(x-\beta)^{i+k}} \prod_{\alpha \in R} (x-\alpha)^{l(\alpha)\cdot(k+1)} \\
&= \sum_{\beta \in R} \sum_{i=1}^{l(\beta)} \binom{-i}{k} \gamma(\beta, i)(x-\beta)^{k(l(\beta)-1)+l(\beta)-i} \prod_{\alpha \in R \setminus \{\beta\}} (x-\alpha)^{l(\alpha)\cdot(k+1)} \\
&\equiv \sum_{i=1}^{l(\beta)} \binom{-i}{k} \gamma(\beta, i)(x-\beta)^{k(l(\beta)-1)+l(\beta)-i} \prod_{\alpha \in R \setminus \{\beta\}} (x-\alpha)^{l(\alpha)\cdot(k+1)} \mod (x-\beta)^{l(\beta)}
\end{aligned}$$

für alle $\beta \in R$. Somit ist

$$\begin{array}{rrcll}
& H^{(k)}\left(\dfrac{h(x)}{f(x)}\right) \cdot f(x)^{k+1} & \equiv & 0 & \mod (x-\beta)^{l(\beta)} \\
\Leftrightarrow & k(l(\beta)-1)+l(\beta)-i & \geq & l(\beta) & \text{für alle } 1 \leq i \leq l(\beta) \\
\Leftrightarrow & k(l(\beta)-1) & \geq & i & \text{für alle } 1 \leq i \leq l(\beta) \\
\Leftrightarrow & l(\beta) & > & 1 & \text{und} \quad k > 1.
\end{array}$$

Ist $k = 1$, so ist

$$H^{(k)}\left(\frac{h(x)}{f(x)}\right) \cdot f(x)^{k+1} \equiv -l(\beta)\gamma(\beta, l(\beta))(x-\beta)^{l(\beta)-1} \cdot (*) \mod (x-\beta)^{l(\beta)},$$

und ist $l(\beta) = 1$, so ist

$$H^{(k)}\left(\frac{h(x)}{f(x)}\right) \cdot f(x)^{k+1} \equiv \binom{-1}{k}\gamma(\beta, 1) \cdot (*) \mod (x-\beta)^{l(\beta)},$$

wobei

$$(*) \equiv \prod_{\alpha \in R \setminus \{\beta\}} (x-\alpha)^{l(\alpha)\cdot(k+1)} \mod (x-\beta)^{l(\beta)}$$

und wobei bekanntlich

$$\binom{-1}{k} = (-1)^k.$$

Nun wenden wir uns der rechten Seite der Gleichung in 2. zu:

$$\begin{aligned} h(x)(f'(x))^k &= \left(\sum_{i=1}^{l(\beta)} \gamma(\beta, i)(x-\beta)^{l(\beta)-i} \cdot \prod_{\alpha \in R \setminus \{\beta\}} (x-\alpha)^{l(\alpha)} \right) \\ &\quad \cdot \left(l(\beta)(x-\beta)^{l(\beta)-1} \prod_{\alpha \in R \setminus \{\beta\}} (x-\alpha)^{l(\alpha)} \right)^k \\ &\equiv \left(\sum_{i=1}^{l(\beta)} \gamma(\beta, i)(x-\beta)^{l(\beta)-i} \cdot \prod_{\alpha \in R \setminus \{\beta\}} (x-\alpha)^{l(\alpha)} \right) \\ &\quad \cdot l(\beta)^k (x-\beta)^{k \cdot (l(\beta)-1)} \prod_{\alpha \in R \setminus \{\beta\}} (x-\alpha)^{k \cdot l(\alpha)} \mod (x-\beta)^{l(\beta)} \end{aligned}$$

Somit ist $h(x)(f'(x))^k \equiv 0$ genau dann, wenn $(x-\beta)^{l(\beta)-i} \cdot (x-\beta)^{k \cdot (l(\beta)-1)}$ von $(x-\beta)^{l(\beta)}$ geteilt wird für alle $1 \leq i \leq l(\beta)$, also wenn

$$\begin{aligned} & l(\beta) - i + k \cdot (l(\beta)-1) \geq l(\beta) \quad \text{für alle } 1 \leq i \leq l(\beta) \\ \Leftrightarrow \quad & k > 1 \quad \text{und} \quad l(\beta) > 1. \end{aligned}$$

Ist $k = 1$, so ist

$$\begin{aligned} h(x)(f'(x)) &= \left(\sum_{i=1}^{l(\beta)} \gamma(\beta, i)(x-\beta)^{l(\beta)-i} \right) \cdot l(\beta)(x-\beta)^{(l(\beta)-1)} \cdot (*) \\ &\equiv l(\beta)\gamma(\beta, l(\beta))(x-\beta)^{l(\beta)-1} \cdot (*) \mod (x-\beta)^{l(\beta)} \end{aligned}$$

und ist $l(\beta) = 1$, so ist

$$\begin{aligned} h(x)(f'(x))^k &= \gamma(\beta, 1)(x-\beta)^{l(\beta)-1} \cdot l(\beta)^k (x-\beta)^{k \cdot (l(\beta)-1)} \cdot (*) \\ &\equiv \gamma(\beta, 1) \cdot (*) \mod (x-\beta)^{l(\beta)}. \end{aligned}$$

Daraus ergibt sich wie gefordert, daß linke und rechte Seite von 2. für jede Nullstelle β modulo $(x-\beta)^{l(\beta)}$ übereinstimmen, also auch modulo f.

3. Folgt aus 2., da für alle $y \in \mathbb{F}_q(x)$ gilt

$$y^{(k)} = k! \cdot H^{(k)}(y).$$

□

Satz 6.26
Sei $f \in \mathbb{F}_q[x]$ ein quadratfreies Polynom mit $\mathrm{Grad}(f) = n$, seien außerdem c und b Polynome, die der Gleichung $c(x)f'(x) + b(x)f(x) = 1$ entsprechen. Es sei die Abbildung ψ gegeben durch $\psi(h) \equiv ch \mod f$ für alle $h \in \mathbb{F}_q[x]$. Dann gilt

1. *Die Einschränkung von ψ auf $\mathcal{N}(f, r)$ ist ein $\mathbb{F}_r$–linearer Isomorphismus nach $\mathcal{B}(f, r)$.*
2. *Ist q ungerade, $h = \sum_{i=1}^t \gamma_i \frac{f}{f_i} f_i' \in \mathcal{N}(f, r)$ mit $\gamma_i \in \mathbb{F}_r$ und ist S die Menge der Quadrate in der multiplikativen Gruppe von $\mathbb{F}_r$, so gilt*

$$\mathrm{ggT}(f, h^{(r-1)/2} - (f')^{(r-1)/2}) = \prod_{\substack{i=1,\\ \gamma_i \in S}}^{t} f_i.$$

Beweis: 1. Das Polynom h ist genau dann in $\mathcal{N}(f,r)$, wenn es der Gleichung $f^r \cdot H^{(r-1)}\left(\frac{h}{f}\right) = h^r$ entspricht und $\mathrm{Grad}(h) < n$ ist. Für jedes solche h ist

$$\begin{aligned}(ch)^r \equiv c^r f^r \cdot H^{(r-1)}\left(\frac{h}{f}\right) \mod f &\equiv c^r h \cdot (f')^{r-1} \mod f = ch \cdot (cf')^{r-1} \\ \equiv ch \cdot 1^{r-1} \mod f &\equiv ch \mod f,\end{aligned}$$

also ist $ch \in \mathcal{B}(f,r)$. Die Multiplikation mit c modulo f ist invertierbar, die Umkehrabbildung ist die Multiplikation mit f'. Also ist ψ ein Isomorphismus über $\mathbb{F}_r$, der von $\mathcal{N}(f,r)$ nach $\mathcal{B}(f,r)$ verläuft.

2. Da für $1 \leq i,j \leq t$ mit $i \neq j$ gilt $\left(\frac{f}{f_i} \cdot f_i'\right) \cdot \left(\frac{f}{f_j} \cdot f_j'\right) \equiv 0 \mod f$, so gilt für jede Zahl $k \in \mathbb{N} \setminus \{0\}$

$$h^k \equiv \sum_{i=1}^{t} \gamma_i^k \left(\frac{f \cdot f_i'}{f_i}\right)^k \mod f,$$

denn es gibt für $h \in \mathcal{N}(f,r)$ Elemente $\gamma_i \in \mathbb{F}_r$ mit

$$\frac{h}{f} = \sum_{i=1}^{t} \gamma_i \frac{f_i'}{f_i} = \sum_{i=1}^{t} \gamma_i \frac{f \cdot f_i'}{f \cdot f_i},$$

also ist dann $h = \sum_{i=1}^{t} \gamma_i \frac{f_i'}{f_i} f$. Darum ist

$$h^{(r-1)/2} - (f')^{(r-1)/2} \equiv \sum_{i=1}^{t} \underbrace{\left(\gamma_i^{(r-1)/2} - 1\right)}_{0 \text{ für } \gamma_i \in S} \left(\frac{f \cdot f_i'}{f_i}\right)^{(r-1)/2} \mod f,$$

also ist $h^{(r-1)/2} - (f')^{(r-1)/2}$ teilbar durch f_i genau dann, wenn $\gamma_i \in S$ ist.

□

Die Räume $\mathcal{N}(f,r)$ und $\mathcal{B}(f,r)$ sind also als $\mathbb{F}_r$–Vektorräume äquivalent, $\mathcal{B}(f,r)$ ist zusätzlich eine Algebra. Das Polynom c ist als Nebenprodukt bei der Überprüfung auf Quadratfreiheit von f mit dem Euklidischen Algorithmus leicht zu erhalten.

Diese Tatsache möchte ich noch einmal besonders hervorheben: Es gibt zwischen den beiden Räumen $\mathcal{B}(f,r)$ und $\mathcal{N}(f,q)$ einen Isomorphismus ψ, der noch dazu leicht durch die Ableitung f' von f zu bestimmen ist. Das versetzt uns in die Lage, die jeweiligen Vorteile der beiden Methoden von Berlekamp und Niederreiter zu kombinieren: Im Fall gerader Charakteristik $p = 2$ bestimmen wir eine Basis des Raumes $\mathcal{N}(f,r)$ und wandeln sie mit ψ in eine Basis der Berlekamp–Unteralgebra $\mathcal{B}(f,r)$ um. Ist die Charakteristik ungerade, so bestimmen wir die Basis von $\mathcal{B}(f,r)$ direkt mit dem Algorithmus von Berlekamp.

Bemerkung 6.27
Alle anderen zur Berlekamp–Unteralgebra $\mathcal{B}(f,r)$ äquivalenten Räume haben ebenfalls faktorisierende Eigenschaften. Es stellt sich also die Frage, welche weiteren äquivalenten Unterräume man bestimmen kann, ohne tatsächlich die Faktorisierung von f zu kennen, und ob sich diese Unterräume effizienter als die beiden hier beschriebenen bestimmen lassen.

Nach Satz 6.26 drängt sich der Verdacht auf, daß Verbesserungen des Algorithmus von Niederreiter zumindest dann möglich sind, wenn in der analogen Situation eine Basis der Berlekamp–Unteralgebra konkret bestimmt werden muß, also bei den Verfahren von Camion, Moenck und Zassenhaus.

Übertragung des Algorithmus von Camion

Punkt 2. des Satzes 6.26 besagt, daß man, um zum Beispiel die Weiterentwicklung aus Abschnitt 5.1 auf Seite 45 des Algorithmus von Berlekamp nach Camion auf Elemente des $\mathcal{N}(f, r)$ anwenden zu können, den Isomorphismus ψ nicht benutzen muß. Ist die Charakteristik p des Körpers $\mathbb{F}_r$ ungerade, so wählt man ein zufälliges Element h des Lösungsraums der Differentialgleichung (10), bildet $\mathrm{ggT}(f, h^{(r-1)/2} - f'^{(r-1)/2})$ und hofft, daß dieser Faktor von f nicht trivial ist.

Ist die Charakteristik gerade, so läßt sich wieder die Abbildung in die multiplikative Untergruppe der Ordnung drei anwenden, wobei es natürlich vorkommen kann, daß man den Körper $\mathbb{F}_r$ erst erweitern muß.

Das typische Verfahren mit der Spur funktioniert im allgemeinen nicht, da sich die Algebra–Struktur von $\mathcal{B}(f, r)$ nicht auf $\mathcal{N}(f, r)$ überträgt. Einzig in dem Spezialfall $q = r = 2$, in dem die Spur gerade die Identität ist und also keine multiplikative Struktur gebraucht wird, ist das Verfahren doch anwendbar und reduziert sich darauf, ein zufälliges Element h des Lösungsraums zu wählen und mit $\mathrm{ggT}(f, h - f')$ auf einen nicht trivialen Faktor von f zu hoffen. Die dadurch erreichte Faktorisierung ist dieselbe, die mit $\mathrm{ggT}(f, h)$ erreicht wird, das Verfahrem ist genau das von Niederreiter.

Beispiel 6.4
Betrachte

$$f(x) = x^{12} + 4x^{11} + 2x^{10} + 2x^9 + 6x^6 + 3x^4 + 3x^3 + 4x^2 + 5 \in \mathbb{F}_7[x].$$

Dann ist aus Beispiel 6.2 auf Seite 89 bekannt, daß die Menge $\{h_1, h_2, h_3, h_4\}$ *mit*

$$\begin{aligned}
h_1(x) &= 4 + 6x + 3x^2 + x^3 + 3x^4 + 3x^5 + 3x^7 + x^8,\\
h_2(x) &= 2 + x + 4x^2 + x^3 + 3x^4 + x^5 + 5x^6 + 2x^7 + x^9,\\
h_3(x) &= 5 + 4x + x^2 + 3x^3 + 4x^5 + 6x^6 + 6x^7 + x^{10} \text{ und}\\
h_4(x) &= 5 + x + 4x^2 + 2x^3 + x^4 + 2x^5 + 4x^7 + x^{11}
\end{aligned}$$

eine Basis des Lösungsraums der Differentialgleichung $H^{(6)}\left(\frac{h}{f}\right) = \frac{h^7}{f^7}$ *ist und daß*

$$f'(x) = 5x^{11} + 2x^{10} + 6x^9 + 4x^8 + x^5 + 5x^3 + 2x^2 + x$$

gilt. Ich wähle die Linearkombination $h(x) = 6h_2(x) + 2h_3(x) + 5h_4(x)$ *und berechne*

$$h(x)^3 - f'(x) \equiv 4x^{11} + 4x^{10} + 2x^8 + 5x^7 + 3x^6 + 6x^4 + 3x^3 + 4x + 6 \mod f(x).$$

Ich erhalte mit

$$\begin{aligned}
b_1(x) = \mathrm{ggT}(f(x), h(x)^3 - f'(x)) &= x^6 + 6x^5 + 5x^4 + 3x^3 + 3x^2 + 6x + 3,\\
b_2(x) = f(x)/b_1(x) &= x^6 + 5x^5 + 2x^4 + 4x^3 + 4x^2 + 6x + 4
\end{aligned}$$

eine nicht triviale partielle Faktorisierung von f*, rechne also weiter mit* $f(x) = b_1(x) \cdot b_2(x)$*.*
Ich wähle als nächstes die Linearkombination $h(x) = 3h_1(x) + 2h_2(x) + 5h_3(x) + 6h_4(x)$*, berechne*

$$h(x)^3 - f'(x) \equiv 2x^{11} + 2x^{10} + x^9 + x^8 + 5x^7 + 3x^6 + x^5 + 5x^4 + 2x^2 + 4x + 3 \mod f(x)$$

und erhalte mit

$$\begin{aligned}
b_3(x) = \mathrm{ggT}(b_1(x), h(x)^3 - f'(x)) &= x^3 + 6x^2 + 5x + 6,\\
b_1(x)/b_3(x) &= x^3 + 4,\\
b_4(x) = \mathrm{ggT}(b_2(x), h(x)^3 - f'(x)) &= x^3 + 6x^2 + 4x + 1 \text{ und}\\
b_2(x)/b_4(x) &= x^3 + 6x^2 + 4x + 4
\end{aligned}$$

alle vier Faktoren von f*. Insgesamt ist also*

$$f(x) = (x^3 + 6x^2 + 5x + 6)(x^3 + 4)(x^3 + 6x^2 + 4x + 1)(x^3 + 6x^2 + 4x + 4).$$

Übertragung des Algorithmus von Moenck

Die Abwandlung des Algorithmus nach Moenck sucht hier für Polynome $h(x) \in \mathcal{N}(f, r)$ die Menge C der $\gamma \in \mathbb{F}_r$, für welche die Bestimmung des $\operatorname{ggT}(f(x), h(x) - \gamma f'(x))$ tatsächlich einen nicht trivialen Faktor ergibt. Die γ werden bestimmt, indem die Nullstellen der Resultanten $R(f(x), h(x) - y \cdot f'(x))$ bestimmt werden.

Beispiel 6.5
Betrachte

$$f(x) = x^{12} + 4x^{11} + 2x^{10} + 2x^9 + 6x^6 + 3x^4 + 3x^3 + 4x^2 + 5 \in \mathbb{F}_7[x].$$

Dann ist aus Beispiel 6.2 auf Seite 89 bekannt, daß die Polynome

$$\begin{aligned} h_1(x) &= 4 + 6x + 3x^2 + x^3 + 3x^4 + 3x^5 + 3x^7 + x^8, \\ h_2(x) &= 5 + x + 4x^2 + 2x^3 + x^4 + 2x^5 + 4x^7 + x^{11}, \\ h_3(x) &= 2 + x + 4x^2 + x^3 + 3x^4 + x^5 + 5x^6 + 2x^7 + x^9 \text{ und} \\ h_4(x) &= 5 + 4x + x^2 + 3x^3 + 4x^5 + 6x^6 + 6x^7 + x^{10} \end{aligned}$$

eine Basis des Lösungsraums der Differentialgleichung $H^{(6)}\left(\frac{h}{f}\right) = \frac{h^7}{f^7}$ *sind und daß*

$$f'(x) = 5x^{11} + 2x^{10} + 6x^9 + 4x^8 + x^5 + 5x^3 + 2x^2 + x$$

gilt. Nun ist die Resultante

$$R(f(x), h_1(x) - y \cdot f'(x)) = 2y^{12} + 2y^{10} + 3y^8 + 5y^6 = 2y^6(y-4)^3(y-3)^3,$$

darum ist die Menge C *hier gegeben als* $C = \{0, 4, 3\}$.

- *Es ist*

$$\begin{aligned} b_1(x) = \operatorname{ggT}(f(x), h(x)) &= x^6 + 6x^5 + 5x^4 + 3x^3 + 3x^2 + 6x + 3, \\ b_2(x) = f(x)/b_1(x) &= x^6 + 5x^5 + 2x^4 + 4x^3 + 4x^2 + 6x + 4, \end{aligned}$$

  rechne weiter mit $f(x) = b_1(x) \cdot b_2(x)$.

- *Es ist*

$$\begin{aligned} b_3(x) = \operatorname{ggT}(b_2(x), h(x) - 4f'(x)) &= x^3 + 6x^2 + 4x + 1, \\ b_4(x) = b_2(x)/b_3(x) &= x^3 + 6x^2 + 4x + 4, \end{aligned}$$

  rechne weiter mit $f(x) = b_1(x) \cdot b_3(x) \cdot b_4(x)$.

Es muß noch ein weiterer Faktor von b_1 *gefunden werden, bilde darum die Resultante*

$$\begin{aligned} R(f(x), h_2(x) - y \cdot f'(x)) &= 2y^{12} + 5y^{11} + y^{10} + 5y^8 + 5y^7 + y^5 + 5y^4 + 2y^3 \\ &= 2y^3(y-6)^3(y-2)^3(y-4)^3 \end{aligned}$$

und erhalte die Menge $C = \{0, 6, 2, 4\}$.

- *Es ist*

$$\begin{aligned} b_5(x) = \operatorname{ggT}(b_1(x), h(x)) &= x^3 + 6x^2 + 5x + 6, \\ b_1(x)/b_5(x) &= x^3 + 4. \end{aligned}$$

Insgesamt ist somit $f(x) = (x^3 + 6x^2 + 5x + 6)(x^3 + 4)(x^3 + 6x^2 + 4x + 1)(x^3 + 6x^2 + 4x + 4)$.

Übertragung des Algorithmus von Zassenhaus

Das Verfahren von Zassenhaus versucht ebenfalls, die Menge C zu bestimmen, aber auf andere Art und Weise: Es müßte das Polynom $G(\frac{h}{f}) = \prod_{\gamma\in C}(\frac{h}{f} - \gamma f')$ in einer Variablen y erzeugt werden, das dann mit Hilfe eines Verfahrens zum Finden der Nullstellen aus Abschnitt 3 auf Seite 15 die Elemente γ bestimmt. Leider ist G hier ein Polynom in den zwei Variablen x und y. Um dem abzuhelfen, könnte man den Ausdruck $\prod_{\gamma\in C}(\frac{h}{f\cdot f'} - \gamma) = \prod_{\gamma\in C}(\frac{\psi(h)}{f} - \gamma)$ betrachten und die Theorie wäre wieder anwendbar. Dies bedeutet aber gerade, die Elemente des Lösungsraums der Differentialgleichung mit ψ in Elemente der Berlekamp–Unteralgebra umzuwandeln. Also gibt es keine Verbesserung nach Zassenhaus direkt für das Verfahren von Niederreiter.

Andere Anknüpfungspunkte der Verfahren von Berlekamp und Niederreiter

Das Verfahren von Cantor und Zassenhaus aus Abschnitt 5.3 auf Seite 58 und das von Shoup aus Abschnitt 5.6 auf Seite 70 nutzen spezielle Eigenschaften der Komponentendarstellung bezüglich der Berlekamp–Unteralgebra aus, die bei der Multiplikation mit f' nicht erhalten bleiben. Darum gibt es keine Verbesserungen des Algorithmus von Niederreiter, die sich an diese Verfahren anlehnen.

Interessante Eigenschaften, die sich aus der Verbindung des Algorithmus von Berlekamp mit dem von Niederreiter ergeben, werden in dem Papier [27] von T. C. Y. Lee und S. A. Vanstone besprochen, beispielsweise, wann sich die Berlekamp–Unteralgebra und der Lösungsraum der Gleichung (10) nicht trivial überschneiden: Genau dann, wenn $\mathrm{ggT}(f(x), f'(x)+1) \neq 1$ ist. Leider ergibt sich aus ihren Untersuchungen keine die Laufzeit eines Faktorisierungs–Algorithmus verbessernde Erkenntnis.

7 Zusammenfassung

Wir haben nun viele verschieden Faktorisierungsmethoden kennengelernt. Es stellt sich die Frage nach einem Vergleich, zum Beispiel über die Komplexitäten bezüglich der Laufzeit:

Verfahren	$\mathcal{O}$
McEliece	$n^2\left(d^2\log(q)+r\right)$
Berlekamp	n^3+tn^2q
Niederreiter	n^3+tn^2q
Camion	$n^3+n^2t\log(t)\log(r)+nt^2\log(t)$
Weiterentwicklung von Camion	$n^3+n^2t\log(t)\log(r)$
Cantor und Zassenhaus	$n^3\log(t)\log(q)$
Moenck	n^n
Zassenhaus	$n^3+n^2\left(t^3+t\log(n)\log(q)\right)$
Shoup	dn^2q

Wächst n, d jedoch nicht, so bieten sich also die Algorithmen von McEliece und Shoup zur Faktorisierung an, wächst mit n auch d, so ist das Verfahren von Shoup besser, da seine Komplexität von d und nicht von d^2 abhängt. Wächst zusätzlich q exponentiell, so ist das Verfahren von McEliece das bessere, da die Komplexität in q nur logarithmisch wächst und man $r = p$ wählen kann.

Welche Laufzeit haben die Algorithmen aber, wenn man nicht an die Grenzen geht? Ich mache einen Vergleich der Laufzeiten, dabei wähle ich die Testpolynome f in Anlehnung an den Artikel [12] von P. Fleischmann und P. Roelse in den Körpern $\mathbb{F}_2, \mathbb{F}_{97}$ und $\mathbb{F}_{19913}$. Die Laufzeiten habe ich mit meinen MapleV4.00c–Programmen auf meinem Pentium mit 200 MHz gemessen, die Programme mit den Tests befinden sich auf der beigelegten CD.

In $\mathbb{F}_2$ sei $\mathrm{Grad}(f) = n$ für $n = 250, 500, 750$ und 1000: Ich lasse die zufälligen Polynome f von Maple so erzeugen, daß sie in gleichgradige Faktoren vom Grad $d = 50$ zerfallen:

Verfahren	$n = 250$	$n = 500$	$n = 750$	$n = 1000$
McEliece	12,099 s	32,491 s	90,735 s	141,554 s
Berlekamp	351,894 s	5883,702 s	> 8 h	
Niederreiter	858,283 s			
Cantor und Zassenhaus	1,470 s	8,124 s	22,358 s	55,785 s
Shoup	458,113 s	1806,808 s	3896,360 s	7757,098 s

An dieser Stelle ist überraschend, daß das Verfahren von Niederreiter eine längere Laufzeit als das Verfahren von Berlekamp hat. Besondere Beachtung ist den Laufzeiten des Verfahres von Cantor und Zassenhaus zu schenken. Die Zeiten sind so kurz, da $d = 50$ gerade ist und ich darum mit der Abbildung in die multiplikative Untergruppe der Ordnung drei arbeite. Ist d ungerade, so arbeite ich mit der Spur und die Laufzeiten bewegen sich in der Größenordnung der Laufzeiten des Verfahrens von McEliece.

Die Verfahren von Camion, Moenck und Zassenhaus haben bei $q = 2$ eine längere Laufzeit als das Verfahren von Berlekamp, da q so klein ist.

In $\mathbb{F}_{97}$ sei $\mathrm{Grad}(f) = n$ mit $n = 25, 50, 75$ und 100, weiterhin sei $d = 5$:

Verfahren	$n = 25$	$n = 50$	$n = 75$	$n = 100$
McEliece	0,325 s	0,918 s	2,064 s	3,478 s
Berlekamp	1,796 s	12,221 s	44,999 s	94,783 s
Niederreiter	3,252 s	24,757 s	87,091 s	226,827 s
Camion	2,574 s	19,597 s	69,316 s	159,081 s
Weiterentwicklung von Camion	2,101 s	13,663 s	62,673 s	115,895 s
Cantor und Zassenhaus	0,083 s	0,375 s	1,776 s	2,632 s
Moenck	2,197 s	16,672 s	86,087 s	217,151 s
Zassenhaus	2,346 s	14,697 s	63,056 s	147,908 s
Shoup	0,601 s	1,597 s	3,942 s	5,750 s

In $\mathbb{F}_{19913}$ sei ebenfalls $\mathrm{Grad}(f) = n$ mit $n = 25, 50, 75$ und 100 und $d = 5$:

Verfahren	$n = 25$	$n = 50$	$n = 75$	$n = 100$
McEliece	35,146 s	68,052 s	122,933 s	195,531 s
Berlekamp	46,852 s	163,508 s	273,848 s	212,024 s
Niederreiter	73,233 s	223,139 s	859,075 s	1088,301 s
Camion	2,359 s	18,639 s	73,104 s	173,865 s
Weiterentwicklung von Camion	2,259 s	19,352 s	50,407 s	138,359 s
Cantor und Zassenhaus	0,402 s	2,261 s	4,847 s	9,529 s
Moenck	2,495 s	21,542 s	68,062 s	156,406 s
Zassenhaus	2,450 s	15,760 s	55,239 s	131,083 s
Shoup	2,063 s	9,654 s	22,850 s	40,080 s

Gewinner in allen Tests ist somit das Verfahren von Cantor und Zassenhaus.

Literatur

[1] B. BEAUZAMY, V. TREVISAN UND P. S. WANG, *Polynomial Factorization: Sharp Bounds, Efficient Algorithms*, Journal of Symbolic Computation, 15 (1993), S. 393–413.

[2] E. R. BERLEKAMP, *Factoring Polynomials over Finite Fields*, Bell System Technical Journal, 46 (1967), S. 1853–1859.

[3] ——, *Factoring Polynomials over Large Finite Fields*, Mathematics of Computation, 24 (1970), S. 713–735.

[4] E. R. BERLEKAMP, H. RUMSEY UND G. SOLOMON, *On the Solution of Algebraic Equations over Finite Fields*, Information and Control, 10 (1967), S. 553–564.

[5] S. BOSCH, *Algebra*, Springer, 1993.

[6] P. CAMION, *A Deterministic Algorithm for Factorizing Polynomials of F_q[X]*, Annals of Discrete Mathematics, 17 (1983), S. 149–157.

[7] ——, *Improving an Algorithm for Factoring Poynomials over a Finite Field and Constructing Large Irreducible Polynomials*, IEEE Transactions on Information Theory, 29 (1983), S. 378–385.

[8] D. G. CANTOR UND H. ZASSENHAUS, *A New Algorithm for Factoring Polynomials over Finite Fields*, Mathematics of Computation, 36 (1981), S. 587–592.

[9] B. CHOR UND R. L. RIVEST, *A Knapsack–Type Public Key Cryptosystem Based on Arithmetic in Finite Fields*, IEEE Transactions on Information Theory, 34 (1988), S. 901–909.

[10] H. COHEN, *A Course in Computational Algebraic Number Theory*, Springer, 1993.

[11] P. FLEISCHMANN, *Connections between the Algorithms of Berlekamp and Niederreiter for Factoring Polynomials over F_q*, Linear Algebra and it's Applications, 192 (1993), S. 101–108.

[12] P. FLEISCHMANN UND P. ROELSE, *Comparative Implementations of Berlekamp's and Niederreiter's Polynomial Factorization Algorithms*, Mathematical Society Lecture Note Series, 233 (1996), S. 73–83.

[13] S. GAO UND J. VON ZUR GATHEN, *Berlekamp's and Niederreiter's Polynomial Factorization Algorithms*, Contemporary Mathematics, 168 (1994), S. 101–116.

[14] K. O. GEDDES, S. R. CZAPOR UND G. LABAHN, *Algorithms for Computer Algebra*, Kluwer Academic Publishers, 1995.

[15] P. GIANNI UND B. M. TRAGER, *Square–Free Algorithms in Positive Characteristic*, Applicable Algebra in Engeneering, Communication and Computing, 7 (1996), S. 1–14.

[16] R. GÖTTFERT, *An Acceleration of the Niederreiter Factorization Algorithm in Characteristic 2*, Mathematics of Computation, 62 (1994), S. 831–839.

[17] D. Y. GRIGOR'EV, *Factorization of Polynomials over a Finite Field and the Solution of Systems of Algebraic Equations*, Journal of Soviet Mathematics, 34 (1986), S. 1762–1803.

[18] H. GUNJI UND D. ARNON, *On Polynomial Factorization over Finite Fields*, Mathematics of Computation, 36 (1981), S. 281–287.

[19] H. HASSE, *Theorie der höheren Differentiale in einem algebraischen Funktionenkörper mit vollkommenem Konstantenkörper*, Journal für reine und angewandte Mathematik, 175 (1936), S. 50–54.

[20] W. KLINGENBERG, *Lineare Algebra und Geometrie*, Springer, 1992.

[21] D. E. KNUTH, *The Art of Computer Programming*, Band 2, Addison–Wesley, 1981.

[22] N. KOBLITZ, *Algebraic Aspects of Cryptography*, Springer, 1998.

[23] S. LANDAU, *Factoring Polynomials over Algebraic Number Fields*, SIAM Journal on Computing, 14 (1985), S. 184–195.

[24] S. LANG, *Algebra*, Addison–Wesley, 1971.

[25] E. LANGE, *Substitutions for Linear Shift Register Sequences and the Factorization Algorithms of Berlekamp and Niederreiter*, Linear Algebra and it's Applications, 249 (1995), S. 217–228.

[26] D. LAZARD, *On Polynomial Factorization*, Lecture Notes in Computer Science: EUROCAM 82, 144 (1982), S. 126–134.

[27] T. C. Y. LEE UND S. A. VANSTONE, *Subspaces and Polynomial Factorizations over Finite Fields*, Applicable Algebra in Engeneering, Communication and Computing, 6 (1995), S. 147–157.

[28] A. K. LENSTRA, *Factorization of Polynomials*, Computational Methods in Number Theory, 1 (1982), S. 169–198.

[29] A. K. LENSTRA, H. W. LENSTRA UND L. LOVASZ, *Factoring Polynomials with Rational Coefficients*, Mathematische Annalen, 261 (1982), S. 515–534.

[30] H. W. LENSTRA, *On the Chor–Rivest Knapsack Cryptosystem*, Journal of Cryptology, 3 (1991), S. 149–155.

[31] R. LIDL UND H. NIEDERREITER, *Introduction to Finite Fields and their Applications*, Cambridge University Press, 1994.

[32] R. J. MCELIECE, *Factorization of Polynomials over Finite Fields*, Mathematics of Computation, 23 (1969), S. 861–867.

[33] ——, *Finite Fields for Computer Scientists and Engineers*, Kluwer Academic Publishers, 1969.

[34] R. J. MCINTOSH, *A Generalization of a Congruential Property of Lucas*, The American Mathematical Monthly, 99 (1992), S. 231–238.

[35] A. F. MENEZES, Editor, *Applications of Finite Fields*, Kluwer Academic Press, 1993.

[36] A. F. MENEZES, P. VAN OORSCHOT UND S. VANSTONE, *Some Computational Aspects of Root Finding in $GF(q^m)$*, Lecture Notes in Computer Science: Symbolic and Algebraic Computation, 358 (1989), S. 259–270.

[37] R. T. MOENCK, *On the Efficiency of Algorithms for Polynomial Factoring*, Mathematics of Computation, 31 (1977), S. 235–250.

[38] H. NIEDERREITER, *A New Efficient Factorization Algorithm for Polynomials over Small Finite Fields*, Applicable Algebra in Engeneering, Communication and Computing, 4 (1992), S. 81–87.

[39] ——, *Factorization of Polynomials and some Linear Algebra Problems over Finite Fields*, Linear Algebra and its Applications, 192 (1993), S. 301–328.

[40] ——, *Factorization of Polynomials over Finite Fields and Characteristic Sequences*, Journal of Symbolic Computation, 16 (1993), S. 401–412.

[41] ——, *Factoring Polynomials over Finite Fields Using Differential Equations and Normal Bases*, Mathematics of Computation, 62 (1994), S. 819–830.

[42] ——, *New Deterministic Factorization Algorithms for Polynomials over Finite Fields*, Contemporary Mathematics, 168 (1994), S. 251–268.

[43] H. NIEDERREITER UND R. GÖTTFERT, *On a New Factorization Algorithm for Polynomials over Finite Fields*, Mathematics of Computation, 64 (1995), S. 347–353.

[44] S. C. POHLIG UND M. E. HELLMANN, *An Improved Algorithm for Computing Logarithms in $GF(p)$ and Its Cryptographic Significance*, IEEE Transactions on Information Theory, 24 (1978), S. 106–111.

[45] M. O. RABIN, *Probabilistic Algorithms in Finite Fields*, SIAM Journal on Computing, 9 (1980), S. 273–280.

[46] R. L. RIVEST, A. SHAMIR UND L. ADLEMAN, *A Method for Obtaining Digital Signatures and Public Key Cryptosystems*, Communications ACM, 21 (1978), S. 120–126.

[47] L. RONYAI, *Factoring Polynomials over Finite Fields*, Journal of Algorithms, 9 (1988), S. 391–400.

[48] L. RONYAI UND A. SZANTO, *Prime–Field–Complete Functions and Factoring Polynomials over Finite Fields*, Computers and Artificial Intelligence, 15 (1996), S. 571–577.

[49] G. SCHEJA UND U. STORCH, *Lehrbuch der Algebra*, Band 2, Teubner, 1988.

[50] B. SCHNEIER, *Angewandte Kryptographie*, Addison–Wesley, 1997.

[51] V. SHOUP, *Factoring Polynomials Using Fewer Random Bits*, Journal of Symbolic Computation, 9 (1990), S. 229–239.

[52] ——, *On the Deterministic Complexity of Factoring Polynomials over Finite Fields*, Information Processing Letters, 33 (1990), S. 261–267.

[53] ——, *A New Polynomial Factorization Algorithm and its Implementation*, Journal of Symbolic Computation, 20 (1995), S. 363–397.

[54] I. E. SHPARLINSKI, *On Bivariate Polynomial Factorization over Finite Fields*, Mathematics of Computation, 60 (1993), S. 787–791.

[55] O. TEICHMÜLLER, *Differentialrechnung bei Charakteristik p*, Journal für reine und angewandte Mathematik, 175 (1936), S. 89–99.

[56] P. VAN OORSCHOT UND S. VANSTONE, *A Geometric Approach to Root Finding in $GF(q^m)$*, IEEE Transactions on Information Theory, 35 (1989), S. 444–453.

[57] S. VAUDENAY, *Cryptoanalysis of the Chor–Rivest Cryptosystem*, Advances in Crptology, Crypto 98, Lecture Notes in Computer Science, 1462 (1998), S. 243–256.

[58] J. VON ZUR GATHEN, *Irreducibility of Multivariate Polynomials*, Journal of Computer and System Sciences, 31 (1985), S. 225–264.

[59] J. VON ZUR GATHEN UND E. KALTOFEN, *Factorization of Multivariate Polynomials over Finite Fields*, Mathematics of Computation, 45 (1985), S. 251–261.

[60] J. VON ZUR GATHEN UND V. SHOUP, *Computing Frobenius Maps and Factoring Polynomials*, Computational Complexity, 2 (1992), S. 187–224.

[61] D. WAN, *Factoring Multivariate Polynomials over Large Finite Fields*, Mathematics of Computation, 54 (1990), S. 755–770.

[62] M. WILLETT, *Factoring Polynomials over a Finite Field*, SIAM Journal of Applicable Mathematics, 35 (1978), S. 333–337.

[63] ——, *Arithmetic in a Finite Field*, Mathematics of Computation, 35 (1980), S. 1353–1359.

[64] H. ZASSENHAUS, *On Hensel Factorization*, Journal of Number Theory, 1 (1969), S. 291–311.

[65] ——, *A Remark on the Hensel Factorization Method*, Mathematics of Computation, 32 (1978), S. 287–292.