Über die Bestimmung der Dimension von Polynomidealen

Diplomarbeit

von

Heinz Kredel

Betreuer:

Prof. Dr. W. Böge

Universität Heidelberg

Fakultät für Mathematik

April 1985

Besonderen Dank möchte ich meinem Lehrer Herrn Prof. Dr. W. Böge aussprechen, auf den viele Anregungen und Korrekturen zurückgehen. Auch bei Herrn Dr. P. Schmidt, der mir besonders für die Algorithmen Anregungen gegeben und Verbesserungsvorschläge gemacht hat, möchte ich mich herzlich bedanken. Auch bei Herrn Dipl. Math. R. Gebauer möchte ich mich für die fruchtvolle Zusammenarbeit bei der Implementierung der Algorithmen (besonders der vielen hier nicht dargestellten) herzlich bedanken.

Weiter ist zu erwähnen, daß ich ohne die Hilfe und Geduld meiner lieben Frau Lioba und meines Sohnes Samuel diese Arbeit unmöglich hätte fertigstellen können.

A. Einleitung

Vorliegende Arbeit beschäftigt sich mit der Bestimmung der Dimension von Polynomidealen. Anlass zu dieser Arbeit ist das Problem die 'Projektion' einer Polynommenge zu bestimmen, das sich in der Collins'schen Quantorenelimination stellt [Collins 1974] [Collins 1975] . Die Menge der 'Projektionspolynome' läßt sich verkleinern, falls man von den Koeffizienten gewisser dabei vorkommender Polynome entscheiden kann, daß sie höchstens endlich viele gemeinsame Nullstellen besitzen.

Zur Festsstellung, wieviele gemeinsame Nullstellen eine gegebene Polynommenge besizt, d.h. welche Dimension das von diesen Polynomen erzeugte Ideal hat, werden hier zwei Wege untersucht:

- zum einen werden mit Hilfe von iterierten Resultantensystemen einvariablige Polynome konstruiert, deren Existenz hinreichend dafür ist, daß die Dimension des Ideals kleiner oder gleich 0 ist.

- zum anderen wird die Gröbner Basis des von den Polynomen erzeugten Ideals mit einem erst 1968 entwickelten Algorithmus gebildet [Buchberger 1970] , aus der dann die Dimension des Ideals abgelesen werden kann.

Der Vorteil der iterierten Resultantensysteme liegt darin, daß ihre Komplexität bekannt ist, und schnelle Algorithmen vorhanden sind, die die Resultanten berechnen [Loos 1982b] . Der Nachteil besteht allerdings darin, daß die angegebeben Bedingungen nur hinreichend dafür sind, daß die Polynome endlichviele gemeinsame Nullstellen besitzen.

Mit Hilfe von Gröbner-Basen erhält man hingegen notwendige und hinreichende Bedingungen dafür, daß ein Polynomideal nur endlichviele gemeinsame Nullstellen besitzt. Allerdings ist die Komplexität des von B. Buchberger angegebenen Algorithmus zur Berechnung von Gröbner-Basen nur für Polynomringe in zwei und seit kurzem auch in drei Variablen bekannt [Buchberger 1982] [Winkler 1984] [Möller & Mora 1984] . In der Praxis stellt sich heraus, daß der Algorithmus sehr komplexe Berechnungen ausführt, die nur bei nicht zu großen Problemen tatsächlich durchgeführt werden können.

In den im Text angegebenen Algorithmen wird eine Pseudo-Programmiersprache verwendet, die sich zwar nah an ALDES [Collins & Loos 1980] und an die implementierten SAC2 [Collins & Loos 1980] [Collins & Loos 1981] Programme anlehnt, aber anstelle von Unterprogrammaufrufen wird teilweise in mathematischer Notation beschrieben was zu tun ist.

Für die Vielzahl der nicht aufgelisteten Programme sei auf [Gebauer & Kredel 1983abcd] [Gebauer & Kredel 1984a] verwiesen.

B. Zur Notation

Im folgenden seien:
K ein (kommutativer) Körper,
N die natürlichen Zahlen einschließlich der Null {0,1,2,3,...}.

Weiter bezeichnen:
a, b, c Elemente von K,
r, n, m, l, k, i, j Elemente von N,

(i), (i,j) Tupel von Zahlen aus N, d.h. für gegebene r, l ε N

$(i) = (i_1,...,i_r) \varepsilon N^r$,

$(i,j) = (i_1,...,i_r,j_1,...,j_l) \varepsilon N^{r+l}$.

$K[x_1,...,x_r]$ sei der Polynomring in r Variablen über K.

Falls erforderlich, werde ich die Struktur des Polynomrings

z.B. durch $(...(K[x_1][x_2])...)[x_r]$ besonders hervorheben.

B.1. Potenzprodukte

u, v oder w seien Potenzprodukte (Abkürzung PP) aus dem Polynomring, d. h.

$$u = x_1^{i(1)} \cdots x_r^{i(r)} \varepsilon K[x_1,...,x_r]$$
$$= x^{(i)} \varepsilon K[x_1,...,x_r].$$

Der totale Grad eines PP wird mit tdeg bezeichnet

$tdeg(u) = tdeg(x^{(i)}) = \Sigma_{k=1,...,r} i_k$.

Die partielle Ordnung der Teilbarkeit der PP sei mit

$<_M$, $>_M$, $\leq_M$, $\geq_M$ bezeichnet:

z. B. $u <_M v$ <==> es gibt $w \neq 1$ mit $u w = v$

z. B. $u \leq_M v$ <==> es gibt w mit $u w = v$

Eine totale Ordnungsrelation $<_T$ auf der Menge aller PP

heißt zulässig, wenn sie die folgenden beiden Eigenschaften besitzt:

$1 = x^{(0)} <_T u$ für alle PP $u \neq 1$

$u <_T v \implies u\,w <_T v\,w$.

Wir schreiben wie üblich auch

$u >_T v$ für $v <_T u$

und $u \geq_T v$ für $u >_T v$ oder $u = v$.

Beispiele für zulässige Ordnungen sind:

die lexikographische Ordnung

$x^{(i)} <_L x^{(j)}$ <==> falls es ein k ($1 \leq k \leq n$) gibt mit

$i_m = j_m$ für m=1,...,k-1 und

$i_k < j_k$

etwa

$$x_1 <_L x_1^2 <_L x_1^3 <_L \ldots <_L x_2 <_L \ldots <_L x_r <_L \ldots,$$

oder auch die graduierte Ordnung

$x^{(i)} <_G x^{(j)}$ <==> falls $tdeg(x^{(i)}) < tdeg(x^{(j)})$

oder falls $tdeg(x^{(i)}) = tdeg(x^{(j)})$

dann $x^{(i)} <_L x^{(j)}$

etwa

$$x_1 <_G \ldots <_G x_r <_G x_1^2 <_G x_1 x_2 <_G \ldots <_G x_r^2 <_G \ldots$$

B.2. Allgemeine Elemente des Polynomrings

Allgemeine Elemente des Polynomrings seien mit f, g oder h bezeichnet:

$f = \Sigma_{j=1,\ldots,m} a_j u_j$

mit $a_j \in K$, u_j Potenzprodukt für j=1,...,m, $m \in N$

und $u_i \neq u_j$ für $i \neq j$,

bzw.

$$f = \Sigma\, a_{(i)}\, x^{(i)}$$

mit $a_{(i)} \varepsilon K$, aber nur endlichviele $\neq 0$.

Ein Polynom der Form $a_{(i)}\, x^{(i)}$ heiße Term oder auch Monom.
Der totale Grad eines Polynoms wird ebenfalls mit tdeg bezeichnet

$$tdeg(f) = \max\{ tdeg(x^{(i)}) : a_{(i)} \neq 0 \}.$$

Wir fixieren für das folgende eine zulässige Ordnung $<_T$ auf der Menge aller PP.

lp(f) sei das 'leading' d.h. das größte oder das höchste Potenzprodukt von f bezüglich $<_T$.

lbcf(f) ist der 'leading base coefficient', das ist der Koeffizient von lp(f) in f. f heißt normiert, falls lbcf(f) = 1 ist.

lbcf(f) lp(f) heißt 'head term' von f.

Für die rekursive Polynomdarstellung legen wir die lexikographische Termordnung zugrunde.

Sei nun $f \varepsilon (...(K[x_1][x_2])...)[x_r]$, x_r heiße Hauptvariable:

deg(f) ist der Grad von f bezüglich x_r,

ldcf(f) ist der Koeffizient von $x_r^{deg(f)}$, es ist

$ldcf(f) \varepsilon (...(K[x_1][x_2])...)[x_{r-1}]$.

Der Grad von f heißt formal, falls zugelassen wird, daß ldcf(f) = 0 ist.

f heißt normiert bezüglich der Hauptvariablen, falls ldcf(f) = 1 ist.

red(f) ist das Reduktum von f: $red(f) = f - ldcf(f)\, x_r^{deg(f)}$.

Das i-te Reduktum von f (i=0,...,deg(f)) ist wie folgt definiert:

$red^0(f) = f$

$red^i(f) = red(red^{i-1}(f))$.

der(f) ist die Ableitung von f bezüglich x_r.

B.3. Teilmengen aus dem Polynomring

Endliche Teilmengen aus dem Polynomring seien mit F, G oder A bezeichnet:

$F = \{ f_1, \ldots, f_n \} \subset K[x_1,\ldots,x_r]$

$A = \{ A_1, \ldots, A_n \} \subset K[x_1,\ldots,x_r]$.

Unendliche Teilmengen aus dem Polynomring, die wir betrachten sind in der Regel Ideale, die mit $\underline{A}$ oder $\underline{B}$ bezeichnet werden:

$\underline{A} = (A_1, \ldots, A_n) \subset K[x_1,\ldots,x_r]$ d. h.

$\underline{A} = \{ \Sigma_{i=1,\ldots,n} h_i A_i : h_i \in K[x_1,\ldots,x_r]$

für $i=1,\ldots,n$ für $n \in N \}$.

ideal(F) bezeichne das von F erzeugte Ideal d. h.

$\text{ideal}(F) = (f_1, \ldots, f_n)$

$\underline{A} + \underline{B} = (A_1, \ldots, A_n, B_1, \ldots, B_m)$,

$\underline{A} \sqcap \underline{B} = \{ f : f \in \underline{A}$ und $f \in \underline{B} \}$.

B.4. Der Restklassenring nach einem Ideal

Mit M sei der Restklassenring des Polynomrings nach dem Ideal $\underline{A}$ bezeichnet:

$$M = K[x_1,\ldots,x_r]_{/\underline{A}}.$$

Die Elemente des Restklassenrings werden mit [f], [g] d.h. mit eckigen Klammern um Elemente des Polynomrings bezeichnet:

$$[f] = f + \underline{A}.$$

M ist ein Vektorraum über K , von im allgemeinen unendlicher Vektorraum Dimension. Die Menge { [u] : u ein PP } ist ein Erzeugendensystem für M.

B.5. SAC2 Computer Algebra System

SAC2 (Symbolic and Algebraic Computation Version 2) ist ein Computer Algebra System, das auf Listenverarbeitung aufbaut. Darin werden Elemente algebraischer Strukturen, zum Beispiel Polynome oder Koeffizienten von Polynomen, durch Listen dargestellt.
SAC2 hat seinen Ursprung in SAC1, einer ebenfalls in FORTRAN geschriebenen Unterprogrammsammlung für Computer Algebra. SAC2 wurde von [Collins & Loos 1980] entwickelt.
SAC2 bietet z.B. Unterprogramme für exakte Arithmetik von ganzen Zahlen, rationalen Zahlen und Polynomen mit solchen Koeffizienten. Desweiteren sind Unterprogramme zur Berechnung von Resultanten, Berechnung von isolierenden Intervallen für reele Nullstellen univariater Polynome und zur Bestimmung der Zerlegung multivariater Polynome in irreduzible Faktoren vorhanden. Für die Beschreibung der informatischen und mathematischen Methoden, die diesen Programmen zugrundeliegen, sei auf [Knuth] und [Buchberger & Collins & Loos] verwiesen.

Entsprechend der 'Einführung in SAC2' [Gebauer & Jagoda 1982a] sei hier nur das Listenkonzept kurz zusammengefaßt:

B.5.1. Definition : ß ist die Basis der SAC2 Zahlendarstellung,

$ß = 2^{29}$ für die IBM 3081 Rechenanlage.

B.5.2. Definition : Atome sind ganze Zahlen aus dem Bereich von -ß bis ß jeweils ausschließlich.

"Listen" werden in SAC2 rekursiv definiert:

B.5.3. Definition : Objekte sind Atome oder Listen.

B.5.4. Definition : Listen sind endliche Folgen von Objekten.

Man kann auch sagen Listen sind geordnete endliche Mengen von Atomen und/oder Listen.

Bezeichnungen

Die leere Liste wird mit '()' bezeichnet.

Nicht leere Listen werden durch '$(a_1,...,a_n)$' bezeichnet,

die a_i (i=1,...,n) sind Objekte.

B.5.5. Definition Listenconstructor: COMP

Sei a ein Objekt, $L = (a_1,...,a_n)$ eine Liste,

dann ist $COMP(a,L) = (a,a_1,...,a_n)$.

B.5.6. Definition : FIRST

Sei L eine Liste, $L = (a_1,\ldots,a_n) \neq ()$ dann ist

$a_1 = \text{FIRST}(L)$.

B.5.7. Definition : RED

Sei L eine Liste, $L = (a_1,\ldots,a_n) \neq ()$ dann ist

$(a_2,\ldots,a_n) = \text{RED}(L)$.

B.5.8. Definition : ADV
Sei L eine Liste, $L \neq ()$ dann gilt
$\text{ADV}(L.a,L') \Longleftrightarrow a = \text{FIRST}(L)$ und $L' = \text{RED}(L)$.

B.5.9. Definition : CONC

Seien $L = (a_1,\ldots,a_n)$ und $L' = (a'_1,\ldots,a'_{n'})$ Listen,

dann ist

$\text{CONC}(L,L') = (a_1,\ldots,a_n,a'_1,\ldots,a'_{n'})$.

B.6. ALDES Programmiersprache

Die angegebenen Computerprogramme sind in Anlehnung an die ALDES Programmiersprache geschrieben und sollten weitgehend ohne zusätzliche Informationen zu verstehen sein.

ALDES (ALgorithm DEScription language) ist eine von [Collins & Loos 1980] entwickelte Programmiersprache, die es gestattet Algorithmen in ALGOL ähnlicher Form zu codieren, und dann in FORTRAN zu übersetzen.

ALDES verfügt über die üblichen Sprachkonstrukte wie etwa:

Zuweisungen
Funktionsaufrufe
Ausdrücke, insbesondere Bedingungen
if then else
repeat until
while do
for do
go to
return

Allerdings besitzt ALDES kein Konzept der Typendeclarationen und Typenüberprüfung. Daher kann von den als Listen vorliegenden Objekten nicht eindeutig festgestellt werden, zu welchen algebraischen Strukturen sie gehören.

Um den Wechsel einer algebraischen Struktur anzudeuten verwende ich die Notation

$$p \leftarrow q \in M$$

Mit der Bedeutung, daß q Element einer Struktur M' ist und durch einen bekannten oder offensichtlichen Isomorphismus zwischen den Strukturen M' und M auf ein Element p aus M abgebildet wird.
In den eigentlichen Programmen werden die Wechsel der algebraischen Strukturen dann durch Aufruf von Unterprogrammen realisiert, die die Listen entsprechend einem Isomorphismus konvertieren.

Um unverständliche Unterprogrammaufrufe einzusparen wird auch der sogenannte Match-Operator (oder auch Unifikator) verwendet. Zum Beispiel

```
R = a lp(R) + R'.
e ← lp(R).
```

steht für

```
DIPMAD(R.a,e,R').
```

oder für

IPFAC(r,A1.s,c,L).

sei vereinfachend

$$A1 = a_1^{e(1)} \cdot \ldots \cdot a_l^{e(l)}.$$

$$L = \{ a_1, \ldots, a_l \}.$$

geschrieben.

Zur Vereinfachung der Schreibweise wird in den Algorithmen oft nicht zwischen Listen und endlichen Mengen unterschieden. Ebenso kann es vorkommen, daß zwischen 0, {} und $\emptyset$ nicht unterschieden wird.

C. Verwendete Sätze und Definitionen

Im folgenden sollen einige wichtige Sätze und Definitionen kurz zusammengestellt werden. Bis auf gekennzeichnete Ausnahmen sind die Definitionen und Sätze [Vd Waerden, I] entnommen.

C.1. Allgemeines über den Polynomring und Ideale

C.1.1. Satz : $K[x]$ ist ein euklidischer Ring.

C.1.2. Folgerung : $K[x]$ ist ein Hauptidealring.

C.1.3. Satz :

In $K[x_1,\ldots,x_r]$ gilt die Eindeutige Zerlegbarkeit in

irreduzible Elemente.
D.h. der Polynomring über einem Körper ist ein ZPE-Ring oder im Englischen ein UFD Ring (Unique Factorisation Domain).

C.1.4. Hilfssatz : Ein Ideal bleibt gleich, falls ein erzeugendes Element um eine Summe anderer Erzeugender versehen mit einem Polynomfaktor abgeändert wird. D.h. falls

$\underline{A} = (A_1,\ldots,A_n)$ und

$\underline{A}' = (A_1 + \Sigma_{j=2,\ldots,n} f_j A_j, A_2,\ldots, A_n)$

wobei $f_j \varepsilon K[x_1,\ldots,x_r]$ $(j=2,\ldots,n)$,

so ist $\underline{A} = \underline{A}'$.

Beweis:

Da $A_j \varepsilon \underline{A}$ $(j=2,\ldots,n)$ und

$B = A_1 + \Sigma_{j=2,\ldots,n} f_j A_j \varepsilon \underline{A}$

folgt $\underline{A}' \subset \underline{A}$.

Umgekehrt: da $A_j \varepsilon \underline{A}'$ $(j=2,\ldots,n)$ und

$$A_1 = [A_1 + \Sigma_{j=2,\ldots,n} f_j A_j] - [\Sigma_{j=2,\ldots,n} f_j A_j]$$

$$= B - [\Sigma_{j=2,\ldots,n} f_j A_j] \varepsilon \underline{A}$$

folgt $\underline{A} \subset \underline{A}'$. ▮

C.1.5. Satz : (Hilbertscher Basissatz) Ideale im Polynomring über einem Ring mit Einselement, in dem ebenfalls der Basissatz gilt, sind endlich erzeugt.

C.1.6. Definition : Das Radikal eines Ideals $\underline{A}$ (in Zeichen rad($\underline{A}$)) ist die Menge aller

$f \in K[x_1,\ldots,x_r]$ für die es einen Exponenten e

(e eine natürliche Zahl) gibt, so daß

$$f^e \in \underline{A}.$$

Insbesondere ist $\underline{A} \subset \text{rad}(\underline{A})$.

C.1.7. Definition : Ein Ideal heißt prim, falls

$M = K[x_1,\ldots,x_r]/\underline{A}$ ein Integritätsring ist.

C.1.8. Bemerkung : Da M ein Integritätsring ist, läßt er sich in einen Körper einbetten, z.B. in den Quotientenkörper, dieser heiße Restklassenkörper des Primideals.

C.2. Nullstellen von Idealen

Neben der Bestimmung von Nullstellen univariater Polynome, stellt sich die Aufgabe die gemeinsamen Nullstellen mehrerer Polynome zu bestimmen. Die folgenden Definitionen und Sätze sind [Gröbner, II, p. 1 ff] entnommen.
Es sei:
L ein Erweiterungskörper von K,
E ein algebraisch abgeschlossener Erweiterungskörper von K,
U ein sogenannter Universalkörper, d.h. ein algebraisch abgeschlossener Erweiterungskörper von K, von abzählbar unendlichem Transzendenzgrad über K,
K(a) der Erweiterungskörper von K mit $a \in U$.

C.2.1. Definition :

Es sei $\underline{A} \subset K[x_1,\ldots,x_r]$ dann sei

$$\text{NST}(\underline{A}) = \{ a \in U^r : f(a) = 0 \text{ für alle } f \in \underline{A} \}$$

und es sei NST(F) = NST(ideal(F)). NST($\underline{A}$) heißt das Nullstellengebilde oder die Nullstellenmenge des Ideals $\underline{A}$.

C.2.2. Satz : Eigenschaften der Nullstellengebilde:

$\underline{A} \subset \underline{B} \Longrightarrow \text{NST}(\underline{B}) \subset \text{NST}(\underline{A})$.

$NST(\underline{A} + \underline{B}) = NST(\underline{A}) \sqcap NST(\underline{B})$.

$NST(\underline{A} \sqcap \underline{B}) = NST(\underline{A}) \sqcup NST(\underline{B})$.

$NST((fg)) = NST((f)) \sqcup NST((g))$.

Sei n eine natürliche Zahl, dann gilt:

$NST(\underline{A}) = NST(rad(\underline{A})) = NST(\underline{A}^n)$.

C.2.3. Satz : Es gilt

$1 \varepsilon \underline{A} \Leftrightarrow NST(\underline{A}) = \emptyset$.

C.2.4. Satz : [Vd Waerden] (Hilbertscher Nullstellensatz): Falls

$f(a_1,\ldots,a_r) = 0$ für alle $(a_1,\ldots,a_r) \varepsilon NST(\underline{A})$,

dann gibt es einen Exponenten e, e eine natürliche Zahl, so daß

$f^e \varepsilon \underline{A}$.

Dieser Satz läßt sich Verallgemeinern:

C.2.5. Satz : [Gröbner, II, p. 7] Falls $NST(\underline{A}) \subset NST(\underline{B})$, dann gibt es einen Exponenten e, e eine natürliche Zahl, so daß

$\underline{B}^e \subset \underline{A}$.

C.2.6. Satz : $NST(\underline{A}) = NST(\underline{B}) \Leftrightarrow rad(\underline{A}) = rad(\underline{B})$.

Beweis:

"<==" $NST(\underline{A}) = NST(rad(\underline{A}))$
$= NST(rad(\underline{B})) = NST(\underline{B})$.

"==>" Es sei $NST(\underline{A}) \subset NST(\underline{B})$ dann gibt es nach [C.2.5., p. 16] eine natürliche Zahl e mit

$\underline{B}^e \subset \underline{A} \Rightarrow rad(\underline{B}^e) \subset rad(\underline{A})$.

Da $rad(\underline{B}^e) = rad(\underline{B})$ gilt weiter

$rad(\underline{B}) \subset rad(\underline{A})$. Durch vertauschen der Rollen von $\underline{A}$ und $\underline{B}$ erhalten wir :
$NST(\underline{B}) \subset NST(\underline{A}) \Rightarrow rad(\underline{A}) \subset rad(\underline{B})$,
zusammen ergibt sich die Behauptung ▌

C.3. Algebraische Abhängigkeit und Unabhängigkeit

Entsprechend [Vd Waerden, I, p. 224 ff.] sei definiert:

C.3.1. Definition : Ein $u \in U$ heißt algebraisch abhängig von $u_1,\ldots,u_n$, falls u algebraisch in bezug auf den Körper $K(u_1,\ldots,u_n)$ ist, d.h. wenn u einer algebraischen Gleichung

$$c_m u^m + \ldots + c_1 u + c_0 = 0$$

mit $c_i \in K(u_1,\ldots,u_n)$ für $i=1,\ldots,m$ genügt, wobei die c_i Polynome in $u_1,\ldots,u_n$ sind, die nicht alle gleich Null sind.

C.3.2. Definition :

$u_1,\ldots,u_n$ heißen algebraisch unabhängig in bezug auf den Körper K, wenn kein u_i für $i=1,\ldots,n$ algebraisch von den übrigen abhängt.

Eine Teilmenge V eines Erweiterungskörpers L von K heißt maximale algebraisch unabhängige Teilmenge von L über K falls V algebraisch unabhängig über K ist, aber jedes $u \in L$ algebraisch abhängig ist von V über K

C.3.3. Satz : Je zwei maximal algebraisch unabhängige Teilmengen V, U von L über K sind entweder beide unendlich oder beide endlich mit gleicher Elementzahl.

C.3.4. Definition : Diese Anzahl der maximal algebraisch unabhängigen Elemente eines Erweiterungskörpers L von K wird Transzendenzgrad des Körpers L in bezug auf K genannt.

C.4. Resultanten und Subresultanten

Resultanten lassen sich ganz allgemein in Polynomringen mit einem (kommutativen) Ring R als Koeffizientenbereich definieren.
Um zu den Sätzen über Resultanten zugelangen, ist es erforderlich, daß R keine Nullteiler enthält, also ein Integritätsring ist.

In den Beweisen wird weiter benötigt, daß der Polynomring ein ZPE-Ring ist. In den hier betrachteten Anwendungen ist R der Polynomring in (r-1) Variablen über einem Körper ($r \geq 1$), also ein Integritätsring mit der ZPE Eigenschaft.

Desweiteren werden Nullstellen von Polynomen aus R[x] betrachtet, wobei die Nullstellen aus R, seinem Quotientenkörper K oder aus einem Erweiterungskörper von K stammen können.
In [Vd Waerden] [Vd Waerden 1931] werden die Resultanten in der Regel für Polynomringe über Körpern betrachtet. [Gröbner] setzt dagegen die ZPE Eigenschaft für R[x] nicht voraus. In seinen Beweisen geht er allerdings zum Quotientenkörper K von R über, und dann ist K[x] wieder ein ZPE Ring. Zur Vereinfachung der Sprechweise werde ich im folgenden die ZPE Eigenschaft für R[x] (und somit auch für R) voraussetzen. Damit wird die Formulierung 'gemeinsamer Faktor' von Polynomen sinnvoll und es muß nicht nur von 'gemeinsamen Nullstellen' gesprochen werden.

C.4.1. Definition : Die Resultante zweier Polynome:

$$A(x) = \Sigma_{i=0,\ldots,m}\ a_i\ x^i,$$

$$B(x) = \Sigma_{i=0,\ldots,n}\ b_i\ x^i$$

$$\begin{aligned} res: R[x] \times R[x] &\longrightarrow R \\ (A,B) &\longrightarrow res(A,B) = det(M) \end{aligned}$$

wobei $\max\{m,n\} > 0$.

M ist die Silvester Matrix gebildet aus n Zeilen der Koeffizienten von A und m Zeilen der Koeffizienten von B:

$$
M = \begin{vmatrix}
a_m & a_{m-1} & \cdots & & a_0 & & & \\
 & a_m & a_{m-1} & \cdots & & a_0 & & \\
 & & \cdots & & & & \cdots & \\
 & & & a_m & a_{m-1} & \cdots & a_1 & a_0 \\
b_n & b_{n-1} & \cdots & & b_0 & & & \\
 & b_n & b_{n-1} & \cdots & & b_0 & & \\
 & & \cdots & & & & \cdots & \\
 & & b_n & b_{n-1} & \cdots & & b_0 & \\
 & & & b_n & b_{n-1} & \cdots & & b_0
\end{vmatrix}
$$

Insbesondere gilt z.B. für n=0:

$$res(A,B) = b_0^{\,m}.$$

C.4.2. Satz : res(A,B) ε ideal({A,B})
d. h. es gibt f, g ε R[x] mit res(A,B) = f A + g B

C.4.3. Satz : res(A,B) = 0 genau dann, falls die Polynome A und B einen gemeinsamen nichtkonstanten Faktor besitzen oder falls beide Anfangskoeffizienten verschwinden.

C.4.4. Satz : res(A B,C) = res(A,C) res(B,C)

C.4.5. Definition : [Collins 1974] Seien A(x) und B(x) wie oben.

Für $0 \le i \le j \le \min\{m,n\}$ sei M_{ij} die Matrix,

die aus M entsteht, falls man die letzten j Zeilen der Koeffizienten von A und die letzten j Zeilen der Koeffizienten von B, sowie die letzten 2j+1 Spalten ausser der Spalte m+n-i-j streicht.

Die j-te Subresultante von A und B ist dann das Polynom

$$S_j(A,B) = \Sigma_{i=0,\ldots,j} \det(M_{ji}) \, x^i$$

mit einem Grad kleiner gleich j.

Der j-th principal subresultant coefficient von A und B ist dann

$psc_j(A,B) = \det(M_{jj})$.

D.h. $psc_j(A,B)$ ist der Koeffizient von x^j in $S_j(A,B)$,

insbesondere ist $psc_0(A,B) = res(A,B)$.

Für die algorithmischen Aspekte der Berechnung von Resultanten und Subresultanten sei auf [Loos 1982b] verwiesen.
Anwendungen des Resultantenkalküls auf die Berechnung von gemeinsamen Nullstellen von Polynommengen finden sich z.B. in [Loos 1982a] .

C.5. Kroneckersche Resultantensysteme

Kroneckerschen Resultantensysteme bieten eine weitere Möglichkeit Variablen zu eliminieren. Der Aufwand zu ihrer tatsächlichen Berechnung ist allerdings auch für heutige Computeranlagen zu groß, um sie praktisch durchzuführen.

Sei wieder R ein (kommutativer) Integritätsring (mit 1) und R[x] ein ZPE-Ring. Gegeben eine Menge F von m Polynomen in einer Variablen mit maximalem Grad n:

$f_1, \ldots, f_m \in R[x]$, $n = \max\{ \deg(f_i) : i=1,\ldots,m \}$.

Mit Hilfe von 2m Unbestimmten, die dem Ring R adjungiert werden, bilde man die zwei Polynome g, h in

$R[u_1,\ldots,u_m,v_1,\ldots,v_m][x]$:

$$g = u_1 f_1 + \ldots + u_m f_m$$

$$h = v_1 f_1 + \ldots + v_m f_m.$$

Nach den Potenzen der u's und v's entwickelt ist die Resultante der beiden Polynome:

$$res(g,h) = \Sigma R_{(i,j)} u^{(i)} v^{(j)}.$$

C.5.1. Definition : [Gröbner, II, p. 14]

Die $R_{(i,j)}$ bilden das

Kroneckersche Resultantensystem zu den Polynomen aus F.

Bezeichnung: $res(F) = \{ \ldots, R_{(i,j)}, \ldots \}$.

C.5.2. Bemerkung : Es gilt ebenfalls

$res(F) \subset ideal(F) \sqcap R$.

und somit auch

$ideal(res(F)) \subset ideal(F) \sqcap R$

C.5.3. Satz : nach [Gröbner, II, p. 14]
res(F) = {0} ist eine notwendige und hinreichende Bedingung für die Existenz eines gemeinsamen Faktors aller Polynome aus F, oder das gleichzeitige Verschwinden aller Anfangskoeffizienten der Polynome aus F, deren (formaler) Grad genau n ist.

Beweis: Haben die Polynome aus F einen gemeinsamen Faktor, oder verschwinden in ihnen die höchsten formalen Anfangskoeffizienten, so

haben auch die beiden Polynome g und h einen gemeinsamen Faktor, bzw. es verschwinden die formalen Anfangskoeffizienten, die Resultante von beiden muß also verschwinden.
Verschwindet umgekehrt die Resultante von g und h, so haben die Polynome g und h einen gemeinsamen Faktor, oder die formalen Anfangskoeffizienten verschwinden.
Dieser gemeinsame Faktor kann nicht von den u's, bzw. von den v's abhängen, denn g hängt nicht von den v's und h hängt nicht von den u's ab, also müssen schon die Polynome aus F einen gemeinsamen Faktor haben.
Verschwinden die formalen Anfangskoeffizienten der Polynome g und h, so verschwinden die formalen Anfangskoeffizienten aller Polynome aus F, deren (formaler) Grad genau n ist. ▌

C.5.4. Bemerkung : Ist bekannt, daß der Anfangskoeffizient von

einem f aus F (etwa f_1) nicht verschwindet,

so kann man schon aus den Polynomen

$$g = f_1$$

$$h = v_2 f_2 + \ldots + v_m f_m.$$

das Kroneckersche Resultantensystem bilden, da nun der Ausnahmefall nicht vorkommen kann.

Der Satz verschärft sich, wenn alle Polynome aus F den maximalen Grad haben. Nach [Vd Waerden 1931] kann man das durch folgendes Verfahren erreichen: Aus jedem f aus F, das nicht den Grad n hat, bildet man die beiden Polynome:

$f' = f\, x^{n-deg(f)}$ und

$f'' = f\, (x - 1)^{n-deg(f)}$.

Die gemeinsamen Faktoren von f' und f'' sind jetzt genau die Faktoren von f. f' sowie f'' haben nun aber den Grad n. Für diese neue Menge F' gilt nun:

C.5.5. Satz : nach [Vd Waerden 1931]
res(F') = {0} ist eine notwendige und hinreichende Bedingung für die Existenz eines gemeinsamen Faktors aller Polynome aus F', oder das gleichzeitige Verschwinden aller Anfangskoeffizienten der Polynome aus F'.

Beweis: Da alle Polynome aus F' nun den gleichen (formalen) Grad haben, folgt die Behauptung aus dem vorhergehenden Satz. ▌

D. Die Dimension eines Ideals

Die Dimension eines Ideals gibt im wesentlichen an, wieviele Nullstellen ein gegebenes Ideal besitzt. Ist die Dimension gleich -1 so hat das Ideal keine Nullstellen. Im Falle der Dimension 0 besteht die Menge der Nullstellen des Ideals aus endlich vielen Punkten (diese endlich vielen Punkte sind natürlich isoliert). Wenn die Dimension gleich i ($1 \leq i \leq r$) (wobei r die Anzahl der Variablen des Polynomrings sei) ist, so enthält die Nullstellenmenge des Ideals mindestens Kurven (i=1), Flächen (i=2) und Hyperflächen (i=r-1).

W. Gröbner gibt in [Gröbner, II, p. 38] eine Definition, die zusammen mit dem Konzept der Gröbner Basen eine konstruktive Bestimmung der Dimension ermöglicht.

Zunächst die Definition

D.0.1. Definition : [Gröbner]

Für ein Ideal $\underline{A} \varepsilon K[x_1,\ldots,x_r]$ sei:

$\dim(\underline{A}) = \max\{ d :$ es gibt $i_1,\ldots,i_d$ mit

$$\underline{A} \sqcap K[x_{i(1)},\ldots,x_{i(d)}] = (0) \}.$$

Für $\underline{A} = (1)$ sei $\dim(\underline{A}) = -1$.

D.0.2. Bemerkung :

Sei $\underline{A} \subset K[x_1,\ldots,x_r]$,

$M = K[x_1,\ldots,x_r]/\underline{A}$.

Ein Polynom $f \neq 0$, das eine algebraische Abhängigkeit von Restklassen dokumentiert

$f([x_{i(1)}],\ldots,[x_{i(j)}]) = [0]$ in M,

liegt in dem Ideal $\underline{A}$:

$f(x_{i(1)},\ldots,x_{i(j)}) \varepsilon \underline{A} - \{0\}$.

D.0.3. Satz : [Gröbner, II, p. 40]

Die Dimension eines Primideals in $K[x_1,\ldots,x_r]$ ist

gleich dem Transzendenzgrad des Restklassenkörpers des Primideals.

Beweis: Sei M der Restklassenkörper und dim($\underline{A}$) = d, beispielsweise sei

$K[x_1,\ldots,x_d] \sqcap \underline{A} = (0)$

dann sind $[x_1]$, ..., $[x_d]$ algebraisch unabhängig in M,

denn einer algebraischen Abhängigkeit dieser Restklassen entspräche ein nicht verschwindendes Polynom aus dem Ideal, also ist der Transzendenzgrad vom M mindestens gleich d.

Ist umgekehrt der Transzendenzgrad von M gleich d,

so sind $[x_{i(1)}]$, ..., $[x_{i(d+1)}]$ sicher

algebraisch abhängig in M, d.h. es gibt ein Polynom f, das diese algebraische abhängigkeit dokumentiert. Diesem entspricht wiederum $f \varepsilon \underline{A}$, also

$K[x_{i(1)},\ldots,x_{i(d+1)}] \sqcap \underline{A} \neq (0)$,

die Dimension des Ideals ist also höchstens gleich d. ▮

Z.B. in der Quantorenelimination ist es wichtig festzustellen, ob ein vorgelegtes Ideal die Dimension 0 oder -1 hat, d.h. ob das Ideal höchstens endlichviele Nullstellen besitzt. Eine wichtige Hilfe bietet der folgende Satz, wenn man die erforderlichen univariaten Polynome konstruieren kann. Doch zunächst ein

D.0.4. Hilfssatz : Ist die Menge der Nullstellen eines Ideals endlich, so sind alle Komponenten einer Nullstelle

$(a_1,\ldots,a_r) \varepsilon$ NST($\underline{A}$) algebraisch über K.

Beweis: Sei eine der Komponenten einer Nullstelle, ohne Einschränkung der Allgemeinheit,

etwa a_1 transzendent

und enthalte etwa die Variable x_1.

Ersetzt man nun in $(a_1,\ldots,a_r)$ x_1 überall

durch ein Körperelement $b \varepsilon K$, so ist dieses

$(a'_1,\ldots,a'_r)$ wieder in NST($\underline{A}$).

$a_1(x_1)$ hat nun als univariates Polynom höchstens

endlich viele Nullstellen, daher gibt es auch nur endlich viele

$c \varepsilon K$ mit $a_1(b) = a_1(c)$.

Damit erhält man für unendlich viele Körperelemente nach

Ersetzung in $(a_1,\ldots,a_r)$ unendlich viele Nullstellen

für das Ideal $\underline{A}$. Dies widerspricht unserer Annahme, alle Komponenten einer Nullstelle müssen somit doch algebraisch sein ▌

D.0.5. Satz : $\underline{A}$ hat genau dann höchstens endlich viele gemeinsame Nullstellen, falls es r einvariablige Polynome ($\neq 0$)

$p_1(x_1)$, $p_2(x_2)$,..., $p_r(x_r)$ ε $\underline{A}$ gibt.

Beweis: "<==="

Seien $p_1(x_1)$,..., $p_r(x_r)$ ε $\underline{A}$ gegeben,

dann gilt:

ideal(p_1,..., p_r) C $\underline{A}$ ==>

NST($\underline{A}$) C NST(ideal(p_1,..., p_r))

nach [C.2.2., p. 15]

ideal(p_1,..., p_r) hat aber höchstens

endlich viele gemeinsame Nullstellen, also hat auch $\underline{A}$ nur endlichviele Nullstellen.

"==>" Ist NST($\underline{A}$) = $\emptyset$, dann gibt es trivialerweise die erforderlichen Polynome, da dann $\underline{A}$ = (1) ist.

Sei $\emptyset \neq$ NST($\underline{A}$) = { $(a_1,\ldots,a_r) \varepsilon U^r$:

$f(a_1,\ldots,a_r) = 0$ für alle $f \varepsilon \underline{A}$ }

die endliche Menge der Nullstellen von $\underline{A}$.
Bezüglich der i-ten Koordinate ($1 \leq i \leq r$) seien die

$a_{i\,1},\ldots,a_{i\,k(i)}$ die verschiedenen Punkte.

Alle $a_{i\,j}$ ($1 \leq i \leq r$, $1 \leq j \leq k(i)$) sind

nach dem vorangegangenen Hilfssatz [D.0.4., p. 24] algebraisch über K.

Sei irr(a,K) das irreduzible Polynom aus K[x] das a als Nullstelle hat. Das Polynom

$$q_i = irr(a_{i\,1},K). \ldots .irr(a_{i\,k(i)},K)$$

wird Null für alle $a_{i\,1},\ldots,a_{i\,k(i)}$ und somit für

alle $(a_1,\ldots,a_r) \varepsilon$ NST($\underline{A}$).

Nach dem Hilbertschen Nullstellensatz gibt es dann einen Exponenten e (e eine natürliche Zahl), so daß

$$p_i = q_i^{e(i)} \varepsilon \underline{A}.$$

Falls also $\underline{A}$ endlich viele Nullstellen hat, gibt es demnach für i=1,...,r nicht verschwindende univariate Polynome. ▌

Zusammenfassend lassen sich nulldimensionale Ideale mit folgendem Satz charakterisieren:

D.0.6. Satz :

Für ein Ideal $\underline{A} \subset K[x_1,\ldots,x_r]$ sind äquivalent:

(1) die Dimension von $\underline{A}$ ist Null,
(2) für i=1,...,r gibt es Polynome

$$0 \neq p_i(x_i) \in \underline{A} \neq (1).$$

(3) $\underline{A}$ hat genau endlich viele Nullstellen d.h.:

die Anzahl der $(a_1, \ldots, a_r) \in U^r$ mit

$f(a_1,\ldots,a_r) = 0$ für alle $f \in \underline{A}$ ist endlich

und nicht Null,
(4) die Vektorraum Dimension von

$K[x_1,\ldots,x_r]/\underline{A}$ ist endlich und nicht Null.

Beweis: "(1) ==> (2)"
Sei Dimension von $\underline{A} = 0$ d.h.

$\underline{A} \sqcap K[x_i] \neq (0)$ für i=1,...,r,

demnach gibt es

$0 \neq p_i(x_i) \in (\underline{A} \sqcap K[x_i])$ für i=1,...,r,

da dim($\underline{A}$) $\neq$ -1 folgt $\underline{A} \neq (1)$.

"(1) <== (2)"
Da es für i=1,...,r Polynome

$0 \neq p_i(x_i) \in \underline{A}$ gibt, folgt

$\underline{A} \sqcap K[x_i] \neq (0)$ also ist die

Dimension von $\underline{A} \leq 0$, da $\underline{A} \neq (1)$ folgt dim($\underline{A}$) = 0.

"(2) <==> (3)"
da $\underline{A} \neq (1)$ und NST($\underline{A}$) $\neq \emptyset$ folgt die Behauptung aus Satz [D.0.5., p. 25]

"(2) ==> (4)"

Sei $0 \neq p_i(x_i) \in \underline{A}$ mit deg $p_i(x_i) = e_i > 0$

für i=1,...,r. Zunächst gilt für beliebiges $k \in N$ [C.1.1., p. 14]

$$x_i^k = f_i(x_i)\, p_i(x_i) + h_i(x_i)$$

mit $\deg h_i(x_i) < \deg p_i(x_i)$.

Im Restklassenring gilt entsprechend

$$[x_i^k] = [f_i(x_i)]\,[p_i(x_i)] + [h_i(x_i)]$$

$$= [h_i(x_i)]$$

$$= \Sigma_{j=1,\ldots,e(i)-1}\ a_j\,[x_i^j].$$

Der Restklassenring als Vektorraum wird von den Bildern aller PP des Polynomrings erzeugt:

$[x_1^{g(1)} \ldots x_r^{g(r)}]$ wobei $g_i \in N$ für $i=1,\ldots,r$.

Jedes dieser Erzeugenden ist aber linear von den

Produkten aller $[x_i^{k(i)}]$, wobei $k_i < e_i$ $i=1,\ldots,r$ ist,

abhängig:

$$[x_1^{g(1)} \ldots x_r^{g(r)}] =$$

$$= [x_1^{g(1)}]. \ldots .[x_r^{g(r)}]$$

$$= [h_1(x_1)]. \ldots .[h_r(x_r)]$$

$$= \Sigma_{k=1,\ldots,l}\ b_k\,[x_1^{j(k(1))} \ldots x_r^{j(k(r))}]$$

mit $l = (e_1 - 1) \ldots (e_r - 1) + 1$.

Damit ist gezeigt, daß die Vektorraumdimension endlich ist, falls es r einvariablige Polynome in $\underline{A}$ gibt. Da $\underline{A} \neq (1)$ ist $[1] \neq [0]$, und somit ist [1] linear unabhängig, also ist die Vektorraum Dimension ≥ 1.

"(4) ==> (2)"
Ist die Vektorraumdimension des Restklassenrings endlich, etwa gleich d, so sind für i=1,...,r die d+1 Elemente

$$[1],\ [x_i],\ \ldots,\ [x_i^{\,d}]$$

linear abhängig, d.h es gilt eine Relation

$$a_{i\,0}\,[1] + a_{i\,1}\,[x_i] + \ldots + a_{i\,d}\,[x_i^{\,d}] = [0]$$

daraus folgt, daß die Polynome

$$p(x_i) = a_{i\,0} + a_{i\,1}\,x_i + \ldots + a_{i\,d}\,x_i^{\,d}$$

im Ideal $\underline{A}$ liegen. Da $d \neq 0$ ist $\underline{A} \neq (1)$. ▌

E. Die Dimension und Resultantensysteme

Resultanten bieten nach [C.4.1., p. 18] ein Hilfsmittel Polynome zukonstruieren, die einerseits in dem betrachteten Ideal liegen und andererseits von einer Variablen weniger abhängen. Damit können sozusagen Variablen eliminiert werden bis die univariaten Polynome gefunden sind (falls es sie gibt).

Zunächst werden 'einfache' Resultantensysteme und dann Kroneckersche Resultantensysteme betrachtet. Anschließend werden mit Hilfe von Resultantensystemen Testverfahren beschrieben, mit denen festgestellt werden kann ob ein vorgelegtes Polynomideal eine Dimension ≤ 0 hat.

E.1. Iterierte Resultantensysteme

Zunächst werden nur Systeme von Resultanten nach [C.4.1., p. 18] betrachtet.

Dort sei nun $R = K[x_1,\ldots,x_{r-1}]$ gesetzt.

Sei $(0) \neq \underline{A} = (A_1,\ldots, A_n) \subset K[x_1,\ldots,x_r]$ $r \geq 0$.

E.1.1. Definition :

Sei $A = \{ A_1,\ldots, A_n \} \subset K[x_1,\ldots,x_r]$.

Dann heißt

$$S(x_{i(1)},\ldots,x_{i(k)}, A) = \{ \text{res}(R,R') \text{ bezüglich } x_{i(1)} : R, R' \in S(x_{i(2)},\ldots,x_{i(k)}, A) \}$$

für $1 \leq k \leq r$ und mit $S(A) = A$ für $k=0$

ein iteriertes Resultantensystem von A bezüglich der

Variablenreihenfolge $x_{i(1)},\ldots,x_{i(k)} \in \{x_1,\ldots,x_r\}$.

In den Anwendungen werden wir später auch Teilmengen von iterierten Resultantensystemen betrachten. Von den $n(n-1)/2$ Resultanten, die aus einer Menge von n Polynomen gebildet werden können, werden dann einige ausgelassen.

E.1.2. Bemerkung : Es ist

$$S(x_{i(1)},\ldots,x_{i(k)}, A) \subset K[\{x_{j(1)},\ldots,x_{j(l)}\}]$$

wobei

$$\{x_{j(1)},\ldots,x_{j(l)}\} =$$

$$= \{x_1,\ldots,x_r\} - \{x_{i(1)},\ldots,x_{i(k)}\}$$

und

$$S(x_{i(1)},\ldots,x_{i(k)}, A) \subset \text{ideal}(A) = \underline{A}$$

da $\text{res}(R,R') \in \underline{A}$ für $R,R' \in \underline{A}$.
Insgesammt ist also:

$$S(x_{i(1)},\ldots,x_{i(k)}, A)$$

$$\subset [\text{ideal}(A) \cap K[\{x_{j(1)},\ldots,x_{j(l)}\}]]$$

mit $\{x_{j(1)},\ldots,x_{j(l)}\}$ wie oben. ▮

E.2. Kroneckersche iterierte Resultantensysteme

Analog zu den 'einfachen' Resultantensystemen sollen nun Kroneckersche iterierte Resultantensysteme betrachtet werden.

Sei $(0) \neq \underline{A} = (A_1, \ldots, A_n) \subset K[x_1, \ldots, x_r]$ $r \geq 0$.

E.2.1. Definition :

Sei $A = \{ A_1, \ldots, A_l \} \subset K[x_1, \ldots, x_r]$.

Dann heißt

$$Sk(x_{i(1)}, \ldots, x_{i(k)}, A) = res(R) \text{ mit}$$

$$R = Sk(x_{i(2)}, \ldots, x_{i(k)}, A)$$

für $1 \leq k \leq r$ und mit $Sk(A) = A$ für $k=0$

ein Kroneckersches iteriertes Resultantensystem von A bezüglich der

Variablenreihenfolge $x_{i(1)}, \ldots, x_{i(k)} \in \{x_1, \ldots, x_r\}$.

E.2.2. Bemerkung: Es ist

$$Sk(x_{i(1)}, \ldots, x_{i(k)}, A) \subset K[\{x_{j(1)}, \ldots, x_{j(l)}\}]$$

wobei

$$\{x_{j(1)}, \ldots, x_{j(l)}\} =$$

$$= \{x_1, \ldots, x_r\} - \{x_{i(1)}, \ldots, x_{i(k)}\}$$

und

$$Sk(x_{i(1)}, \ldots, x_{i(k)}, A) \subset ideal(A) = \underline{A}$$

da $res(A) \subset \underline{A}$.
Insgesammt ist also:

$$Sk(x_{i(1)}, \ldots, x_{i(k)}, A)$$

$$\subset [ideal(A) \sqcap K[\{x_{j(1)}, \ldots, x_{j(l)}\}]]$$

mit $\{x_{j(1)}, \ldots, x_{j(l)}\}$ wie oben. ∎

E.3. Test auf Dimension ≤ 0

Im folgenden wird für iterierte Resultantensysteme ein Testverfahren beschrieben, mit dem festgestellt werden kann, ob die Dimension ≤ 0 ist. Anstelle der 'einfachen' Resultantensysteme können auch Kroneckersche Resultantensysteme benutzt werden, die Algorithmen sind aber nur auf die ersteren zugeschnitten.

E.3.1. Satz : Falls die r iterierten Resultantensysteme bezüglich der zyklischen Permutation der Variablenreihenfolge von r-1 Variablen von A nicht nur aus der Null bestehen, so hat ideal(A) höchstens endlichviele Nullstellen.

Beweis: Nach Voraussetzung gibt es die von Null verschiedenen einvariabligen Polynome:

$p_1(x_1) \in S(x_2,\ldots,x_r, A)$

$p_2(x_2) \in S(x_3,\ldots,x_r,x_1, A)$

...

$p_r(x_r) \in S(x_1,\ldots,x_{r-1}, A).$

Nach der Bemerkung [E.1.2., p. 30] gilt dann:

$p_1(x_1),\ldots, p_r(x_r) \in \underline{A},$

die Behauptung folgt dann aus Satz [D.0.5., p. 25] ▮

E.3.2. Bemerkung: Die Umkehrung des Satzes muß im allgemeinen nicht richtig sein. Das heißt, falls für ein (oder mehrere)

$S(x_{i(1)},\ldots,x_{i(r-1)}, A) = \{ 0 \}$ gilt,

so folgt nicht, daß ideal(A) unendlich viele gemeinsame Nullstellen hat.

Beispiel:

$A = \{ f_1, f_2, f_3 \}$

$= \{ x\,y\,z + 1,\ x\,y\,z,\ x\,y \} \subset K[x,y,z].$

S sei das Resultantensystem, das alle n(n-1)/2 mögliche Resultanten enthält:

$S(., A) = \{ res(f_1,f_2),\ res(f_2,f_3),\ res(f_1,f_3) \}$

Es ist

$S(z, A) = \{ -x\,y,\ x\,y,\ x\,y \}$

$S(y,z, A) = \{ 0,\ 0,\ 0 \}$

$$S(y, A) = \{ -xz, 0, -x \}$$
$$S(x,y, A) = \{ 0, 0, 0 \}$$

$$S(x, A) = \{ -yz, 0, -y \}$$
$$S(z,x, A) = \{ 0, 0, -y \}$$

weitere Möglichkeiten für Resultantensysteme sind

$$S(x,z, A) = \{ 0, 0, 0 \}$$
$$S(z,y, A) = \{ 0, 0, -x \}$$
$$S(y,x, A) = \{ 0, 0, 0 \}$$

aber $\underline{A}$ = ideal(A) hat überhaupt keine gemeinsame Nullstellen da $1 \varepsilon \underline{A}$:

$$1 = (xyz + 1) - (xyz).$$

Denn aus res(A,B) = 0 folgt nur, daß A und B einen gemeinsamen Faktor haben, oder in ihnen beide Anfangskoeffizienten verschwinden.

Variante von Satz [E.3.1., p. 33] nach P. Schmidt.

E.3.3. Satz : Wenn zu jedem j, $1 \leq j \leq r$ eine Anordnung

$i_1 < \ldots < i_{r-1}$ von $\{1,\ldots,j-1,j+1,\ldots,r\}$

existiert, so daß

$S(x_{i(1)},\ldots,x_{i(r-1)}, A) \neq \{0\}$ gilt,

dann folgt dim ideal(A) ≤ 0.

Beweis: Nach Voraussetzung gibt es zu jedem j, $1 \leq j \leq r$ die von Null verschiedenen einvariabligen Polynome:

$p_1(x_j) \varepsilon S(x_{i(1)},\ldots,x_{i(r-1)}, A)$

Nach der Bemerkung [E.1.2., p. 30] gilt dann:

$p_1(x_1),\ldots, p_r(x_r) \varepsilon \underline{A}$,

die Behauptung folgt dann aus Satz [D.0.5., p. 25] ▮

Das obige Beispiel zeigt, daß selbst die Umkehrung dieses Satzes falsch ist.

E.4. Test auf Dimension ≤ 0 bei zusätzlicher Faktorisierung

In praktisch vorkommenden Fällen sind die Polynome oft faktorisierbar, wobei die einzelnen Faktoren zusätzlich Exponenten größer 1 haben können. Damit verringert sich der Aufwand für die nachfolgende Berechnung der Resultanen beträchtlich. Bei den folgenden Betrachtungen ist es nicht wesentlich, daß die Faktoren irreduzibel sind, man könnte ebensogut nur quadratfreie Faktoren verwenden.

Wenn die Polynome in $A = \{A_1,\ldots,A_n\}$ die Zerlegung

$$A_i = A_{i\,1}^{e(i\,1)} \cdot \ldots \cdot A_{i\,j(i)}^{e(i\,j(i))} \qquad i=1,\ldots,n$$

$$e_{i\,l} > 0 \quad \text{für } l=1,\ldots,j_i, \quad \text{für } i=1,\ldots,n.$$

besitzen, so sei

$$A'_i = A_{i\,1} \cdot \ldots \cdot A_{i\,j(i)} \qquad i=1,\ldots,n$$

und $A' = \{A'_1,\ldots,A'_n\}$.

E.4.1. Hilfssatz : NST(A) = NST(A').

Beweis: Alle Polynome aus A' liegen in dem Radikal von ideal(A) (nach [C.1.6., p. 15]), ebenso liegen alle Polynome aus A in dem Radikal von Ideal(A'). Somit gilt rad(ideal(A)) = rad(ideal(A')), die Behauptung folgt dann aus [C.2.6., p. 16] .

E.4.2. Folgerung : NST(A) endlich <==> NST(A') endlich.

Es sei

$$T = \{\ \{A_{1\,k(1)},\ldots,A_{n\,k(n)}\} : k_i=1,\ldots,j_i,\ i=1,\ldots,n\ \}$$

die Menge der n-Tupel von Faktoren der Polynome in A.

E.4.3. Hilfssatz : Falls NST(B) für alle $B \in T$ und für ein $i \in \{1,\ldots,r\}$ nur endlichviele

x_i-Koordinaten besitzt, so hat NST(A) nur

endlichviele x_i-Koordinaten.

Beweis: Wegen [C.2.2., p. 15] gilt

$$NST(A') = \bigsqcup_{B \in T} NST(B),$$

insbesondere gilt das dann für die einzelnen Koordinaten.

E.4.4. Bemerkung : Nach [E.1.2., p. 30] besitzt NST(B)

höchstens endlichviele x_i-Koordinaten, falls für $i \neq j$

$$NST(S(x_j, B))$$

nur endlichviele x_i-Koordinaten besitzt.

E.5. Abschwächung des Tests auf Dimension ≤ 0

Eine Möglichkeit, den Test auf Dimension ≤ 0 eines Polynomideals, weniger aufwendig zu gestalten, ergibt sich durch weglassen von Resultanten bei der Bestimmung des Resultantensystems. Denn in dem Satz [E.3.1., p. 33] ist nur gefordert, daß die Resultantensysteme nicht nur aus der Null bestehen, es müssen daher nicht alle möglichen Resultanten berechnet werden.

Eine weitere Möglichkeit, den Aufwand zu verringern, besteht nach [Böge 1984] darin, von vornherein auf die Berechnung des iterierten Resultantensystems zuverzichten, falls weniger Polynome vorhanden sind, als der Polynomring Variablen hat. Denn man beachte, daß ein iteriertes Resultantensystem 'in der Regel' nur aus der Null besteht, falls weniger Polynome in der Ausgangsmenge enthalten sind, als der Polynomring Variable hat.

Ein Nachteil besteht aber darin, daß dann weitere Fälle, wo die Polynome nur endlichviele gemeinsame Nullstellen haben nicht mehr entdeckt werden.

Beispiel

$A = \{ x - 1, x - 3 \}$ in $K[x,y,z]$

da A nur 2 Elemente enthält würde das iterierte Resultantensystem nicht berechnet, ob wohl die Polynome keine gemeinsame Nullstelle haben, und somit das iterierte Resultantensystem nicht aus der Null besteht.

F. Auf Resultanten aufbauende Algorithmen

Entsprechend dem Satz [E.3.1., p. 33] werden von dem Algorthmus PFZT alle zyklischen Permutaionen der Variablenreihenfolge erzeugt. Der Algorithmus PFZT1 berechnet dann für eine feste Variablenreihenfolge ein iteriertes Resultantensystem.

F.0.1. Algorithmus PFZT: nach [Collins & Loos 1981]

t ← PFZT(r,A)

[Polynomial finitely many common zeros test.

$A = (A_1,\ldots,A_n)$, $n \geq 1$, is a list of

nonzero polynomials over $K[x_1,\ldots,x_r]$ in r variables, $r \geq 1$.

t = 1 if it has been determined that
the polynomials in A have only finitely many common zeros.
t = 0 if either it has not been determined how many
common zeros the polynomials in A have, or if it has been
determined that they have infinitely many common zeros.]

(1) [Initialise.] t ← 1. i ← 1.

$P \leftarrow \{x_1,\ldots,x_r\}$.

(2) [Test for finitely many distinct x_i - coordinates

among the common zeros.]
A' ← A. A'' ← ().
repeat ADV(A'.A1,A').
if i > 1 then A1 ← A1 ε K[P].
if A1 ε K then go to 4.

if A1 ε $K[x_i]$ then go to 3.

A'' ← COMP(A1,A'')
until A' = (). A'' ← INV(A'').
b ← PFZT1(r,A''). if b = 0 then (t ← 0. go to 4).

(3) [Check for completion.] i ← i + 1.
if i ≤ r then

($P \leftarrow \{x_i,\ldots,x_r,x_1,\ldots,x_{i-1}\}$

[Cyclic permutation of the variables.]
go to 2).

(4) [finish.] return ▌

In dem Algorithmus PFZT1 wird im Step 2 die Menge T, der Tupel aller Faktoren, nach [E.4., p. 35] bestimmt. In Step 3 wird dann für ein Tupel von Faktoren von Eingabepolynomen ein Resultantensystem berechnet. Mit diesem Resultantensystem als Eingabe ruft sich der Algorithmus rekursiv auf, bis nur noch Polynome in einer Variablen übrigbleiben.

Soll auf die Berechnung der Faktoren der Polynome verzichtet werden, so ist (mit entsprechenden Modifikationen) nur Step 2 wegzulassen.

Soll das Resultantensystem nach [E.5., p. 36] berechnet werden, so ist am Ende von Step 1 eine Abfrage

if n < r then (t ← 0. go to 5).

einzufügen.

F.0.2. Algorithmus PFZT1: nach [Collins & Loos 1981]

t ← PFZT1(r,A)

[Polynomial finitely many common zeros test, one variable.

$A = (A_1,\ldots,A_n)$, $n \geq 0$, with

$A_i = A_i(x_1,\ldots,x_r)$ for i=1,...,n is a

list of nonzero polynomials in r variables, $r \geq 1$.
t = 1 if it has been determined that the common zeros of

$A_1,\ldots,A_n$ have only finitely many distinct x_1 coordinates.

t = 0 if either A = (), or if it has not been determined

how many distinct x_1 coordinates the common zeros of

$A_1,\ldots,A_n$ have, or if it has been determined that

they have infinitely many distinct x_1 coordinates.]

```
(1)  [A = () or r = 1.] t ← 1.
     if A = () then ( t ← 0. go to 5 ).
     if r = 1 then ( t ← 1. go to 5 ).
```

(2) [Factor each A_i.] r' ← r - 1.

T' ← (). A' ← A.
repeat ADV(A'.A1,A').

$A1 = a_1^{e(1)} \cdot \ldots \cdot a_l^{e(l)}$.
$L = \{ a_1, \ldots, a_l \}$

T' ← COMP(L,T')
until A' = ().

$T \leftarrow \{ (b_1,\ldots,b_n) : b_i \in L_i, S' = (L_1,\ldots,L_n) \}$

(3) [Test tuples of factors.]
C ← (). P ← ().
I ← next tupel of T. T ← T - I.
repeat ADV(I.B1,I).
if deg(B1) = 0 then C ← COMP(ldcf(B1),C)
else P ← COMP(B1,P)
until I = ().
R ← (). if P ≠ () then ADV(P.P1,P).
while P ≠ () do (ADV(P.P2,P).
R1 ← PRES(r,P1,P2). [Resultant of P1 and P2]
if R1 ∈ K-{0} then go to 4.

if R1 ∈ $K[x_1]$ then go to 4.

if R1 ≠ 0 then R ← COMP(R1,R)).
L ← CONC(C,R).
b ← PFZT1(r',L). if b = 0 then (t ← 0. go to 5)).

(4) [Test for completition.] if T ≠ ∅ then go to 3.
(5) [Finish.] return ▌

G. Die Dimension und Gröbner-Basen

Im folgenden werden zunächst die Grundgedanken zu den Gröbner-Basen ausgeführt. Anschießend werden mit Hilfe der Eigenschaften von Gröbner-Basen Verfahren zur Bestimmung der Dimension entwickelt.

G.1. Reduktion von Polynomen

Für das folgende sei wieder eine zulässige Ordnung

$<_T$ für die PP fixiert.

[Buchberger 1976a] folgend wird nun die Reduktion on Polynomen behandelt. Zunächst für ein einzelnes Polynom:

Sei $f, g, h \in K[x_1,\ldots,x_r]$

G.1.1. Definition : g heißt reduzibel bezüglich f, falls es einen Term in g gibt, der Vielfaches des lp(f) ist. Andernfalls heißt g irreduzibel bezüglich f.

(Nicht zuverwechseln mit irreduzibel bezüglich der Zerlegung von g in irreduzible Faktoren)

G.1.2. Definition : Falls g reduzibel bezüglich f ist, also

$lp(f) <_M x^{(i)}$ für einen Term $a_{(i)}x^{(i)}$ in g,

so heißt ein Polynom

$$g - b\, x^{(j)}\, f \quad \text{mit } b \in K$$

reduzierte Form von g bezüglich f,

wobei $a_{(i)}x^{(i)} = b\, x^{(j)}\, lp(f)$.

Dieses Polynom enthält den reduzierten Term nicht mehr. Im allgemeinen kann es mehrere Terme in g geben die Vielfaches des lp(f) sind.

Ist $a_{(i)}x^{(i)}$ der bezüglich $<_T$ größte
Term in g mit $a_{(i)}x^{(i)} = b\, x^{(j)}\, lp(f)$, so wird die

reduzierte Form von g bezüglich f mit RF(g,f) bezeichnet, mit der Vereinbarung RF(g,f) = g falls g irreduzibel bezüglich f ist.

G.1.3. Bemerkung : Durch diese Festlegung ist RF(g,f) eindeutig bestimmt, der größte Term in RF(g,f), der Vielfaches von lp(f) ist, ist jetzt

kleiner als der

entsprechende $a_{(i)}\, x^{(i)}$ von g.

Betrachtet man die Folge der Abbildungen, die durch

$RF(.,f):K[x_1,\ldots,x_r] \longrightarrow K[x_1,\ldots,x_r]$

$h_i \longrightarrow RF(h_i,f) = h_{i+1}$

$h_{i+1} = h_i - a_i\, u_i\, f$ mit $h_0 = g$

für i=0,1,2,... definiert sind, dann gilt der

G.1.4. Satz : Es gibt ein $n \varepsilon N$, so daß in der zuvor definierten Folge der Polynome

$h_0, h_1, \ldots, h_n, \ldots$

$h_n = h_{n+1}$ gilt.

Dieses Polynom ist eindeutig bestimmt (bis auf einen Faktor aus K), und wird mit NF(g,f) bezeichnet, d.h.

$NF(g,f) = g - \Sigma_{i=1,\ldots,n}\, a_i\, u_i\, f = g - h\, f.$

Beweis: [Buchberger 1976a, p. 22] ▌

G.1.5. Bemerkung : Ker NF(.,f) = ideal({f})

Beweis: Es gilt
NF(g,f) = g - h f = 0 <==>
g = h f <==>
g ε ideal({f}). ▌

Die Reduktion von Polynomen bezüglich einer Menge von Polynomen

$F = \{f_1,\ldots,f_m\} \subset K[x_1,\ldots,x_r]$

bringt nun aber einige Schwierigkeiten mit sich:

G.1.6. Definition : g heißt reduzibel bezüglich F, falls es ein

i (1≤i≤m) gibt, so daß g reduzibel bezüglich f_i ist.

g heißt irreduzibel bezüglich F, falls g für alle i

(1≤i≤m) irreduzibel bezüglich f_i ist.

Es sei $h = a_{(i)} x^{(i)}$ der bezüglich der Termordnung

größte Term in g, der Vielfaches eines lp(f) für ein $f \varepsilon F$ ist. Im

allgemeinen kann h Vielfaches mehrerer lp(f) für verschiedene f ε F sein. Wählt man aus all diesen Polynomen aus F, die zur Reduktion verwendet werden können, eins aus, so erhält man ein ausgezeichnetes Polynom RF(g,F) = RF(g,f).

Legt man zusätzlich die Auswahl auf ein Schema fest, so kann man wie oben auch hier eine Folge von Abbildungen und dadurch ein Polynom NF(g,F) definieren, und es gilt wieder der

G.1.7. Satz : Es gibt ein n ε N, so daß

in der Folge $h_{i+1} = RF(h_i,F)$ i=0,1,2,... gilt:

$h_n = h_{n+1}$, d.h. h_n ist irreduzibel bezüglich F,

und wird NF(g,F) genannt:

$$NF(g,F) = g - \Sigma_{j=1,\ldots,k} a_j u_j f_{i(j)}$$

$$\text{mit } i_j \varepsilon \{1,\ldots,m\}$$

$$= g - \Sigma_{i=1,\ldots,m} h_i f_i$$

Beweis: [Buchberger 1976a, p. 22] ▮

Bei einem anderen Auswahlschema würde man möglicherweise zu verschiedenen reduzierten Formen kommen. So könnte etwa Anstelle des Polynoms mit dem kleinsten Index das Polynom mit dem größten Index, oder auch das Polynom mit dem bezüglich der Termordnung größten lp ausgewählt werden.
Mit anderen Worten NF(g,F) = NF(g,F,Auswahl). Es gibt demnach viele Abbildungen entsprechend [G.1.7., p. 42] , eine solche Abbildung sei mit NF(.,F) bezeichnet.

Die Reduktion von f auf eine bezüglich F irreduzible Form ist also für beliebiges F eine mengenwertige Abbildung, d.h. eine Transformation (vergl. [Samuel & Zariski]). Diese Transformation sei mit NT(.,F) bezeichnet. NT(f,F) ist also die Menge aller NF(f,F) für verschiedene 'Reduktionswege', d.h. für verschiedene NF(.,F).

Das zweite Problem der Abbildungen NF(.,F) ist, daß

$$\text{Ker NF}(.,F) \neq \text{ideal}(F)$$

für ein festgewähltes NF(.,F).

Beispiel

$$F = \{f_1,f_2\} = \{ x - y,\ x^2 - z \} \text{ mit } x >_L y >_L z$$

und falls eine Auswahl von Polynomen zur Reduktion möglich ist, so

wähle dasjenige aus, das den kleinsten Index hat.

Es ist nun zwar

$f_2 \in \text{ideal}(F)$ sogar $f_2 \in F$ aber

$$NF(f_2,F) = (f_2 - x f_1) - y f_1$$
$$= x^2 - z - x^2 + x y - x y + y^2$$
$$= y^2 - z \neq 0.$$

Und es ist auch z.B.

$f_2 - x f_1 = x y - z = f \in \text{ideal}(F)$ aber

$$NF(f,F) = x y - z - x y + y^2$$
$$= y^2 - z \neq 0$$

G.2. Gröbner-Basen

Die zwei Probleme:
die Definition von NF(.,F) ist von dem Auswahlschema abhängig,
für festes Auswahlschema, d.h. für festes NF(.,F) ist
Ker NF(.,F) ≠ ideal(F)
wurden von Buchberger 1968 durch das Konzept der Gröbner-Basen gelöst [Buchberger 1970] . Es wurde von ihm ein Algorithmus angegeben, der für ein festes Auswahlschema bei der Reduktion die Menge F so in eine Menge G umformt, daß gilt:

(1) ideal(F) = ideal(G)
und
(2) das Bild von NT(f,G) besteht für alle Polynome f
nur aus einem Element.

G.2.1. Definition : Eine Menge von Polynomen G mit diesen Eigenschaften heißt nach Buchberger Gröbner-Basis.

Eine Gröbner-Basis von F sei mit GB(F) bezeichnet,

und GB(A) stehe für $GB(\{A_1,\ldots,a_n\})$.

(2) entspricht der Aussage G3 in dem Theorem 3.3. in [Buchberger 1976a]

Falls die Anzahl der Bildelemente von NT(f,G) für alle Polynome f gleich 1 ist, so ist damit nicht gesagt, daß es nur einen einzigen 'Reduktionsweg' gibt, d.h. NF(.,G) ist damit noch nicht eindeutig bestimmt, wohl aber ist

damit NF(f,G) eindeutig bestimmt.

Das zweite Problem wird durch folgenden Satz gelöst:

G.2.2. Satz : G ist genau dann eine Gröbner Basis, wenn für alle NF(.,G) gilt:
$f \in ideal(G) \iff NF(f,G) = 0.$
Das entspricht der Aussage G6 in Satz 3.7. in [Buchberger 1976a]

Daraus folgt, daß jedes

$$NF(.,G):K[x_1,\ldots,x_r] \longrightarrow K[x_1,\ldots,x_r]$$
$$f \longrightarrow NF(f,G)$$

eine Abbildung mit Kern
$Ker(NF(.,G)) = ideal(G) = ideal(F)$ ist.

Eine für die Berechnung von G geeignete Charakterisierung von Gröbner-Basen gibt der folgende

G.2.3. Satz : G ist eine Gröbner-Basis genau dann, wenn für alle $f, g \in G$ und ein festes NF(.,G) gilt:
$NF(SP(f,g), G) = 0.$
Das entspricht der Aussage G2 in Theorem 3.3. in [Buchberger 1976a]

wobei SP(f,g) folgendermaßen definiert ist:

G.2.4. Definition S-Polynom: Sei $u = lcm(lp(f),lp(g))/lp(g)$ und $v = lcm(lp(f),lp(g))/lp(f)$ [lcm = least common multiple], dann ist

$$SP(f,g) := lbcf(g)\ v\ f - lbcf(f)\ u\ g.$$

Damit ist nun ein konstruktiver Weg gefunden, die Menge F in eine Menge G, mit der Eigenschaft, daß G Gröbner-Basis ist, umzuformen. Für den Algorithmus siehe [H.1.1.] .

G.2.5. Definition : F heißt reduzibel, falls es ein $f \in F$ gibt, das reduzibel bezüglich $F - \{f\}$ ist.

F heißt irreduzibel, falls alle $f \in F$ irreduzibel bezüglich $F - \{f\}$ sind.

Damit ergibt sich eine Eindeutigkeit für Gröbner Basen:

G.2.6. Satz : [Schrader 1976, Ergänzung zur Diplomarbeit] [Buchberger 1976b, Theorem 1.6.]
Es seien G und H Gröbner Basen zu F bzw. F' mit
$ideal(F) = ideal(F')$. Sind G und H jeweils irreduzibel, so ist $G = H$ (bis auf Faktoren aus K).

Alle obigen Betrachtungen gelten für eine feste Termordnung, für verschiedene Termordnungen sind im Allgemeinen die Gröbner-Basen verschieden.

In den folgenden Abschnitten sollen die Verfahren beschrieben werden, mit denen die Dimension eines Polynomideals in den Fällen -1, 0, ≥ 1 bestimmt werden kann.

G.3. Bestimmung der Dimension -1

Zunächst soll geklärt werden, ob ein Polynomideal überhaupt Nullstellen besitzt.

Es sei $(0) \neq \underline{A} \subset K[x_1,\ldots,x_r]$ $r \geq 0$ ein Polynomideal.

G.3.1. Satz : [Buchberger 1984]
Ein Polynomideal hat genau dann keine Nullstelle, wenn 1 in der Gröbner Basis liegt.

Beweis: Ein Ideal $\underline{A}$ hat genau dann keine Nullstelle, falls $1 \in \underline{A}$. Sei G die Gröbner Basis von $\underline{A}$, dann ist ideal(G) = $\underline{A}$ und

$$\begin{aligned} & 1 \in \underline{A} \\ \Leftrightarrow \quad & 1 \in \text{ideal}(G) \\ \Leftrightarrow \quad & NF(1,G) = 0 \\ \Leftrightarrow \quad & 1 \in G. \end{aligned}$$ ∎

Dieser Satz gilt für eine beliebige (aber fest gewählte) zulässige Termordnung.

G.4. Bestimmung der Dimension 0

Ob ein Polynomideal endlich viele Nullstellen besitzt, wird durch folgende Überlegungen beantwortet. Die Aussagen dieses Abschnitts gelten für eine beliebige (aber fest gewählte) zulässige Termordnung.

G.4.1. Lemma: [Buchberger 1984, Lemma 6.7.]

Sei $\underline{A} \subset K[x_1,\ldots,x_r]$ ein Ideal. G sei eine

Gröbner Basis von $\underline{A}$.
B := { [u] : u ein Potenzprodukt das kein Vielfaches eines größten Potenzprodukts eines Polynoms aus G ist, $[u] = u + \underline{A}$ }
Dann ist B eine linear unabhängige Vektorraumbasis von

$K[x_1,\ldots,x_r]/_{\underline{A}}$.

Beweis: Angenommen es gäbe eine lineare Abhängigkeit:

$$c_1 [u_1] + c_2 [u_2] + \ldots + c_l [u_l] = 0$$

für einige $[u_i] \in B$ und nicht alle $c_i = 0$.

dann ist

$$f = c_1 u_1 + c_2 u_2 + \ldots + c_l u_l \in \underline{A}$$

und NF(f,G) = 0 nach der Eigenschaft der Gröbner Basis.
f ist aber schon in Normalform bezüglich G

da keines der u_i ($1 \leq i \leq l$) bezüglich G

reduziert werden kann, d. h. NF(f,G) = f,
also ist f = 0.

$$\Longrightarrow c_1 = c_2 = \ldots = c_l = 0$$

im Widerspruch zur Annahme. B ist auch ein Erzeugendensystem für den Vektorraum, denn [f] = [NF(f,G)] und NF(f,G) ist eine Linearkombination von u's, mit $[u] \in B$. ▌

G.4.2. Satz : [Buchberger 1984, Methode 6.9.]
$\underline{A}$ ($\neq (1)$) hat genau dann endlich viele Nullstellen, falls in der Gröbner Basis G von $\underline{A}$ für alle i ($1 \leq i \leq r$) ein

Potenzprodukt der Form $x_i^{k(i)}$ mit $k(i) > 0$

unter den größten Potenzprodukten der Polynome aus F enthalten ist.

Beweis: Mit der Äquivalenz:
$\underline{A}$ hat endlich viele Nullstellen <==>
die Vektorraum Dimension von

$K[x_1,\ldots,x_r]/_{\underline{A}}$ ist endlich

(nach Satz [D.0.6., p. 27]), folgt die Behauptung aus dem Lemma [G.4.1., p. 46] , denn die

$[1],\ [x_1],\ [x_1^{\,2}],\ \ldots,\ [x_1^{\,k(1)-1}],$

$[x_2],\ [x_2^{\,2}],\ \ldots,\ [x_2^{\,k(2)-1}],$

$\ldots$

$[x_r],\ [x_r^{\,2}],\ \ldots,\ [x_r^{\,k(r)-1}]$

und ihre Produkte bilden ein Erzeugendensystem für den

Vektorraum $K[x_1,\ldots,x_r]/_{\underline{A}}$,

dessen Dimension damit endlich ist.
Falls umgekehrt für ein i, etwa i=j, kein größtes

Potenzprodukt der Form $x_j^{\,k(j)}$ in der Gröbner Basis

enthalten ist, so ist die unendliche Menge

$\{\ [1],\ [x_j],\ [x_j^{\,2}],\ [x_j^{\,3}],\ldots\ \}$

in $K[x_1,\ldots,x_r]/_{\underline{A}}$ linear unabhängig,

die Vektorraum Dimension also unendlich. ▌

G.5. Bestimmung der Dimension ≥ 1

Die übrigen Fälle, in denen die Dimension ≥ 1 ist, sind nun nicht mehr sofort aus einer Gröbner-Basis ablesbar. Es ist nun im Allgemeinen notwendig mehrere Gröbner-Basen bezüglich verschiedener Variablenreihenfolgen zu berechnen. Zudem kann jetzt nur noch die lexikographische Termordnung verwendet werden.

G.5.1. Lemma: [Trinks 1978] nach [Buchberger 1984, Methode 6.10.]
Sei F eine Gröbner Basis bezüglich der rein lexikographischen Termordnung der Polynome.
Ohne Einschränkung der Allgemeinheit

gelte $x_1 <_L x_2 <_L \cdots <_L x_r$.

Dann gilt für i=1,...,r:

$$\text{ideal}(F) \sqcap K[x_1,\ldots,x_i] = \text{ideal}(F \sqcap K[x_1,\ldots,x_i])$$

wobei das letztere als Ideal in $K[x_1,\ldots,x_i]$

zu betrachten ist.

Beweis:

Sei $f \in \text{ideal}(F) \sqcap K[x_1,\ldots,x_i]$

das heißt insbesondere $f \in \text{ideal}(F)$ und somit ist f reduzierbar bezüglich F. Also

$$f = \Sigma\, b_j\, u_j\, f_j \quad \text{in } K[x_1,\ldots,x_r]$$

mit $f_j \in F$, u_j Potenzprodukte, $b_j \in K$

und $lp(u_j f_j) \leq_L lp(f)$.

Wegen $f \in K[x_1,\ldots,x_i]$ folgt

$lp(f_j) \in K[x_1,\ldots,x_i]$, denn ein PP in f ist

Vielfaches eines $lp(f_k)$. Wegen der lexikographischen

Termordnung ist damit aber ganz $f_k \in K[x_1,\ldots,x_i]$.

Zusammen gilt also

$$f = \Sigma\, b_j\, u_j\, f_j \text{ in } K[x_1,\ldots,x_i]$$

d.h. $f \in \text{ideal}(F \sqcap K[x_1,\ldots,x_i])$.

Damit ist

$(\text{ideal}(F) \sqcap K[x_1,\ldots,x_i]) \subseteq \text{ideal}(F \sqcap K[x_1,\ldots,x_i])$

gezeigt.

$\text{ideal}(F \sqcap K[x_1,\ldots,x_i]) \subseteq \text{ideal}(F) \sqcap K[x_1,\ldots,x_i]$

gilt sowieso, da das Ideal links gerade von den Polynomen aus F erzeugt wird, die nur von den Variablen

$x_1, \ldots, x_i$ abhängen, und da $F \subseteq \text{ideal}(F)$

folgt die Behauptung. ▌

G.5.2. Folgerung: Wenn man die Gröbner Basis von A bezüglich aller 'benötigten' Variablenreihenfolgen

$\{x_{i(1)},\ldots,x_{i(d)},x_{i(d+1)},\ldots,x_{i(r)}\}$

bildet, wobei

$\{x_{i(1)},\ldots,x_{i(d)}\}$ aus der Menge aller d-elementigen

Teilmengen ohne Wiederholungen (d=0,...,r) von

$\{x_1,\ldots,x_r\}$ sind, so erhält man alle Variablen

von denen A nicht abhängt und somit die Dimension von A.

Beispiel:

Für A In $K[x_1,x_2,x_3]$ müssen

3 Gröbner Basen von A berechnet werden. Und zwar

GB(A) bezüglich $\{x_1,x_2,x_3\}$

um dim(A) = -1 (d.h. es gibt keine Nullstelle) und

um dim(A) = 0 (d.h. es gibt endlich viele Nullstellen)

entscheiden zu können.
Treffen diese Fälle nicht zu, so ist die Dimension sicher ≥ 1.

Dann müssen die folgenden Gröbner-Basen berechnet werden:

GB($\underline{A}$) bezüglich $\{x_2, x_3, x_1\}$

GB($\underline{A}$) bezüglich $\{x_3, x_1, x_2\}$.

Damit kennt man:

$\underline{A} \sqcap K[x_1]$

$\underline{A} \sqcap K[x_2]$

$\underline{A} \sqcap K[x_3]$

und:

$\underline{A} \sqcap (K[x_1, x_2] = K[x_2, x_1])$

$\underline{A} \sqcap (K[x_2, x_3] = K[x_3, x_2])$

$\underline{A} \sqcap (K[x_1, x_3] = K[x_3, x_1])$

H. Auf Gröbner-Basen aufbauende Algorithmen

H.1. Der Algorithmus von B. Buchberger

Aus Satz [G.2.3., p. 44] ergibt sich folgender Algorithmus: Betrachte die Menge aller Paare von Polynomen aus G, berechne ihre S-Polynome und reduziere sie auf Normalform. Ist die Normalform eines S-Polynoms nicht 0, so füge das Polynom zu der Menge G hinzu und betrachte wieder alle Paare von Polynomen aus G. Falls sich alle S-Polynome auf 0 reduzieren, so ist die Bedingung des Satzes [G.2.3., p. 44] erfüllt, G also eine Gröbner Basis.

H.1.1. Algorithmus DIGB: nach [Buchberger 1984]

```
          G ← DIGB(F)
[Compute the Gröbner Basis for the set F.
G and F are sets of polynomials in r variables.
ideal(G) = ideal(F) and G is a Gröbner Basis. ]

(1) [Initialise. ]
     G ← F.
     B ← { {f,g} : f, g ε G, f ≠ g }.
(2) [Augment S-polynomials if necessary. ]
     while B ≠ ∅ do (
           select {f,g} ε B.
           B ← B - {f,g}.
           h' ← DIPSP(f,g).
           h  ← DIPNF(h',G). [for a fixed NF]
           if h ≠ 0 then (
              B ← B ⊔ { {g,h} : g ε G }.
              G ← G ⊔ {h} ) ) ▌
```

Zum Beweis der partiellen Korrektheit und der Terminierung siehe [Buchberger 1970] .

Ist von einer Menge F schon bekannt, daß sie eine Gröbner Basis ist, so kann das bei der Bildung der Menge B in Step 1 berücksichtigt werden. Step 2 bleibt ungeändert:

H.1.2. Algorithmus DIGBA:

```
          G ← DIGBA(p,F)
[Compute the Gröbner Basis for the set {p} ⊔ F.
G and F are sets of polynomials in r variables.
p is a polynomial in r variables. F is a Gröbner Basis.
ideal(G) = ideal(F)+ideal(p) and G is a Gröbner Basis. ]

(1) [Initialise. ]
     G ← F.
     B ← { {p,g} : g ε G }. G ← COMP(p,G).
```

```
(2) [Augment S-polynomials if necessary. ]
    while B ≠ ∅ do (
          select {f,g} ε B.
          B ← B - {f,g}.
          h' ← DIPSP(f,g).
          h  ← DIPNF(h',G). [for a fixed NF]
          if h ≠ 0 then (
              B ← B ⊔ { {g,h} : g ε G }.
              G ← G ⊔ {h} ) ) ▌
```

H.2. S-Polynome

Aus der Definition [G.2.4., p. 44] der S-Polynome ergibt sich unmittelbar der folgende Algorithmus (Step 3). Step 1 und Step 2 behandeln lediglich einige Spezialfälle. Die Koeffizienten der Polynome sollen aus Körpern stammen.

H.2.1. Algorithmus DIPSP: nach [G.2.4., p. 44]

```
                          C ← DIPSP(A,B)
[Distributive polynomial S-polynomial. A and B are
polynomials in distributive representation. C is
the S polynomial of A and B. ]

(1)  [A=0 or B=0. ] C ← 0.
     if A = 0 or B = 0 then return.
     A = a lp(A) + A'. B = b lp(B) + B'.
     e ← lp(A). f ← lp(B).
     if A' = 0 and B' = 0 then return.

(2)  [A'=0 or B'=0. ] g ← lcm(e,f).
     if A' = 0 then ( g = f' f. c ← - a.
                      C ← B' c f'. return ).
     if B' = 0 then ( g = e' e.
                      C ← A' b e'. return ).

(3)  [A'≠0 and B'≠0. ]
     g = f' f. g = e' e.
     C ← A' b e' - B' a f' ▌
```

Falls die Koeffizienten aus Körpern stammen, die Quotientenkörper von Integritätsringen sind, so läßt sich die Arithmetik in diesen Quotientenkörpern gänzlich vermeiden.

Sind diese Koeffizientenringe zuzätzlich noch ZPE-Ringe (d.h. gilt in ihnen die Eindeutige Zerlegbarkeit in irreduzible Elemente), und ist diese Zerlegung konstruktiv ausführbar, so ergeben sich ähnliche Modifikationen wie sie im Folgenden für die ganzen Zahlen durchgeführt wurden.

H.2.2. Algorithmus DIIPSP:

```
                          C ← DIIPSP(A,B)
[Distributive integral polynomial S-polynomial. A and B are
polynomials in distributive representation. C is
the S polynomial of A and B. ]

(1)  [A=0 or B=0. ] C ← 0.
     if A = 0 or B = 0 then return.
     A = a lp(A) + A'. B = b lp(A) + B'.
     e ← lp(A). f ← lp(A).
     if A' = 0 and B' = 0 then return.

(2)  [A'=0 or B'=0. ]
     g ← lcm(e,f). c' ← gcd(a,b).
     if A' = 0 then ( g = f' f. a ← - a / c'.
                      C ← B' a f'. return ).
     if B' = 0 then ( g = e' e. b ← b / c'.
                      C ← A' b e'. return ).

(3)  [A'≠0 and B'≠0. ]
     a ← a / c'. b ← b / c'.
     g = f' f. g = e' e.
     C ← A' b e' - B' a f' ▌
```

Man beachte die Analogie wie hier die PP f' und e' gebildet werden und die Koeffizienten b und a. c' wird gerade so gewählt, daß alle Divisionen in den ganzen Zahlen aufgehen. Alle Rechenoperationen zwischen Basiskoeffizienten finden somit nur noch in den ganzen Zahlen statt.

H.3. Normalform

Der folgende Algorithmus implementiert das in [G.1.7., p. 42] definierte Reduktionsverfahren für Polynomringe über Körpern. In der äusseren repeat-Schleife werden alle Terme des zureduzierenden Polynoms auf Teilbarkeit getestet und gegebenenfalls bezüglich P reduziert. Nicht reduzierbare Terme werden in dem Polynom R abgespeichert. Abhängig von der Operation 'select Q ε P'' erhält man verschiedene Reduktionen.

Für die Rechenzeit am günstigsten hat es sich herausgestellt, das Polynom mit den wenigsten Termen als erstes auszuwählen.

Denkbar wäre es auch die Menge aller Normalformen zu berechnen, d.h. alle möglichen Selektionen vorzunehmen, und dann aus dieser Menge ein günstigstes Polynom herauszusuchen.

H.3.1. Algorithmus DIPNOR:

```
                        R ← DIPNF(P,S)
[Distributive polynomial normal form. P is a list
of non zero polynomials in distributive
representation in r variables. S is a distributive
polynomial. R is a polynomial such that S is reducible to R
modulo P and R is in normal form with respect to P. ]

(1)  [S=0 or P empty. ]
     if S = 0 or P = ∅ then ( R ← S. return ).
(2)  [Reduction step. ] R ← 0. S' ← S.
     repeat   P' ← P.
          S' = a lp(S') + S''. e ← lp(S'). S' ← S''.
          repeat
               select Q ε P'.
               P' ← P' - {Q}. Q = b lp(Q) + Q'.
               f  ← lp(Q).

               s  ← f ≤M e.

               until P' = ∅ or s = true.
          if s = false then R ← R + a e
                       else if Q' ≠ 0 then (
                            e = g f. c ← a / b.
                            S' ← S' - Q' c g ).
          until S' = 0. ▌
```

Ähnlich wie bei den S-Polynomen läßt sich auch die Normalform [G.1.7., p. 42] so abändern, daß Arithmetik in Quotientenkörpern vermieden wird.

Da die Polynome in der Gröbner Basis nur bis auf einen Faktor aus dem

Koeffizientenkörper bestimmt sind, können alle Polynome mit dem kleinsten gemeinsamen Vielfachen der Nenner der Basiskoeffizienten multipliziert werden.

Ist die Division 'c ← a / b' im Algorithmus DIPNF nicht ausführbar, so muß nun der schon reduzierte Teil R des Polynoms und der noch zu reduzierende Teil S' mit einer geeigneten Zahl d hochmultipliziert werden. d wird gerade so bestimmt, daß

lp(S') b - lp(Q') a g = 0 gilt.

Hieraus ergibt sich auch die Bedeutung des Eingabeparameters R': in der Regel kann R' unbeschadet gleich 0 sein, soll jedoch von einem Polynom nur ein Teil reduziert werden, so muß auch der nicht zu reduzierende Teil mit eingegeben werden, damit er ebenfalls mit hochmultipliziert werden kann.

Diese Modifikation ist aber nicht zuverwechseln mit einer Berechnung von Gröbner Basen über den ganzen Zahlen, oder Koeffizientenbereichen die tatsächlich nur Ringe sind.

H.3.2. Algorithmus DIIPNF:

```
                        R ← DIIPNF(P,R',S)
[Distributive integral polynomial normal form. P is a list
of non zero integral polynomials in distributive
representation in r variables. R' + S is a distributive integral
polynomial. R is a polynomial such that R' + S is reducible to R
modulo P and R is in normal form with respect to P. ]

(1)  [S=0 or P empty. ]
     if S = 0 or P = ∅ then ( R ← R' + S. return ).
(2)  [Reduction step. ] R ← R'. S' ← S.
     repeat   P' ← P.
              S' = a lp(S') + S''. e ← lp(S'). S' ← S''.
              repeat
                    select Q ε P'.
                    P' ← P' - {Q}. Q = b lp(Q) + Q'.
                    f  ← lp(Q).

                    s  ← f ≤M e.

                    until P' = ∅ or s = true.
              if s = false then R ← R + a e
                           else if Q' ≠ 0 then (
                                e = g f. d ← gcd(a,b).
                                a  ← a / d. b ← b / d.
                                R  ← R b.
                                S' ← S' b - Q' a g )
              until S' = 0 ▌
```

H.4. Algorithmen zur Bestimmung der Dimension -1, 0, ≥ 1

Der folgende Algorithmus verwendet die Sätze [G.3., p. 45] und [G.4., p. 46] zur Bestimmung der Dimension eines Polynomideals aus seiner Gröbner Basis.

In Step 2 wird entsprechend Satz [G.3., p. 45] fest gestellt, ob 1 in der Gröbner Basis enthalten ist.

In Step 3 werden entsprechend Satz [G.4., p. 46] die einvariabligen Headterme gezählt. Wenn es r einvariablige Headterme gibt, dann ist die Dimension gleich 0.

H.4.1. Algorithmus DIGBZT:

t ← DIGBZT(G)
[Distributive polynomials Gröbner basis finitly manny common zero test. G is an irreducible Gröbner basis.
t = -1 or 0 if dimension(ideal(G)) equal -1 or 0,
t = 1 if dimension(ideal(G)) ≥ 1.]

(1) [trivial cases.] t ← 1.
if G = () then return.
r ← number of variables.
if r = 0 then (t ← -1. return).

(2) [Dimension = -1.] ADV(G.q,G').
e ← lp(q).

$e = x^{(i)}$. $v = \{ j : i_j > 0 \quad 1 \le j \le r \}$.

if v = () then (t ← -1. return).
if RED(v) = () then V ← 1 else V ← 0.

(3) [Count univariate headterms.]
while G' ≠ () do (ADV(G'.q,G').
e ← lp(q).

$e = x^{(i)}$. $v = \{ j : i_j > 0 \quad 1 \le j \le r \}$.

if RED(v) = () then V ← V + 1).

if V = r then t ← 0 ▌

Der Algorithmus 'DIFZT' ist ähnlich wie der Algorithmus 'PFZT', auf die uns interesierende Anwendung in der Quantorenelimination zugeschnitten. Dabei wird eine (zunächst leere) Menge sukzessive um ein Polynom vergrössert, bis die Dimension des von diesen Polynomen erzeugten Ideals kleiner oder gleich 0 ist.

Daher wird in Step 2 auch die schon berechnete Gröbner Basis weiter verwendet. In Step 3 wird der Algorithmus 'DIGBZT' verwendet, um die Dimension zu bestimmen.

H.4.2. Algorithmus DIFZT:

DIFZT(r,p,S.t,S')
[Distributive polynomial finitly many common zero test.
p is polynomial. r is the number of variables if
p is in recursive representation, r = -1 if p is in distributive
representation. S is a Gröbner basis. S' is the Gröbner basis
of ideal(S,p). t = -1 or 0 if dimension(ideal(S')) equal -1 or 0,
t = 1 if dimension(ideal(S')) ≥ 1. if r ≤ 1 then t = 0.]

(1) [Convert to distributive representation.]
if 0 ≤ r ≤ 1 then (t ← 0. return).
if r < 0 then p' ← p

else p' ← p ε $K[x_1,...,x_r]$.

(2) [Gröbner basis for ideal(S,p').] S' ← DIGBA(p',S).

(3) [Dimension ideal(S').] t ← DIGBZT(S') ▮

I. Resultantensysteme und Gröbner-Basen

I.1. Reduzierbarkeit von Resultantensystemen

I.1.1. Bemerkung : Sei G die Gröbner-Basis von $\underline{A}$ = ideal(F), dann ist

$$\text{ideal}(G) \sqcap K[x_1] = \text{ideal}(G \sqcap K[x_1]).$$

Da $K[x_1]$ ein Hauptidealring ist, wird das Ideal

rechts von genau einem Polynom erzeugt. Dieses muß sich bezüglich GB($\underline{A}$) auf Null reduzieren lassen (nach Eigenschaft der Gröbner Basen), das geht aber nur, falls es selbst in GB($\underline{A}$) enthalten ist. Da die Polynome in GB($\underline{A}$) 'irreduzibel' sind, kann neben diesem auch kein

weiteres Polynom aus $K[x_1]$ mehr vorkommen.

I.1.2. Folgerung : Es war

$$S(x_2, \ldots, x_r, F) \subset (\text{ideal}(F) \sqcap K[x_1])$$
$$= \text{ideal}(G \sqcap K[x_1]) =: (p(x_1)).$$

also

$$S(x_2, \ldots, x_r, F) \subset \text{ideal}(p(x_1)).$$

Alle iterierten Resultantensysteme bezüglich irgend einer

Permutation von $x_2, \ldots, x_r$ liegen also in ideal($p(x_1)$),

diese Polynome sind somit durch $p(x_1)$ teilbar.

Allgemeiner gilt für i=1,...,r:

$$S(x_{i+1}, \ldots, x_r, F) \subset (\text{ideal}(F) \sqcap K[x_1,\ldots,x_i])$$
$$= \text{ideal}(G \sqcap K[x_1,\ldots,x_i])$$

d.h. jedes $f \in S(x_{i+1}, \ldots, x_r, F)$

läßt sich bezüglich $G' = \text{ideal}(G \sqcap K[x_1,\ldots,x_i])$

auf Null reduzieren.

I.2. Nullstellen von normierten iterierten Resultantensystemen

Nach dem vorhergehenden ist insbesondere (für i=1,...,r)

$$\text{ideal}(S(x_{i+1}, \ldots, x_r, F)) \subset \text{ideal}(G \sqcap K[x_1,\ldots,x_i]).$$

I.2.1. Bemerkung : Nach [C.2.2., p. 15] gilt dann:

$$\text{NST}(\text{ideal}(G \sqcap K[x_1,\ldots,x_i])) \subset \text{NST}(\text{ideal}(S(x_{i+1}, \ldots, x_r, F))).$$

Aber im Allgemeinen gibt es Nullstellen von

$\text{ideal}(S(x_{i+1}, \ldots, x_r, F))$, die keine Nullstellen von

$\text{ideal}(G \sqcap K[x_1,\ldots,x_i])$ sind.

Z.B. solche, für die alle Anfangskoeffizienten von F verschwinden.

Die obigen Überlegungen gelten sowohl für die 'einfachen' wie auch für die Kroneckerschen Resultantenssyteme. Die Kroneckerschen Resultantensysteme sollen nun weiter betrachtet werden.

Die unangenehme Eigenschaft, der Resultanten, daß ihr Verschwinden auch auf das Verschwinden der höchsten Koeffizienten zurück zuführen sein kann, läßt sich durch Normierung bezüglich der Hauptvariablen von wenigstens einem Polynom aus F beheben.

I.2.2. Hilfssatz : [Vd Waerden 1931, p. 8] Ein nicht verschwindendes Polynom

$f \in K[x_1,\ldots,x_r]$, vom Grade n in x_r,

kann durch eine Variablensubstitution so in ein Polynom

$f' \in K[x_1',\ldots,x_r']$ übergeführt werden,

daß der Koeffizient von $x_r^{\text{tdeg}(f')}$ eine

von 0 verschiedene Konstante ist. Ohne Einschränkung der Allgemeinheit kann dieses Element dann auf 1 normiert werden.

Beweis: Die Substitution sei

$$x_1 = x_1' + u_1 x_r'$$

$$x_2 = x_2' + u_2 x_r'$$

$$\ldots$$

$$x_r = \quad u_r x_r'$$

wobei die u_i $(i=1,\ldots,r)$ Unbestimmte sind,

die dem Körper K adjungiert werden.
Dann ist

$$f(x_1,\ldots,x_r) = f'(x_1',\ldots,x_r') =$$

$$= f^*(u_1,\ldots,u_r)\, x_r'^n + \ldots$$

$f^*(u_1,\ldots,u_r)$ als Polynom in den u_i $(i=1,\ldots,r)$

ist nun von Null verschieden. ▌

I.2.3. Bemerkung :

Anstelle der Unbestimmten $u_1,\ldots,u_r$ kann man

auch passend gewählte Elemente aus K nehmen, für die

$f^*(u_1,\ldots,u_r)$ nicht verschwindet.

Für die Existenz solcher Elemente siehe [Gröbner, II, p. 8, I, p. 78f]

I.2.4. Bemerkung : Die obige Substitution ist eine reguläre linear homogene Abbildung der Variablen:

$$x_i \dashrightarrow \Sigma_{j=1,\ldots,r}\, v_{i\,j}\, x_j \quad \text{für } i=1,\ldots,r$$

$$= 1 \quad \text{für } j=i,\ 1 \leq i \leq r-1$$

wobei $v_{i\,j} = u_i$ für $j=r,\ 1 \leq i \leq r$

$$= 0 \quad \text{sonst.}$$

mit $v_{i\,j} \in K$ $(i,j=1,\ldots,r)$ und $\det(v_{i\,j}) \neq 0$

falls $u_r \neq 0$.

Jede Nullstelle von f geht durch dieselbe lineare Abbilung in eine

Nullstelle von f' über, und man erhält so auch alle Nullstellen von f'.

Wegen $\det(v_{i\ j}) \neq 0$ gilt auch die Umkehrung.

I.2.5. Satz : [Vd Waerden 1931, p. 9] Befindet sich unter den Polynomen aus F wenigstens eins, das bezüglich der Hauptvariablen normiert ist, so ist das Verschwinden des Kronekerschen Resultantensystems nach [C.5.3., p. 21] notwendig und hinreichend für die Existenz einer gemeinsamen Nullstelle der Polynome aus F.

Beweis: Nach [C.5.3., p. 21] können wegen der Normierung nicht alle Anfangskoeffizienten der Polynome aus F verschwinden. Es bleibt dann nur der Fall, daß die Polynome aus F eine gemeinsame Nullstelle haben, falls das Kroneckersche Resultantensystem verschwindet. ▌

I.2.6. Satz : [Vd Waerden 1931, p. 9]
Ist das iterierte Kroneckerschen Resultantensystem

$Sk(x_{i+1},\ldots,x_r, F)$ normiert bezüglich x_i,

so läßt sich jede Nullstelle des folgenden Kroneckerschen Resultantensystems

$Sk(x_i,x_{i+1},\ldots,x_r, F)$ auf mindestens eine

Weise zu einer Nullstelle

von $Sk(x_{i+1},\ldots,x_r, F)$ ergänzen (i=1,...,r).

Beweis: Wegen der Normierung können nicht alle Anfangskoeffizienten der Polynome aus

$Sk(x_{i+1},\ldots,x_r, F)$ nach einsetzen einer

Nullstelle von $Sk(x_i,x_{i+1},\ldots,x_r, F)$

verschwinden. Es bleibt demnach immer eine algebraische Gleichung von mindestens dem Grad 1, zu der dann eine Nullstelle gefunden werden kann. Diese zusammen mit der von

$Sk(x_i,x_{i+1},\ldots,x_r, F)$ ist dann eine

Nullstelle von $Sk(x_{i+1},\ldots,x_r, F)$ (i=1,...,r) ▌

I.2.7. Folgerung : Unter den Voraussetzungen der Normierung von Satz [I.2.5., p. 61] gilt:

$NST(\ ideal(Sk(x_{i+1},\ldots,x_r, F)\ )\ )\ C$

$$NST(\ ideal(\ G \sqcap K[x_1,\ldots,x_i]\ )\ )\ (i=1,\ldots,r).$$

Beweis: Wegen der Normierung läßt sich jede Nullstelle

$(a_1,\ldots,a_i)$ von $ideal(Sk(x_{i+1},\ldots,x_r,\ F\ )\ )$

zu einer Nullstelle $(a_1,\ldots,a_r)$ von ideal(F) fortsetzen.

Da ideal(F) = ideal(G) ist

$(a_1,\ldots,a_i)$ auch Nullstelle von

$ideal(\ G\ ) \sqcap K[x_1,\ldots,x_i] =$

$ideal(\ G \sqcap K[x_1,\ldots,x_i]\ )\ (i=1,\ldots,r).$

I.2.8. Folgerung : Unter den Voraussetzungen der Normierung von Satz [I.2.5., p. 61] gilt:

$$NST(\ ideal(Sk(x_{i+1},\ldots,x_r,\ F\ )\ )\ ) =$$

$$NST(\ ideal(\ G \sqcap K[x_1,\ldots,x_i]\ )\ )\ (i=1,\ldots,r).$$

Beweis: Folgt aus [I.2.7., p. 61] und [I.2.1., p. 59] . ▌

I.2.9. Bemerkung : Unter den Voraussetzungen der Normierung von Satz [I.2.5., p. 61] gilt:

$$rad(\ ideal(Sk(x_{i+1},\ldots,x_r,\ F\ )\ )\ ) =$$

$$rad(\ ideal(\ G \sqcap K[x_1,\ldots,x_i]\ )\ )\ (i=1,\ldots,r)$$

wegen [I.2.8., p. 62] und [C.2.6., p. 16] ▌

I.3. Weitere Eigenschaften der Gröbner-Basen

I.3.1. Hilfssatz :

Sei $\underline{A} \subset K[x_1,\ldots,x_r]$ ein Ideal und

$p = p' q^e \in \underline{A}$, e eine natürliche Zahl.

Falls $1 \in \underline{A} + (q)$ so ist auch $p' \in \underline{A}$.

Beweis:

Es sei $\underline{A} = (f_1,\ldots,f_n)$, $p \in \underline{A}$ bedeutet es gibt

$g_i \in K[x_1,\ldots,x_r]$ $(i=1,\ldots,n)$ mit

$p = \Sigma_{j=1,\ldots,n}\ g_j f_j$.

$1 \in \underline{A} + (q)$ bedeutet: es gibt

$h_i \in K[x_1,\ldots,x_r]$ $(i=1,\ldots,n+1)$ mit:

$1 = \Sigma_{j=1,\ldots,n}\ h_j f_j + h_{n+1} q$.

Multiplikation mit $p' q^{e-1}$ ergibt

$$p' q^{e-1} = p' q^{e-1} (\Sigma_{j=1,\ldots,n}\ h_j f_j + h_{n+1} q)$$

$$= \Sigma_{j=1,\ldots,n}\ p' q^{e-1} h_j f_j + h_{n+1} p$$

$$= \Sigma_{j=1,\ldots,n}\ p' q^{e-1} h_j f_j + h_{n+1} (\Sigma_{j=1,\ldots,n}\ g_j f_j)$$

$$p' q^{e-1} = \Sigma_{j=1,\ldots,n}\ h_j f_j$$

also ist $p' q^{e-1} \in \underline{A}$. Nach e-maliger Anwendung folgt $p' \in \underline{A}$. ∎

I.3.2. Satz :

Sei $GB(\underline{A}) = \{f_1,\ldots,f_n\}$ die irredizible

Gröbner Basis von $\underline{A}$ C $K[x_1,\ldots,x_r]$.

Für $p = p' \; q \; \varepsilon \; GB(\underline{A})$ $q \neq 1$, gilt nicht

$$1 \; \varepsilon \; \underline{A} + (\; q \;).$$

Beweis: Wäre $1 \; \varepsilon \; \underline{A} + (\; q \;)$ so folgt nach dem Hilfssatz [I.3.1., p. 63] , daß $p' \; \varepsilon \; \underline{A}$; aber es ist $NF(p',GB(\underline{A})) \neq 0$:
denn angenommen $NF(p',GB(\underline{A})) = 0$, so gäbe es ein

$f \; \varepsilon \; \{f_1,\ldots,f_n\}$, $f \neq p$ mit

$lp(f) \leq_M lp(p') <_M lp(p)$ im Widerspruch zur

Irreduzibiliät von p. ▌

Dieser Satz illustriert besonders deutlich die Vorteile der Gröbner Basen gegenüber den Resultantensystemen. Denn ist $p = res(f,g) = p' \; q \; \varepsilon \; \underline{A}$ so kann es sein, daß $1 \; \varepsilon \; \underline{A} + (q)$ ist. Zum Beispiel dann, wenn q Faktor der höchsten Koeffizienten von f und g ist. Dieser Ausnahmefall läßt sich bei Resultanten nur durch Normierung beheben, für Gröbner Basen kann dieser Ausnahmefall auch ohne Normierung nicht eintreten.

J. Rechenzeitvergleiche zwischen PFZT und DIFZT

Bezeichnungen in den Beispielen:

'Resultanten' = die entsprechenden Ergebnisse mit PFZT gerechnet. Dies ist der Collins'sche Algorithmus in einer verbesserten Form.

'Resultanten Boe' = die entsprechenden Ergebnisse mit PFZT gerechnet. Die iterierten Resultantensysteme von weniger Polynomen als Variablen werden sofort Null gesetzt.

'Gröbner B lex' = die entsprechenden Ergebnisse mit DIFZT gerechnet. Berechnung mit Hilfe von Gröbner Basen, mit lexikographischer Termordnung.

'Gröbner B grad' = die entsprechenden Ergebnisse mit DIFZT gerechnet. Berechnung mit Hilfe von Gröbner Basen, mit graduierter Termordnung.

'G B grad int' = die entsprechenden Ergebnisse mit DIFZT gerechnet. Berechnung mit Hilfe von Gröbner Basen, mit graduierter Termordnung und ohne Arithmetik rationaler Zahlen.

Die Angabe 'dim =' bedeutet:

-1 entspricht 'keine Nullstellen'

0 entspricht 'endlich viele Nullstellen'

≤0 entspricht 'keine oder endlich viele Nullstellen'

≥1 entspricht 'unendlich viele Nullstellen'

? entspricht 'unendlich viele Nullstellen oder nicht festgestellt wie viele Nullstellen es gibt'

Time entspricht der Rechenzeit in Millisekunden auf einer IBM 3081D.

Cells entspricht den verbrauchten Speicherzellen.

Es wird eine Menge von Polynomen jeweils um ein Polynom vergrößert und dann festgestellt, welche Dimension das von ihnen erzeugte Ideal hat. Stellt ein Algorithmus fest, daß die Dimension kleiner oder gleich 0 ist, so wird kein neues Polynom mehr hinzugenommen:

```
F ← ().
repeat  generate a new random polynomial p.
        F ← COMP(p,F).
        compute dim ideal( F ) using different methods.
        until dim ≤ 0.
```

J.1. Polynome in 2 Variablen

Bei nur zwei Variablen ist es noch möglich 'dicht besetzte' Polynome zu betrachen. Die Zeit- und Speicherplatzangaben sind für jedes neu hinzugekommene Polynom angegeben. 'Dense' Polynome führen zu hohen Rechenzeiten bei den Gröbner Basen - Algorithmen:

K[x,y]

$p_1 = -5\,x^3\,y^4 - 14\,y^4 - 11\,x^3\,y^2$

Resultanten	dim = ?	Time = 445	Cells = 2594
Gröbner B lex	dim ≥ 1	Time = 8	Cells = 43

$p_2 = -13\,x^2\,y^4 - x^2\,y^3 - 2\,x\,y^2$

Resultanten	dim = ?	Time = 807	Cells = 5594
Gröbner B lex	dim ≥ 1	Time = 1503	Cells = 7515

$p_3 = 12\,x^3\,y^2 + 8\,x^2\,y^2 - 7\,y^2$

Resultanten	dim = ?	Time = 1368	Cells = 12862
Gröbner B lex	dim ≥ 1	Time = 9400	Cells = 86421

$p_4 = 9\,x^2\,y^3 + 7\,x^3 - 15\,x$

Resultanten	dim ≤ 0	Time = 8126	Cells = 76732
Gröbner B lex	dim = 0	Time = 19	Cells = 83

K[x,y]

$p_1 = -3\,x\,y^4 - 2\,y^2 - 2\,x^2 - 14\,x^1$

Resultanten	dim = ?	Time = 398	Cells = 781
Gröbner B lex	dim ≥ 1	Time = 9	Cells = 53

$p_2 = -15\,y^4 + 6\,y^3 - 7\,y^2 + 9\,x\,y$

Resultanten	dim ≤ 0	Time = 2116	Cells = 20337
Gröbner B lex	dim = 0	Time = 45818	Cells = 429087

J.2. Polynome in 3 Variablen

Die nächsten beiden Beispiele zeigen den Einfluß großer Koeffizienten. Die Resultanten-Algorithmen sind für grössere Eingabekoeffizienten langsamer als die Gröbner Basen - Algorithmen. Die Zeit- und Speicherplatzangaben sind für jedes neu hinzugekommene Polynom angegeben.

$K[x_1, x_2, x_3]$

$q_1 = (-508694266\ x_1\ x_2 - 268696740)$

Resultanten Boe	dim = ?	Time = 0	Cells = 5
G B grad int	dim ≥ 1	Time = 2	Cells = 25
Gröbner B grad	dim ≥ 1	Time = 2	Cells = 29
Resultanten	dim = ?	Time = 40	Cells = 603
Gröbner B lex	dim ≥ 1	Time = 1	Cells = 29

$q_2 = (473206071\ x_1\ x_3 - 522797016\ x_1^2\ x_2 - 265217482\ x_2)$

Resultanten Boe	dim = ?	Time = 0	Cells = 6
G B grad int	dim ≥ 1	Time = 178	Cells = 713
Gröbner B grad	dim ≥ 1	Time = 172	Cells = 574
Resultanten	dim = ?	Time = 79	Cells = 1254
Gröbner B lex	dim ≥ 1	Time = 161	Cells = 448

$q_3 = (513784118\ x_1^2\ x_3^2 - 473995974\ x_2^2)$

Resultanten Boe	dim ≤ 0	Time = 7892	Cells = 168814
G B grad int	dim = 0	Time = 288	Cells = 5583
Gröbner B grad	dim = 0	Time = 431	Cells = 8745
Resultanten	dim ≤ 0	Time = 7878	Cells = 168813
Gröbner B lex	dim = 0	Time = 378	Cells = 6072

Ein ähnliches Beispiel mit kleineren Koeffizienten:

$q_1 = (-14\ x_1\ x_2\ -8)$

Resultanten Boe	dim = ?	Time = 0	Cells = 5
G B grad int	dim ≥ 1	Time = 2	Cells = 25
Gröbner B grad	dim ≥ 1	Time = 2	Cells = 29
Resultanten	dim = ?	Time = 38	Cells = 587
Gröbner B lex	dim ≥ 1	Time = 1	Cells = 29

$q_2 = (14\ x_1\ x_3\ -15\ x_1^2\ x_2\ -7\ x_2)$

Resultanten Boe	dim = ?	Time = 0	Cells = 6
G B grad int	dim ≥ 1	Time = 161	Cells = 301
Gröbner B grad	dim ≥ 1	Time = 162	Cells = 353
Resultanten	dim = ?	Time = 77	Cells = 1228
Gröbner B lex	dim ≥ 1	Time = 157	Cells = 309

$q_3 = (15\ x_1^2\ x_3^2\ -13\ x_2^2)$

Resultanten Boe	dim ≤ 0	Time = 1889	Cells = 38473
G B grad int	dim = 0	Time = 181	Cells = 1170
Gröbner B grad	dim = 0	Time = 192	Cells = 1342
Resultanten	dim ≤ 0	Time = 1932	Cells = 38472
Gröbner B lex	dim = 0	Time = 78	Cells = 845

Hier noch ein Beispiel mit 'dense' Polynomen:

$$p_1 = (15\, x_1^2\, x_3^2 + 14\, x_1\, x_3 - 13\, x_2^2 - 15\, x_1^2\, x_2 - 14\, x_1\, x_2 - 7\, x_2 - 8)$$

Resultanten Boe	dim = ?	Time = 0	Cells = 5
G B grad int	dim ≥ 1	Time = 2	Cells = 66
Gröbner B grad	dim ≥ 1	Time = 3	Cells = 94
Resultanten	dim = ?	Time = 1074	Cells = 34471
Gröbner B lex	dim ≥ 1	Time = 4	Cells = 94

$$p_2 = (-9\, x_1^2\, x_2\, x_3^2 + 11\, x_1\, x_2^2\, x_3 + 13\, x_1^2\, x_3 - 8\, x_1\, x_2 + 25\, x_2 - 2\, x_1 - 7)$$

Resultanten Boe	dim = ?	Time = 1	Cells = 6
G B grad int	dim ≥ 1	Time = 718	Cells = 18541
Gröbner B grad	dim ≥ 1	Time = 1131	Cells = 18737
Resultanten	dim = ?	Time = 1910	Cells = 54628
Gröbner B lex	dim ≥ 1	Time = 33129	Cells = 693954

$$p_3 = (-13\, x_3 - 8\, x_2 - 9\, x_1^2 + 7\, x_1 - 13)$$

Resultanten Boe	dim ≤ 0	Time = 15307	Cells = 339726
G B grad int	dim = 0	Time = 3418	Cells = 108863
Gröbner B grad	dim = 0	Time = 12207	Cells = 247762
Resultanten	dim ≤ 0	Time = 15357	Cells = 340023

J.3. Polynome in 4 Variablen

Bei 4 Variablen müssen die Polynome schon 'sparse' gewählt werden, um zu Ergebnissen zu kommen. Die Zeit- und Speicherplatzangaben sind für jedes neu hinzugekommene Polynom angegeben.

K[x,y,z,w]

Ein Beispiel, das sich mit lexikographischer Gröbner Basis nicht berechnen ließ:

$$p_1 = 15\,y^3\,z^3\,w^3 + 7\,x\,y\,z^3\,w^3 - 2\,x\,y\,z^2\,w^2 - 8\,y\,z^2\,w^2 + 3\,x\,z^2\,w^2 + 6\,z^2\,w^2 - 15\,x^2\,y\,z\,w$$

$$p_2 = -4\,x^2\,y^3\,z^3\,w^3 + 6\,x\,y^2\,w^3 - 9\,x^3\,y^3\,w^2 - 7\,x^3\,y^3\,w^2 - 14\,x\,y^3\,w^2 - 6\,x^3\,y\,z\,w - 14\,x\,z$$

Das Programm wurde abgebrochen, da nicht mehr genug Speicherplatz zur Verfügung stand.

Ein Beispiel mit 'sparse' Polynomen und mittelgroßen Koeffizienten.

$$p_1 = 18951\,w - 17256\,x^2 - 5819$$

Resultanten	dim = ?	Time = 467	Cells = 1304
Gröbner B lex	dim ≥ 1	Time = 10	Cells = 60

$$p_2 = 19073\,y^2\,z + 48267$$

Resultanten	dim = ?	Time = 633	Cells = 2648
Gröbner B lex	dim ≥ 1	Time = 163	Cells = 86

$$p_3 = -9443\,w^5 + 6106\,w^4 - 8569\,w$$

Resultanten	dim = ?	Time = 3915	Cells = 43338
Gröbner B lex	dim ≥ 1	Time = 361	Cells = 2667

$$p_4 = 13669\,z^5 - 29474\,y^3\,z^3 + 24215\,y^2$$

Resultanten	dim ≤ 0	Time = 10284	Cells = 113859

Gröbner B lex dim = 0 Time = 496 Cells = 2720

J.4. Polynome in 6 Variablen

Die Zeit- und Speicherplatzangaben sind für jedes neu hinzugekommene Polynom angegeben. Zwei Beispiele mit 6 Variablen die sich auch mit Resultanten nicht berechnen ließen:

K[x,y,z,u,v,w]

$$p_1 = -2\,x\,y\,z^2\,u^2\,v^3\,w^3 + 4\,x\,y^3\,u^3\,v^2\,w^3 + 5\,x^2\,y^2\,z\,u\,w^3 - 14\,z\,v^2\,w^2 + 7\,x\,y\,z^3\,u^3\,v^3\,w - 7\,x\,y\,u$$

$$p_2 = -14\,x\,z\,v^3\,w^3 - 2\,y\,u\,v^2\,w^2 - x^2\,y^2\,u\,v\,w^2 + 12\,x^2\,z^3\,u^3\,w^2 - 4\,x^2\,y^3\,z^3\,u^3\,v^3\,w + 15\,y^2\,z^2\,u$$

Das Programm wurde abgebrochen, da nicht mehr genug Speicherplatz zum Weiterrechnen zur Verfügung stand.

J.4.1. Bemerkung : Das gleiche Beispiel ohne Faktorisierung und Betrachtung aller Faktortupel in dem Resultanten Algorithmus: Nach mehr als 154 Garbage Collections wurde das Programm mit Zeitüberschreitung abgebrochen.

Beispiele mit 'sparse' Polynomen sind wieder berechenbar:

$p_1 = 5\ v^2 - 2$

Resultanten	dim = ?	Time = 788	Cells = 3986
Gröbner B lex	dim ≥ 1	Time = 9	Cells = 56

$p_2 = -\ 15\ y - 15$

Resultanten	dim = ?	Time = 1039	Cells = 5816
Gröbner B lex	dim ≥ 1	Time = 153	Cells = 99

$p_3 = 6\ u + 2$

Resultanten	dim = ?	Time = 1446	Cells = 9401
Gröbner B lex	dim ≥ 1	Time = 142	Cells = 130

$p_4 = -\ 12\ u\ w + 2\ u^2$

Resultanten	dim = ?	Time = 2686	Cells = 19975
Gröbner B lex	dim ≥ 1	Time = 120	Cells = 227

$p_5 = -\ w - 4$

Resultanten	dim ≤ 0	Time = 4954	Cells = 39757
Gröbner B lex	dim = -1	Time = 46	Cells = 178

Und etwas 'dichtere' Polynome:

$p_1 = 5\ v^2 - 2$

Resultanten	dim = ?	Time = 800	Cells = 4275
Gröbner B lex	dim ≥ 1	Time = 9	Cells = 56

$p_2 = -15\ x\ y^2 - 15$

Resultanten	dim = ?	Time = 1053	Cells = 6513
Gröbner B lex	dim ≥ 1	Time = 152	Cells = 100

$p_3 = 6\ u + 2$

Resultanten	dim = ?	Time = 1450	Cells = 10089
Gröbner B lex	dim ≥ 1	Time = 139	Cells = 131

$p_4 = -\ 12\ u^2\ w + 2\ u^2$

Resultanten	dim = ?	Time = 2809	Cells = 21921
Gröbner B lex	dim ≥ 1	Time = 118	Cells = 286

$p_5 = -\ z\ w^2 - 4$

Resultanten	dim = ?	Time = 3321	Cells = 27259
Gröbner B lex	dim ≥ 1	Time = 74	Cells = 257

$p_6 = -\ 2\ x\ y - 9\ x^2$

Resultanten	dim ≤ 0	Time = 19128	Cells = 163676
Gröbner B lex	dim = 0	Time = 348	Cells = 680

Die Zeit- und Speicherplatzangaben sind kumulativ für jedes neu hinzugekommene Polynom angegeben.

$K[x_1, x_2, x_3, x_4, x_5, x_6]$

$p_1 = (-15\, x_1\, x_2^2 + 3)$

Resultanten Boe	dim = ?	Time = 0	Cells = 8
G B grad int	dim ≥ 1	Time = 1	Cells = 43
Gröbner B grad	dim ≥ 1	Time = 2	Cells = 47
Resultanten	dim = ?	Time = 121	Cells = 2082
Gröbner B lex	dim ≥ 1	Time = 2	Cells = 47

$p_2 = (6\, x_3\, x_6^2 - 15\, x_4\, x_6 - 4)$

Resultanten Boe	dim = ?	Time = 1	Cells = 17
G B grad int	dim ≥ 1	Time = 77	Cells = 148
Gröbner B grad	dim ≥ 1	Time = 77	Cells = 163
Resultanten	dim = ?	Time = 438	Cells = 8405
Gröbner B lex	dim ≥ 1	Time = 72	Cells = 163

$p_3 = (-14\, x_2\, x_3\, x_5 + 11\, x_1^2\, x_3\, x_5 + 15\, x_1^2)$

Resultanten Boe	dim = ?	Time = 1	Cells = 27
G B grad int	dim ≥ 1	Time = 396	Cells = 2004
Gröbner B grad	dim ≥ 1	Time = 416	Cells = 2277
Resultanten	dim = ?	Time = 1068	Cells = 21094
Gröbner B lex	dim ≥ 1	Time = 1015	Cells = 9690

$p_4 = (- x_4 x_5 - 7 x_4^2 + 11 x_3)$

Resultanten Boe	dim = ?	Time = 1	Cells = 38
G B grad int	dim ≥ 1	Time = 1014	Cells = 12039
Gröbner B grad	dim ≥ 1	Time = 1098	Cells = 13026
Resultanten	dim = ?	Time = 2242	Cells = 45770
Gröbner B lex	dim ≥ 1	Time = 1159	Cells = 12172

$p_5 = (5 x_4^2 x_5 + x_3 - 6)$

Resultanten Boe	dim = ?	Time = 2	Cells = 50
G B grad int	dim ≥ 1	Time = 4559	Cells = 110537
Gröbner B grad	dim ≥ 1	Time = 5664	Cells = 106216
Resultanten	dim = ?	Time = 73858	Cells = 2655719
Gröbner B lex	dim ≥ 1	Time = 66972	Cells = 1288803

$p_6 = (15 x_3 x_5^2 + 3 x_3^2 x_4^2)$

Resultanten Boe	dim = ?	Time = 2319	Cells = 50549
G B grad int	dim = 0	Time = 107707	Cells = 3252954
Gröbner B grad	dim = 0	Time = 624098	Cells = 13687979
Resultanten	dim = ?	Time = 76180	Cells = 2706301

J.4.2. Bemerkung : Die Gröbner Basen Algorithmen entscheiden schon hier auf Dimension gleich Null, während die Resultanten Algorithmen dies noch nicht feststellen können.

J.5. Polynome in 8 Variablen

Die Zeit- und Speicherplatzangaben sind für jedes neu hinzugekommene Polynom angegeben.

$K[x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8]$

$p_1 = -4\, x_2^2\, x_7^2 - 2$

Resultanten	dim = ?	Time = 1219	Cells = 8651
Gröbner B lex	dim ≥ 1	Time = 10	Cells = 69

$p_2 = 15\, x_6 + 7$

Resultanten	dim = ?	Time = 2141	Cells = 17338
Gröbner B lex	dim ≥ 1	Time = 130	Cells = 115

$p_3 = 2\, x_8 - 8\, x_2$

Resultanten	dim = ?	Time = 2638	Cells = 22462
Gröbner B lex	dim ≥ 1	Time = 97	Cells = 150

$p_4 = -7\, x_6 - 8\, x_3$

Resultanten	dim = ?	Time = 3733	Cells = 33035
Gröbner B lex	dim ≥ 1	Time = 55	Cells = 228

$p_5 = -2$

Resultanten	dim ≤ 0	Time = 50	Cells = 9
Gröbner B lex	dim = -1	Time = 37	Cells = 141

J.6. Zusammenfassung

Der Algorithmus nach [F., p. 37] ist nicht viel langsamer als der Algorithmus nach [E.5., p. 36] , da der erstere auch schon bei der Berechnung eines Resultantensystems bezüglich einer zyklischen Permutation der der Variablen, das Verschwinden feststellt, und dann die Berechnung abbricht.

Die Berechnung mit Gröbner-Basen lohnt sich immer dann, falls die Polynome sehr 'sparse' sind, denn dann ist die Wahrscheinlichkeit größer, daß die eingegebenen Polynome schon eine Gröbner-Basis bilden.

Über den ganzen Zahlen benötigt die Berechnung von Gröbner-Basen in einigen Fällen nur 20 Prozent der Zeit wie über den rationalen Zahlen gerechnet.

Die größte Einsparung ergibt sich aber durch Verwendung der graduierten Termordnung Anstelle der lexicographischen Termordnung bei der Berechnung von Gröbner-Basen.

Die Benutzung reiner Resultantensysteme nach [E.3., p. 33] führt nicht weiter als die Berechnung mit zusätzlicher Faktorisierung nach [E.4., p. 35] wie das Beispiel [J.4.1., p. 72] zeigt.

Vom mathematischen Standpunkt aus sollte der Berechnung der Dimension mit Hilfe von Gröbner Basen der Vorzug gegenüber der Berechnung mit Hilfe von iterierten Resultantensystemen gegeben werden. Besonders das Beispiel [J.4.2., p. 76] zeigt, daß mit Gröbner Basen früher festgestellt werden kann ob die Dimension kleiner gleich 0 ist, auch wenn das wie in diesem Beispiel, mit einer höheren Rechenzeit erkauft werden muß.

Auch vom 'computational' Standpunkt ist den Gröbner Basen zumindest, wenn sie mit dem 'schnellsten' Algorithmus, d.h. mit Verwendung der graduierten Termordnung unter Vermeidung rationalzahliger Arithmetik, berechnet werden der Vorzug zugeben. In dieser Hinsicht ist besonders das Beispiel [POL9, Anhang 1] zu erwähnen, wo es mit mit Hilfe von Gröbner Basen möglich war eine Stufe in der Resolution weiter zurechnen, als es bislang möglich war.

Allerdings ist bei der Bewertung der praktischen Verwendbarkeit von Algorithmen Vorsicht am Platze, da die Rechenzeiten erstens von sehr vielen (unter Umständen nicht bekannten oder nicht beachteten) zusätzlichen Faktoren abhängt und zweitens alle Algorithmen stark von den Eingabeparametern abhängen, d.h. schon geringfügig 'dichtere' Polynome können zum Scheitern eines Verfahrens führen. Der 'beste' Algorithmus in diesem Sinn ist dann der, der ein gegebenes Problem überhaupt löst (auch wenn er dazu lang braucht).

K. Anwendung in der Quantorenelimination

K.1. Definition der Projektion bzw. Resolution

Die in den vorhergehen Abschnitten besprochenen Algorithmen sollen nun auf ein Problem in der Quantorenelimination angewendet werden. Hier wird zunächst nur kurz angegeben, wie sich die Frage nach der Dimension eines Polynomideals in dem Collins'schen Verfahren zur Quantorenelimination in der elementaren reellen Algebra stellt. Für die Quantorenelimination selbst sei hier auf [Collins 1974] [Collins 1975] verwiesen. In [Buchberger & Collins & Loos] befindet sich ein umfangreicheres Literaturverzeichnis, zusammengestellt von Collins, dazu.

Sei $\emptyset \neq A = \{ A_1, \ldots, A_n \} \subset (\ldots(K[x_1][x_2])\ldots)[x_r]$ $r \geq 2$.

K.1.1. Definition : [Collins 1974] [Böge 1983a]

$$R(A) := \{ red^k(B) : B \in A, \; k \geq 0 \text{ und } deg(red^k(B)) \geq 0 \}$$

$$L(A) := \{ ldcf(B) : B \in R \}$$

$$S'(A) := \{ psc_k(B, der(B)) : B \in R \text{ und } 0 \leq k < deg(\, der(B)\,) \}$$

$$S(A) := \{ psc_k(B_1, B_2) : B_1, B_2 \in R, \; B_1 \neq B_2 \text{ und } 0 \leq k < \min\{ deg(B_1), deg(B_2) \} \}$$

dann sei

$$Proj(A) := L(A) \sqcup S'(A) \sqcup S(A)$$

die Projektion von A.

Nach [Collins 1981] kommt man aber schon mit einer kleineren Menge von Polynomen, d. h. mit einem kleineren

Proj(A) aus.

Die folgende Definition ist aus den uns vorliegenden Computerprogrammen zusammengestellt, und soll illustrieren, an welchen Stellen die Bestimmung der Dimension von Polynomidealen benötigt wird.

K.1.2. Definition : [Collins 1981] Im folgenden sei $B \varepsilon A$.

$$R''(B) := \{ red^k(B) : 0 \leq k \text{ und } \deg(red^k(B)) \geq 0 \}$$

$$R'(B) := \{ red^0(B),..., red^k(B) \varepsilon R''(B) :$$

und k sei minimal mit

$$\dim \text{ideal}(ldcf(red^0(B)),..., ldcf(red^k(B))) \leq 0 \}$$

Dann sei

$$R^*(A) := \{ red^i(B) \varepsilon R'(B) : B \varepsilon A \}$$

weiter sei

$$L^*(A) := \{ ldcf(B) : B \varepsilon R^*(A) \}$$

Die Mengen der psc's seien definiert als:

$$P'(B,C) := \{ psc_k(B,C) : 0 \leq k < \min\{ \deg(B), \deg(C) \} \}$$

Für eine Menge

$$F := \{ F_1,..., F_l \}$$

$$F \subset (...(K[x_1])[x_2]...)[x_{r-1}] \quad r \geq 2$$

und Polynome B, C definieren wir

$$P(B,C,F) := \{ psc_0(B,C),..., psc_k(B,C) \varepsilon P'(B,C) :$$

und k sei minimal mit

$$\dim \text{ideal}(\{ psc_0(B,C),..., psc_k(B,C) \} \cup F) \leq 0 \}$$

damit sei

$$D(B) := \{ C \varepsilon P(red^i(B), der(red^i(B)), F) :$$

$$F = \{ ldcf(red^0(F)),..., ldcf(red^{i-1}(F)) \}$$

$$, \; red^{i}(B) \in R'(B) \; \}$$

und somit

$$S'^{*}(A) := \{ C \in D(B) : B \in A \}$$

Für Polynome B und C sei

$$R'(B,C) := \{ (red^{i}(B), red^{j(i)}(C)) :$$

$$red^{i}(B) \in R'(B), \; red^{j(i)}(C) \in R'(C)$$

und i, j(i) minimal mit

$$\dim ideal(ldcf(red^{0}(B)),..., ldcf(red^{i}(B)),$$

$$ldcf(red^{0}(C)),..., ldcf(red^{j(i)}(C))) \leq 0 \; \}$$

damit definieren wir

$$R(B,C) := \{ D \in P(red^{i}(B), red^{j(i)}(C), F),$$

$$F = \{ ldcf(red^{0}(B)),..., ldcf(red^{i-1}(B)),$$

$$ldcf(red^{0}(C)),..., ldcf(red^{j(i)-1}(C)) \}$$

$$: (red^{i}(B), red^{j(i)}(C)) \in R'(B,C) \; \}$$

Schließlich sei

$$S^{*}(A) := \{ D \in R(B,C) : B,C \in A, \; B \neq C \},$$

die Projektion von A ist dann wieder durch

$$Proj^{*}(A) := L^{*}(A) \sqcup S'^{*}(A) \sqcup S^{*}(A)$$

definiert.

K.1.3. Bemerkung : Es gilt

$$R(A)^{*} \subset R(A)$$

$$L(A)^{*} \subset L(A)$$

$$S'(A)^{*} \subset S'(A)$$

$$S(A)^{*} \subset S(A)$$

und somit

$$Proj^{*}(A) \subset Proj(A).$$

In praktisch vorkommenden Fällen wird dabei die Anzahl der 'Projektionspolynome' wesentlich verringert,

d.h. $Proj^{*}(A)$ ist wesentlich kleiner als $Proj(A)$.

(Die betrachteten Polynome haben oft schon konstante Koeffizienten).

Wie im Abschnitt [G.3., p. 45] [G.4., p. 46] [H.4., p. 56] beschrieben, kann mit Hilfe von Gröbner-Basen genau festgestellt werden, ob die Dimension der von den obigen Polynommengen erzeugten Ideale kleiner gleich Null ist, d.h. ob die Ideale höchstens endlich viele Nullstellen besitzen. Somit kann die Definition [K.1.2., p. 80] für die 'Projektion' verwendet werden.

Ebenso kann mit Hilfe von Resultantensystemen (Abschnitt [E., p. 30]) sicher festgestellt werden, wann die Ideale, die von den obigen Polynommengen erzeugt werden, höchstens die Dimension Null haben.

Wird mit Resultantensystemen nicht festgestellt, daß die Dimension ≤ 0 ist (obwohl das der Fall ist),

vergrösert sich nur die Menge $Proj^{*}(A)$,

die damit erst recht zur Definition der 'Projektion' ausreicht, da schon die 'Projektionsmenge' $Proj(A)$ nach [K.1.1., p. 79] ausreicht.

Dieser Sachverhalt läßt sich weiter ausnutzen, in dem man die Frage nach der Dimension in bestimmten Fällen offen läßt, wie bei [Böge 1984] siehe [E.5., p. 36] . Man kann den Test, zum Feststellen der Dimension, so zu wählen versuchen, daß sein Aufwand und der durch Vergrößerung von

$Proj^{*}(A)$ entstehende, zusammen möglichst klein werden.

K.2. Rechenzeitvergleiche für die Resolution

Die in den vorigen Abschnitten entwickelten Verfahren, sollen nun in die 'Projektionprocedure' nach [Collins 1981] eingebaut und verglichen werden.

Dabei wird die 'Projectionprocedure', bzw. nach der Bezeichnung von Böge 'Resultionprocedure', mit der Einsparung von unnötigen Folge-psc's bei vorliegen von Quantorenblöcken, nach [Böge 1983b] und [Böge 1983c] verwendet. Für die vollständige Beschreibung dieser Programme sei auf [Gebauer & Kredel 1984a] verwiesen.

In den folgenden 3 Beispielen werden die Rechenzeiten und der Speicherplatzverbrauch der Programme PFZT und DIFZT, so wie einer nach [Böge 1984] abgeschwächten Modifikation von PFZT (siehe [E.5., p. 36]) zusammengestellt.

PFZT - Collins = Abschätzung der Dimension mit Hilfe von iterierten Resultantensystemen. Das ist der der ursprüngliche Collins'sche Algorithmus in einer verbesserten Form.

DIFZT = Berechnung der Dimension mit Hilfe von Gröbner Basen, mit graduierter Termordnung und ohne Arithmetik rationaler Zahlen.

PFZT - Böge = PFZT - Collins, nur werden die iterierten Resultantensysteme von weniger Polynomen als Variablen sofort Null gesetzt.

Die Zeitangaben sind in Millisekunden, gerechnet auf einer IBM 3081D.

Die Angaben zum Speicherplatzverbrauch sind in Tausend Zellen.

Die Angaben in der Zeile 'Resultion' beziehen sich auf den Aufwand zur Elimination jeweils einer Variablen. Das ist im wesentlichen der Aufwand für die Berechnung der Ableitungen, Hauptkoeffizienten der Subresultanten und der Feststellung der Dimension. Ebenfalls enthalten ist der Aufwand für die Verwaltung der sogennanten 'location tables'.

Die Angaben in der Zeile 'Basis' beziehen sich auf den Aufwand zur Aufbereitung der Resolutionspolynome zu einer aus irreduziblen Polynomen bestehenden 'Basis'- Polynommenge. Das ist im wesentlichen der Aufwand für quadratfreie Faktorisierung, Berechnung der paarweisen grösten gemeinsamen Teiler und dann der irreduziblen Faktorisierung der Polynome.

Zu beachten ist, daß in dem Aufwand, sowohl bei 'Resultion' wie auch bei 'Basis', die Verwaltung der sogennanten Herkunftslisten der Polynome mit enthalten ist.

K.3. Beispiel POL1 in 4 Variablen

	PFZT - Böge		DIFZT	
Step	Time	Cells	Time	Cells
3 Variables				
Resultion	80	1.5	59	1.7
Basis	492	11.4	482	11.4
2 Variables				
Resultion	534	10.8	186	4.9
Basis	813	16.9	805	16.9
1 Variable				
Resultion	96	1.6	53	1.4
Basis	151	2.1	145	2.1
Total	2452	46.6	1965	40.7

K.4. Beispiel POL6 in 5 Variablen

	PFZT - Böge		DIFZT		PFZT - Collins	
Step	Time	Cells	Time	Cells	Time	Cells
4 Variables						
Resultion	337	6.1	282	7.7	1807	34.5
Basis	3643	77.0	3617	77.0	3644	77.0
3 Variables						
Resultion	846	14.3	698	17.0	2655	48.4
Basis	7166	151.6	7159	151.6	7169	151.6
2 Variables						
Resultion	2081	45.9	855	17.1	3406	74.8
Basis	2307	44.8	2291	44.8	2306	44.8
1 Variable						
Resultion	372	6.3	231	5.8	371	6.3
Basis	453	7.4	463	7.4	450	7.4
Total	19789	361.3	18189	336.1	24394	452.7

K.5. Beispiel POL9 in 7 Variablen

	PFZT - Böge		DIFZT		PFZT - Collins	
Step	Time	Cells	Time	Cells	Time	Cells
6 Variables						
Resultion	121	1.9	65	2.0	343	7.0
Basis	804	21.1	794	21.0	803	21.1
5 Variables						
Resultion	156	2.7	151	3.4	8450	294.6
Basis	7880	282.8	7850	282.8	7923	293.4
4 Variables						
Resultion	1167	29.1	2097	56.6	218840	8074.1
Basis	118591	4321.5	118369	4321.5	118640	4321.4
3 Variables						
Resultion	567577	18845.6	406578	15845.3	585316	19356.4
Basis	53311	1339.0	53209	1323.2	53347	1338.8
2 Variables						
Resultion			349843	8494.6		
Basis			1242170	34138.9		
1 Variable						
Resultion						
Basis						
Total						

In den beiden Verfahren 'PFZT - Böge' und 'PFZT - Collins' versagte das Programm zur Berechnung der Resultante, die Stufe '2 Variables' wurde daher mit diesen Verfahren nicht erreicht. Die Stufe '1 Variable' wurde daher auch mit 'DIFZT' nicht weiter berechnet.

K.6. Zusammenfassung

Für die angegebenen Beispiele unterscheiden sich die Rechenzeiten zwischen den Algorithmen nach [E.5., p. 36] und [H.4., p. 56] unwesentlich.

Die Einsparung nach [E.5., p. 36] macht sich hier aber gegenüber dem Algorithmus nach [F., p. 37] in einigen Fällen (etwa Beispiel [K.5., p. 84] , Zeile: 4 Variables) stark bemerkbar.

Zu beachten ist die Zeile '2 Variables' in Beispiel [K.5., p. 84] die auftretenden Resultanten liesen sich nicht berechnen.
Das hat seine Ursache darin, daß die verwendeten Resultantenalgorithmen die Resultante modulo verschiedener großer Primzahlen berechnen, und sodann die einzelnen modularen Resultanten nach dem Chinesischen Restsatz wieder zu der Resultante über den ganzen Zahlen zusammengesetzt wird. Die dabei verwendeten Abschätzungen werden aber bei wachsenden Koeffizienten und Graden der Polynome so groß, daß die von SAC2 verwendeten Primzahllisten nicht mehr ausreichen, und der Algorithmus stopt. Mit einem 'einfachen', d.h. nicht modularen Algorithmus, währe das Problem aber zulösen.

Insgesammt ist festzustellen, daß ein wesentlicher Teil der Rechenzeit für die Bestimmung der Dimension verwendet wird. Die Berechnung der psc's und der 'Buchführung' nimmt hingegen weniger Zeit in Anspruch.

Die praktisch gleichen Zeiten für die Basis Berechnung deuten daraufhin, daß die erzeugten 'Projektions/Resolutions' Polynommengen gleich sind. Die größere mathematische Genauigkeit der Gröbner-Basen Algorithmen schlägt hier also nicht zu Buche. (Aus den vorliegenden Computerausdrucken ist ersichtlich, daß die erzeugten Polynommengen tatsächlich gleich sind.)

Anhang 1. Verwendete Symbole und Bezeichnungen

←	Zuweisung (in Algorithmen)
▌	Ende von Algorithmen und Beweisen
⊓	Zeichen für Durchschnitt von Mengen
⊔	Zeichen für Vereinigung von Mengen
-	Zeichen für Differenz von Mengen
C	Zeichen für Inclusion von Mengen im Sinne von 'enthalten oder gleich'
=	Zeichen für Identität von Mengen
=	Zeichen für den 'Matchoperator' in Algorithmen
#	Zeichen für die Anzahl der Elemente von Mengen
ε	Zeichen für 'Element von'
Ø	Zeichen für die leere Menge

<==> 'genau dann wenn'

==> 'wenn dann'

Bei Doppelindizes werden die zweiten Indizes teilweise durch Klammern gekennzeichnet:

k(i) entspricht also dann k_i.

Anhang 2. Referenzen

2.1. Lehrbücher

[Vd Waerden]: V. d. Waerden, 1971,1967
Algebra I, II
Springer: Heidelberg 1971, 1967

[Samuel & Zariski]: P. Samuel, O. Zariski, 1958,1960
Commutative Algebra I, II
Van Nostrand: Princeton 1958, 1960

[Gröbner]: W. Gröbner, 1968,1970
Algebraische Geometrie I, II
Bibliographisches Institut: Mannheim 1968, 1970

[Knuth]: D. E. Knuth, 1973,1980
The art of computer programming I, II
Addision-Wesley: Reading USA 1973, 1980 (II 2nd ed)

[Buchberger & Collins & Loos]: B. Buchberger, G. Collins, R. Loos, 1982
Computer Algebra - Symbolic and Algebraic Computation
Springer: Wien, 1982

2.2. Spezielle Literatur zum Thema

[Vd Waerden 1931]: V. d. Waerden, 1931
Algebra II
Springer: Berlin 1931

[Buchberger 1970]: B. Buchberger, 1970
Ein algorithmisches Kriterium für die Lösbarkeit eines algebraischen Gleichungssystems.
Aequationes mathematicae, vol. 4, p. 374-383, 1970

[Collins 1974]: G. Collins, May 1974
Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition - Preliminary Report
Proceedings of the EUROSAM 74 Conference, SIGSAM Bulletin, Vol 8, No. 3 Aug 1974

[Collins 1975]: G. Collins, May 1975
Quantifier Elimination for Real Closed Fields by Cylindrical Algebraic Decomposition
Automatentheorie und Formale Sprachen, GI-Fachtagung, Mai 1975
Lecture Notes in Computer Science 33, 134-183, Springer-Verlag: Berlin Heidelberg New York 1975

[Buchberger 1976a]: B. Buchberger, August 1976
A theoretical basis for the reduction of polynomials to canonical forms.
ACM SIGSAM Bulletin, Vol. 10, No. 3, August 1976, pp. 19-29

[Schrader 1976]: R. Schrader, 1976
Zur Konstruktiven Idealtheorie (Diplomarbeit)
Mathematisches Institut II, Universität Karlsruhe 1976

[Buchberger 1976b]: B. Buchberger, November 1976
Some Properties of Gröbner-Bases for Polynom Ideals.
ACM SIGSAM Bulletin, Vol. 10, No. 4, November 1976, pp. 19-24

[Trinks 1978]: W. L. Trinks, 1978
Über Buchbergers Verfahren Systeme algebraischer Gleichungen zu lösen.
J. of Number Theory, vol. 10, pp. 475-488, 1978

[Buchberger 1979]: B. Buchberger, 1979
A criterion for detecting unnecessary reductions in the construction of Gröbner Bases (Dedicated to the 80-th. birthday of Prof. W. Gröbner)
Proc. of the Eurosam 79, Lecture Notes in Comp.Sci.72, 3-21, Springer Verlag, 1979

[Collins & Loos 1980]: G. E. Collins, R. G. Loos, 1980
ALDES and SAC-2 now available.
SAC2 - Symbolic and Algebraic Computation Version 2, a computer algebra system, ALDES - ALgorithm DEScription language
ACM SIGSAM Bulletin Vol. 14, No. 2, 1980

[Buchberger & Winkler 1981]: B. Buchberger, F. Winkler, September 1981
An algorithm for constructing canonical bases (Gröbner Bases) of polynomial ideals.
CAMP-Publ.Nr. 81-10.0, Johannes Kepler Universität Linz, September 1981

[Collins & Loos 1981]: G. E. Collins, R. G. Loos, November 1981
SAC2 Erweiterungen zur Quantorenelimination. Unveröffentlichte Programme.
G. E. Collins, Computer Science Department, University of Wisconsin, 1210 W. Dayton Street, Madison, WI 53706, U.S.A
R. G. K. Loos, Institut für Informatik I, Universität Karlsruhe, Zirkel 2, D-7500 Karlsruhe,

[Loos 1982a]: R. Loos, 1982
Computing in Algebraic Extentions
in [Buchberger & Collins & Loos] 173-187

[Loos 1982b]: R. Loos, 1982
Generalised Polynomial Remainder Sequences
in [Buchberger & Collins & Loos] 115-137

[Gebauer & Jagoda 1982a]: R. Gebauer, G. Jagoda, 1982
Einführung in SAC-2
Institut für Angewandte Mathematik, Universität Heidelberg, 1982

[Gebauer & Jagoda 1982b]: R. Gebauer, G. Jagoda, 1982
SAC-2 Kwic Index
Institut für Angewandte Mathematik, Universität Heidelberg, 1982

[Buchberger 1982]: B. Buchberger, 1982
A Note on the Complexity of Constructing Groebner-Bases
Proc. European Computer Algebra Conf., EUROCAL '83, London,
1983,
Lecture Notes in Computer Science 162, 137-145, Springer-Verlag:
Berlin Heidelberg New York Tokyo, 1983

[Gebauer & Kredel 1983a]: R. Gebauer, H. Kredel, April 1983
Common Distributive Polynomial System.
Institut für Angewandte Mathematik, Universität Heidelberg,
04/1983

[Böge 1983a]: W. Böge, April 1983
Korrekturen zu Müller und Collins. Manuskript.
Institut für Angewandte Mathematik, Universität Heidelberg,
April 1983

[Böge 1983b]: W. Böge, Mai 1983
Ein Beispiel zur Nichtlinearen Optimierung und
Quantorenelimination. Skript zu einer Vorlesung über
Nichtlineare Optimierung
Institut für Angewandte Mathematik, Universität Heidelberg, Mai
1983

[Gebauer & Kredel 1983b]: R. Gebauer, H. Kredel, Juni 1983
Distributive Integral Polynomial System.
Institut für Angewandte Mathematik, Universität Heidelberg,
06/1983

[Böge 1983c]: W. Böge, August 1983
Quantifierelimination for the Elementary Real Algebra.
Diskusionsvorlage zu einem Workshop über Quantorenelimination.
Institut für Angewandte Mathematik, Universität Heidelberg,
August 1983

[Gebauer & Kredel 1983c]: R. Gebauer, H. Kredel, August 1983
Distributive Rational Polynomial System.
Institut für Angewandte Mathematik, Universität Heidelberg,
08/1983

[Gebauer & Kredel 1983d]: R. Gebauer, H. Kredel, Dezember 1983
Distributive Arbitrary Domain Polynomial System.
Institut für Angewandte Mathematik, Universität Heidelberg, 12/1983

[Buchberger 1984]: B. Buchberger, 1984
Groebner Bases: An Algorithmic Method in Polynomial Ideal Theory.
Chapter 6 in : Recent Trends in Multidimensional Systems Theory.
D. Reidel Publishing Company, to appear 1984

[Böge 1984]: W. Böge, 1984
Private Mitteilung
Institut für Angewandte Mathematik, Universität Heidelberg, 1984

[Gebauer & Kredel 1984a]: R. Gebauer, H. Kredel, 1984
Programme zur Resultion von W. Böge
Institut für Angewandte Mathematik, Universität Heidelberg, 1984

[Winkler 1984]: F. Winkler, 1984
On the Complexity of the Groebner-Basis Algorithm over K[x,y,z]
Proc. EUROSAM 84, Cambridge July 1984
Lecture Notes in Computer Science 174, 184-194, Springer-Verlag: Berlin Heidelberg New York Tokyo, 1984
also available as: Techn. Rep. Nr. CAMP 83-25.0, Inst. für Math., Univ. Linz, Austria, 1983

[Möller & Mora 1984]: M. Möller, F. Mora, July 1984
Upper and lower bounds for the degree of Groebner Bases
Proc. EUROSAM 84, Campridge, July 1984
Lecture Notes in Computer Science 174, 172-183, Springer-Verlag: Berlin Heidelberg New York Tokyo, 1984