Seminar 6: Polynomarithmetik: Faktorisierung

Ausarbeitung

13.12.2007

Autor: Stefan Hasenbanck

Version: 1.0

1 Inhaltsverzeichnis

2 Einleitung

Diese Seminararbeit beschäftigt sich mit der Faktorzerlegung und der Quadratfreien Faktorisierung von Polynomen.

Diese Techniken können in Computeralgebrasystemen verwendet werden, um Brüche in den Polynomen vorkommen zu kürzen (siehe Kapitel: „Rationale Funktionen“). Zudem dienen sie als Grundlage für bessere Algorithmen und werden zum Teil in anderen Gebieten wie der Integration verwendet.

3 Der erweiterte euklidische Algorithmus

3.1 Ziel dieses Abschnitts

Man benötigt einen Algorithmus zum Bestimmen des größten gemeinsamen Teilers (hier abgekürzt mit ggT) zweier Polynome für die Quadratfreie Faktorisierung (siehe zugehöriges Kapitel).

Zudem reicht uns der ggT zweier Polynome, um einen Bruch der Polynome enthält zu kürzen (siehe Kapitel: „Rationale Funktionen“).

Der ggT zweier Polynome ist definiert als das Produkt aller gemeinsamen Nullstellen.

3.2 Finden des ggT mit dem euklidischen Algorithmus

Um den ggT zweier beliebiger Polynome zu erhalten, kann man auf den schon aus der Vorlesung GTI bekannten euklidischen Algorithmus zurückgreifen.

Euklidischer Algorithmus zum Finden des ggT zweier Polynome in Rekursiver Darstellung (in Java ähnlicher Syntax):

```
ggT(Polynom a, Polynom b)
{
   /* Sonderfälle */
   if (a == 0) and (b == 0) return UNENDLICH;
   if (b==0) return a;
   /* Standardablauf */
   if (a mod b = 0) return b;
   /*else*/
   return ggT(b, a mod b);
}
```

Da die Polynomdivision mit Rest aus Analysis bekannt ist, wird hier darauf nicht näher eingegangen.

> Für die Polynomdivision mit Rest empfehle ich:
> „http://www.arndt-bruenner.de/mathe/scripts/polynomdivision.htm“ ,
> da sie alle Zwischenergebnissen darstellt.

Es folgt ein Beispiel bei dem wir feststellen:

- dass der ggT den wir durch den euklidischen Algorithmus erhalten bis auf einen konstanten Vorfaktor eindeutig ist
- das bei dem euklidischen Algorithmus die Zahlen in den Koeffizienten stark anwachsen, was als Expression Swell bezeichnet wird, hierauf wird später im Kapitel nochmals eingegangen.

3.3 Beispiel für die Anwendung des euklidischen Algorithmus

$(x-2) * (x-6)^2 * 4 = 4x^3-56x^2+240x-288$

$(x-2) * (x+3)*(x-7)^2 = x^4-13x^3+29x^2+133x-294$

ggT($4x^3-56x^2+240x-288$, $x^4-13x^3+29x^2+133x-294$) ->

$(4x^3-56x^2+240x-288) / (x^4-13x^3+29x^2+133x-294) = 0$ Rest $(4x^3-56x^2+240x-288)$

$(x^4-13x^3+29x^2+133x-294) / (4x^3-56x^2+240x-288) = ¼ x + ¼$ Rest $-17x^2 + 145x - 222$

$$(4x^3-56x^2+240x-288) / (-17x^2 + 145x - 222) = -\frac{4}{17}x + \frac{372}{289} \text{ Rest } \frac{324}{289}x - \frac{648}{289}$$

$$(-17x^2 + 145x - 222) / (\frac{324}{289}x - \frac{648}{289}) = -\frac{4913}{324}x + \frac{10693}{108} \text{ Rest } 0$$

$$\text{ggT}(4x^3-56x^2+240x-288, x^4-13x^3+29x^2+133x-294) = \frac{324}{289}x - \frac{648}{289} =$$

$$\text{ggT}((x-2) * (x-6)^2 * 4 , (x-2) * (x+3)*(x-7)^2) = \frac{324}{289} *(x-2)$$

Wie man sofort sieht, bekommen wir nach einer Normierung das Ergebnis, welches wir erwartet haben.

> Unter normieren versteht man, das Ergebnis mit einer Konstanten zu multiplizieren, so dass der Koeffizient vor der Variablen mit dem höchsten Exponenten zu 1 wird.

Dabei wird immer normiert, dadurch ist die Aussage von

ggT(4*(x-1) , 4*(x-2)) = 1

lediglich, dass 4x-4 und 4x-8 keine gemeinsame Nullstelle haben, bzw. die Polynome keinen gemeinsamen nicht trivialen Teiler haben – unter trivialen Teilern versteht man an dieser Stelle konstante Teiler.

3.4 Der erweiterte euklidische Algorithmus

Der erweiterte euklidische Algorithmus bestimmt ein a und ein b, so dass gilt:

$$a * x + b * y = ggt(x, y)$$

Im einfachsten Fall erhält man die Bézoutkoeffizienten a und b, indem man eine sukzessive Rückwärtsauflösung der Schritte vom euklidischen Algorithmus macht.

Man kann auch wieder gleich bei der Berechnung die Bézoutkoeffizienten bestimmen, in der Weise wie man es von den ganzen Zahlen gewöhnt ist.

3.5 Beispiel für den erweiterten euklidischen Algorithmus:

Wir haben als Schritte im euklidischen Algorithmus durchgeführt:

$(x^4-13x^3+29x^2+133x-294)$ / $(4x^3-56x^2+240x-288)$ = ¼ x + ¼ Rest $-17x^2 + 145x - 222$

$(4x^3-56x^2+240x-288)$ / $(-17x^2 + 145x - 222) = -\frac{4}{17}x + \frac{372}{289}$ Rest $\frac{324}{289}x - \frac{648}{289}$

$$\frac{289}{324} * \left(\frac{324}{289}x - \frac{648}{289}\right) = (x-2)$$

Daher erhalten wir:

(x-2) = 289/324 * (324/289x-648/289)

($(4x^3-56x^2+240x-288)$ - (($-17x^2 + 145x - 222$) * (-4/17x + 372/289)))

($(x^4-13x^3+29x^2+133x-294)$ - ($(4x^3-56x^2+240x-288)$ *(¼ x + ¼)))

(x-2) = 289/324 * ($(4x^3-56x^2+240x-288)$ - (($(x^4-13x^3+29x^2+133x-294)$ - ($(4x^3-56x^2+240x-288)$ *(¼ x + ¼))) * (-4/17x + 372/289)))

Nun fassen wir zusammen:

(x-2) = 289/324 * ($(4x^3-56x^2+240x-288)$ - (($(x^4-13x^3+29x^2+133x-294)$ * (-4/17x + 372/289)) - ($(4x^3-56x^2+240x-288)$ *(-1/17 x^2 + 76/289 x + 93/289))))

(x-2) = 289/324 * (($(4x^3-56x^2+240x-288)$ + ($(4x^3-56x^2+240x-288)$ *(-1/17 x^2 + 76/289 x + 93/289))) - $(x^4-13x^3+29x^2+133x-294)$ * (-4/17x + 372/289))

(x-2) = 289/324 * ((-1/17 x^2 + 76/289 x + 382/289) * $(4x^3-56x^2+240x-288)$ - (-4/17x + 372/289) * $(x^4-13x^3+29x^2+133x-294)$)

(x-2) = (-17/324 x^2 + 19/81 x + 191/162) * $(4x^3-56x^2+240x-288)$ - (-17/81x + 31/27) * $(x^4-13x^3+29x^2+133x-294)$

Die Bézoutkoeffizienten sind somit:

=> a = (-17/324 x^2 + 19/81 x + 191/162) b = (-17/81x + 31/27)

Man sieht es gilt:

$$(x-2) = (-\frac{17}{324}x^2 + \frac{19}{81}x + \frac{191}{162}) * (4x^3 - 56x^2 + 240x - 288) - (-\frac{17}{81}x + \frac{31}{27}) * (x^4 - 13x^3 + 29x^2 + 133x - 294)$$

3.6 Das ermitteln des ggT mit dem euklidischen Algorithmus bei mehreren Polynomen

Das Ermitteln des ggT bei mehreren Polynomen funktioniert genauso wie bei das Ermitteln des ggT bei mehreren ganzen Zahlen.

ggT(Poly1, Poly2, Poly3) = ggT(Poly1, ggT(Poly2,Poly3))

3.7 Das Problem des Expression Swells

Man hat bei der Ermittlung des ggT mit dem euklidischen Algorithmus gesehen, dass die Zahlen in den Koeffizienten stark angewachsen sind.

Unter der Problemklasse des „expression swells“ versteht man alle Probleme, bei denen die Länge der Zahl(en) stark anwächst.

Dieses kann zu Speicherproblemen führen und wie in dem Seminar „Langzahlarithmetik: Addition und Multiplikation“ zu sehen ist, ist die Komplexität der Berechnung auch abhängig von der Länge der Zahl.

Da der ggT normiert wird, handelt es sich hier um ein Teil-Problem, welches man als „intermediate expression swell“ bezeichnet -> sprich die Werte wachsen nur während der Berechnung stark an.

Man kann dieses Problem beim euklidischen Algorithmus ein wenig einschränken.

Dafür macht man vor Beginn des Algorithmus aus jedem Polynom ein primitives Polynom, damit die Koeffizienten möglichst kurz sind.

Zudem macht man, in jeden Schritt, aus dem Rest auch ein primitives Polynom.

Beispiel:

ggT($4x^3-56x^2+240x-288$, $x^4-13x^3+29x^2+133x-294$) ->

($4x^3-56x^2+240x-288$) / ($x^4-13x^3+29x^2+133x-294$) = 0 Rest ($4x^3-56x^2+240x-288$)

($x^4-13x^3+29x^2+133x-294$) / ($4x^3-56x^2+240x-288$) = ¼ x + ¼ Rest $-17x^2 + 145x - 222$

($4x^3-56x^2+240x-288$) / ($-17x^2 + 145x - 222$) = $-\frac{4}{17}x + \frac{372}{289}$ Rest $\frac{324}{289}x - \frac{648}{289}$ =x-2

($-17x^2 + 145x - 222$) / (x-2) =-17x + 111 Rest 0

ggT($4x^3-56x^2+240x-288$, $x^4-13x^3+29x^2+133x-294$) = x-2

Man sieht, dass das Problem geringer wird, jedoch kann man auf diese Weise den intermediate expression swell nicht vollständig verhindern.

3.8 Resümee

Wenn es darum geht, den ggT zweier Polynome zu finden, so ist der euklidische Algorithmus schneller und einfacher zu realisieren als die vollständige Faktorisierung.

Zudem erhalten wir auch wieder die Bézoutkoeffizienten, was uns die vollständige Faktorisierung nicht bieten kann.

4 Eindeutige Faktorzerlegung

4.1 Ziel dieses Abschnitts

In diesem Abschnitt werden zwei Algorithmen vorgestellt, welche eine eindeutige Faktorzerlegung von Polynomen vornehmen, der eine in $\mathbb{Z}_p$ und der andere in $\mathbb{Z}$, wobei ein Polynom in $\mathbb{Z}_p$ nicht die gleiche Zerlegung besitzen muss wie in $\mathbb{Z}$.

Die eindeutige Faktorzerlegung kann verwendet werden, um Nullstellen zu finden oder um Polynome in eine andere Darstellungsform zu bringen um dann effektiver weiterrechnen zu können.

4.2 Allgemeingültige Grundlagen für das zerlegen in Faktoren in $\mathbb{Z}_p$ und $\mathbb{Z}$

4.2.1 Kandidatenpolynome

Im weiteren Verlauf betrachten wir Algorithmen, welche eine Menge an Polynomen testweise von dem zu untersuchenden Polynom dividiert – ist der Rest 0, so ist das testweise dividierte Polynom ein Teiler des zu untersuchenden Polynoms.

Die Menge an möglichen Teilern bezeichnet man als Kandidatenpolynome.

4.2.2 Was ist eine eindeutige Zerlegung

Es gibt eine eindeutige Zerlegung in irreduzible Polynome der Form:

$$a(x) = \prod_{k=1}^{n} p_k(x)$$

wobei für jede weitere Zerlegung in irreduzible Polynome der Form

$$a(x) = \prod_{k=1}^{m} q_k(x)$$

gilt das m = n ist und es eine Zuordnung von p_j zu q_i gibt, die bis auf einen Konstanten Faktor identisch ist.

Zu beachten ist, dass jedes $p_j(x)$ und $q_i(x)$ irreduzibel sein muss.

4.2.3 Grad der Teilpolynome

Der mögliche Grad der Teilpolynome ergibt sich aus der Formel $x^n * x^m = x^{n+m}$.

Dabei sieht man, dass wenn das zu untersuchende Polynom ein Grad von n hat, es maximal ein Teilpolynom geben kann, welches einen Grad größer als $\frac{n}{2}$ hat, da sonst der Grad des zu untersuchenden Polynoms größer sein müsste als n.

Daher muss man nur Teilpolynome bis zu einem Grad von $\frac{n}{2}$ testen und sollte dann das zu untersuchende Polynom nicht 1 sein, so kann man ohne Beschränkung der Allgemeinheit sagen, dass der Rest von dem zu untersuchenden Polynom ein irreduzibler Teiler des zu untersuchenden Polynoms ist.

4.2.4 Umwandeln von $\mathbb{Q}$ [x] Polynomen in $\mathbb{Z}$ [x] Polynome

Ein Polynom ist aus $\mathbb{Q}$ [x], wenn mindestens ein Koeffizient aus $\mathbb{Q}$ ist und die restlichen Koeffizienten aus $\mathbb{Q}$ oder $\mathbb{Z}$ sind.

Eine Umwandlung kann erfolgen, indem man alle Koeffizienten zu Koeffizienten aus $\mathbb{Z}$ umwandelt. Dazu muss man lediglich das Polynom mit dem kgV (kleinsten gemeinsamen Vielfachen) der Nenner aller Koeffizienten des Polynoms multiplizieren.

Beispiel:

$x^4-13/7x^3+29/7x^2+19x-42 \quad * 7 \quad = \quad x^4-13x^3+29x^2+133x-294$

4.3 Eindeutige Faktorzerlegung in $\mathbb{Z}_p$

Es wird ein Polynom in einer Restklasse betrachten, daher beschränkt sich die Anzahl der möglichen Koeffizienten auf eine endliche Menge.

Man kann nun eine Liste aller Kandidatenpolynome erzeugen, indem man alle Polynome mit einem Grad <= $\frac{n}{2}$, wobei der Grad des zu untersuchenden Polynoms n ist, und allen möglichen Koeffizienten notiert.

Diese Liste enthält redundante Informationen, da die Polynome nicht normiert sind, darum würde man als nächstes eine Normierung durchführen und alle doppelten Polynome entfernen.

Die Liste enthält nun noch nicht irreduzible Polynome, jedoch wird nun einfach mit dem kleinsten Polynom das testweise Dividieren begonnen. Da man mit dem kleinsten Polynom beginnt, fallen alle nicht irreduziblen Polynome automatisch weg.

4.3.1 Algorithmus der Eindeutige Faktorzerlegung in $\mathbb{Z}_p$

1. Schritt: Liste der Kandidatenpolynome erstellen.

2. Schritt: Liste durch Normierung verkleinern.

3. Schritt: Alle Kandidatenpolynome testweise von dem zu untersuchenden Polynom dividieren

4.3.2 Beispiel für die Eindeutige Faktorzerlegung in $\mathbb{Z}_p$

Beispiel für x^2+6x+2 in $\mathbb{Z}_7$:

1. Liste ungekürzt:

1, 2, 3, 4, 5, 6, x, 2x, 3x, 4x, 5x, 6x,

x+1, x+2, x+3, x+4, x+5, x+6,

2x+1, 2x+2, 2x+3, 2x+4, 2x+5, 2x+6,

3x+1, 3x+2, 3x+3, 3x+4, 3x+5, 3x+6,

4x+1, 4x+2, 4x+3, 4x+4, 4x+5, 4x+6,

5x+1, 5x+2, 5x+3, 5x+4, 5x+5, 5x+6,

6x+1, 6x+2, 6x+3, 6x+4, 6x+5, 6x+6

2. Liste in Normiert:

1, x,

x+1, x+2, x+3, x+4, x+5, x+6

3. Testen

Ergebnis: $x^2+6x+2 = (x + 3)^2$

4.3.3 Resümee der Eindeutige Faktorzerlegung in $\mathbb{Z}_p$

Leider ist das Verfahren für große p und/oder große n sehr aufwändig und daher in der Praxis nur begrenzt einsetzbar.

Zudem ist das Verfahren nur für Restklassen einsetzbar, allerdings für alle Restklassen und für alle Polynome.

4.4 Eindeutige Faktorzerlegung in $\mathbb{Z}$

Da die Menge der möglichen Koeffizienten in $\mathbb{Z}$ unendlich groß ist, benutzen wir einen anderen Ansatz, welcher auch die Menge der Kandidatenpolynome auf endlich viele beschränkt. Dieser Ansatz wird in dem Algorithmus von Kronecker verwendet.

4.4.1 Grundlagen der Eindeutige Faktorzerlegung in $\mathbb{Z}$

Der Ansatz ist, aus dem zu untersuchenden Polynom Rückschlüsse auf die Teilpolynome zu bekommen.

Ist das zu untersuchende Polynom aus $\mathbb{Z}$[x], so können seine Teilpolynome auch nur aus $\mathbb{Z}$[x] sein.

Wenn ein Teilpolynom aus $\mathbb{Q}$[x] ist und das zu untersuchende Polynom aus $\mathbb{Z}$[x], so muss ein anderes Teilpolynom existieren, welches einen konstanten Faktor enthält, welcher das Teilpolynom aus $\mathbb{Q}$[x] in ein Polynom aus $\mathbb{Z}$[x] umwandeln kann.

Beispiel:

$$(2x-2)*(x-\frac{1}{2}) = 2*(x-1)*(x-\frac{1}{2}) = (x-1)*(2x-1) = 2x^2-3x+1$$

Ein Polynom, welches Teilpolynome aus $\mathbb{Q}$[x] enthält und für das der eben beschriebene Sachverhalt nicht gilt, kann nur aus $\mathbb{Q}$[x] sein.

Zudem ist aus Analysis bekannt, dass Polynome aus $\mathbb{Z}$[x] für jeden x-Wert aus $\mathbb{Z}$ einen y-Wert aus $\mathbb{Z}$ liefert.

Und es ist logisch, dass gilt: $f(x_i) = a(x_i)*b(x_i) \,|\, \forall x_i \wedge \forall f(x) = a(x)*b(x)$

Da ein Polynom aus $\mathbb{Z}$[x] nur aus Teilpolynomen aus $\mathbb{Z}$[x] bestehen kann, folgt dass jeder y-Wert Teiler enthalten muss, die den y-Werten der Teilpolynome an dem jeweiligen x-Wert entsprechen.

$$b(x)\,|\,a(x) \rightarrow b(x_i)\,|\,a(x_i) \;\; \forall i \in \mathbb{Z}$$

Ein Polynom n-ten Grades kann mit n+1 Stützstellen durch die Interpolation eindeutig bestimmt werden.

Beweis:

- Ein Polynom mit dem Grad n hat genau n Nullstellen, davon maximal n im Reellen und die Restlichen im Komplexen.
- Nehmen wir an: wir haben zwei Polynomen (f(x), g(x)), die den Grad n haben und für (n+1) x-Werte haben sie jeweils die gleichen y-Werte.
- Dann wissen wir, dass a(x)=f(x)-g(x) an den (n+1) x-Werten den y-Wert 0 haben muss.
- Da wir aber wissen, dass a(x) den höchsten Grad von f(x) und g(x) haben muss, können wir daraus schließen, dass a(x) das Nullpolynom sein muss. Woraus natürlich folgt, dass f(x) = g(x) gelten muss.

4.4.2 Kronecker Algorithmus für die Eindeutige Faktorzerlegung in $\mathbb{Z}$

1. Schritt: das zu untersuchende Polynom in ein Polynom aus $\mathbb{Z}$[x] umwandeln

2. Schritt: einsetzen von (n/2)+1 x-Wert aus $\mathbb{Z}$, wobei n der Grad des zu untersuchenden Polynoms ist

3. Schritt: alle Teiler der aus den (n/2)+1 x-Werten erhaltenden y-Werte notieren

4. Schritt: alle möglichen Kombinationen aller Teiler der y-Werte bilden sortiert nach zugehörigen x-Wert

5. Schritt: durch Interpolation mithilfe der x-Werte und den Teilern der y-Werte die möglichen Kandidatenpolynome ermitteln

6. Schritt: die Kandidatenpolynome per Division testen.

Wenn der Rest 0 ist, ist das Kandidatenpolynom wirklich ein Teiler, wenn er ungleich 0 ist nicht.

Wenn das zu untersuchende Polynom während des Testens 1 wird, so sind alle Teilpolynome gefunden und der Algorithmus kann abgebrochen werden, wenn es nicht 1 wird, so ist der Rest am Ende des Testens ein irreduzibles Teilpolynom des ursprünglichen Polynoms.

4.4.3 Beispiel für die Eindeutige Faktorzerlegung in $\mathbb{Z}$

1. Schritt: das zu untersuchende Polynom in ein Polynom aus $\mathbb{Z}[x]$ umwandeln

$1/7x^4-13/7x^3+29/7x^2+19x-42 \quad * 7 \quad = \quad x^4-13x^3+29x^2+133x-294$

2. Schritt: einsetzen von (n/2)+1 x-Wert aus $\mathbb{Z}$, wobei n der Grad des zu untersuchenden Polynoms ist

Wir haben ein Polynom des Grades 4, also müssen wir 3 x-Werte einsetzen.

{(6,36), (3,96) , (5,96)}

3. Schritt: alle Teiler der aus den (n/2)+1 x-Werten erhaltenden y-Werte notieren

36 = 1,2,3,4,6,9,12,18,36

96 = 1,2,3,4,6,8,12,16,24,32,48,96

96 = 1,2,3,4,6,8,12,16,24,32,48,96

4. Schritt: alle möglichen Kombinationen aller Teiler der y-Werte bilden sortiert nach zugehörigen x-Wert

Werte Kombination Nr.	X=6	X=3	X=5
1	1	1	1
2	1	1	2
...	...	...	...
1296	36	96	96

5. Schritt: durch Interpolation mithilfe der x-Werte und den Teilern der y-Werte die möglichen Kandidatenpolynome ermitteln

X=6	X=3	X=5	Kandidatenpolynom
1	1	1	1
1	1	2	1+1/2*(3-x)*(-6+x)
...	...	...	...
36	96	96	36+(-20 (-20 *(-3+x))) *(-6+x)

6. Schritt: die Kandidatenpolynome per Division testen.

$1/7 * (x^4-13x^3+29x^2+133x-294) = 1/7 * (x-2) * (x+3)*(x-7)^2$

Dabei bricht der Algorithmus früher ab, da es keine Nullstelle, gibt die einen Grad > n/2 hat, wobei n der Grad des zu untersuchenden Polynoms ist.

4.4.4 Resümee der Eindeutige Faktorzerlegung in $\mathbb{Z}$

Um die Laufzeit von dem Kronecker Algorithmus relativ gering zu halten, sollte man folgendes beachten:

- Man kann den größten gemeinsamen Teiler aller Koeffizienten ausklammern, um unnötige Teiler zu vermeiden
- Wenn a(0) = 0, dann kann mindestens ein x ausgeklammert werden, dieses kann die Anzahl der Werte verringern, die zur Berechnung einer Interpolation benötigt werden.
- Bei der Wahl der x_i sollte man versuchen, Zahlen mit wenigen Teilern zu finden. Bei $a(x_i) = 0$ kann man $(x-x_i)$ aus a(x) ausklammern. Allerdings darf man die 0 als ermittelten Wert nicht zulassen, da alle Zahlen ein Teiler von 0 sind.
- Kandidatenpolynome die $\in \mathbb{Q}[x]$ aber nicht $\in \mathbb{Z}[x]$ sind, können von vornherein ausgeschlossen werden.

Leider ist die Komplexität sehr groß, da man, wie man sieht, auch bei Polynomen mit einen kleinen Grad und relativ kleinen Koeffizienten sehr viele Interpolationen durchführen müssen.

Auch für die Eindeutige Faktorzerlegung in $\mathbb{Z}$ gibt es bessere Algorithmen.

4.5 Resümee der Eindeutige Faktorzerlegung

Nun sind Algorithmen für die Zerlegung in $\mathbb{Z}$ und in $\mathbb{Z}_p$ bekannt, leider sind beide hier vorgestellten Algorithmen nicht sehr effektiv, können jedoch für kleine Probleme genügen und können auch als Grundlage für andere Zerlegungen dienen.

5 Quadratfreie Faktorisierung

5.1 Ziel dieses Abschnitts

In diesem Abschnitt sollen die Grundlagen der Quadratfreien Faktorisierung, sowie der Algorithmus zur Quadratfreien Faktorisierung in $\mathbb{Z}$ erklärt werden.

Die Quadratfreie Faktorisierung dient als Vorschritt zur eindeutigen Faktorzerlegung oder wird in der Integration verwendet.

5.2 Begriffserklärungen

Die Quadratfreie Faktorisierung von einen Polynom a(x) ist gegeben durch:

$$a(x) = \prod_{k=1}^{m} a_k^k(x)$$

wobei m der Grad von a(x) ist und $a_k(x)$ nur in irreduzible Polynome zerlegt werden kann, die den Grad 1 haben.

Zudem muss gelten: $ggT(a_k(x), a_j(x)) = 1 \,|\forall k \neq j$.

Das bedeutet $a_k(x)$ enthält jeweils alle Faktoren, die k-mal in der Eindeutigen Faktorzerlegung vorkommen.

Beispiel:

$$a(x) = (x-2)(x-5)(x+2)^2 x^2 (x-11)^2 (x+17)^3$$

$$(a_1(x))^1 = ((x-2)(x-5))^1$$

$$(a_2(x))^2 = ((x+2)x(x-11))^2$$

$$(a_3(x))^3 = ((x+17))^3$$

Als Quadratfreien Teil von a(x) bezeichnet man:

$$qf(a(x)) = \prod_{k=1}^{m} a_k(x)$$

Beispiel:

$$a(x) = (x-2)(x-5)(x+2)^2 x^2 (x-11)^2 (x+17)^3$$

Dann ist der Quadratfreie Teil von $qf(a(x)) = (x-2)(x-5)(x+2)\ x\ (x-11)\ (x+17)$

5.3 Grundlagen der Quadratfreien Faktorisierung

Als Grundlage dient die Ermittlung des ggT von zwei Polynomen (welche weiter oben vorgestellt wurde).

Des weiteren benutzen wir Ableitungsregeln die aus Analysis bekannt sein sollten.

Ableitungsregeln:
(Konstantenregel)
f(x) = c -> f‘(x) = 0

(Potenzregel)
$f(x) = x^n \rightarrow f'(x) = n*x^{n-1}$
(Potenzregel)
$f(x) = (a(x))^n \rightarrow f'(x) = a(x)' * n * (a(x))^{n-1}$

(Produktregel)
f(x) = (a(x)*b(x)) -> f’(x) = a’(x)*b(x)
+ a(x)*b’(x)
(Produktregel)
f(x) = (a(x)*b(x)*c(x)) -> f’(x) = a’(x)*b(x)*c(x)
+ a(x)*b’(x)*c(x)
+ a(x)*b(x)*c’(x)
und so weiter …

Aus der Analysis sollte auch bekannt sein, dass bei der Ableitung eines Polynoms alle n-fachen Nullstellen zu (n-1)-fachen Nullstellen werden.

Hier nun ein Beispiel zum Ableiten eines Polynoms:
$f(x) = (x-2) * (x+3)*x^2*(x-7)^2 *(x^2+11)^3$

f‘(x) = 1 * (x+3)*x²*(x-7)² *(x²+11)³
+(x-2) * 1*x²*(x-7)² *(x²+11)³
+(x-2) * (x+3)*2x*(x-7)² *(x²+11)³
+(x-2) * (x+3)*x²*2*(x-7) *(x²+11)³
+(x-2) * (x+3)*x²*(x-7)² *2x*3*(x²+11)²

Wenn wir die (n-1)-fachen Nullstellen aus dem Polynom herausziehen, so sieht man, dass genau diese den ggT(f(x) , f'(x)) ausmachen.

f'(x) = x*(x-7)*(x²+11)² *
(
1 * (x+3)*x*(x-7) *(x²+11)
+(x-2) * 1*x*(x-7) *(x²+11)
+(x-2) * (x+3)*2*(x-7) *(x²+11)
+(x-2) * (x+3)*x*2 *(x²+11)
+(x-2) * (x+3)*x*(x-7) *2x*3
)

ggT(f(x) , f'(x)) = x*(x-7)*(x²+11)²

Dieses gilt nur uneingeschränkt für Polynome aus $\mathbb{Z}[x]$. Wenn man Polynome aus $\mathbb{Z}_p[x]$ hat, dann gilt für $(a(x))^p$

$f(x) = (a(x))^p \rightarrow f'(x) = a(x)' * p * (a(x))^{p-1}$

da aber p mod p = 0 ist, gilt auch f'(x) = 0.

Daher wird hier nur die Quadratfreien Faktorisierung in $\mathbb{Z}$ betrachtet.

5.4 Der Algorithmus für die Quadratfreien Faktorisierung in $\mathbb{Z}$

Beispiel folgt.

1. Schritt: Man setzt einen Index (i) auf 1.
2. Schritt: Man belegt eine Variable mit einem Polynom.
3. Schritt: Man bildet die Ableitung und legt diese auf eine andere Variable.
4. Schritt: Man bestimmt den ggT von dem Polynom und der Ableitung und speichert ihn zwischen.
5. Schritt: Man bestimmt den quadratfreien Teil von dem Polynom, indem man das Polynom durch den zwischengespeicherten ggT teilt.
6. Schritt: Man bestimmt nun ein Polynom, welches alle (i)-fachen Nullstellen enthält. Indem man den quadratfreien Teil durch den ggT von dem quadratfreien Teil und dem zwischengespeicherten ggT teilt.
7. Schritt: Nun setzt man die Variable des Polynoms auf den zwischengespeicherten ggT und erhöht (i) um 1, dann fängt man bei Schritt 3. wieder an.

Der Algorithmus bricht ab, wenn die Ableitung 0 ist.

5.5 Beispiel für Quadratfreie Faktorisierung in $\mathbb{Z}$

In diesem Beispiel soll der Ablauf klar werden, daher wird hier mit Polynomen gerechnet, die in ihre Faktoren aufgeteilt sind. Die Rechenschritte sind so einfacher nachzuvollziehen, allerdings geht dieses natürlich auch mit den ausmultiplizierten Polynomen.

1. Schritt: Man setzt einen Index (i) auf 1.
i := 1;

2. Schritt: Man belegt eine Variable mit einem Polynom.
f := (x-2) * (x+3)*x²*(x-7)² *(x²+11)³

3. Schritt: Man bildet die Ableitung und legt diese auf eine andere Variable.
df := f' = x*(x-7)*(x²+11)² *((x+3)*x*(x-7) *(x²+11)+(x-2) * x*(x-7) *(x²+11)+(x-2) * (x+3)*2*(x-7) *(x²+11)+(x-2) * (x+3)*x*2 *(x²+11)+(x-2) * (x+3)*x*(x-7) *2x*3)

4. Schritt: Man bestimmt den ggT von dem Polynom und der Ableitung und speichert ihn zwischen.
ggTeiler := ggT(f, df) = x*(x-7)*(x²+11)²

5. Schritt: Man bestimmt den quadratfreien Teil von dem Polynom indem man das Polynom durch den zwischengespeicherten ggT teilt.
quadFrei := f / ggTeiler = (x-2) * (x+3)*x*(x-7) *(x²+11)

6. Schritt: Man bestimmt nun ein Polynom, welches alle (i)-fachen Nullstellen enthält. Indem man den quadratfreien Teil durch den ggT von dem quadratfreien Teil und dem zwischengespeicherten ggT teilt.
polNullstellenDesGrades[i] := quadFrei / ggT(quadFrei, ggTeiler) =

$$\frac{(x-2) * (x+3)*x*(x-7) *(x^2+11)}{x*(x-7)*(x^2+11)} = (x-2)*(x+3)$$

7. Schritt: Nun setzt man die Variable des Polynoms auf den zwischengespeicherten ggT und erhöht (i) um 1, dann fängt man bei Schritt 3. Wieder an.
f := ggTeiler = x*(x-7)*(x²+11)²
i := i+1 = 2;

Nun werden die Schritte wiederholt.

df := f' = (x²+11) * ((x-7)*(x²+11) + x*(x²+11) + x * (x-7) * 2x * 2)
ggTeiler := ggT(f, df) = (x²+11)
quadFrei := f / ggTeiler = x*(x-7) *(x²+11)
polNullstellenDesGrades[i] := quadFrei / ggT(quadFrei, ggTeiler) = x*(x-7)
f := ggTeiler = (x²+11)
i := i+1 = 3;

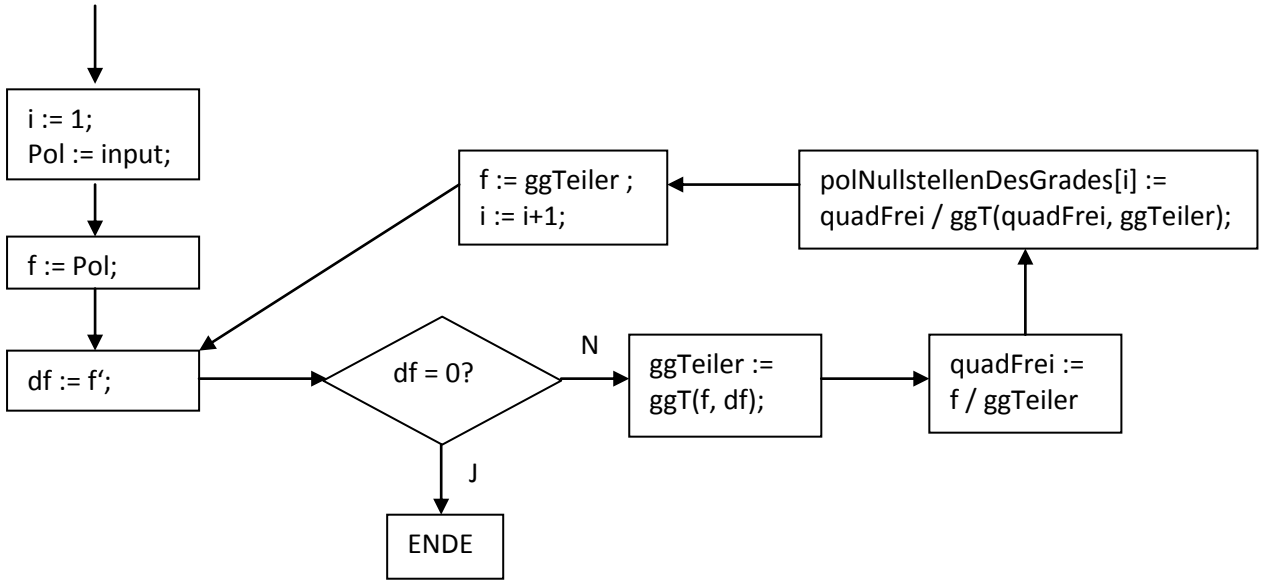
df := f‘ =2x
ggTeiler := ggT(f, df) = 1
quadFrei := f / ggTeiler = (x^2+11)
polNullstellenDesGrades[i] := quadFrei / ggT(quadFrei, ggTeiler) = (x^2+11)
f := ggTeiler = 1
i := i+1 = 4;

df := f‘ = 0 -> Abbruch des Algorithmus

Die Ergebnisse sind dann:
polNullstellenDesGrades[1] = (x-2)*(x+3)
polNullstellenDesGrades[2] = x*(x-7)
polNullstellenDesGrades[3] = (x^2+11)

Diese Ergebnisse würden jedoch natürlich auch in ausmultiplizierter Form vorliegen, so dass man nun eine Eindeutige Faktorzerlegung durchführen müsste.

5.6 Ablaufplan der Quadratfreie Faktorisierung in $\mathbb{Z}$



5.7 Resümee der Quadratfreie Faktorisierung in $\mathbb{Z}$

Es treten die bekannten Schwierigkeiten der Teilschritte auf (ggT – intermediate expression swell).

Es funktioniert nur bei Polynomen aus $\mathbb{Z}$[x].
Die Erweiterung auf $\mathbb{Z}_p$[x] gibt es in einen späteren Seminar.

Wenn der Grad aller Nullstellen = 1 ist, gibt der Algorithmus keine Verbesserung, kostet jedoch Zeit.
Wenn der Grad der Nullstellen unterschiedlich ist oder > 1, erleichtert der Algorithmus die Faktorisierung immens.

Bei einem Polynom $(x-1)*(x-2)^{14}$ bekommt man für die 2-fachen bis 13-fachen Nullstellen die Konstante 1.
So zu sagen die Zerlegung von $(x-1)*1^2*1^3*1^4*1^5*1^6*1^7*1^8*1^9*1^{10}*1^{11}*1^{12}*1^{13}*(x-2)^{14}$.

Es gibt eine Verbesserung von diesem Algorithmus welche von Yun entwickelt wurde, welcher vor allem auf Verbesserung der Komplexität des ggT ausgerichtet ist.

Allgemein ist es sinnvoll vor der Eindeutigen Faktorzerlegung eine Quadratfreie Faktorisierung durchzuführen, sie geht vergleichsweise schnell und kann die Laufzeit extrem verbessern.

6 Rationale Funktionen

6.1 Grundlagen

Wie wir im Seminar „Langzahlarithmetik: Vereinfachen von Brüchen“ gesehen haben, liegen Brüche nach der Addition/Subtraktion oder Multiplikation/Division zweier Brüche nicht zwingend in einer gekürzten Form vor.
Daher kann man nach solchen Operationen den ggT auf den Nenner und Zähler anwenden. Das Ergebnis kann man benutzen, um Nenner und Zähler zu kürzen.
Das gleiche gilt für Funktionen.

Beispiel: $\frac{x^3-2x^2-5x+6}{x-1}=x^2-x-6$

Wobei wir x^2-x-6 als Standarddarstellung bezeichnen.

Hierbei ist darauf zu achten, dass Information verloren gehen können, z.B. die oben gezeigte Standarddarstellung entfernt eine hebbare Lücke, so dass die Polynome sich unterscheiden.

6.2 Resümee Rationale Funktionen

Man muss aufpassen, was man mit dem Polynom aussagen will, so kann bei einen praktischen Beispiel ein Bruch mit Polynomen entstanden sein, der mathematisch gesehen Lücken enthält, wobei jedoch in der Realität die Funktion durchgängig definiert ist.

Genauso gut kann es sein, dass man nur einen Wertebereich betrachtet, dann wäre eine hebbare Lücke außerhalb dieses Bereichs uninteressant.

Allerdings sollte man vorher überlegen ob man eine Lücke entfernen will oder nicht.

7 Literaturverzeichnis

Es wurde verwendet:

Koepf, Wolfram: Computeralgebra Eine algorithmisch orientierte Einführung, Springer-Verlag, 2006

Kaplan, Michael: Computeralgebra, Springer-Verlag, 2005

Als Einführung in das Thema erscheinen folgende Bücher weniger geeignet:

Klose, Jürgen: Schnelle Polynomarithmetik zur exakten Lösung des Fermat-Weber-Problems, Universität Erlangen, 1993

Klotz, Gerhard: Faktorisierung von Matrizen mit maximaler Genauigkeit, Universität Karlsruhe, 1987

Zu empfehlen ist:

http://www.arndt-bruenner.de/mathe/mathekurse.htm

http://www-madlener.informatik.uni-kl.de/teaching/ss2007/ca/ca.html

Die Folien zur Vorlesung, zu finden unter der Überschrift „Vorlesung“ ganz am Ende kurz vor der Überschrift „Übungen“, enthalten viele Bilder, die einige Sachverhalte sehr deutlich darstellen.