REELLE QUANTORENELIMINATION DURCH PARAMETRISCHES ZÄHLEN VON NULLSTELLEN

Andreas Dolzmann

24. November 1994

Eingereicht als Diplomarbeit am Lehrstuhl Prof. Dr. Weispfenning
an der Fakultät für Mathematik und Informatik der Universität Passau

ZUSAMMENFASSUNG

In dieser Arbeit wird ein Quantoreneliminationsverfahren für die Theorie der reellen Zahlen und seine Implementierung im Computeralgebrasystem *MAS* beschrieben. Das Verfahren basiert hauptsächlich auf dem Zählen reeller Nullstellen nulldimensionaler Ideale und der Berechnung von umfassenden Gröbnerbasen. Das Verfahren wurde von WEISPFENNING [Wei93a] eingeführt.

Inhaltsverzeichnis

KAPITEL 1

Grundlagen

1.1. Terme und Polynome

In dieser Arbeit wird ein Verfahren zur Quantorenelimination in der Theorie der reellen Zahlen und seine Implementierung im Computeralgebrasystem *MAS* vorgestellt. Wir lassen als Sprache für die Terme nur die Sprache der angeordneten Ringe zu. Die Division nehmen wir als partielle Operation aus. Als zusätzliche Konstanten stehen die ganzen Zahlen zur Verfügung. Als Relationen zum Aufbau atomarer Formeln lassen wir alle Relationen aus $\{<, \leqslant, =, \neq, \geqslant, >\}$ zu.

Wir werden im weiteren Terme und Polynome in einem Polynomring über $\mathbb{Z}$ miteinander identifizieren. Dabei sind dann die Variablen im Sinne der Logik die Unbestimmten des Polynomrings. Zusätzlich setzen wir voraus, daß atomare Formeln so normiert sind, daß auf der rechten Seite der Relation 0 steht.

Betrachte man z. B. die atomare Formel $X + 2Y^2 > 3XY$, so hat sie normalisiert die Form $X + 2Y^2 - 3XY > 0$. In einigen Schritten des Eliminationsverfahren werden wir dann die linke Seite der atomaren Formel als Polynom in $\mathbb{Z}[X, Y]$ betrachten.

1.2. Parametrisierte Probleme

Wir werden Polynome mit unbestimmten Koeffizienten benutzen, d. h., wir werden Aussagen machen, die für alle Polynome, deren Koeffizienten gewissen Beziehungen genügen, gelten. Solche Polynome nennen wir auch *parametrisiert*. Wir werden sie als Polynome im Polynomring

$$R[U_1, \ldots, U_m][X_1, \ldots, X_n] \quad \text{bzw.} \quad R(U_1, \ldots, U_m)[X_1, \ldots, X_n]$$

repräsentieren, wobei R der Grundring ist. Die Problematik dabei ist, daß man sich nicht direkt auf die Koeffizienten beziehen darf. Während Additionen, Subtraktionen und Multiplikationen in dem entsprechenden Polynomring durchzuführen sind, ist die Division durch Parameter nur dann erlaubt, falls sie ungleich Null sind. Ist dies nicht durch die Voraussetzungen gegeben, muß man eine Fallunterscheidung nach dem Verschwinden des Parameters machen.

Definition 1.1.
Seien R, S Ringe, $R[\underline{U}][\underline{X}] := R[U_1, \ldots, U_m][X_1, \ldots, X_n]$ multivariater Polynomring über R. Jeder Ringhomomorphismus $\varphi : R[\underline{U}] \longrightarrow S$ bestimmt eindeutig einen Ringhomomorphismus $\psi : R[\underline{U}][\underline{X}] \longrightarrow S[\underline{X}]$ mit $\psi|_{R[\underline{U}]} = \varphi$ und $\psi(X_i) = X_i$. Einen solchen Homomorphismus ψ nennen wir *Spezialisierung* der Parameter $U_1, \ldots, U_m$.

In den meisten Fällen dieser Arbeit wird S ein Erweiterungskörper von R und der Homomorphismus φ eine Erweiterung der natürlichen Einbettung von R in S sein. Damit ist φ eindeutig durch die Bilder der Unbestimmten U_i festgelegt. Somit kann eine solche Spezialisierung durch die Angabe eines Tupels $(a_1, \ldots, a_m)$ festgelegt werden. Mit der Spezialisierung $\underline{a} \in S^m$ referieren wir auf den eindeutig bestimmten Ringhomomorphismus $\psi : R[\underline{U}][\underline{X}] \longrightarrow S[\underline{X}]$ mit

- $\psi|_R = \mathrm{id}$,
- $\psi(X_i) = X_i$,
- $\psi(U_i) = a_i$.

Bemerkung 1.2.
Für ein Polynom $f \in R[\underline{U}]$ und eine Spezialisierung $\underline{a} \in S$ bezeichne $f(\underline{a})$ die Auswertung von f an der Stelle $\underline{a}$.

In den praktischen Anwendungen in dieser Arbeit wird der Ring R entweder $\mathbb{Z}$ oder $\mathbb{Q}$ sein, und der Körper S wird meistens $\mathbb{R}$ oder $\mathbb{C}$ sein.

Anwendungen von Spezialisierung auf ein Polynom f bzw. einer Menge von Polynomen F notieren wir als $f(\underline{a}, \underline{X})$ bzw. $F(\underline{a}, \underline{X})$ oder als $\sigma_{\underline{a}}(f)$ bzw. $\sigma_{\underline{a}}(F)$. Dabei verwenden wir die erste Schreibweise insbesondere dann, wenn die Auswertung eines Polynoms bzw. einer Polynommenge unter einer Spezialisierung im Vordergrund steht. Ist nur die Spezialisierung der Parameter relevant, benutzen wir die zweite Notation.

Ist es offensichtlich, daß durch Betrachtung von parametrisierten Polynomen keine Probleme auftreten, so werden wir uns die Freiheit nehmen, die Aussagen für „normale“ Polynome im Polynomring $\mathbb{R}[\underline{X}]$ zu beweisen.

1.3. Umfassende Gröbnerbasen und Gröbnersysteme

Im folgenden wird die Theorie der umfassenden Gröbnerbasen, so weit sie in dieser Arbeit gebraucht wird, kurz vorgestellt. Dabei werden wir auf strengere Formalisierungen und alle Beweise verzichten und nur einen Überblick geben. Näheres ist in [Wei92] zu finden.

Definition 1.3.
Sei $F \subseteq \mathbb{Q}[U_1, \ldots U_m][X_1, \ldots X_n]$. Eine endliche Menge $G \subseteq \mathrm{Id}(F) \subseteq \mathbb{Q}[\underline{U}][\underline{X}]$ heißt *umfassende Gröbnerbasis* falls für jede Spezialisierung $\underline{a} \in K^m$ in einen Oberkörper $K \supseteq \mathbb{Q}$ die Menge $\sigma_{\underline{a}}(G)$ eine Gröbnerbasis für $\mathrm{Id}(\sigma_{\underline{a}}(F))$ ist.

Das zentrale Problem bei der Berechnung von umfassenden Gröbnerbasen ist die Tatsache, daß Koeffizienten unter Spezialisierungen verschwinden können. Insbesondere ist die Bestimmung des Headterms eines parametrisierten Polynoms i. a. nicht möglich. Das folgende Beispiel zeigt, daß die Gröbnerbaseneigenschaft unter Spezialisierungen nicht erhalten bleibt.

Beispiel 1.4.
Sei $G := \{X + 1, UY + X\} \subseteq \mathbb{Q}[U, X, Y]$. Dann ist G nach dem ersten Buchbergerkriterium Gröbnerbasis bezüglich jeder Termordnung „$<$“ mit $X < Y$. Die Menge G bleibt eine Gröbnerbasis im Polynomring $\mathbb{Q}(U)[X, Y]$ (bei Anwendung der natürlichen Bijektion). Betrachtet man jedoch U als Parameter, so ist zwar G Gröbnerbasis für die Spezialisierung $U = 1$ aber nicht für die Spezialisierung $U = 0$, da $1 \in \mathrm{Id}(\{X + 1, X\})$ aber 1 sich nicht bezüglich G auf 0 reduziert.

Zentrale Lösungsidee bei der Berechnung von umfassenden Gröbnerbasen mit Hilfe von *Gröbnersystemen* ist es, eine Fallunterscheidung über das Verschwinden der Koeffizienten zu machen. Diese Fallunterscheidung wird zur Bestimmung der höchsten

Koeffizienten benutzt. Die Arithmetik ist jedoch die des Polynomringes $\mathbb{Q}(U)[X]$, d. h., bei Berechnungen werden alle Terme der Polynome berücksichtigt, also auch die, deren Koeffizienten in der Fallunterscheidung verschwinden.

Definition 1.5.
Ein Paar (R, G) von Polynommengen $R, G \subseteq \mathbb{Q}[U]$ heißt *Bedingung*. Eine Bedingung heißt von einer Spezialisierung $\underline{a}$ *erfüllt*, falls die folgende Konjunktion gilt.

$$\Big(\bigwedge_{p \in R} p(\underline{a}) \neq 0 \Big) \wedge \Big(\bigwedge_{p \in G} p(\underline{a}) = 0 \Big)$$

Definition 1.6.
Ein *Gröbnersystem* über einer initialen Bedingung (R_I, G_I) für eine Polynommenge F ist eine endliche Menge von Paaren

$$S := \{((R_\ell, G_\ell), P_\ell)\},$$

so daß für jede Spezialisierung $\underline{a}$, die die initiale Bedingung (R_I, G_I) erfüllt, genau ein Element $((R_\ell, G_\ell), P_\ell) \in S$ existiert, so daß

(1) (R_ℓ, G_ℓ) von $\underline{a}$ erfüllt ist.
(2) $\sigma_{\underline{a}}(P)$ Gröbnerbasis für $\mathrm{Id}(\sigma_{\underline{a}}(F))$ ist.
(3) $\underline{a}$ eindeutig den Headterm von jedem $p \in P_\ell$ festlegt.

Definition 1.7.
(1) Unter einem Gröbnersystem verstehen wir ein Gröbnersystem über $(\emptyset, \emptyset)$.
(2) Jedes Element aus S bezeichnen wir als *Ast* vom Gröbnersystem S.
(3) Ist jedes P_ℓ unter der Spezialisierung $\underline{a}$ eine reduzierte Gröbnerbasis, so heißt S *reduziertes Gröbnersystem*.

Bemerkung 1.8.
Die Vereinigung der Polynommengen aller Fälle

$$\bigcup \{ P \mid ((R, G), P) \in S \}$$

eines Gröbnersystems S für $\mathrm{Id}(F)$ bildet eine umfassende Gröbnerbasis für $\mathrm{Id}(F)$. Umgekehrt kann man aus einer gegebenen umfassenden Gröbnerbasis auch ein Gröbnersystem berechnen.

Definition 1.9.
Entfernt man alle Monome in allen Polynomen aller Gröbnerbasen eines Gröbnersystems, die unter den jeweiligen Bedingungen der Fallunterscheidung des Gröbnersystems verschwinden, so nennt man das entstehende Gröbnersystem *grün gefärbt*.

Bemerkung 1.10.
In jedem Zweig eines Gröbnersystems ist die Dimension des Ideales $\mathrm{Id}(\sigma(F))$ für jede Spezialisierung, die die entsprechende Bedingung erfüllt, gleich. Sie kann algorithmisch bestimmt werden. Gleiches gilt für die Menge der unabhängigen Variablenmengen.

KAPITEL 2

Parametrisches Zählen reeller Nullstellen

In den Arbeiten von Becker und Wörmann [BW91] bzw. Pedersen, Roy und Szpirglas [PRS93] wurde ein Verfahren beschrieben, um die Anzahl der reellen Nullstellen eines nulldimensionalen Ideales unter der Berücksichtigung einer Nebenbedingung zu ermitteln. Dieses Verfahren läßt sich mit einer Methode, die von Ben-Or, Kozen und Reif [BKR86] eingeführt wurde, auf eine endliche Menge von Nebenbedingungen verallgemeinern.

Neben den Originalarbeiten wurde für die folgenden beiden Abschnitte insbesondere die Diplomarbeit von Lippold [Lip93] benutzt. Dort findet man auch ausführliche Beweise der hier zitierten Sätze. Weitergehende algebraische Grundlagen finden sich auch in [BW93b].

2.1. Zählen reeller Nullstellen unter einer Nebenbedingung

Definition 2.1.
Sei $P \subseteq K[X_1, \ldots, X_n]$ und L Oberkörper von K. Dann Dann heißt $\underline{\alpha} \in L^n$ *Nullstelle* von P, falls $f(\underline{\alpha}) = 0$ für jedes $f \in P$.

Die Menge

$$V_L(P) := \{\, \underline{\alpha} \in L^n \mid \underline{\alpha} \text{ Nullstelle von } P \,\}$$

heißt die *Varietät* von P in L.

Ist $L = \mathbb{R}$ so sprechen wir von *reellen Nullstellen* einer Menge P.

Bemerkung 2.2.
Es ist leicht zu sehen, daß eine Nullstelle von P auch eine Nullstelle von $\mathrm{Id}(P)$ und umgekehrt ist. Insbesondere gilt $|V_L(P)| = |V_L(\mathrm{Id}(P))|$.

Satz 2.3 (Pedersen, Roy, Szpirglas).
Sei K angeordneter Körper, R reell abgeschlossener Erweiterungskörper von K und C der algebraische Abschluß von R. Sei I nulldimensionales Ideal in $K[\underline{X}]$ und $B := \{v_1, \ldots, v_b\}$ Basis des K-Vektorraums $K[\underline{X}]/I$. Für $h \in K[\underline{X}]$ sei

$$Q_h := (\mathrm{spur}(m_{h \cdot v_i \cdot v_j}))_{1 \leqslant i,j \leqslant b},$$

wobei

$$m_f : K[\underline{X}]/I \to K[\underline{X}]/I \quad \textit{definiert durch} \quad g + I \mapsto (f \cdot g) + I$$

die Multiplikation mit f im Restklassenring $K[\underline{X}]/I$ ist.

Dann gilt:

(1) Q_h *ist eine reelle, symmetrische, quadratische Matrix*
(2) $\operatorname{rang}(Q_h) = |\{\, \alpha \in V_C(I) \mid h(\alpha) \neq 0 \,\}|$
(3) $\operatorname{sig}(Q_h) = |\{\, \alpha \in V_R(I) \mid h(\alpha) > 0 \,\}| - |\{\, \alpha \in V_R(I) \mid h(\alpha) < 0 \,\}|$ □

Bemerkung 2.4.
Die Eigenwerte einer reellen, symmetrischen $b \times b$ Matrix Q sind sämtlich reell. Insbesondere hat also das zugehörige charakteristische Polynom nur reelle Nullstellen, die die Eigenwerte der Matrix Q sind. Rang und Signatur lassen sich in diesem Fall mit der Zeichenregel von Descartes bestimmen. Sei π_Q die Anzahl der positiven Eigenwerte von Q mit Vielfachheiten gezählt, und ν_Q die Anzahl der negativen Eigenwerte von Q ebenfalls mit ihren Vielfachheiten gezählt. Dann gilt:

(1) $\operatorname{rang}(Q) = \pi_Q + \nu_Q$
(2) $\operatorname{sig}(Q) = \pi_Q - \nu_Q$

Korollar 2.5.
Aus obigem Satz 2.3 *ergeben sich folgende Spezialfälle:*

(1) $\operatorname{sig}(Q_{h^2}) = |\{\, \alpha \in V_{\mathbb{R}}(I) \mid h(\alpha) \neq 0 \,\}|$
(2) $\operatorname{sig}(Q_1) = |V_{\mathbb{R}}(I)|$ □

Bemerkung 2.6.
Mit diesen Ergebnissen lassen sich die Nullstellen von I unter einer gegebenen Nebenbedingung h zählen, indem man das folgende Gleichungssystem löst.

$$\begin{pmatrix} 1 & 1 & 1 \\ -1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} |\{\, \alpha \in V_{\mathbb{R}}(I) \mid h(\alpha) < 0 \,\}| \\ |\{\, \alpha \in V_{\mathbb{R}}(I) \mid h(\alpha) = 0 \,\}| \\ |\{\, \alpha \in V_{\mathbb{R}}(I) \mid h(\alpha) > 0 \,\}| \end{pmatrix} = \begin{pmatrix} \operatorname{sig}(Q_1) \\ \operatorname{sig}(Q_h) \\ \operatorname{sig}(Q_{h^2}) \end{pmatrix}$$

Dieses Gleichungssystem ist regulär und hat somit eine eindeutig bestimmte Lösung, da

$$\begin{vmatrix} 1 & 1 & 1 \\ -1 & 0 & 1 \\ 1 & 0 & 1 \end{vmatrix} = 2$$

gilt.

Sind überhaupt keine Nebenbedingungen gegeben, so erhält man die Anzahl der reellen Nullstellen aus der Berechnung der Signatur der Matrix Q_1.

Aufbauend auf obigen Satz 2.3 läßt sich ein Verfahren zum Zählen reeller Nullstellen unter einer Nebenbedingung implementieren. Eine solche Implementierung wurde von LIPPOLD [Lip93] im Computeralgebrasystem *MAS* durchgeführt. Die Arithmetik im Restklassenring $K[\underline{X}]/I$ wird in diesem Programm mittels einer auf Gröbnerbasen basierenden Technik implementiert. Der eigentliche Algorithmus läuft nach folgendem Schema ab:

Algorithmus 2.7.
(1) *Berechnung einer reduzierten Gröbnerbasis G von I.*
(2) *Berechnung der reduzierten Terme*

$$\left\{ t + I \;\middle|\; t \text{ Term, ex. kein } t' \in \operatorname{HT}(G) : t' \mid t \right\},$$

die eine Basis des K-Vektorraums $K[\underline{X}]/I$ bilden.
(3) *Berechnung der kombinierten Strukturkonstanten.*
(4) *Berechnung der Signaturen.*
(5) *Lösen des Gleichungssystems.*

2.2. Zählen reeller Nullstellen unter mehreren Nebenbedingungen

Das in dem letzten Abschnitt geschilderte Verfahren zum Zählen von reellen Nullstellen unter einer Nebenbedingung läßt sich auch auf die Problemstellung des Zählens reeller Nullstellen unter endlich vielen Nebenbedingungen übertragen. Das dazu benutzte kombinatorische Argument wurde von Ben-Or, Kozen und Reif [BKR86] eingeführt.

Definition 2.8.

(1) Sei A die $n_1 \times n_2$ Matrix und B eine $m_1 \times m_2$ Matrix. Dann ist das rechte Tensor-Produkt $A \otimes B$ eine $n_1 \cdot m_1 \times n_2 \cdot m_2$ Matrix, die entsteht, wenn man jeden Eintrag a_{ij} durch $a_{ij} \cdot B$ ersetzt.

(2) Die m-te Tensor-Potenz $A^{(m)}$ einer Matrix A definiert man rekursiv durch $A^{(1)} := A$ und $A^{(m)} := A \otimes A^{(m-1)}$.

Proposition 2.9.
Sei A eine $m \times m$ Matrix und B eine $n \times n$ Matrix. Dann gilt

$$\det(A \otimes B) = (\det A)^n \cdot (\det B)^m.$$

Insbesondere gilt also

$$det(A^{(p)}) = (\det(A))^{p'}$$

für ein geeignetes $p' \in \mathbb{N}$.

Beweis. Siehe Lancaster [LT85] (Seite 408 ff.). □

Im folgenden sei I nulldimensionales Ideal in $K[\underline{X}]$ und $h_1, \ldots, h_s \in K[\underline{X}]$. Mit $\sigma(f)$ bezeichnen wir die Signatur der Matrix Q_f, die bereits im letzten Abschnitt eingeführt wurde.

Satz 2.10.
Sei $Z_s := (z_1, \ldots, z_{3^s})^t$ der Spaltenvektor mit Einträgen

$$z_i := \left| \left\{ \alpha \in V_{\mathbb{R}}(I) \;\middle|\; \bigwedge_{j=1}^{s} \operatorname{sign}(h_j(\alpha)) = \varrho_{ij} \right\} \right|$$

wobei $(\varrho_{i1}, \ldots, \varrho_{is})$ das bezüglich der invers lexikographischen Ordnung auf $\mathbb{Z}^s$ i-te Element der Menge $\{-1, 0, 1\}^s$ ist, sei $S_s := (\sigma_1, \ldots, \sigma_{3^s})^t$ der Spaltenvektor mit Einträgen $\sigma_i := \sigma(h_1^{e_1}, \ldots, h_s^{e_s})$ wobei $(e_1, \ldots, e_s)$ das bezüglich der invers lexikographischen Ordnung auf $\mathbb{N}^s$ i-te Element der Menge $\{0, 1, 2\}^s$ ist und

$$M := \begin{pmatrix} 1 & 1 & 1 \\ -1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}.$$

Dann gilt

$$M^{(s)} \cdot Z_s = S_s. \quad \square$$

Bemerkung 2.11.
Aus $\det(I) = 2$ folgt $\det(M^{(s)}) = 2^p$ für ein geeignetes $p \in \mathbb{N}$. Somit hat die Gleichung aus Satz 2.10 eine eindeutige Lösung Z_s. Aus dieser läßt sich die Anzahl der reellen Nullstellen unter den gegebenen Nebenbedingungen ablesen.

2.3. Existenz reeller Nullstellen parametrischer Ideale

Mit dem letzten Satz hat man prinzipiell auch im parametrischen Fall die Möglichkeit, die Nullstellen eines Ideals unter Nebenbedingungen zu zählen. Jedoch hängt der Vektor S_s i. a. von den Parametern ab. So müßte man also für jeden in Frage kommenden Vektor S_s ein Gleichungssystem der Größe 3^s lösen, was praktisch kaum durchführbar ist.

Verzichtet man auf die gleichzeitige Berechnung der Anzahl reeller Nullstellen unter allen Kombinationen von Nebenbedingungen $h_i > 0$, $h_i = 0$ und $h_i < 0$ und beschränkt sich auf eine Kombination von Nebenbedingungen mit den Relationen „=“ und „$\neq$“, so muß man das Gleichungssystem nicht lösen.

Grundlage dieses Ergebnisses ist das entsprechende Resultat für endlich viele Nebenbedingungen der Form $f_j > 0$. Es findet sich in der Arbeit von Becker und Wörmann [BW91] und in der Arbeit von Weispfenning [Wei93a]. Während in der zuletzt genannten Arbeit das Ergebnis durch strukturelle Überlegung aus dem Satz 2.10 abgeleitet wird, wählen wir hier einen direkten Zugang.

Im folgenden soll eine Methode vorgestellt werden, die es erlaubt festzustellen, ob ein nulldimensionales Ideal $I = \mathrm{Id}(\{f_i\})$ mindestens eine reelle Nullstellen α besitzt, die

$$\bigwedge_{i=1}^{s} h_i(\alpha) > 0 \wedge \bigwedge_{i=s+1}^{t} h_i(\alpha) \neq 0$$

erfüllt. In diesem Fall muß das oben erwähnte Gleichungssystem nicht gelöst werden. Wir werden dabei als Grundkörper $\mathbb{Q}$, als reell abgeschlossenen Erweiterungskörper $\mathbb{R}$ und als algebraisch abgeschlossenen Erweiterungskörper $\mathbb{C}$ voraussetzen.

Um Produkte von Potenzen von Polynomen kurz zu notieren, verwenden wir im weiteren folgende Notationen. Für $e \in \mathbb{N}^t$ und $v \in \mathbb{R}$ sei

$$h^e := \prod_{i=1}^{t} h_i^{e_i} \quad \text{und} \quad h^e(v) := \prod_{i=1}^{t} (h_i(v))^{e_i}.$$

Lemma 2.12.
Sei $s < t \in \mathbb{N}$, $f_1, \ldots, f_t \in \mathbb{R}[\underline{X}]$, und

$$E_t := \underbrace{\{1,2\} \times \cdots \times \{1,2\}}_{s \text{ mal}} \times \underbrace{\{2\} \times \cdots \times \{2\}}_{t-s \text{ mal}}.$$

Dann gilt für jedes $V \subseteq \mathbb{R}^n$

$$\sum_{e \in E_t} \left(\left| \left\{ v \in V \;\middle|\; \prod_{i=1}^{t} h_i^{e_i}(v) > 0 \right\} \right| - \left| \left\{ v \in V \;\middle|\; \prod_{i=1}^{t} h_i^{e_i}(v) < 0 \right\} \right| \right) =$$
$$2^s \cdot \left| \left\{ v \in V \;\middle|\; \bigwedge_{i=1}^{s} h_i(v) > 0 \wedge \bigwedge_{i=s+1}^{t} h_i(v) \neq 0 \right\} \right|.$$

Beweis. Sei ϱ_i gleich „$>$“ für $1 \leqslant i \leqslant s$ und ϱ_i gleich „$\neq$“ für $s+1 \leqslant i \leqslant t$.

Beweis über Induktion nach t. Für $t = 1$ und $s = 1$, d. h. ϱ_1 gleich „$>$“ folgt

$$\begin{aligned}\sum_{e\in E_1} (|\{\, v \in V \mid h^e(v) > 0 \,\}| - |\{\, v \in V \mid h^e(v) < 0 \,\}|) \\ &= |\{\, v \in V \mid h_1(v) > 0 \,\}| - |\{\, v \in V \mid h_1(v) < 0 \,\}| + \\ &\quad |\{\, v \in V \mid h_1^2(v) > 0 \,\}| - |\{\, v \in V \mid h_1^2(v) < 0 \,\}| \\ &= |\{\, v \in V \mid h_1(v) > 0 \,\}| - |\{\, v \in V \mid h_1(v) < 0 \,\}| + \\ &\quad |\{\, v \in V \mid h_1(v) \neq 0 \,\}| \\ &= |\{\, v \in V \mid h_1(v) > 0 \,\}| - |\{\, v \in V \mid h_1(v) < 0 \,\}| + \\ &\quad |\{\, v \in V \mid h_1(v) > 0 \,\}| + |\{\, v \in V \mid h_1(v) < 0 \,\}| \\ &= 2^s \cdot |\{\, v \in V \mid h_1(v) > 0 \,\}|.\end{aligned}$$

Dabei benutzt man, daß $h_i(v)^2 > 0$ äquivalent zu $h_i(v) \neq 0$ ist.

Für $t = 1$ und $s = 0$, d. h. ϱ_1 gleich „$\neq$“ folgt

$$\begin{aligned}\sum_{e\in E_1} (|\{\, v \in V \mid h^e(v) > 0 \,\}| - |\{\, v \in V \mid h^e(v) < 0 \,\}|) \\ &= |\{\, v \in V \mid h_1^2(v) > 0 \,\}| - |\{\, v \in V \mid h_1^2(v) < 0 \,\}| \\ &= 2^s \cdot |\{\, v \in V \mid h_1(v) \neq 0 \,\}| \quad .\end{aligned}$$

Im Induktionsschritt $t > 1$ führt man die Behauptung dadurch auf die Induktionsannahme zurück, daß man zu jedem durch $e \in E_t$ gegebenen Summanden die zu $(e_1, \ldots, e_{t-1}) \in E_{t-1}$ gehörenden Summanden betrachtet.

Im Fall $s < t$, d. h. ϱ_t gleich „$>$“ gilt

$$\begin{aligned}&\sum_{e\in E_t} (|\{\, v \in V \mid h^e(v) > 0 \,\}| - |\{\, v \in V \mid h^e(v) < 0 \,\}|) = \\ &\sum_{e\in E_{t-1}} (|\{\, v \in V \mid h^e(v) \cdot h_t(v) > 0 \,\}| - |\{\, v \in V \mid h^e(v) \cdot h_t(v) < 0 \,\}| + \\ &\qquad |\{\, v \in V \mid h^e(v) \cdot h_t(v)^2 > 0 \,\}| - |\{\, v \in V \mid h^e(v) \cdot h_t(v)^2 < 0 \,\}|).\end{aligned}$$

Für jedes $e \in E_{t-1}$ gilt

$$\begin{aligned}&|\{\, v \in V \mid h^e(v) \cdot h_t(v) > 0 \,\}| - |\{\, v \in V \mid h^e(v) \cdot h_t(v) < 0 \,\}| + \\ &|\{\, v \in V \mid h^e(v) \cdot h_t(v)^2 > 0 \,\}| - |\{\, v \in V \mid h^e(v) \cdot h_t(v)^2 < 0 \,\}| \\ &= |\{\, v \in V \mid h^e(v) > 0 \wedge h_t(v) > 0 \,\}| + |\{\, v \in V \mid h^e(v) < 0 \wedge h_t(v) < 0 \,\}| - \\ &\quad |\{\, v \in V \mid h^e(v) > 0 \wedge h_t(v) < 0 \,\}| - |\{\, v \in V \mid h^e(v) < 0 \wedge h_t(v) > 0 \,\}| + \\ &\quad |\{\, v \in V \mid h^e(v) > 0 \wedge h_t(v) \neq 0 \,\}| - |\{\, v \in V \mid h^e(v) < 0 \wedge h_t(v) \neq 0 \,\}| \\ &= |\{\, v \in V \mid h^e(v) > 0 \wedge h_t(v) > 0 \,\}| + |\{\, v \in V \mid h^e(v) < 0 \wedge h_t(v) < 0 \,\}| - \\ &\quad |\{\, v \in V \mid h^e(v) > 0 \wedge h_t(v) < 0 \,\}| - |\{\, v \in V \mid h^e(v) < 0 \wedge h_t(v) > 0 \,\}| + \\ &\quad |\{\, v \in V \mid h^e(v) > 0 \wedge h_t(v) > 0 \,\}| + |\{\, v \in V \mid h^e(v) > 0 \wedge h_t(v) < 0 \,\}| - \\ &\quad |\{\, v \in V \mid h^e(v) < 0 \wedge h_t(v) > 0 \,\}| - |\{\, v \in V \mid h^e(v) < 0 \wedge h_t(v) < 0 \,\}| \\ &= 2 \cdot (|\{\, v \in V \mid h^e(v) > 0 \wedge h_t(v) > 0 \,\}| - |\{\, v \in V \mid h^e(v) < 0 \wedge h_t(v) > 0 \,\}|) \quad .\end{aligned}$$

Definiert man $W := \{\, v \in V \mid h_t(v) > 0 \,\}$, so gilt

$$\begin{aligned}\sum_{e\in E_t} (|\{\, v \in V \mid h^e(v) > 0 \,\}| - |\{\, v \in V \mid h^e(v) < 0 \,\}|) = \\ 2 \cdot \sum_{e\in E_{t-1}} (|\{\, v \in W \mid h^e(v) > 0 \,\}| - |\{\, v \in W \mid h^e(v) < 0 \,\}|).\end{aligned}$$

Mit der Induktionsannahme ist die letzte Summe gleich

$$2 \cdot 2^{s-1} \left| \left\{ v \in W \;\middle|\; \bigwedge_{i=1}^{t-1} h_i(v) \; \varrho_i \; 0 \right\} \right| = 2^s \cdot \left| \left\{ v \in V \;\middle|\; \bigwedge_{i=1}^{t} h_i(v) \; \varrho_i \; 0 \right\} \right|$$

Im Fall $t > 1$ und $s < t$, d. h. ϱ_t gleich „$\neq$“ gilt

$$\sum_{e \in E_t} (|\{ v \in V \mid h^e(v) > 0 \}| - |\{ v \in V \mid h^e(v) < 0 \}|) =$$
$$\sum_{e \in E_{t-1}} (|\{ v \in V \mid h^e(v) \cdot h_t(v)^2 > 0 \}| - |\{ v \in V \mid h^e(v) \cdot h_t(v)^2 < 0 \}|).$$

Für jedes $e \in E_{t-1}$ gilt

$$\begin{aligned} &|\{ v \in V \mid h^e(v) \cdot h_t(v)^2 > 0 \}| - |\{ v \in V \mid h^e(v) \cdot h_t(v)^2 < 0 \}| \\ &= |\{ v \in V \mid h^e(v) > 0 \wedge h_t(v) \neq 0 \}| - |\{ v \in V \mid h^e(v) < 0 \wedge h_t(v) \neq 0 \}|. \end{aligned}$$

Definiert man $W := \{ v \in V \mid h_t(v) \neq 0 \}$, so gilt

$$\sum_{e \in E_t} (|\{ v \in V \mid h^e(v) > 0 \}| - |\{ v \in V \mid h^e(v) < 0 \}|) =$$
$$\sum_{e \in E_{t-1}} (|\{ v \in W \mid h^e(v) > 0 \}| - |\{ v \in W \mid h^e(v) < 0 \}|).$$

Mit der Induktionsannahme ist die letzte Summe gleich

$$2^s \cdot \left| \left\{ v \in W \;\middle|\; \bigwedge_{i=1}^{t-1} h_i(v) \; \varrho_i \; 0 \right\} \right| = 2^s \cdot \left| \left\{ v \in V \;\middle|\; \bigwedge_{i=1}^{t} h_i(v) \; \varrho_i \; 0 \right\} \right|. \quad \square$$

Satz 2.13.
Unter den Voraussetzungen des Lemmas 2.12 *gilt*

$$\sum_{e \in E_t} \operatorname{sig}\left(Q_{\prod_{i=1}^{t} h_i^{e_i}}\right) = 2^s \cdot \left| \left\{ \alpha \in V_{\mathbb{R}}(I) \;\middle|\; \bigwedge_{i=1}^{s} h_i(\alpha) > 0 \wedge \bigwedge_{i=s+1}^{t} h_i(\alpha) \neq 0 \right\} \right|.$$

Beweis. Wähle in Lemma 2.12 $V := V_{\mathbb{R}}(I)$. Mit dem Satz von Pedersen, Roy und Szpirglas (Satz 2.3) gilt dann

$$\begin{aligned} \sum_{e \in E_t} \operatorname{sig}(Q_{h^e}) &= \sum_{e \in E_t} \left(|\{ \alpha \in V_{\mathbb{R}}(I) \mid h^e(\alpha) > 0 \}| - |\{ \alpha \in V_{\mathbb{R}(I)} \mid h^e(\alpha) < 0 \}| \right) \\ &= 2^s \cdot \left| \left\{ \alpha \in V_{\mathbb{R}}(I) \;\middle|\; \bigwedge_{i=1}^{s} h_i(\alpha) > 0 \wedge \bigwedge_{i=s+1}^{t} h_i(\alpha) \neq 0 \right\} \right|. \quad \square \end{aligned}$$

Mit diesem Resultat ist die Anzahl der Nullstellen exakt bestimmt. Für die Quantorenelimination ist jedoch nur von Bedeutung, ob ein Ideal überhaupt reelle Nullstellen besitzt.

Proposition 2.14.
Bezeichne Z die Anzahl der Nullstellen des Ideals unter den Nebenbedingungen. Dann gilt

$$Z > 0 \Leftrightarrow \sum_{e \in E_t} \operatorname{sig}(Q_{h^e}) \neq 0 \Leftrightarrow \sum_{e \in E_t} \operatorname{sig}(Q_{h^e}) > 0 \Leftrightarrow \sum_{e \in E_t} \operatorname{sig}(Q_{h^e}) \geqslant 2^s$$

Beweis. Die Äquivalenzen folgen aus dem Satz 2.13 und daraus, daß die Anzahl der Nullstellen größer gleich 0 ist. $\square$

Wie oben bereits erwähnt wurde, kann die Signatur von Q_{h^e} als Differenz der Anzahlen der positiven und negativen Nullstellen des zugehörigen charakteristischen Polynoms mittels der Vorzeichenregel von Descartes berechnet werden.

Definition 2.15.
Sei $f \in \mathbb{R}[X]$. Dann definiert man den Typ $\tau(f)$ des Polynoms f als die Differenz der Anzahlen der positiven und negativen Nullstellen von f.

Bemerkung 2.16.
Mit der obigen Definition des Types eines Polynoms gilt

$$\operatorname{sig}(Q_h) = \tau(\chi_{Q_h}),$$

wobei χ_{Q_h} das charakteristische Polynom der Matrix Q_h bezeichnet.

Proposition 2.17.
Seien $f_1, \ldots, f_t \in \mathbb{R}[X]$. Dann gilt

$$\sum_{i=1}^{t} \tau(f_i) = \tau\Big(\prod_{i=1}^{t} f_i\Big)$$

Beweis. Jede Nullstelle eines der Polynome f_i ist Nullstelle des Produktes. Da ein Körper keine Nullteiler besitzt, muß umgekehrt jede Nullstelle des Produktes von einer Nullstelle einer der Faktoren herkommen. Da die Nullstellen mit Vielfachheiten gezählt werden, folgt hieraus die Gleichheit. □

Korollar 2.18.
Bezeichne χ_e das charakteristische Polynom von Q_{h^e}. Dann gilt

$$Z > 0 \Leftrightarrow \tau\Big(\prod_{e \in E_t} \chi_e\Big) \neq 0 \Leftrightarrow \tau\Big(\prod_{e \in E_t} \chi_e\Big) > 0 \Leftrightarrow \tau\Big(\prod_{e \in E_t} \chi_e\Big) \geqslant 2^s. \quad \square$$

Um zu entscheiden, ob ein Ideal reelle Nullstellen hat, die gewisse Nebenbedingungen erfüllen, muß man also die Bedingung $\tau(f) \neq 0$ in Abhängigkeit der Parameter charakterisieren.

KAPITEL 3

Typformeln

Ein zentraler Punkt im Eliminationsverfahren ist es, univariate Polynome zu charakterisieren, die genau so viele positive wie negative Nullstellen haben. Dabei müssen jedoch nur solche reellen Polynome betrachtet werden, die ausschließlich reelle Nullstellen besitzen.

Wird von der Anzahl der Nullstellen gesprochen, so ist damit die Anzahl der Nullstellen mit ihrer Vielfachheit gemeint. Somit hat ein univariates Polynom vom Grad d, das ausschließlich reelle Nullstellen hat, genau d reelle Nullstellen. Unter Nullstellen verstehen wir im folgenden immer reelle Nullstellen, sofern nicht explizit von komplexen Nullstellen gesprochen wird. Mit „echt komplexen“ Nullstellen bezeichnen wir Nullstellen aus $\mathbb{C} \setminus \mathbb{R}$.

Die Erzeugung von Typformeln wird in der Originalarbeit [Wei93a] nicht beschrieben. Die Idee, Typformeln mittels der Vorzeichenregel von Descartes, angewendet auf die Signums von Koeffiziententupel, zu berechnen geht ebenso auf Weispfenning zurück, wie wie die rekursive Zurückführung von Typformeln höheren Grades auf Typformeln niedrigeren Grades. Die effiziente Bestimmung von Typformeln stützt sich auf seine Idee, das Verhalten von Koeffizienten c_i und c_{i+2} mit $c_i \cdot c_{i+2} = -1$ möglichst weitgehend auszunutzen.

3.1. Grundlagen und Definition

Definition 3.1.
Sei $f \in \mathbb{R}[X]$. Dann bezeichne

(1) $\mathrm{Nst}_+(f)$ die Anzahl der positiven reellen Nullstellen von f, wobei diese mit ihren Vielfachheiten gezählt werden.
(2) $\mathrm{Nst}_-(f)$ die Anzahl der negativen reellen Nullstellen von f, wobei diese mit ihren Vielfachheiten gezählt werden.
(3) $\mathrm{Nst}_0(f)$ die Vielfachheit der Nullstelle 0 von f.
(4) $\mathrm{Nst}_{\mathbb{C}}(f)$ die Anzahl der komplexen Nullstellen mit Vielfachheiten von f, die nicht reell sind.

Bemerkung 3.2.
Mit den obigen Definitionen ist der Typ eines univariaten Polynoms (siehe Definition 2.15) gegeben durch

$$\tau(f) := \mathrm{Nst}_+(f) - \mathrm{Nst}_-(f).$$

Definition 3.3.
Sei $d > 0$ und $\chi = X^d + \sum_{i=0}^{d-1} c_i X^i \in \mathbb{Q}[c_0, \ldots, c_{d-1}][X]$ ein normiertes, univariates Polynom mit unbestimmten Koeffizienten. Sei $A \subseteq \mathbb{R}^d$ die Menge alle Spezialisie-

rungen, für die $\sigma_{\underline{a}}(\chi)$ nur reelle Nullstellen besitzt. Eine quantorenfreie Formel $T_d(\underline{c})$ mit Variablen $c_0, \ldots, c_{d-1}$ heißt *Typformel* für Polynome des Grades d (bezüglich des Typs 0), falls für jede Spezialisierung $\underline{a} \in A$

$$T_d(\underline{a}) \quad \text{gdw.} \quad \tau(\sigma_{\underline{a}}(\chi)) = 0$$

gilt. Eine Typformel $T_d(\underline{c})$ heißt *strikt*, falls

$$T_d(\underline{a}) \quad \text{gdw.} \quad \tau(\sigma_{\underline{a}}(\chi)) = 0 \wedge c_0(\underline{a}) \neq 0$$

gilt.

3.2. Existenz und Konstruktion

Ziel ist es, möglichst kurze Typformeln zu finden, da die Typformeln mit gewissen Substitutionen das Ergebnis der Quantorenelimination bilden. Zunächst ist es jedoch noch nicht einmal klar, ob es solche Formeln gibt bzw. diese automatisch bestimmt werden können. Einen Beweis, daß Typformeln für Polynome eines beliebigen Grades existieren und automatisch zu bestimmen sind, wird im folgenden auf der Grundlage der Vorzeichenregel von Descartes gegeben.

Definition 3.4.
Sei

$$f = \sum_{i=0}^{d} a_i X^i \quad \text{und} \quad \underline{a} := (a_0, \ldots, a_d),$$

dann heißt $\underline{a}$ das Koeffiziententupel von f. Mit

$$\overline{\underline{a}} := \overline{(a_0, a_1, a_2, \ldots, a_d)} := (a_0, -a_1, a_2, \ldots, (-1)^d \cdot a_d)$$

bezeichnen wir das Koeffiziententupel von $f(-X) = \sum_{i=0}^{d} (-1)^i a_i X^i$.

Im weiteren werden wir auch bei Aussagen über beliebigen Elementen aus $\mathbb{R}^{d+1}$ von Koeffiziententupeln und Koeffizienten sprechen.

Definition 3.5.
Sei $\underline{a} = (a_0, \ldots, a_d) \in \mathbb{R}^{d+1}$ und

$$\begin{aligned}\mathrm{Vzw}'(a_0, a_1, \ldots, a_d) :=& \\ &\{(i,j) \in \mathbb{N}^2 \mid 0 \leqslant i < j \leqslant d, a_i \cdot a_j < 0 \text{ und } a_k = 0 \text{ für } (i < k < j)\}\end{aligned}$$

dann ist die Anzahl der Vorzeichenwechsel von $\underline{a}$ definiert durch

$$\mathrm{Vzw}(a_0, a_1, \ldots, a_d) := |\mathrm{Vzw}'(a_0, a_1, \ldots, a_d)|.$$

Anschaulich gesagt, zählt man die Vorzeichenwechsel zwischen benachbarten Einträgen im Tupel, nachdem man eventuell vorhandene Einträge gleich Null gestrichen hat.

Bemerkung 3.6.
Für $0 < i$ mit $c_i \neq 0$ gilt

$$\mathrm{Vzw}'(a_0, \ldots, a_d) = \mathrm{Vzw}'(a_0, \ldots, a_i) \cup \mathrm{Vzw}'(a_i, \ldots, a_d)$$

und somit auch $\mathrm{Vzw}(a_0, \ldots, a_d) = \mathrm{Vzw}(a_0, \ldots, a_i) + \mathrm{Vzw}(a_i, \ldots, a_d)$.

Lemma 3.7.
Für $\underline{a} \in \mathbb{R}^{d+1}$ *gilt* $\mathrm{Vzw}(\underline{a}) < d + 1$. □

Satz 3.8 (Descartes 1637).
Für ein reelles Polynom $f = \sum_{i=0}^{d} a_i X^i$ *gilt*

$$\mathrm{Nst}_+(f) = \mathrm{Vzw}(a_0, \ldots, a_d) - 2k$$

für geeignetes $k \in \mathbb{N}$. *Hat* f *ausschließlich reelle Nullstellen, so gilt die Aussage für* $k = 0$.

Beweis. Einen Beweis findet man z. B. in [SS88]. □

Korollar 3.9.
Unter den obigen Voraussetzungen gilt

$$\mathrm{Nst}_-(f) = \mathrm{Vzw}(a_0, -a_1, a_2, \ldots, (-1)^d \cdot a_d) - 2k'$$

für ein geeignetes $k' \in \mathbb{N}$. *Hat* f *ausschließlich reelle Nullstellen, so gilt die Gleichung für* $k' = 0$.

Beweis. Die Gleichung folgt aus der Vorzeichenregel von Descartes (Satz 3.8) für das Polynom $f(-X)$, der Spiegelung von f an der y-Achse. □

Lemma 3.10.
Sei $f = \sum_{i=0}^{d} a_i X^i$. *Dann sind folgende Aussagen äquivalent.*

- $a_0 = \cdots = a_{\ell-1} = 0$ *und* $a_\ell \neq 0$.
- *Das Koeffiziententupel von* f *ist* $(0, \ldots, 0, a_\ell, \ldots, a_d)$ *mit* $a_\ell \neq 0$.
- 0 *ist Nullstelle von* f *mit Vielfachheit* ℓ.

Beweis. Ausklammern von X^ℓ bzw. Multiplikation mit X^ℓ. □

Definition 3.11.
Sei $\underline{a} \in \mathbb{R}^{d+1}$. Dann ist der Typ von $\underline{a}$ definiert durch

$$\tau(\underline{a}) := \mathrm{Vzw}(\underline{a}) - \mathrm{Vzw}(\overline{\underline{a}}).$$

Bemerkung 3.12.
Ist $\underline{a}$ Koeffiziententupel eines Polynoms f mit ausschließlich reellen Nullstellen, so gilt nach der Vorzeichenregel von Descartes $\tau(\underline{a}) = \tau(f)$.

Lemma 3.13.
Sei $f \in \mathbb{R}[X]$ *univariates Polynom vom Grad* d. *Dann gilt*

(1) $d = \mathrm{Nst}_+(f) + \mathrm{Nst}_-(f) + \mathrm{Nst}_0(f) + \mathrm{Nst}_{\mathbb{C}}(f)$.
(2) $\mathrm{Vzw}(\underline{a}) + \mathrm{Vzw}(\overline{\underline{a}}) + \mathrm{Nst}_0(f) < d \Rightarrow \mathrm{Nst}_{\mathbb{C}}(f) > 0$

Beweis. Gleichung (1) ist nur die Formulierung der Tatsache, daß ein reelles Polynom vom Grad d genau d komplexe Nullstellen besitzt.

Zu (2): Angenommen f habe nur reelle Nullstellen, d. h. $\mathrm{Nst}_{\mathbb{C}}(f) = 0$. Dann gilt

$$d > \mathrm{Vzw}(\underline{a}) + \mathrm{Vzw}(\overline{\underline{a}}) + \mathrm{Nst}_0(f) = \mathrm{Nst}_+(f) + \mathrm{Nst}_-(f) + \mathrm{Nst}_0(f) + \underbrace{\mathrm{Nst}_{\mathbb{C}}}_{=0} = d,$$

ein Widerspruch. □

Bemerkung 3.14.
Sei $f(X) = X^2 - X + 1 = (X - 1/2)^2 + 3/4$. Dann hat f die echt komplexen Nullstellen

$$\frac{1}{2} \cdot (1 + i \cdot \sqrt{3}) \quad \text{und} \quad \frac{1}{2} \cdot (1 - i \cdot \sqrt{3}).$$

Für das Koeffiziententupel $\underline{a} = (1, -1, 1)$ von f gilt $\mathrm{Vzw}(\underline{a}) + \mathrm{Vzw}(\overline{a}) = 2$. Dies zeigt, daß die Umkehrung des Lemmas 3.13 (2) im allgemeinen falsch ist.

a	$\overline{a}$	$\mathrm{Vzw}(a) + \mathrm{Vzw}(\overline{a})$
$(-1,-1)$	$(-1,1)$	1
$(-1,0)$	$(-1,0)$	0
$(-1,1)$	$(-1,-1)$	1
$(0,-1)$	$(0,1)$	0
$(0,1)$	$(0,-1)$	0
$(1,-1)$	$(1,1)$	1
$(1,0)$	$(1,0)$	0
$(1,1)$	$(1,-1)$	1

ABBILDUNG 1. Summe von Vorzeichenwechseln

Proposition 3.15.
Sei $\underline{a} \in \mathbb{R}^{d+1}$. *Dann gilt* $\mathrm{Vzw}(\underline{a}) + \mathrm{Vzw}(\overline{\underline{a}}) \leqslant d$.

Beweis. Ist $\underline{a} = \underline{0}$, so gilt $\mathrm{Vzw}(\underline{a}) + \mathrm{Vzw}(\overline{\underline{a}}) = 0 \leqslant d$. Sei also $\underline{a} \neq \underline{0}$. Beweis durch Induktion nach d der Länge des Tupels $\underline{a}$.

$d = 0$: Es gibt keinen Vorzeichenwechsel.

$d = 1$: Abbildung 1 zeigt das Ergebnis anhand der Signaturen von a_i.

$d \rightsquigarrow d+1$: Ist $a_{d+1} = 0$, so gilt

$$\mathrm{Vzw}(a_0, \ldots, a_d, 0) + \mathrm{Vzw}(\overline{a_0, \ldots, a_d, 0}) = \mathrm{Vzw}(a_0, \ldots, a_d) + \mathrm{Vzw}(\overline{a_0, \ldots, a_d}) \leqslant d < d+1$$

Dies beweist die Behauptung für $a_{d+1} = 0$. Also sei $a_{d+1} \neq 0$. Definiere

$$i := \max\{\, 0 \leqslant j \leqslant d \mid a_j \neq 0 \,\}.$$

Dieses Maximum existiert, da $0 \neq \underline{a}$ und somit die Menge nicht leer ist und sie nach Definition endlich ist.

Ist $d+1$ gerade und i ungerade oder umgekehrt, dann ist entweder $a_i \cdot a_{d+1} < 0$ oder $(-1)^i \cdot a_i \cdot (-1)^{d+1} \cdot a_{d+1} < 0$. Also gilt

$$\mathrm{Vzw}(a_0, \ldots, a_{d+1}) + \mathrm{Vzw}(\overline{a_0, \ldots, a_{d+1}}) = \mathrm{Vzw}(a_0, \ldots, a_d) + \mathrm{Vzw}(\overline{a_0, \ldots, a_d}) + 1 \leqslant d+1.$$

Ist d gerade und i ungerade oder umgekehrt, dann ist $i < d$. Also gilt

$$\mathrm{Vzw}(a_0, \ldots, a_i, \underbrace{0, \ldots, 0}_{(d-i+1)}) + \mathrm{Vzw}(\overline{a_0, \ldots, a_i, \underbrace{0, \ldots, 0}_{(d-i+1)}}) = \mathrm{Vzw}(a_0, \ldots, a_i) + \mathrm{Vzw}(\overline{a_0, \ldots, a_i}) \leqslant i < d.$$

Daraus folgt

$$\mathrm{Vzw}(a_0, \ldots, a_{d+1}) + \mathrm{Vzw}(\overline{a_0, \ldots, a_{d+1}}) \leqslant \underbrace{\mathrm{Vzw}(a_0 \ldots, a_d) + \mathrm{Vzw}(\overline{a_0, \ldots, a_d}) + 2}_{<d+2} \leqslant d+1 \qquad \square$$

Damit haben wir alle Grundlagen zusammen, um die Existenz von Typformeln zu beweisen. Da bei der Definition von Typformeln nur Polynome mit ausschließlich reellen Nullstellen berücksichtigt werden, können wir die Anzahl der positiven und negativen Nullstellen eines Polynoms mit der Vorzeichenregel von Descartes bestimmen. Diese nimmt nur Bezug auf die Vorzeichen der Koeffizienten eines Polynoms.

Für Polynome mit unbestimmten Koeffizienten reicht es somit aus, eine Fallunterscheidung nach den Vorzeichen der unbestimmten Koeffizienten zu machen.

Die Menge aller Polynome, deren Koeffizienten gewissen Vorzeichenbedingungen genügen, werden wir jeweils durch ein Polynom mit speziellen Koeffizienten repräsentieren.

Definition 3.16.

(1) Die Menge $C := \{-1, 0, 1, J\} \subset \mathbb{Z}[J]$ heißt die Menge der *charakteristischen Koeffizienten.*

(2) Mit $C' \subset C$ bezeichnen wir die Menge $\{-1, 0, 1\}$.

(3) Sei $f := \sum_{i=0}^{d} a_i X^i$ Polynom in $\mathbb{Z}[J][X]$ mit $a_i \in C$. Dann ist die Menge der zu f *kompatiblen* Polynome definiert durch

$$\Big\{ g \in \mathbb{R}[X] \;\Big|\; g = \sum_{i=0}^{d} b_i X^i \text{ und } \bigwedge_{\substack{0 \leqslant i \leqslant d \\ a_i \in \mathbb{Z}}} \operatorname{sign}(a_i) = \operatorname{sign}(b_i) \Big\}.$$

Koeffizienten $a_i = J$ des Polynoms f liefern also keine Vorzeichenbedingungen; sie können frei gewählt werden.

(4) Die Formel

$$\bigwedge_{\substack{0 \leqslant i \leqslant d \\ a_i \in \mathbb{Z}}} \operatorname{sign}(a_i) = \operatorname{sign}(b_i)$$

für Polynome $g = \sum_{i=0}^{d} b_i X^i$ bezeichnen wir als *Kompatibilitätsbedingung.*

(5) Sei $\underline{c} \in C^{d+1}$, dann heißt $\underline{d} \in \mathbb{R}^{d+1}$ *kompatibel* zu $\underline{c}$, falls

$$\bigwedge_{i=0}^{d} c_i = \operatorname{sign}(d_i).$$

Bemerkung 3.17.

(1) Die Menge aller univariaten Polynome mit Koeffizienten aus C und einer Gradschranke d ist endlich.

(2) Ist f ein Polynom mit Koeffizienten aus C', so haben alle kompatiblen Polynome, die ausschließlich reelle Nullstellen haben, den gleichen Typ. Dies folgt aus der Typbestimmung mit Hilfe der Vorzeichenregel von Descartes.

(3) Insbesondere ist kein Polynom f mit $\tau(f) = 0$, das ausschließlich reelle Nullstellen hat, zu einem Koeffiziententupel $\underline{c} \in {C'}^{d+1}$ mit $\tau(\underline{c}) = 0$ kompatibel.

Zur automatischen Bestimmung von Typformeln werden alle Polynome mit charakteristischen Koeffizienten aus $C' := \{-1, 0, 1\}$ mit einem maximal Grad d und ihr Typ bestimmt. Die Disjunktion der Kompatibilitätsbedingungen der Polynome, die den Typ 0 haben, ist dann eine Typformel für Polynome des Grades d.

Bei der sukzessiven Bestimmung von Typformeln für Polynome bis zu einem gewissen Maximalgrad kann man gewisse Fälle auf bereits berechnete Typformeln zurückführen.

Proposition 3.18.

(1) *Setze für ungerades* $d > 2$

$$T_d(c_0, \ldots, c_{d-1}) :\equiv (c_0 = 0) \wedge T_{d-1}(c_1 \ldots, c_{d-1})$$

(2) *und für gerades* $d > 3$

$$\begin{aligned} T_d(c_0, \ldots, c_{d-1}) :\equiv & \big((c_0 = 0) \wedge (c_1 = 0) \wedge T_{d-2}(c_2 \ldots, c_{d-1})\big) \\ & \vee (c_0 \neq 0) \wedge T'_d(c_0, \ldots, c_{d-1}) \end{aligned}$$

wobei T'_d *eine strikte Typformel für Polynome vom Grad* d *ist.*

Dann ist $T_d(\underline{c})$ *Typformel für Polynome vom Grad* d.

Beweis. Sei $\chi := X^d + \sum_{i=0}^{d-1} c_i X^i$ und habe χ nur reelle Nullstellen. Zu (1): Sei $d = 2e+1$ ungerade und $\tau(\chi) = 0$. Dann gilt

$$\begin{aligned} & \mathrm{Nst}_+(\chi) + \mathrm{Nst}_-(\chi) + \mathrm{Nst}_+(\chi) = 2e+1 \\ \Longrightarrow\ & \mathrm{Nst}_0(\chi) = 2e+1-2\,\mathrm{Nst}_+(\chi) \\ \Longrightarrow\ & \mathrm{Nst}_0(\chi) > 0 \\ \Longrightarrow\ & c_0 = 0 \end{aligned}$$

Ausklammern von X ergibt

$$\chi = g \cdot X \quad \text{wobei} \quad g = X^{d-1} + \sum_{i=1}^{d-1} c_i X^{i-1}.$$

Da $0 = \tau(\chi) = \tau(g) + \tau(X) = \tau(g)$, ist $T_{d-1}(c_1, \ldots, c_{d-1})$ erfüllt und somit auch $T_d(\underline{c})$.

Gilt umgekehrt $T_d(\underline{c})$, so ist zum einen $T_{d-1}(c_1, \ldots, c_{d-1})$ erfüllt und zum anderen gilt $c_0 = 0$. Es folgt

$$0 = \tau\Big(X^{d-1} + \sum_{i=1}^{d-1} c_i X^{i-1}\Big) = \underbrace{\tau(X)}_{=0} + \tau\Big(X^{d-1} + \sum_{i=1}^{d-1} c_i X^{i-1}\Big) = \tau\Big(X^d + \sum_{i=1}^{d-1} c_i X^i\Big).$$

Also ist $T_d(\underline{c})$ Typformel.

Zu (2): Sei d gerade. Mit einer Fallunterscheidung nach c_0 folgt

$$T_d(\underline{c}) \quad \text{gdw.} \quad (c_0 = 0 \wedge T_{d-1}(c_1, \ldots, c_{d-1})) \vee (c_0 \neq 0 \wedge T'_d(c_0, \ldots, c_{d-1})).$$

Mit (1) folgt die Behauptung. □

Proposition 3.19.

- $T_1(c_0) :\equiv c_0 = 0$
- $T_2(c_0, c_1) :\equiv c_1 = 0 \vee c_0 < 0$

sind Typformeln für Polynome vom Grad 1 *bzw.* 2.

Beweis. Für $d = 1$ ist f eine Gerade, und somit ist die Aussage trivial.

Für $d = 2$ ist f eine Parabel. Ist $c_1 = 0$ liegt ihr Scheitelpunkt auf der y-Achse und somit muß $\tau(f) = 0$ sein. Für $c_0 < 0$ folgt die Aussage aus der Zeichenregel von Descartes angewandt auf $(c_0, c_1, 1)$. Sei umgekehrt $f = x^2 + c_1 x + c_0$ eine Parabel mit $\tau(f) = 0$. Ist die Parabel symmetrisch zu y-Achse, gilt $c_1 = 0$. Besitzt f nur reelle Nullstellen, aber keine doppelte Nullstelle an 0, so existiert eine Nullstelle $-a_1 < 0$ und eine Nullstelle $a_2 > 0$. Es folgt

$$f = x^2 + (a_2 - a_1)x - (a_2 \cdot a_1) = x^2 + c_1 x + c_0$$

und somit $c_0 < 0$. Insgesamt erhalten wir $c_1 = 0 \vee c_0 < 0$. □

Bemerkung 3.20.
Im folgenden seien – ohne Beweis – einige von Hand optimierte Typformeln angegeben.

- $T_3(\underline{c}) :\equiv c_0 = 0 \wedge (c_2 = 0 \vee c_1 < 0)$
- $T'_4(\underline{c}) :\equiv c_0 > 0 \wedge (c_2 < 0 \vee (c_1 c_3 < 0))$
- $T'_6(\underline{c}) :\equiv c_0 < 0 \wedge \Big(\big((c_2 > 0 \vee c_3 c_1 < 0) \wedge (c_4 < 0 \vee c_5 c_3 < 0)\big) \vee (c_5 c_1 > 0 \wedge c_4 c_2 < 0)\Big)$

Eingabe: Der Grad d (d gerade), der Polynome für die T'_d berechnet werden soll.

Ausgabe: Eine strikte Typformel T'_d.

```
IF d ≡ 0 (mod 4) THEN c_{0,d} := 1 ELSE c_{0,d} := −1 END
ψ := FALSE
FOR EACH t ∈ {c_{0,d}} × {−1, 0, 1}^{d−1} × {1} DO
    π := Vzw(t_0, ..., t_d)
    ν := Vzw(t_0, −t_1 ..., (−1)^d t_d)
    IF π + ν = d AND π − ν = 0 THEN
        φ := TRUE
        FOR EACH 0 ⩽ i < d DO
            φ := φ ∧ (c_i ϱ(t_i) 0)
        END
        ψ := ψ ∨ φ;
    END
END
RETURN ψ
```

Dabei ist

$$\varrho(c) := \begin{cases} < & \text{falls} \quad c = -1 \\ = & \text{falls} \quad c = 0 \\ > & \text{falls} \quad c = 1 \end{cases}$$

ABBILDUNG 2. Algorithmus zur Bestimmung von $T_d(\underline{c})$

Lemma 3.21.
Sei $\chi = \sum_{i=0}^{d} c_i X^i$ Polynom vom Typ 0 mit ausschließlich reellen Nullstellen und $c_d = 1$ sowie $c_0 \neq 0$. Dann gilt:

(1) *Ist $d \equiv 2 \pmod 4$, dann ist $c_0 < 0$.*
(2) *Ist $d \equiv 0 \pmod 4$, dann ist $c_0 > 0$.*

Beweis. Nach der Vorzeichenregel von Descartes muß das Koeffiziententupel von χ genau $d/2$ Vorzeichenwechsel besitzen. Ist $d \equiv 2 \pmod 4$, so ist $d/2$ ungerade. Von $a_d > 0$ ausgehend, kommt man aber mit einer ungeraden Anzahl von Vorzeichenwechseln auf $c_0 < 0$. Analoges gilt für $d \equiv 0 \pmod 4$. □

Algorithmus 3.22.
Der in Abbildung 2 gezeigte Algorithmus berechnet eine strikte Typformel $T'_d(\underline{c})$ für Polynome des Grades d.

Beweis. Die Termination des Algorithmus ist klar, da die `FOR EACH`-Schleifen über endliche Mengen laufen.

Zur Korrektheit: Nach Lemma 3.21 werden alle in Frage kommenden Koeffiziententupel durchlaufen. Nach der Vorzeichenregel von Descartes wird der Typ des durch das Koeffiziententupel festgelegte Polynoms korrekt bestimmt, sofern es ausschließlich reelle Nullstellen hat. Andere Polynome müssen jedoch nach der Definition der Typformeln nicht berücksichtigt werden. Nach 3.13 und 3.15 ist $\pi + \nu = d$ notwendig dafür, daß das zugehörige Polynom nur reelle Nullstellen besitzt. Die Bedingung $\pi + \nu = 0$ garantiert, daß das zugehörige Polynom, falls es ausschließlich reelle Nullstellen hat, den Typ 0 besitzt. □

Bemerkung 3.23.
Der Algorithmus 3.22 beweist zum einen die Existenz von Typformeln für Polynome vom beliebigen Grad, da nach Proposition 3.18 und 3.19 die Berechnung der strikten

	$\pi+\nu=d$			$\pi+\nu\neq d$
d	$\pi-\nu=0$	$\pi-\nu<0$	$\pi-\nu>0$	
2	3	0	0	0
4	13	1	1	12
6	63	11	11	158
8	321	86	86	1694
10	1683	594	594	16812
12	8989	3871	3871	160416
14	48639	24437	24437	1496810
16	265729	151308	151308	13780562

ABBILDUNG 3. Anzahl von Koeffiziententupel

Typformeln T'_d zur Konstruktion von Typformeln ausreicht. Andererseits zeigt der Algorithmus, daß Typformeln algorithmisch konstruiert werden können.

Bemerkung 3.24.
Der Algorithmus liefert eine erste grobe Abschätzung für die Länge der strikten Typformeln. Da $3^{(d-1)}$ Tupel durchlaufen werden, können auch nur so viele einen Typ gleich Null habe. Da aus jedem Tupel ein Disjunktionsglied der Länge d (für eine DNF) entsteht, enthält eine strikte Typformel für Polynome des Grades d maximal $d \cdot 3^{d-1}$ atomare Formeln.

Beispiel 3.25.
Die Abbildung 3 zeigt die Anzahl der Koeffiziententupel in Abhängigkeit vom Typ des zugehörigen Polynoms, die vom obigen Algorithmus durchlaufen werden.

3.3. Effiziente Bestimmung von Typformeln

Im Abschnitt 3.2 wurde ein Algorithmus präsentiert, mit dem man strikte Typformeln bestimmen kann. Eine obere Abschätzung für die Anzahl der klassifizierten Koeffiziententupel war 3^{d-1}, und damit ergab sich als Abschätzung für die Anzahl der atomaren Formeln in der disjunktiven Normalform $d \cdot 3^{d-1}$. Im folgenden Abschnitt soll eine Optimierung dieses Verfahrens vorgestellt werden. Wir werden uns auf die Bestimmung des Teiles T'_d (siehe 3.18) beschränken. Dabei werden wir Polynome mit charakteristischen Koeffizienten aus $C = \{-1, 0, 1, J\}$ betrachten.

Definition 3.26.
Sei $\underline{c} \in C'^{d+1}$, d gerade. Dann heißt $\underline{c}$ *gut*, falls folgende Bedingungen erfüllt sind.

(1) $c_d = 1$
(2) $c_0 = c_{0,d} := \begin{cases} -1 & \text{für } d \equiv 2 \pmod 4 \\ 1 & \text{für } d \equiv 0 \pmod 4 \end{cases}$
(3) $\mathrm{Vzw}(\underline{c}) + \mathrm{Vzw}(\overline{\underline{c}}) = d$
(4) $\mathrm{Vzw}(\underline{c}) - \mathrm{Vzw}(\overline{\underline{c}}) = 0$

Proposition 3.27.
Sei $\underline{c} \in C'^{d+1}$ *gutes Koeffiziententupel. Dann gibt es kein* $0 < i < j < d$ *mit* $c_i = \cdots = c_j = 0$, $c_{i-1} \neq 0$, *und* $c_{j+1} \neq 0$

Beweis. Angenommen es gäbe ein solches i und j. Dann gilt

$$\begin{aligned}
&\mathrm{Vzw}(c_0,\ldots,c_{i-1},0,\ldots,0,c_{j+1},\ldots,c_d) + \mathrm{Vzw}(\overline{c_0,\ldots,c_{i-1},0,\ldots,0,c_{j+1},\ldots,c_d})\\
&= \mathrm{Vzw}(c_0,\ldots,c_{i-1}) + \mathrm{Vzw}(c_{i-1},0,\ldots,0,c_{j+1}) + \mathrm{Vzw}(c_{j+1},\ldots,c_d) +\\
&\quad \mathrm{Vzw}(c_0,\ldots,(-1)^{i-1}\cdot c_{i-1}) + \mathrm{Vzw}((-1)^{i-1}\cdot c_{i-1},0,\ldots,0,(-1)^{j+1}\cdot c_{j+1}) +\\
&\quad \mathrm{Vzw}((-1)^{j+1}\cdot c_{j+1},\ldots,(-1)^d\cdot c_d) =: V.
\end{aligned}$$

Ist i gerade und j ungerade oder umgekehrt, so gilt

$$\begin{aligned}
V &\leqslant \mathrm{Vzw}(c_0,\ldots,c_{i-1}) + \mathrm{Vzw}(c_{j+1},\ldots,c_d) +\\
&\qquad \mathrm{Vzw}(c_0,\ldots,(-1)^{i-1}\cdot c_{i-1}) + \mathrm{Vzw}((-1)^{j+1}\cdot c_{j+1},\ldots,(-1)^d\cdot c_d) + 1\\
&= \mathrm{Vzw}(c_0,\ldots,c_{i-1},c_{j+1},\ldots,c_d) + \mathrm{Vzw}(\overline{c_0,\ldots,c_{i-1},c_{j+1},\ldots,c_d})\\
&\leqslant d-(j-i)-1+1 = d-(j-i).
\end{aligned}$$

Da nach Voraussetzung $j-i>1$, gilt $V \leqslant d-1$, ein Widerspruch zur Definition eines guten Koeffiziententupels.

Sind i und j gerade oder i und j ungerade, so gilt

$$\begin{aligned}
V &\leqslant \mathrm{Vzw}(c_0,\ldots,c_{i-1}) + \mathrm{Vzw}(c_{j+1},\ldots,c_d) +\\
&\qquad \mathrm{Vzw}(c_0,\ldots,(-1)^{i-1}\cdot c_{i-1}) + \mathrm{Vzw}((-1)^{j+1}\cdot c_{j+1},\ldots,(-1)^d\cdot c_d) + 2\\
&= \mathrm{Vzw}(c_0,\ldots,c_{i-1},0,c_{j+1},\ldots,c_d) + \mathrm{Vzw}(\overline{c_0,\ldots,c_{i-1},0,c_{j+1},\ldots,c_d})\\
&\leqslant d-(j-i)-1+2 = d-(j-i)+1.
\end{aligned}$$

Da nach Voraussetzung $j-i>2$, gilt $V \leqslant d-1$, ein Widerspruch zur Definition eines guten Koeffiziententupels. □

Definition 3.28.
Sei $\underline{c} \in C^{d+1}$ und $0<i<d$ mit $c_{i-1}\cdot c_{i+1} = -1$. Dann heißt i *Jokerposition* von $\underline{c}$.

Im folgenden wird gezeigt, wie man mittels Jokerpositionen die Berechnung von Typformeln optimieren kann. Wir werden nun den Koeffizienten J als zusätzlichen charakteristischen Koeffizient zulassen und jeweils die Menge der zu einem Koeffiziententupel kompatiblen Polynome betrachten.

Definition 3.29.
Sei $\underline{c} = (c_0\ldots,c_d) \in C^{d+1}$ mit folgenden Eigenschaften:

- $c_0, c_d \in C'$
- Für alle $0<i<d$ mit $c_i = J$ gilt $c_{i-1}\cdot c_{i+1} = -1$, d. h. i ist Jokerposition.

Dann definieren wir

(1) $\overline{(c_0,\ldots,c_d)} := (e_0,\ldots,e_d)$, wobei

$$e_i := \begin{cases} (-1)^i\cdot c_i & \text{falls } c_i \in \{-1,0,1\} \\ J & \text{falls } c_i = J \end{cases}$$

(2) $\mathrm{Vzw}(c_0,\ldots,d_d) := \mathrm{Vzw}(e_0,\ldots,e_d)$, wobei

$$e_i := \begin{cases} c_i & \text{falls } c_i \in \{-1,0,1\} \\ 0 & \text{falls } c_i = J \end{cases}$$

(3) $\tau(\underline{c}) = \mathrm{Vzw}(\underline{c}) - \mathrm{Vzw}(\overline{\underline{c}})$

(c_0, c_1, c_2)	$\mathrm{Vzw}(\underline{c})$
$(-1,-1,1)$	1
$(-1,1,1)$	1
$(1,-1,-1)$	1
$(1,1,-1)$	1

ABBILDUNG 4. Vorzeichenwechsel in einem Tripel.

Lemma 3.30.
Sei $(c_0, c_1, c_2) \in C'^3$ *und* $c_0 \cdot c_2 = -1$. *Dann gilt* $\mathrm{Vzw}(c_0, c_1, c_2) = 1$.

Beweis. Für $c_1 = 0$ ist die Aussage nach Definition von $\mathrm{Vzw}(c_0, 0, c_2)$ erfüllt. Die anderen Fälle ersieht man aus Abbildung 4. □

Lemma 3.31.
Sei $\underline{c} \in C^{d+1}$, *so daß* $c_0, c_d \in C'$ *und* $c_{i-1} \cdot c_{i+1} = -1$, *falls* $c_i = J$. *Sei* $\underline{e} \in C'^{d+1}$ *kompatibel zu* $\underline{c}$. *Dann gilt*

(1) $\mathrm{Vzw}(\underline{c}) = \mathrm{Vzw}(\underline{e})$.
(2) $\mathrm{Vzw}(\overline{\underline{c}}) = \mathrm{Vzw}(\overline{\underline{e}})$.
(3) $\tau(\underline{c}) = \tau(\underline{e})$.

Beweis. Zu (1): Wir zeigen $\mathrm{Vzw}(\underline{c}) = \mathrm{Vzw}(\underline{e})$ über Induktion nach

$$k = |\{\, 0 < j < d \mid c_j = J \,\}|.$$

Für alle $c_j \neq J$ gilt nach Definition von kompatibel $c_j = e_j$. Für $k = 0$ ist somit die Aussage trivial. Man sieht leicht, daß falls $\underline{e}$ zu $\underline{c}$ kompatibel ist, auch $(e_i, \ldots, e_j)$ kompatibel zu $(c_i, \ldots, c_j)$ ist. Sei $k > 0$ und $c_j = J$. Es gilt $c_{j-1} = e_{j-1}$ und $c_{j+1} = e_{j+1}$, da nach Voraussetzung $c_{j-1} \cdot c_{j+1} = -1$ und somit $c_{j-1}, c_{j+1} \in \{-1, 1\}$. Also gilt

$$\begin{aligned}
\mathrm{Vzw}(c_0, \ldots, c_d) &= \mathrm{Vzw}(c_0, \ldots, c_{j-1}) + \mathrm{Vzw}(c_{j-1}, J, c_{j+1}) + \mathrm{Vzw}(c_{j+1}, \ldots, c_d) \\
&= \mathrm{Vzw}(c_0, \ldots, c_{j-1}) + \mathrm{Vzw}(c_{j-1}, 0, c_{j+1}) + \mathrm{Vzw}(c_{j+1}, \ldots, c_d) \\
&= \mathrm{Vzw}(c_0, \ldots, c_{j-1}) + \mathrm{Vzw}(c_{j-1}, e_j, c_{j+1}) + \mathrm{Vzw}(c_{j+1}, \ldots, c_d)
\end{aligned}$$

und mit der Induktionsannahme

$$\begin{aligned}
&= \mathrm{Vzw}(e_0, \ldots, e_{j-1}) + \mathrm{Vzw}(e_{j-1}, e_j, e_{j+1}) + \mathrm{Vzw}(e_{j+1}, \ldots, e_d) \\
&= \mathrm{Vzw}(e_0, \ldots, e_d).
\end{aligned}$$

Falls $c_{j-1} \cdot c_{j+1} = -1$, gilt auch

$$(-1)^{j-1} c_{j-1} \cdot (-1)^{j+1} c_{j+1} = -1.$$

Also gilt $\mathrm{Vzw}(\overline{\underline{c}}) = \mathrm{Vzw}(\overline{\underline{e}})$, die Aussage aus (2). Somit folgt (3):

$$\tau(\underline{c}) = \mathrm{Vzw}(\underline{c}) - \mathrm{Vzw}(\overline{\underline{c}}) = \mathrm{Vzw}(\underline{e}) - \mathrm{Vzw}(\overline{\underline{e}}) = \tau(\underline{e}). \quad \square$$

Korollar 3.32.
Sei $\underline{c} \in C^{d+1}$, *so daß* $c_i = J \Rightarrow c_{i-1} \cdot c_i + 1 = J$, *dann ist der Typ aller Polynome, die ein zu* $\underline{c}$ *kompatibles Koeffiziententupel besitzen, gleich. Insbesondere gibt es zu solch einem* $\underline{c}$ *mit* $\tau(\underline{c}) = 0$ *kein Polynom* f *mit ausschließlich reellen Nullstellen und* $\tau(f) \neq 0$. □

Bemerkung 3.33.
Für die Konstruktion von Typformeln bedeutet eine Jokerposition, daß der entsprechende Koeffizient nicht in eine Vorzeichenbedingung einbezogen werden muß. Wir werden an diesen Positionen den charakteristischen Koeffizienten J einsetzen. Zu beachten ist, daß durch das Ersetzen eines charakteristischen Koeffizienten aus C' durch den charakteristischen Koeffizienten J auch Jokerpositionen zerstört werden können. Umgekehrt können auch durch das Ersetzen eines charakteristischen Koeffizienten J Jokerpositionen entstehen.

Beispiel 3.34.
Sei $\underline{c} = (1, 1, -1, -1)$. Dann sind 1 und 2 Jokerpositionen. Alle zu $(1, J, -1, -1)$ bzw. zu $(1, 1, J, -1)$ kompatiblen Polynome haben dann den gleichen Typ, wie $\underline{c}$. Jedoch gibt es in diesen beiden Koeffizententupeln keine Jokerpositionen mehr.

Ersetzt man in $(1, 1, J, -1, 1)$ den Koeffizienten J durch -1, so sind 1, 2 und 3 Jokerpositionen, jedoch war ursprünglich nur 2 eine Jokerposition.

Proposition 3.35.
Sei $\underline{c} \in {C'}^{d+1}$ gutes Koeffizententupels und sei $0 < i < d$ mit $c_i = 0$. Dann ist i Jokerposition von $\underline{c}$.

Beweis. Nach Proposition 3.27 gilt $c_{i-1} \neq 0$ und $c_{i+1} \neq 0$. Angenommen i sei keine Jokerposition, d. h. $c_{i-1} = c_{i+1}$. Dann gilt

$$\mathrm{Vzw}(c_0, \ldots, c_{i-1}, 0, c_{i-1}, \ldots, c_d) + \mathrm{Vzw}(\overline{c_0, \ldots, c_{i-1}, 0, c_{i+1}, \ldots, c_d}) =$$
$$\mathrm{Vzw}(c_0, \ldots, c_{i-1}, \ldots, c_d) + \mathrm{Vzw}(\overline{c_0, \ldots, c_{i-1}, \ldots, c_d}) < d,$$

ein Widerspruch. □

Diese beiden Eigenschaften des charakteristischen Koeffizienten 0 in guten Tupeln zeigen, daß er bei der Bestimmung von guten Koeffizententupeln vernachlässigt werden kann, sofern jede Jokerposition wirklich unspezifiziert gelassen wird. Diese Eigenschaft nutzen wir im folgenden Algorithmus aus.

Algorithmus 3.36.
Der in Abbildung 5 *gezeigte Algorithmus wird zur Berechnung von Koeffizententupeln zur Bestimmung von Typformeln benutzt.*

Beweis. Die Korrektheit und Termination des Algorithmus wird in den folgenden drei Lemmata 3.38 und 3.39 und 3.40 bewiesen. □

Korollar 3.37.
(1) *Für $c \in D$ ist i in $(c, c_i, \ldots, c_d)$ keine Jokerposition.*
(2) *Für $c \in D \setminus \{-1, 1\}$ ist i in $(c, c_i, \ldots, c_d)$ Jokerposition.*

Beweis. Folgt aus der Spezifikation von D. □

Lemma 3.38.
Der Algorithmus 3.36 *terminiert.*

Beweis. Wir betrachten den Baum der rekursiven Aufrufe des Algorithmus. Da alle `FOR`-Schleifen nach der Spezifikation von D nur über endliche Mengen laufen, ist dieser Baum nur endlich verzweigt.

Im folgenden wird mit Induktion über i gezeigt, daß die Länge $|\underline{c}| := d - i + 1$ des übergebenen Koeffizententupels $\underline{c}$ bei jedem rekursiven Aufruf streng isoton wächst und durch $d+1$ nach oben beschränkt ist. Bezeichne $\underline{c}'$ das neue Koeffizententupel.

$i = 0$: Es wird kein erneuter rekursiver Aufruf gemacht, $\underline{c}$ hat die Länge $d + 1$.

Eingabe: Eine gerade, natürliche Zahl $d \in \mathbb{N}$. Ein Tupel $(c_i, \ldots, c_d) \neq ()$ mit

(1a) $c_k = J \Longleftrightarrow c_{k-1} \cdot c_{k+1} = -1$ für $i < k < d$ und
(1b) $c_i \in \{-1, 1\}$ und
(1c) $i = 0 \Longrightarrow c_0 = c_{0,d}$ und
(1d) $i = 1 \Longrightarrow c_2 = J$ und
(1e) $i = d \Longrightarrow c_d = 1$

Eine Menge $D \subseteq \{-1, 1\}$, so daß entweder

(2a) $i = d$ und $D = \{-1, 1\}$ oder
(2b) $i = 0$ und $D = \emptyset$ oder
(2c) $0 < i < d$, $c_{i+1} \in \{-1, 1\}$ und $D = \{c_{i+1}\}$ oder
(2d) $0 < i < d$, $c_{i+1} = J$ und $D = \{-1, 1\}$.

Ausgabe: Die Menge

$$M := \{\, (e_0, \ldots, e_d) \in C^{d+1} \mid e_k = J \Longleftrightarrow e_{k-1} \cdot e_{k+1} = -1 \ (i < k < d),\ c_0 = c_{0,d},\ c_d = 1,\ \mathrm{Vzw}(\underline{c}) + \mathrm{Vzw}(\overline{c}) = d,\ \tau(\underline{c}) = 0,\ e_k = c_k \ (i \leqslant k \leqslant d) \text{ und } e_0 = c_{0,d} \,\} .$$

```
1  PROCEDURE ctj(d, (c_i, ..., c_d), D)
2  BEGIN
3      π := Vzw(c_i, ..., c_d);
4      ν := Vzw((-1)^i c_i, ..., (-1)^d c_d);
5      IF d ≡ 0 (mod 4) THEN c_{0,d} := 1 ELSE c_{0,d} := -1 END
6      IF (π > d/2) OR (ν > d/2) THEN RETURN ∅; END
7      IF i = 0 THEN
8          IF π + ν = d AND π = ν THEN
9              RETURN {(c_0, ..., c_d)};
10         ELSE
11             RETURN ∅;
12         END
13     ELSIF i = 1 THEN
14         RETURN ctj(d, (c_{0,d}, c_i ..., c_d), ∅);
15     ELSIF i = 2 THEN
16         IF c_{0,d} · c_i = -1 THEN
17             RETURN ctj(d, (c_{0,d}, J, c_2, ..., c_d), ∅);
18         ELSE
19             R := ∅;
20             FOR EACH c ∈ D DO
21                 R := R ∪ ctj(d, (c_{0,d}, c, c_2, ..., c_d), ∅);
22             END
23             RETURN R;
24         END
25     ELSE
26         R := ctj(d, (-c_i, J, c_i, ..., c_d), {-1, 1});
27         FOR EACH c ∈ D DO
28             R := R ∪ ctj(d, (c, c_i, ..., c_d), {c_i});
29         END
30         RETURN R;
31     END
32 END ctj;
```

ABBILDUNG 5. Verbesserter Algorithmus zur Berechnung von charakteristischen Koeffiziententupeln

$i = 1$: Das Koeffiziententupel $\underline{c}$ wird um den Eintrag $c_{0,d}$ erweitert. Somit gilt

$$|\underline{c}| < |\underline{c}'| = |\underline{c}| + 1 = d + 1.$$

$i = 2$: Das Koeffiziententupel wird um die Einträge $c_{0,d}$ und c_1 erweitert. Somit gilt

$$|\underline{c}| < |\underline{c}'| = |\underline{c}| + 2 = d + 1.$$

$i > 2$: Da $i \geqslant 2$, gilt $|\underline{c}| \leqslant d - 1$. Das Koeffiziententupel wird bei jedem Aufruf um mindestens einen und um maximal zwei Einträge erweitert. Also gilt

$$|\underline{c}| < |\underline{c}'| = |\underline{c}| + 1 \leqslant d \quad \text{bzw.} \quad |\underline{c}| < |\underline{c}'| = |\underline{c}| + 2 \leqslant d + 1.$$

Dies beweist, daß jeder Ast des Baums endlich ist. Somit ist nach Königs Lemma der Baum endlich und der Algorithmus terminiert. □

Lemma 3.39.
Im Algorithmus 3.36 *genügen sämtliche rekursiven Aufrufe den Spezifikationen.*

Beweis. Bezeichne $(c_j, \ldots, c_d)$ das übergebene Tupel und D_0 die übergebene Menge. Das dem rekursiven Aufruf übergebene Tupel bezeichnen wir mit $(c_i, \ldots, c_d)$, wobei $i < j$. Dies ist nach den rekursiven Aufrufen im Algorithmus zulässig, da das neue Tupel durch Erweiterung des übergebenen entsteht. Man beachte, daß sich die Bedingungen der `IF`-Abfrage ab Zeile 7 auf j bezieht.

Die Implikation in (1e) ist für jeden rekursiven Aufruf erfüllt, da die Prämisse nach der Voraussetzung $\underline{e} \neq ()$, in diesen nicht erfüllt ist. Nach den Voraussetzungen gilt die Äquivalenz in (1a) für $i < k < d$.

Zum Aufruf in Zeile 14: Die Äquivalenz in (1a) folgt aus

$$j = 1 \Longrightarrow c_2 = J \Longrightarrow c_0 \cdot c_2 \neq -1 \quad \text{und} \quad c_1 \in \{-1, 1\}.$$

Zum Aufruf in Zeile 17: Nach der `IF`-Abfrage gilt die Äquivalenz (1a) für $k = 1$. Für $k = 2$ ist $c_1 = J$ und nach Voraussetzung $c_2 \in \{-1, 1\}$. Somit gilt (1a).

Zum Aufruf in Zeile 21: Da die Bedingung in der `IF`-Abfrage nicht erfüllt ist, und da $c_1 \in D \subseteq \{-1, 1\}$ ist, gilt die Äquivalenz (1a) für $k = 1$. Für $k = 2$ ist entweder $k = d$ oder die Äquivalenz gilt nach Korollar 3.37 (1). Also ist (1a) erfüllt.

Die Gültigkeit der anderen Voraussetzungen zeigten wir gemeinsam für die Aufrufe in den Zeilen 14, 17 und 21. Nach der Definition von $c_{0,d}$ gilt (1b) und (1d). Die Implikation (1c) gilt, da die Prämisse nicht erfüllt ist. Nach dem Aufrufschema gilt (2b).

Zum Aufruf in Zeile 26: Für $k = i + 1 = j - 1$ gilt die Äquivalenz (1a) nach dem Aufrufschema. Für $k = i + 2 = j$ gilt die Äquivalenz, da $c_i \in \{-1, 1\}$. Nach Voraussetzung gilt $c_i = -c_j \in \{-1, 1\}$. Ist $j = 3$, so ist $i = 1$ und nach dem Aufrufschema gilt die Implikation (1c). Da $j > 2$ ist $i > 0$ und somit ist die Prämisse von (1d) nicht erfüllt. Nach dem Aufrufschema gilt (2d).

Zum Aufruf in Zeile 28: Für $k = j = i + 1$ gilt die Äquivalenz nach Korollar 3.37 (1). Nach Voraussetzung gilt $c_i \in D \subseteq \{-1, 1\}$. Da $j > 2$ ist $i > 1$ und somit ist die Prämisse von (1c) und (1d) nicht erfüllt. Da $c_i = c_j \in \{-1, 1\}$, gilt nach dem Aufrufschema (2c). □

Lemma 3.40.
Die Spezifikation der Ausgabemenge des Algorithmus 3.36 *ist korrekt.*

Beweis. Beweis über Induktion nach i. Für $i = 0$ ist nach der `IF`-Abfrage das zurückgegeben Tupel $\underline{c}$ gut. Nach den Voraussetzungen an $\underline{c}$ gilt also die Spezifikation.

Im Induktionsschritt gilt die Bedingung der Äquivalenz nach der Voraussetzung an das übergebene Tupel. Die Gleichheit $c_0 = c_{0,d}$ folgt ebenfalls aus den Voraussetzungen an $\underline{c}$.

Die Anzahl der Vorzeichenwechsel ist nach den Voraussetzungen an $\underline{c}$ definiert. Somit garantiert die Bedingung der `IF`-Abfrage in Zeile 8, daß nur Koeffiziententupel zurückgegeben werden, die den Vorzeichenwechselbedingungen genügen. Nach der Spezifikation von $\underline{c}$ gilt $c_d = 1$.

An keiner anderen Stelle wird ein Koeffiziententupel zurückgegeben. Somit gilt $\mathrm{ctj}(d, \underline{c}, D) \subseteq M$.

Für $i = 1$ wird das Tupel um $c_{0,d}$ erweitert. Alle anderen Koeffizienten sind nicht zulässig.

Für $i = 2$ wird das Tupel um $c_{0,d}$ und $c_1 = J$ erweitert, falls $c_{0,d} \cdot c_2 = -1$. Da $c_{0,d}$ eindeutig festgelegt ist, ist das in diesem Fall die einzig mögliche Erweiterung. Ist $c_0 \cdot c_2 \neq -1$, wird das Tupel um die Koeffizienten $c_{0,d}$ und $c_1 = c$ für jedes $c \in D$ erweitert. Dabei ist $c_{0,d}$ nach Spezifikation festgelegt. Für $c \in \{-1, 1\} \setminus D$ wäre nach Korollar 3.37 (2) die Stelle i eine Jokerposition aber $c_i \neq J$, ein Widerspruch.

Für $i > 2$ wird zum einen das Tupel um $c_i = -c_j$ und $c_{i+1} = J$ erweitert. In diesem Fall ist c_i durch J festgelegt. Zum anderen wird das Tupel um $c_i = c$ für jedes $c \in D$ erweitert. Für $c \in \{-1, 1\} \setminus D$ wäre nach Korollar 3.37 (2) die Stelle i eine Jokerposition aber $c_i \neq J$, ein Widerspruch. Bei dieser Erweiterung kann der Koeffizient c_0 nicht entstehen.

Ist $\pi > d/2$ oder $\nu > d/2$ so folgt aus $\pi + \nu = d$, daß $\pi - \nu \neq 0$. Also kann in diesem Fall kein Tupel $(\ldots, c_j, \ldots, c_d)$ die geforderten Vorzeichenwechselbedingungen erfüllen. Ohne die `IF`-Abfrage in Zeile 6 würde also die Bedingung der `IF`-Abfrage in Zeile 8 nicht erfüllt sein. Somit ist die `IF`-Abfrage in Zeile 6 nur eine Optimierung und ändert nichts an der Ausgabemenge M. □

Beispiel 3.41.
Die Ausgabe von $\mathrm{ctj}(4, (1), \{-1, 1\})$ ist

$$\{(1, J, -1, J, 1), (1, 1, J, -1, 1), (1, -1, J, 1, 1)\}.$$

Lemma 3.42.
Sei $\underline{c} \in C'^{d+1}$ ein gutes Koeffiziententupel. Dann gibt es ein $\underline{e} \in \mathrm{ctj}(d, (1), \{-1, 1\})$, so daß $\underline{c}$ zu $\underline{e}$ kompatibel ist.

Beweis. Wir definieren eine endliche Folge $\underline{e}^{(n)}$ von Koeffiziententupeln in C^{d+1}. Sei $\underline{c} \in C'^{d+1}$ ein beliebiges, gutes Koeffiziententupel. Dann definieren wir

$$(e_0^0, \ldots, e_d^0) := (c_0, \ldots, c_d).$$

Die weiteren Folgenglieder definieren wir induktiv. Sei

$$I_n := \{\, 0 < j < d \mid e_{j-1}^{(n)} \cdot e_{j+1}^{(n)} = -1 \,\}.$$

Ist $I_n = \emptyset$, dann definiere $\underline{e} := \underline{e}^{(n)}$ und sonst

$$i_n := \min(I_n) \quad \text{und} \quad \left(e_0^{(n+1)}, \ldots, e_d^{(n+1)}\right) := \left(e_0^{(n)}, \ldots, e_{i_n-1}^{(n)}, J, e_{i_n+1}^{(n)}, \ldots, e_d^{(n)}\right)$$

Nach Konstruktion ist dann $\underline{e}$ zu $\underline{c}$ kompatibel, da nur Koeffizienten durch den Koeffizient J ersetzt werden. Durch die Definition von i_n wird garantiert, daß $\underline{e}$ die Voraussetzungen des Lemmas 3.31 erfüllt. Somit gilt $\mathrm{Vzw}(\underline{c}) = \mathrm{Vzw}(\underline{e})$ und $\mathrm{Vzw}(\overline{c}) = \mathrm{Vzw}(\overline{e})$. Darüber hinaus gilt $e_j = J \iff c_{j-1} \cdot c_{j+1} = -1$ und da $\underline{c}$

Eingabe: Die Menge M, wie vom Algorithmus 3.43 berechnet.

Ausgabe: Eine strikte Typformel T'_d.

```
ψ := FALSE
FOR EACH t ∈ M DO
    φ := TRUE
    FOR EACH 0 ⩽ i < d DO
        φ := φ ∧ ζ_i
    END
    ψ := ψ ∨ φ;
END
RETURN ψ
```

Dabei ist

$$\zeta_i := \begin{cases} (c_i < 0) & \text{falls } c = -1 \\ (c_i > 0) & \text{falls } c = 1 \\ \texttt{TRUE} & \text{falls } c = J \end{cases}$$

ABBILDUNG 6. Algorithmus zur Bestimmung von $T'_d(\underline{c})$ aus M

ein gutes Koeffiziententupel war, gilt $c_0 = c_{0,d}$ und $c_d = 1$. Insgesamt erhalten wir $\underline{e} \in \mathrm{ctj}(d, (1), \{-1, 1\})$. □

Aus der Menge $M := \mathrm{ctj}(d, (1), \{-1, 1\})$ kann man leicht eine strikte Typformel $T'_d(\underline{a})$ berechnen.

Algorithmus 3.43.
Der Algorithmus aus Abbildung 6 *berechnet eine strikte Typformel $T'_d(\underline{c})$ in disjunktiver Normalform für Polynome des Grades d.*

Beweis. Die Termination ist klar. Zur Korrektheit: Sei $f = X^d + \sum_{i=0}^{d-1} a_i X^i$ Polynom mit ausschließlich reellen Nullstellen und $\tau(f) = 0$. Dann ist $\tau(\underline{c}) = 0$ für $\underline{c} := (\mathrm{sign}(a_0), \ldots, \mathrm{sign}(a_{d-1}), 1)$, und darüber hinaus ist $\underline{c}$ sogar ein gutes Koeffiziententupel. Also gibt es ein $\underline{e} \in M$, so daß $\underline{c}$ kompatibel zu $\underline{e}$ ist. Somit gibt es ein Disjunktionsglied in $T'_d(\underline{c})$, das erfüllt ist.

Gilt umgekehrt $\tau(f) \neq 0$, so ist $\tau(\underline{c}) \neq 0$ und es gibt es kein $\underline{e} \in M$, so daß $\underline{c}$ kompatibel zu $\underline{e}$ ist. Es folgt, daß kein Disjunktionsglied von $T_d(\underline{c})$ erfüllt ist. □

Beispiel 3.44.
Für $M := \mathrm{ctj}(4, (1), \{-1, 1\})$ ist die Ausgabe des obigen Algorithmus

$$(c_0 > 0 \wedge c_2 < 0) \vee (c_0 > 0 \wedge c_1 > 0 \wedge c_3 < 0) \vee (c_0 > 0 \wedge c_1 < 0 \wedge c_3 > 0).$$

Lemma 3.45.
Sei d gerade, $c_{0,d} := 1$ falls $d \equiv 0 \pmod 4$ bzw. $c_{0,d} := -1$ falls $d \equiv 2 \pmod 4$. Sei

$$Q := \{\, (c_{0,d}, c_1, \ldots, c_{d-1}, 1) \in \{-1, 1, J\} \mid c_i = J \Longleftrightarrow c_{i-1} \cdot c_{i+1} = -1 \ (0 < i < d) \,\}.$$

Dann gilt $|Q| \leqslant 2^{d-1}$.

Beweis. Zum Beweis konstruieren wir eine injektive Abbildung

$$\varphi : M \longrightarrow \{c_{0,d}\} \times \{-1, 1\}^{d-1} \times \{1\}.$$

Die Idee bei der Konstruktion ist es, die Koeffizienten $c_j = J$ so durch Elemente aus $\{1, -1\}$ zu ersetzen, daß maximal eine neue Jokerposition entsteht. Dazu werden wir Blöcke von aufeinanderfolgenden Koeffizienten betrachten, bei denen jeder zweite Koeffizient gleich J ist (z. B. $(1, J, -1, J, 1, 1)$, $(-1, -1, J, 1, J, 1, J, -1, 1)$). Es

ist klar, daß durch das Ersetzen eines Koeffizienten c_i nur i und $i+1$ neue Jokerpositionen werden können.

Sei $\underline{c} := (c_0, \ldots, c_d) \in Q$. Zur Definition von $\varphi(\underline{c})$ definieren wir eine endliche Folge $\underline{c}^{(n)}$. Setze $\underline{c}^{(0)} := \underline{c}$. Die weiteren Folgenglieder definieren wir induktiv, falls noch ein Koeffizient $c_k = J$ existiert. Ansonsten setzen wir $\varphi(\underline{c}) := c^{(n)}$.

Sei $i_n := \min\{\, 0 < k < d \mid c_i^{(n)} = J \,\}$ und

$$j_n := \max\{\, i_n \leqslant k := i_n + 2m < d \mid c_\ell = J \text{ für alle } i_n \leqslant \ell := i_n + 2m \leqslant k \,\}.$$

Sei

$$r_n := \begin{cases} 1 & \text{falls } i_n = 1 \text{ und } j_n = d-1 \\ c^{(n)}_{j_n+2} & \text{falls } i_n = 1 \text{ und } j_n < d-1 \\ c^{(n)}_{i_n-2} & \text{falls } i_n > 1 \text{ und } j_n = d-1 \\ c^{(n)}_{i_n-2} & \text{falls } i_n > 1 \text{ und } j_n < d-1 \text{ und } c^{(n)}_{i_n-2} = c^{(n)}_{j_n+2} \\ c^{(n)}_{i_n-2} & \text{falls } i_n > 1 \text{ und } j_n < d-1 \text{ und } c^{(n)}_{i_n-2} \neq c^{(n)}_{j_n+2} \end{cases}$$

und sei

$$c_k^{(n+1)} := \begin{cases} r_n & \text{falls } i_n \leqslant k \leqslant j_n \text{ und } c_k^{(n)} = J \\ c_k^{(n)} & \text{sonst.} \end{cases}$$

Mit diesen Definitionen gelten für jedes $\underline{e} := \underline{c}^{(n)}$ die folgenden Aussagen.

(1) Ist $c_k \in \{-1, 1\}$, dann ist $c_k = e_k$.
(2) Ist k Jokerposition von $\underline{c}$, so ist k Jokerposition von $\underline{e}$.
(3) Es gibt kein k, so daß $k-1$, k und $k+1$ Jokerpositionen von $\underline{e}$ sind.
(4) Sind k und $k+1$ Jokerpositionen von $\underline{e}$, dann ist k Jokerposition von $\underline{c}$.
(5) Ist k Jokerposition aber weder $k-1$ noch $k+1$ Jokerposition von $\underline{e}$, dann ist k Jokerposition von $\underline{c}$.

Beweis über Induktion nach n. Man sieht leicht, daß für $\underline{c} = \underline{c}^{(0)}$ alle Aussagen gelten.

Die Aussagen (1) und (2) sind klar, da nur Koeffizienten $c_k = J$ ersetzt werden.

Zu (3): Angenommen es gäbe ein k, so daß $k-1$, k und $k+1$ Jokerpositionen von $\underline{c}^{(n+1)}$ aber nicht von $\underline{c}^{(n)}$ sind.

Es gilt $c_\ell^{(n)} = c_k^{(n+1)}$ für $\ell < i_n$ und für $\ell > j_n$.

Ist $i_n < j_n$, dann ist $i_n - 1$ keine Jokerposition von $\underline{c}^{(n+1)}$. Angenommen $i_n - 1$ wäre eine Jokerposition, dann ist $i_n > 1$. Somit gilt $c_{i_n}^{(n+1)} = c_{i_n-2}^{(n)} \in \{-1, 1\}$. Es folgt $c_{i_n}^{(n+1)} \cdot c_{i_n-2}^{(n+1)} = 1$, ein Widerspruch.

Ebenso ist für $i_n < k := i_n + 2m \leqslant j_n$ auch i_{k-1} keine Jokerposition von $\underline{c}^{(n+1)}$: Es gilt $c_{k-2}^{(n)} = c_k^{(n)} = J$, also $c_{k-2}^{(n+1)} = c_k^{(n+1)} = r_n$. Somit kann k keine Jokerposition sein.

Genau dann sind j_n und j_{n+1} Jokerpositionen von $\underline{c}^{(n+1)}$, wenn $1 < i_n$ und $j_n < d-1$ und $c_{i_n-2}^{(n)} \neq c_{j_n+2}^{(n)}$. Seien j_n und $j_n + 1$ Jokerpositionen. Dann ist $j_n < d-1$. Da j_n Jokerposition ist, ist $c_{j_n}^{(n+1)} \neq c_{j_n+2}^{(n+1)}$, es folgt $c_{i_n-2}^{(n)} \neq c_{j_n+2}^{(n)}$. Die Umkehrung ist nach Definition von r_n trivial.

Ist $j_n < d-2$, dann ist $j_n + 2$ keine Jokerposition von $\underline{c}^{(n+1)}$. Andernfalls gälte $c_{j_n+2} = J$ nach Definition von Q, ein Widerspruch zur Definition von j_n.

Zusammenfassend erhalten wir, daß (3) für $\underline{c}^{(n+1)}$ gilt, da (3) für $\underline{c}^{(n)}$ nach Induktionsvoraussetzung gilt.

Aus der Gültigkeit von (3) und daraus, daß $\ell - 1$ für $i_n \leqslant \ell := i_n + 2m \leqslant j_n$ keine Jokerposition ist, folgt die Gültigkeit von (4).

Die Aussage (5) folgt daraus, daß durch das Ersetzen des Koeffizienten c_k höchstens $k-1$ und $k+1$ Jokerpositionen werden können.

Seien $\varphi(\underline{c}) = \varphi(\underline{e}) =: \underline{m}$ für $\underline{c}, \underline{e} \in Q$. Dann ist $\underline{c} = \underline{e}$. Sind $c_k, e_k \in \{-1, 1\}$, dann gilt $c_k = m_k = e_k$. Sei o. B. d. A. $c_k = J$. Ist $k-1$ und $k+1$ keine Jokerposition von $\underline{m}$, dann ist k auch Jokerposition von $\underline{e}$ und somit $e_k = J$. Sind k und $k+1$ Jokerpositionen von $\underline{m}$, dann ist k auch Jokerposition von $\underline{e}$ und somit $e_k = J$. Angenommen k und $k-1$ sind Jokerpositionen von $\underline{m}$, dann ist $k-1$ Jokerposition von $\underline{c}$ und somit $c_k \neq J$, ein Widerspruch.

Also ist φ eine injektive Abbildung. Somit folgt

$$|\varphi(Q)| \leqslant \{c_{0,d}\} \times \{-1, 1\}^{d-1} \times \{1\} = 2^d - 1. \quad \square$$

Beispiel 3.46.
Zum besseren Verständnis seien hier einige Beispiele zu Definitionen des Beweises angegeben.

Das Koeffiziententupel $\underline{c} := c^{(n)}$ hat für $i = i_n$ sowie $j = j_n$ und unter Vernachlässigung der Fälle $i = 1$ sowie $j = d-1$ die Form

$$(c_0, \ldots, c_{i-2}, c_{i-1}, J, -c_{i-1}, J, c_{i-1}, \ldots, J, c_j, c_{j+1} \ldots, c_d),$$

wobei $c_0, \ldots, c_{i-1}, c_{j+1} \in \{-1, 1\}$. Wir ersetzen die Koeffizienten $c_k = J$ entweder durch c_{i-2} oder c_{j+2}.

Im folgenden wird zu jeder möglichen Definition von r_n ein Beispiel gebracht.

- $(1,\ J,-1,\ J,\ 1,\ J,-1) \rightsquigarrow (1,\ 1,-1,\ 1,\ 1,\ 1,-1)$
- $(1,\ J,-1,-1,-1,-1,-1) \rightsquigarrow (1,-1,-1,-1,-1,-1,-1)$
- $(1,-1,\ 1,-1,\ 1,\ J,-1) \rightsquigarrow (1,-1,\ 1,-1,\ 1,-1,-1)$
- $(1,\ 1,\ 1,\ J,-1,\ 1,-1) \rightsquigarrow (1,\ 1,\ 1,\ 1,-1,\ 1,-1)$
- $(1,\ 1,\ J,-1,\ J,\ 1,-1) \rightsquigarrow (1,\ 1,\ 1,-1,\ 1,\ 1,-1)$

Bemerkung 3.47.
Im ersten Fall der Definition von r_n im Beweis des Lemmas 3.45 kann auch $r_n = -1$ gesetzt werden. Es folgt, $|Q| < 2^{d-1}$.

Die Aussage (3) über $\underline{e}$ zeigt, daß die Komplexitätsabschätzung verbessert werden kann. So gilt z. B. $(1, 1, -1, -1, 1) \notin \varphi(Q)$. Darüberhinaus ist kein Tupel der Form

$$(\ldots, 1, 1, -1, -1, 1, \ldots) \quad \text{bzw.} \quad (\ldots, -1, -1, 1, 1, -1, \ldots)$$

im Bild von Q unter φ enthalten.

Korollar 3.48.
Da $M \subseteq Q$, ist $|M| \leqslant |Q|$. $\square$

Beispiel 3.49.
In Abbildung 7 wird die Anzahl der als gut klassifizierten Koeffiziententupel des naiven Algorithmus 3.22 der des verbesserten Algorithmus 3.36 gegenübergestellt. Sei $N := \{\, c_{0,d}\} \times \{-1, 0, 1\} \times \{1\} \mid \tau(\underline{c}) = 0 \,\}$. Zum weiteren Vergleich werden auch die Anzahlen der atomaren Formeln der aus N erzeugten, strikten Typformel T'_d und der aus M erzeugten strikten Typformel S'_d (jeweils in disjunktiver Normalform) angegeben. Zusätzlich wird die Kardinalität der Menge Q aufgeführt. Ausgelassene Werte konnten aus Speicherplatzgründen (32 MByte) nicht berechnet werden.

	naiver Algorithmus			verbesserter Algorithmus			
d	3^{d-1}	$\lvert N\rvert$	$\lvert T'_d\rvert$	$\lvert M\rvert$	$\lvert S'_d\rvert$	$\lvert Q\rvert$	2^{d-1}
2	3	3	6	1	1	1	2
4	27	13	52	3	8	5	8
6	243	63	252	9	37	19	32
8	2187	321	1284	27	150	63	128
10	19683	1683	6732	83	581	211	512
12	177147	8989	35956	259	2186	715	2048
14	1594323	48639	194556	817	8071	2412	8192
16	14348907	265729	1062916	2599	29412	8189	32768
18	129140163			8323	106151	27701	131072
20	1162261467			26797	380272	93713	524288
22	10460353203			86695		317031	2097152
24	94143178827			281287		1072507	8388608

ABBILDUNG 7. Vergleich der beiden Algorithmen

Bemerkung 3.50.
Die Erzeugung der guten Koeffiziententupel mit dem Algorithmus 3.36 geschieht rekursiv durch Verlängern eines Anfangsstückes eines Koeffiziententupels. Dies kann man zu einer weiteren Verringerung der Anzahl der atomaren Formeln von T'_d benutzen. In diesem Fall baut man rekursiv eine Formel der Form

$$c_d > 0 \wedge ((c_{d-1} = -1 \wedge \dots) \vee (c_{d-2} = -1 \wedge \dots) \vee (c_{d-1} = 1 \wedge \dots))$$

auf. Zusätzlich kann man die Ungleichung $c_{0,d} \varrho\, 0$ ausklammern.

Beispiel 3.51.
Die strikte Trennungsformel $T'_6(\underline{c})$ hat in diesem Fall die Form

$$\begin{aligned} &c_0 > 0 \wedge (\\ &\quad (c_4 < 0 \wedge (c_2 > 0 \vee c_1 > 0 \wedge c_3 < 0 \vee c_1 < 0 \wedge c_3 > 0)) \vee \\ &\quad (c_5 < 0 \wedge (c_3 > 0 \wedge (c_1 < 0 \vee c_2 > 0) \vee c_1 < 0 \wedge c_2 < 0 \wedge c_4 > 0)) \vee \\ &\quad (c_5 > 0 \wedge (c_3 < 0 \wedge (c_1 > 0 \vee c_2 > 0) \vee c_1 > 0 \wedge c_2 < 0 \wedge c_4 > 0))). \end{aligned}$$

Diese Formel hat 21 atomare Formeln. In disjunktiver Normalform enthält sie 37 atomare Formeln.

3.4. EINE VERALLGEMEINERUNG VON TYPFORMELN

Eine Variante des hier vorgestellten Algorithmus benutzt Formeln, die Polynome mit einem Typ t charakterisieren.

Definition 3.52.
Sei $d > 0$ und $\chi = X^d + \sum_{i=0}^{d-1} c_i X^i \in \mathbb{Q}[c_0, \dots, c_{d-1}][X]$ normiertes, univariates Polynom mit unbestimmten Koeffizienten. Sei $-d \leqslant t \leqslant d \in \mathbb{N}$. Seien $A \subseteq \mathbb{R}^d$ die Menge aller Spezialisierungen für die $\sigma_{\underline{a}}(\chi)$ nur reelle Nullstellen besitzt. Eine quantorenfreie Formel $T_{t,d}(\underline{c})$ mit Variablen $c_0, \dots, c_{d-1}$ heißt *Typformel bezüglich des Typs* t für Polynome des Grades d, falls für jede Spezialisierung $\underline{a} \in A$

$$T_d(\underline{a}) \quad \text{gdw.} \quad \tau(\sigma_{\underline{a}}(\chi)) = t$$

gilt.

Eine genaue Dursicht der hier vorgestellten Resultate zeigt, daß sie ebenfalls verallgemeinert werden können. Die rekursive Zurückführung der Typformeln T_d sowie die Bestimmung von $c_{0,d}$ können ebenfalls in Abhängigkeit von d verallgemeinert werden. Alle wichtigen Resultate der effizienten Bestimmung von Typformeln stützen sich nur auf die Nicht-Existenz von echt komplexen Nullstellen und nicht auf den Typ 0.

Auch die Komplexitätsabschätzung 3.48 der Anzahl der Koeffiziententupel, die zur Bestimmung von einer strikten Trennungsformel herangezogen werden, bleibt bestehen. Dies folgt daraus, daß sie aus der Abschätzung der Kardinalität der Menge Q abgeleitet wurde und sich nicht auf den Typ bezieht.

KAPITEL 4

Der Eliminationsalgorithmus

4.1. Auffinden nicht-trivialer Gleichungen

Das hier geschilderte Verfahren zur reellen Quantorenelimination setzt voraus, daß in der zu eliminierenden Formel mindestens eine nicht-triviale Gleichung vorkommt. Ist dies nicht der Fall, so kann das Verfahren nicht direkt angewendet werden.

In diesem Fall werden wir nicht-triviale Gleichungen suchen, die erfüllt sind, wenn die Eingabeformel erfüllt ist. Im folgenden wird hierzu ein Verfahren dargestellt, das dieses Problem im univariaten Fall behandelt. Dabei wird eine Disjunktion aus Formeln konstruiert. Die einzelnen Argumente dieser Disjunktion werden entweder quantorenfreie Formeln oder Formeln in einer gewissen Normalform sein. Die entsprechende Umformung stammt von Weispfenning ist aber in der Originalarbeit [Wei93a] noch nicht komplett ausgeführt. Das Zulassen der zusätzlichen Relation „$\neq$“ hat in diesem Zusammenhang kaum Auswirkungen.

Im folgenden werden wir Polynome $f \in \mathbb{R}[X]$ und die durch sie gegebene Abbildung

$$f : \mathbb{R} \longrightarrow \mathbb{R} \quad \text{mit} \quad a \longmapsto \varphi_a(f)$$

miteinander identifizieren. Dabei bezeichnet φ_a den Auswertungshomomorphismus an der Stelle a.

Definition 4.1.
Sei $f \in R[X]$.

- Dann bezeichne $f \neq 0$ eine quantorenfreie Formel in den Koeffizienten, die äquivalent zu der Bedingung ist, daß f nicht-trivial, d. h. nicht das Nullpolynom, ist.
- Dann bezeichne $f = 0$ eine quantorenfreie Formel in den Koeffizienten, die äquivalent zu der Bedingung ist, daß f trivial, d. h. das Nullpolynom, ist.
- Dann bezeichne $f(\infty) > 0$ bzw. $f(-\infty) > 0$ eine quantorenfreie Formel in den Koeffizienten, die äquivalent zu der Bedingung ist, daß

$$\lim_{x \to \infty} f(x) > 0 \quad \text{bzw.} \quad \lim_{x \to -\infty} f(x) > 0.$$

Lemma 4.2.
Für $h = \sum_{i=0}^{d} a_i X^i$ seien quantorenfreie Formeln $\varphi_i(\underline{a})$ und $\psi_i(\underline{a})$ rekursiv definiert durch

- $\varphi_0(\underline{a}) :\equiv a_0 > 0$
- $\varphi_i(\underline{a}) :\equiv \Big(a_i > 0 \vee \big(a_i = 0 \wedge \varphi_{i-1}(\underline{a})\big)\Big)$ *für* $0 < i \leqslant d$.

bzw.

- $\psi_0 :\equiv a_0 > 0$.
- $\psi_i(\underline{a}) :\equiv \Big(a_i > 0 \vee \big(a_i = 0 \wedge \psi_{i-1}(\underline{a})\big)\Big)$ *für* $0 < i \leqslant d$, *i gerade.*
- $\psi_i(\underline{a}) :\equiv \Big(a_i < 0 \vee \big(a_i = 0 \wedge \psi_{i-1}(\underline{a})\big)\Big)$ *für* $0 < i \leqslant d$, *i ungerade.*

Dann gilt

(1) $h(\infty) > 0$ *gdw.* $\varphi_d(\underline{a})$ *sowie*
(2) $h(-\infty) > 0$ *gdw.* $\psi_d(\underline{a})$.

Beweis. Der Beweis folgt aus den beiden folgenden Äquivalenzen.

(1) φ_d gdw. $\operatorname{sign}(\mathrm{HC}(f)) > 0$
(2) ψ_d gdw. $\operatorname{sign}(\mathrm{HC}(f)) > 0 \wedge \deg(f) = 2 \cdot e$ für $e \in \mathbb{N}$. □

Lemma 4.3.
Sei $f = \sum_{i=0}^{d} a_i X^i \in \mathbb{R}[X]$. *Dann gilt*

$$f \neq 0 \quad \textit{gdw.} \quad \bigvee_{j=0}^{d} a_j \neq 0 \qquad \textit{sowie} \qquad f = 0 \quad \textit{gdw.} \quad \bigwedge_{j=0}^{d} a_j = 0. \quad \square$$

Lemma 4.4.
Seien $g_1, \ldots, g_s, h_1, \ldots, h_t \in \mathbb{R}[\underline{U}][X]$ *parametrisierte Polynome und* $\underline{u}$ *eine Spezialisierung der Parameter. Sei*

$$\varphi(\underline{u}, x) :\equiv \Big(\bigwedge_{j=1}^{s} g_j(\underline{u}, u) > 0\Big)$$

und $\psi(\underline{u})$ *äquivalent zu* $\exists x(\varphi(\underline{u}, x))$. *Dann gilt*

$$\exists x\left(\varphi(\underline{u}, x) \wedge \Big(\bigwedge_{k=1}^{t} h_k(\underline{u}, x) \neq 0\Big)\right) \quad \textit{gdw.} \quad \psi(\underline{u}) \wedge \bigwedge_{k=1}^{t} h_k \neq 0.$$

Beweis. Gelte die linke Seite der Äquivalenz. Dann gibt es zumindest einen Punkt x_0 mit $\bigwedge h_k(\underline{u}, x_0) \neq 0$, somit ist kein h_k das Nullpolynom. Es folgt die Gültigkeit der rechten Seite der Äquivalenz, da die Nichttrivialitätsbedingungen unabhängig von X sind.

Gilt umgekehrt die rechte Seite der Äquivalenz an einem Punkt x_0, so ist keines der h_k das Nullpolynom. Sei

$$N := \bigcup \{\, x \in \mathbb{R} \mid f_i(\underline{u}, x) = 0 \text{ für ein } i \,\},$$

dann ist N endlich und $\bigwedge h_k(\underline{u}, x) \neq 0$ ist für alle $x \in N$ nicht erfüllt, aber an allen $\mathbb{R} \setminus N$ erfüllt. Da ψ an x_0 gilt, gilt $\bigwedge g_j(\underline{u}, x_0) > 0$. Aus der Stetigkeit von Polynomen folgt, daß $\bigwedge g_j(\underline{u}, x_0) > 0$ für jedes $x \in I$ gilt, wobei I ein offenes, nicht leeres Intervall ist. Somit gilt

$$\varphi(\underline{u}, x) \wedge \Big(\bigwedge_{k=1}^{t} h_k(\underline{u}, x) \neq 0\Big)$$

an allen Punkten $x \in I \setminus N$. Da I unendlich ist, ist $I \setminus N \neq \emptyset$ und somit gilt die linke Seite der Äquivalenz. □

Definition 4.5.
Sei $f = \sum_{i=0}^{d} a_i X^i \in \mathbb{R}[\underline{U}][X]$ parametrisiertes Polynom. Dann ist die *formale Ableitung* von f nach der Variablen X folgendermaßen definiert

$$\frac{\partial f}{\partial X} := \sum_{i=0}^{d-1} a_{i+1} X^i.$$

Satz 4.6.
Seien $g_1, \ldots, g_s, h_1, \ldots, h_t \in \mathbb{R}[\underline{U}][X]$ parametrisierte Polynome und $\underline{u}$ eine Spezialisierung der Parameter. Seien

$$\varphi_1 \equiv \bigwedge_{j=1}^{s} g_j(\underline{u}, \infty) > 0,$$

$$\varphi_2 :\equiv \bigwedge_{j=1}^{s} g_j(\underline{u}, -\infty) > 0,$$

$$\varphi_3 :\equiv \bigvee_{\ell=1}^{s} \exists x \Big(\bigwedge_{j=1}^{s} g_j(\underline{u}, x) > 0 \wedge \frac{\partial g_\ell}{\partial X}(\underline{u}, x) = 0 \Big),$$

$$\varphi_4 :\equiv \bigvee_{1 \leqslant k < \ell \leqslant s} \exists x \Big(\bigwedge_{j=1}^{s} g_j(\underline{u}, x) > 0 \wedge (g_k - g_\ell)(\underline{u}, x) = 0 \Big),$$

sowie

$$\varphi_3' :\equiv \bigvee_{\ell=1}^{s} \exists x \Big(\bigwedge_{j=1}^{s} g_j(\underline{u}, x) > 0 \wedge \frac{\partial g_\ell}{\partial X} \neq 0 \wedge \frac{\partial g_\ell}{\partial X}(\underline{u}, x) = 0 \Big),$$

$$\varphi_4' :\equiv \bigvee_{1 \leqslant k < \ell \leqslant s} \exists x \Big(\bigwedge_{j=1}^{s} g_j(\underline{u}, x) > 0 \wedge (g_k - g_\ell)(\underline{u}, x) = 0 \wedge (g_k - g_\ell) \neq 0 \Big).$$

Dann gilt

$$\exists x \big(\bigwedge_{j=1}^{s} g_j(\underline{u}, x) > 0 \wedge \bigwedge_{k=1}^{t} h_k(\underline{u}, x) \neq 0 \big) \text{ gdw. } \big(\bigwedge_{k=1}^{t} h_k \neq 0 \big) \wedge \big(\varphi_1 \vee \varphi_2 \vee \varphi_3 \vee \varphi_4 \big)$$
$$\text{gdw. } \big(\bigwedge_{k=1}^{t} h_k \neq 0 \big) \wedge \big(\varphi_1 \vee \varphi_2 \vee \varphi_3' \vee \varphi_4' \big).$$

Beweis. Nach Lemma 4.4 reicht es aus, die folgenden Äquivalenzen zu beweisen.

$$(1)\ \exists x \Big(\bigwedge_{j=1}^{s} g_j(\underline{u}, x) > 0 \Big) \quad \text{gdw.} \quad (2)\ \varphi_1 \vee \varphi_2 \vee \varphi_3 \vee \varphi_4 \quad \text{gdw.} \quad (3)\ \varphi_1 \vee \varphi_2 \vee \varphi_3' \vee \varphi_4'$$

Zum Beweis (2)⇒(1) bzw. (3)⇒(1): Gilt φ_1 für gegebene Parameter $\underline{u}$, so gilt (1) für ein genügend großes x. Analog dazu gilt (1) für ein genügend kleines x, falls φ_2 gilt. Da (1) in jeder Konjunktion in φ_3, φ_4, φ_3' oder φ_4' als Subkonjunktion enthalten ist, gilt (1) falls eine der Disjunktionen φ_3, φ_4, φ_3' und φ_4' gilt.

Zum Beweis (1)⇒(2) bzw. (1)⇒(3): Die Erfüllungsmenge von (1) ist der Schnitt aller Erfüllungsmengen von $g_j > 0$. Diese sind, da die g_j Polynome sind, endliche Vereinigungen von offenen Intervallen. Somit ist auch die Erfüllungsmenge von (1) eine Vereinigung endlich vieler – möglicherweise unechter – offener Intervalle. Enthält sie das Intervall $]a, \infty[$, $]-\infty, a[$ oder $]-\infty, \infty[$, so ist φ_1 oder φ_2 erfüllt. Ansonsten enthält die Erfüllungsmenge mindestens ein maximales, beschränktes Intervall $]a, b[$

mit $a \neq b$. Da $]a,b[$ maximal ist, sind a und b Nullstellen mindestens eines Polynoms g.

Ist g ein Polynom mit den Nullstellen a und b, so existiert nach dem Satz von Rolle ein

$$x \in]a,b[\quad \text{mit} \quad \frac{\partial g}{\partial X}(\underline{u},x) = 0.$$

In diesem Fall gilt also eine der Konjunktionen der Disjunktion φ_3. Da g zwei Nullstellen besitzt, gilt $\deg(g) > 1$ und somit $\frac{\partial g_l}{\partial X} \neq 0$. Also gilt auch φ_3'.

Gibt es kein solches Polynom g, so gibt es zwei Polynome $g_a \neq g_b$ mit

$$g_a(a) = 0 = g_b(b) \quad \text{aber} \quad g_a(b) \neq 0 \neq g_b(a).$$

Es gilt $g_a(b) > 0$. Angenommen es wäre $g_a(b) < 0$. Da $g_a(x)$ für alle $x \in]a,b[$ gäbe nach dem Zwischenwertsatz ein $x_0 \in]x,b[$ mit $g_a(b) = 0$, ein Widerspruch zu der Voraussetzung. Ebenso ist $g_b(a) > 0$. Also gilt

$$(g_a - g_b)(a) = -g_b(a) < 0 \quad \text{und} \quad (g_a - g_b)(b) = g_a(b) > 0.$$

Folglich muß es nach dem Zwischenwertsatz eine Stelle $x \in]a,b[$ geben, so daß $(g_a - g_b)(\underline{u},x) = 0$. Also gilt eine der Konjunktionen in der Disjunktion φ_4. Da g_a und g_b unterschiedliche Nullstellen haben, gilt insbesondere $g_a - g_b \neq 0$. Somit gilt auch φ_4'. □

Bemerkung 4.7.
In Satz 4.6 kann der Laufbereich des Index ℓ aus der Disjunktion in φ_3 bzw in φ_3' unter der Berücksichtigung des Grades von g_ℓ eingeschränkt werden. Es genügt, die $1 \leqslant \ell \leqslant s$ zu betrachten, die $\deg(g_\ell) > 1$ erfüllen.

Beweis. Die Formel φ_3 bzw. φ_3' deckt den Fall ab, daß die Endpunkte des maximalen Intervalls $]a,b[$ durch die Nullstellen eines Polynoms gegeben sind. Somit muß g_ℓ ein Polynom mit $\deg(g_\ell) > 1$ sein, da es sonst nicht einmal zwei Nullstellen besitzt. □

4.2. Die Dimension von Idealen

Die folgende Definition der Dimension eines Ideals und das Resultat des Lemmas 4.10 sind [BW93b] entnommen. Dort findet man ebenso Algorithmen zur Berechnung der Dimension eines Ideales und zur Bestimmung maximal unabhängiger Variablenmengen.

Definition 4.8.
Sei $I \subseteq K[X_1,\ldots,X_n]$ echtes Ideal und $U := \{U_1,\ldots,U_m\} \subseteq \{X_1,\ldots,X_n\}$. Dann heißt $\{U_1,\ldots,U_m\}$ *unabhängig* modulo I, falls $I \cap K[U_1,\ldots,U_m] = \{0\}$. Ist $\{U_1,\ldots,U_m\}$ unabhängig modulo I und gibt es keine echte Obermenge von U, die unabhängig modulo I ist, so heißt U *maximal unabhängig* modulo I.

Die Dimension eines Ideals ist definiert durch

$$\dim(I) := \max\{\,|U| \mid U \subseteq \{X_1,\ldots,X_n\} \text{ unabhängig modulo } I\,\}.$$

Zusätzlich definieren wir $\dim(K[\underline{X}]) := -1$.

Korollar 4.9.
Sei $I \subseteq K[X_1,\ldots,X_n]$ Ideal.

(1) *Es gilt $-1 \leqslant \dim(I) \leqslant n$.*
(2) *Ein Ideal I hat genau dann die Dimension n, wenn $I = \{0\}$.*

Beweis. Aussage (1) ist nach Definition klar, da Teilmengen von $\{X_1, \ldots, X_n\}$ betrachtet werden.

Zu (2): Angenommen $\dim(I) = n$. Die einzige unabhängige Menge der Kardinalität n ist $\{X_1, \ldots, X_n\}$. Also ist $I \cap K[X_1, \ldots, X_n] = I$. Es folgt $I = \{0\}$. Die Replikation ist trivial. □

Lemma 4.10.
Sei $I \in \mathbb{R}[X_1, \ldots, X_n]$ Ideal, $\{X_1, \ldots, X_d\}$ maximal unabhängig modulo I. Dann ist das von der Menge I im Ring $\mathbb{R}(X_1, \ldots, X_d)[X_{d+1} \ldots X_n]$ erzeugte Ideal nulldimensional.

Beweis. Vergleiche [BW93b] Lemma 7.47, sowie die Definitionen der dort benutzten Begriffe. □

Bemerkung 4.11.
Sei S ein Gröbnersystem des Ideals $\mathrm{Id}(F)$. Dann können in jedem Ast des Gröbnersystems die Dimension und maximal unabhängige Mengen bestimmt werden. Das Resultat aus Lemma 4.10 kann dann in einem etwas anderen Sinne auch auf parametrisierte Polynome angewendet werden. In diesem Fall betrachtet man die Variablen einer maximal unabhängigen Menge als zusätzliche Parameter. Sei P die Polynommenge des jeweiligen Astes des Gröbnersystems. Sei P' die Polynommenge, die aus P entsteht, indem man die Polynome P aus in natürlicher Weise als parametrisierte Polynome in $\mathbb{Q}[U_1, \ldots, U_m, X_1, \ldots, X_d][X_{d+1}, \ldots, X_n]$ auffaßt. Dann ist im allgemeinen für eine beliebige Spezialisierung $\underline{a} \in \mathbb{Q}^{m+d}$ weder $\sigma_{\underline{a}}(P)$ eine Gröbnerbasis für $\mathrm{Id}(\sigma_{\underline{a}}(P))$ noch ist $\mathrm{Id}(\sigma_{\underline{a}}(P))$ nulldimensional.

Beispiel 4.12.
Sei $f = UX^2Y + XY \in \mathbb{Q}[U][X, Y]$, $0 \neq a \in \mathbb{Q}$ eine Spezialisierung. Dann ist $\dim(\mathrm{Id}(\sigma_a(f))) = 1$.Die Menge $\{Y\}$ ist eine maximal unabhängig modulo $\mathrm{Id}(\sigma_a(f))$. Betrachtet man f im Polynomring $\mathbb{Q}[U, Y][X]$, so ist $\dim(\mathrm{Id}(f)) = 0$. Faßt man aber U und Y als Parameter auf, dann gilt zwar $\dim(\mathrm{Id}(\sigma_a(f))) = 0$ für eine Spezialisierung $a \in Q^2$ mit $a_1 \cdot a_2 \neq 0$ aber nicht für eine Spezialisierung mit $a_1 \cdot a_2 = 0$, z. B. $0 \neq a_1 \in \mathbb{Q}$ beliebig und $a_2 = 0$.

4.3. Die Normalform zur Elimination

Das eigentliche Eliminationsverfahren, wie es hier vorgestellt wird, kann nur eine existenzquantifizierte Konjunktion über atomare Formeln mit den Relationen „$<$“, „$=$“ und „$\neq$“ eliminieren. Im folgenden wird gezeigt, wie die Quantoren einer beliebigen Formel erster Stufe eliminiert werden können.

Bemerkung 4.13.
Eine Formel heißt in positiver Normalform, falls sie den Operator „$\neg$“ nicht enthält. Jede Formel über der Sprache der angeordneten Ringe läßt sich in positive Normalform überführen, falls man – wie wir – alle ordnungstheoretischen Relationen zuläßt. Dazu zieht man zunächst unter der Anwendung der Regel von de Morgan alle Negationen bis vor die atomaren Formeln. Anschließend zieht man die Negationen in die atomaren Formeln. Dabei ändert sich entsprechend die Relation der atomaren Formel. Diese Änderung ist in Abbildung 8 zusammengefaßt.

Bemerkung 4.14.
Jede Formel über der Sprache der geordneten Ringe läßt sich in eine äquivalente Formel überführen, die keine Negationen enthält, und die in den atomaren Formeln nur die Relationen $=$, $\neq$ und $>$ enthält. Dazu werden in der gegebenen Formel die atomaren Formeln wie in der Abbildung 9 angegeben umgeformt. Wir sprechen in diesem Fall von *normalisierten Relationen*. Formt man dabei nur die atomaren

aus	wird
$\neg(t < 0)$	$(t \geqslant 0)$
$\neg(t \leqslant 0)$	$(t > 0)$
$\neg(t = 0)$	$(t \neq 0)$
$\neg(t \neq 0)$	$(t = 0)$
$\neg(t \geqslant 0)$	$(t < 0)$
$\neg(t > 0)$	$(t \leqslant 0)$

ABBILDUNG 8. Änderung der atomaren Formeln bei Negation

aus	wird
$(t < 0)$	$(-t > 0)$
$(t \leqslant 0)$	$(-t > 0) \vee (t = 0)$
$(t = 0)$	$(t = 0)$
$(t \neq 0)$	$(t \neq 0)$
$(t \geqslant 0)$	$(t > 0) \vee (t = 0)$
$(t > 0)$	$(t > 0)$

ABBILDUNG 9. Normalisierung der atomaren Formeln

Formeln um, die abhängig von einer gegebenen Variablenmenge X sind, so sprechen wir von bezüglich X normalisierten Relationen.

Algorithmus 4.15.
Seien

$$\alpha :\equiv \exists \underline{x}\big(\bigwedge_{i=1}^{r} f_i(\underline{u}, \underline{x}) = 0 \wedge \bigwedge_{j=1}^{s} g_j(\underline{u}, \underline{x}) > 0 \wedge \bigwedge_{k=1}^{t} h_k(\underline{u}, \underline{x}) \neq 0\big)$$

sowie

$$\beta :\equiv \big(\bigwedge_{\ell=1}^{q} p_\ell \ \varrho_\ell \ 0\big),$$

wobei $\varrho_\ell \in \{<, \leqslant, =, \neq, \geqslant, >\}$. *Dann bezeichne* `qenf`$(\beta, \alpha)$ *einen Algorithmus, der eine quantorenfreie Formel* ψ *zurückgibt, so daß*

$$\beta \wedge \alpha \Longleftrightarrow \beta \wedge \psi.$$

Unter dieser Voraussetzung eliminiert der in Abbildung 10 *gezeigte Algorithmus die Quantoren einer Formel erster Stufe über der Sprache der angeordneten Ringe.*

Beweis. Der Algorithmus terminiert, da alle Schleifen über endliche Mengen laufen. Die Statements vor der `FOR`-Schleife ab Zeile 11 sind Äquivalenzumformungen. Schleifeninvariante der `FOR`-Schleife von Zeile 11 bis Zeile 32 ist

$$\big(Q_1 \underline{x}_1 \ldots Q_b \underline{x}_b \psi\big) \equiv \varphi.$$

Dabei bezeichne $Q_i \underline{x}_i$ ein Block von Existenzquantoren bzw. einen Block von Allquantoren mit den gebundenen Variablen $\underline{x}_i$. Zu Beginn der `FOR`-Schleife ist die Gültigkeit klar. In jedem Durchlauf wird jeweils ein Quantorenblock eliminiert. Wird dabei ein Allquantorenblock eliminiert, so wird dieser unter Anwendung von

$$\forall x(\varphi) \quad \text{gdw.} \quad \neg\exists x(\neg\varphi)$$

Eingabe: Eine Formel φ erster Stufe über der Sprache der angeordneten Ringe.

Ausgabe: Eine quantorenfreie zu φ äquivalente Formel ψ erster Stufe über der Sprache der angeordneten Ringe.

```
1  PROCEDURE qe
2  BEGIN
3      φ' := „Zu φ äquivalente Formel ohne die Wahrheitswerte TRUE und
           FALSE“
4      φ' := „Zu φ' äquivalente Formel mit Quantoren und ausschließlich den
           Operationen ∧, ∨ und ¬.“
5      φ' := „Zu φ' äquivalente Formel in deren atomaren Formeln die rechte
           Seite der Relation gleich 0 ist.
6      φ' := „Zu φ' äquivalente Formel in pränexer Normalform mit b Quanto-
           renblöcke.“
7      FOR 1 ⩽ ℓ ⩽ b DO
8          Q_ℓ := „Der ℓ-te (von innen gezählt) Quantorenblock von φ'.“
9      END
10     ψ := „Die Matrix von φ'.“
11     FOR 1 ⩽ ℓ ⩽ b DO
12         „Benenne Variablen so um, daß x die von Q_ℓ gebundenen Variablen
               sind.“
13         „Benenne Variablen so um, daß u die restlichen Variablen sind.“
14         IF „Q_ℓ ist Block von Existenzquantoren“ THEN
15             ψ := ¬ψ
16         END
17         ψ := „Zu ψ äquivalente Formel in positiver Normalform.“
18         ψ := „Normalisiere die Relationen der atomaren Formeln bezüglich
               den gebundenen Variablen.“
19         γ := „Eine DNF von ψ (ohne die Normalisierung der Relationen zu
               zerstören).“
20         ψ := FALSE
21         FOR EACH Konjunktion γ' aus γ DO
22             Sei β die Konjunktion der atomaren Formeln aus γ', die nicht
                   von x abhängen.
23             Sei α die existenzquantifizierte Konjunktion der atomaren
                   Formeln aus γ', die von x abhängen.
24             γ' := qenf(β, α)
25             ψ := ψ ∨ γ'
26         END
27         IF „Q_ℓ ist Block von Existenzquantoren“ THEN
28             ψ := ¬ψ
29         END
30         ψ := „positive Normalform ohne TRUE und FALSE von ψ.“
31         IF ψ = TRUE OR ψ = FALSE THEN RETURN ψ END
32     END
33     RETURN ψ
34 END qe
```

ABBILDUNG 10. Elimination einer beliebigen Formel

eliminiert. Dieses wird im Algorithmus durch die `IF`-Anweisungen 14 und 27 verwirklicht. Beim Aufruf von `qenf` in Zeile 24 kann γ' in die entsprechende Normalform gebracht werden, da die Relationen normalisiert sind. □

Bemerkung 4.16.
Die Normalisierung der Relation muß vor dem Bilden der disjunktiven Normalform durchgeführt werden, da sie eine disjunktive Normalform zerstören kann. Das Bilden der disjunktiven Normalform kann dagegen ohne Zerstörung der Normalisierung durchgeführt werden. Zum Beispiel erhält die naive Bildung einer disjunktiven Normalform durch iterierte Anwendung des Distributivgesetzes die Normalisierung der Relationen. Im Gegensatz dazu erhält die Normalisierung der Relationen die positive Normalform.

4.4. Elimination eines Existenzquantorenblocks

In diesem Kapitel werden wir die Elimination der Quantoren einer Formel mit folgender Form vorstellen.

$$\exists \underline{x}\Big(\bigwedge_{i=1}^{r} f_i(\underline{u},\underline{x}) = 0 \wedge \bigwedge_{j=1}^{s} g_j(\underline{u},\underline{x}) > 0 \wedge \bigwedge_{k=1}^{t} h_k(\underline{u},\underline{x}) \neq 0 \Big).$$

Diese Form nennen wir *Eliminationsnormalform*. Wir werden jedoch bei der Elimination einer Formel in Eliminationsnormalform zusätzlich eine Konjunktion atomarer Formeln der Form

$$\Big(\bigwedge_{\ell=1}^{q} p_\ell(\underline{u}) \ \varrho_\ell \ 0 \Big),$$

berücksichtigen, wobei $\varrho_\ell \in \{<, \leqslant, =, \neq, \geqslant, >\}$ und $p_\ell(\underline{u})$ konstant bezüglich $\underline{x}$ ist. Nach Abschnitt 4.3 reicht dies aus, um jede Formel erster Stufe zu eliminieren.

Zur Elimination fassen wir die Terme auf der linken Seite der Relationen als parametrisierte Polynome über $\underline{X}$, dargestellt als Polynome im Polynomring $\mathbb{Z}[\underline{U}][\underline{X}]$ auf.

Abbildung 11 gibt einen Überblick über den Algorithmus. Im folgenden werden wir alle Schritte ausführlicher darstellen.

Gibt es in der Formel keine atomare Formeln der Form $(f_i = 0)$, d. h. $r = 0$, so wird wie in Abschnitt 4.1 beschrieben vorgegangen. Dazu werden zunächst alle bis auf den innersten Quantor, der o. B. d. A. x_n bindet, vernachlässigt. Unter Anwendung des Satzes 4.6 wird die Matrix der Formel in die Form (3) dieses Satzes umgeformt. Die einzelnen Konjunktionen der Disjunktionen φ_3' und φ_4' werden unter der Berücksichtigung von

$$\big(\bigwedge_{k=1}^{t} h_k \neq 0 \big)$$

mittels rekursiven Aufrufen des Eliminationsverfahrens eliminiert. Sei ψ die Konjunktion der Ergebnisse dieser Eliminationen. Nach Elimination von x_n wird mit einem zweiten rekursiven Aufruf angewendet auf

$$\exists x_1 \ldots \exists x_n \Big(\big(\bigwedge_{k=1}^{t} h_k \neq 0 \big) \wedge \varphi_1 \wedge \varphi_2 \wedge \psi \Big)$$

der restliche Quantorenblock eliminiert.

Ist $r \neq 0$, d. h. eine nicht-triviale Gleichung ist in der Eingabe vorhanden, so wird aus der Konjunktion der atomarem Formeln $(p_\ell \ \varrho_\ell \ 0)$ eine initiale Bedingung abgeleitet. Sei A die Menge der atomaren Formeln in dieser Konjunktion. Seien o. B. d. A. die

p_ℓ so geordnet, daß für $1 \leqslant \ell \leqslant q'$ die Relation ϱ_ℓ aus $\{<, =, \neq, >\}$ sind. Dann ist die initiale Bedingung bestimmt durch

$$G_I := \{\, p \mid (p = 0) \in A \,\}$$
$$R_I := \{\, p \mid (p < 0) \in A \text{ oder } (p \neq 0) \in A \text{ oder } (p > 0) \in A \,\}.$$

Atomare Formeln mit Relationen aus $\{\leqslant, \geqslant\}$ werden also nicht berücksichtigt. Danach wird ein reduziertes, grüngefärbtes Gröbnersystem von $\{f_1, \ldots, f_r\}$ über der initialen Bedingung (R_I, G_I) berechnet. An dieser Stelle kann sich die Berücksichtigung der atomaren Formeln durch die Bildung der initialen Bedingung positiv auswirken. Man betrachte zum Beispiel die Formel

$$\exists x(ax^2 + bx + c = 0 \wedge a > 0).$$

Bei Berücksichtigung von $(a > 0)$ durch die initiale Bedingung $(a \neq 0)$ enthält das berechnete Gröbnersystem nur einen Fall. Vernachlässigt man die atomare Formel $(a > 0)$, so müssen 4 Fälle des Gröbnersystems bearbeitet werden, von denen 3 durch die Bedingung $(a \neq 0)$ ausgeschlossen werden.

Im weiteren werden zunächst alle Äste $((R, G), P)$ des Gröbnersystems getrennt behandelt. Als erstes wird die Dimension $\dim(I)$ sowie eine maximale Menge unabhängiger Variablen des Ideals $I := \mathrm{Id}(\{\, f_i \mid 1 \leqslant i \leqslant r\})$ berechnet. Nach Bemerkung 1.10 sind die Dimension und die Menge der maximal unabhängigen Variablenmengen eindeutig in jedem Ast eines Gröbnersystems bestimmt. Nach der Dimension unterscheiden wir vier Fälle bei der Vorgehensweise zur Elimination.

Im ersten Fall $\dim(I) = -1$ gilt definitionsgemäß $I = \mathbb{R}[\underline{U}][\underline{X}]$. Da in diesem Fall $1 \in I$ ist, hat I weder reelle noch komplexe Nullstellen. Somit ist das Ergebnis der Elimination im jeweiligen Ast des Gröbnersystems `FALSE`.

Im zweiten Fall $\dim(I) = 0$ zählen wir die Anzahl der reellen Nullstellen des Ideals I unter den Nebenbedingungen

$$\bigwedge_{j=1}^{s} g_j(\underline{u}, \underline{x}) > 0 \wedge \bigwedge_{k=1}^{t} h_k(\underline{u}, \underline{x}) \neq 0.$$

Da die Nullstellen eines Ideals auch Nullstellen der erzeugenden Menge sind, folgt aus der Existenz einer reellen Nullstelle von I die Existenz eines reellen $\underline{x}$, das die Matrix der zu eliminierenden Formel erfüllt. Somit ist die Existenz von reellen Nullstellen des Ideals unter den Nebenbedingungen äquivalent zur Gültigkeit der Existenzaussage.

Zum Zählen der reellen Nullstellen benutzen wir das Resultat des Korollars 2.18 aus Kapitel 2. Beim Zählen der reellen Nullstellen benötigen wir, daß P reduzierte Gröbnerbasis ist. Mit Hilfe von P wird in der Implementierung die Arithmetik in $K[\underline{X}]/I$ durchgeführt. Sei Z die Anzahl der Nullstellen von I, die die gegebenen Nebenbedingungen erfüllen. Dann gilt

$$Z > 0 \Leftrightarrow \tau\Big(\prod_{e \in E_s} \chi_e\Big) \neq 0,$$

wobei χ_e das charakteristische Polynom der Matrix Q_{h^e} bezeichnet. Wir benutzen Typformeln wie in Kapitel 3 eingeführt, um eine quantorenfreie Formel für die Typbedingung des charakteristischen Polynoms zu erhalten. Sei

$$\chi := \prod_{e \in E_s} \chi_e = Y^d + \sum_{\ell=0}^{d-1} c_\ell Y^\ell.$$

Dann hat χ wie in Abschnitt 2.1 erwähnt nur reelle Nullstellen. Somit gilt

$$\tau(\chi) = 0 \quad \text{gdw.} \quad T_d(c_0, \ldots, c_{d-1}).$$

Zur Elimination berechnet man also das Produkt der charakteristischen Polynome. Die Koeffizienten dieses Polynoms sind Elemente aus $\mathbb{Q}(\underline{u})$. Die Nenner der Koeffizienten sind jedoch unter jeder Spezialisierung, die die gegebenen Bedingung des Gröbnersystems erfüllen, ungleich 0. Dies ist durch die Konstruktion der Algorithmen zur Bestimmung der charakteristischen Polynome gewährleistet, da die höchsten Koeffizienten der Polynome in P nach den Bedingungen des Gröbnersystems nicht verschwinden. Die einzelnen Koeffizienten des Polynoms werden in die entsprechende Typformel eingesetzt. Damit wir konventionsgemäß nur Polynome in $\mathbb{Z}[\underline{U}, \underline{X}]$ erhalten, müssen wir die einzelnen atomaren Formeln umformen. Eine atomare Formel $(f/g \; \varrho \; 0)$ formen wir dabei nach folgenden Regeln um:

(1) Ist $\varrho \in \{=, \neq\}$ ersetzen wir sie durch $(f \; \varrho \; 0)$. Die Äquivalenz folgt aus der Multiplikation mit g.
(2) Ist $\varrho \in \{<, \leqslant, \geqslant, >\}$ ersetzen wir sie durch $(f \cdot g \; \varrho \; 0)$. Die Äquivalenz folgt aus der Multiplikation mit g^2.

Die Negation der so entstandenen Formel ist das Ergebnis der Elimination im jeweiligen Ast des Gröbnersystems.

Im dritten Fall $0 < \dim(I) := d < n$ teilt man die Menge der gebunden Variablen in die Menge der unabhängigen Variablen und der abhängigen Variablen auf. Sei nach Umnumerierung der Variablen $\{x_{n-d+1}, \ldots, x_n\}$ die gewählte maximale Menge unabhängiger Variablen und $\{x_1, \ldots, x_{n-d}\}$ ihr Komplement bezüglich der Menge der quantifizierten Variablen. Dann eliminiert man zunächst

$$\exists x_1 \ldots \exists x_{n-d} \Big(\bigwedge_{\ell} p_\ell(\underline{u}, \underline{x}) \; \varrho_\ell \; 0 \wedge \bigwedge_{i=1}^{r} f_i(\underline{u}, \underline{x}) = 0 \wedge \bigwedge_{j=1}^{s} g_j(\underline{u}, \underline{x}) > 0 \wedge \bigwedge_{k=1}^{t} h_k(\underline{u}, \underline{x}) \neq 0 \Big).$$

Ist ψ das Ergebnis dieser Elimination, so eliminiert man mit einem zweiten rekursiven Aufruf die Formel $\exists x_{n-d+1} \ldots \exists x_n(\psi)$.

Der vierte und letzte Fall ist $\dim(I) = n$, d. h., für jede Spezialisierung $\underline{a}$, die die Bedingung des Astes des Gröbnersystems erfüllen, ist $\sigma_{\underline{a}}(I) = \{0\}$. Es gibt also keine nicht-trivialen Gleichungen in den gebundenen Variablen. Auch in diesem Fall müssen nicht-triviale Gleichungen zu der Konjunktion hinzugefügt werden. Wir werden diesen Fall wiederum auf einen rekursiven Aufruf der Quantorenelimination zurückführen und die Formel

$$\exists x_1 \ldots \exists x_n \Big(\Big(\bigwedge_{\ell} p_\ell(\underline{u}, \underline{x}) \; \varrho_\ell \; 0 \Big) \wedge \Big(\bigwedge_{j=1}^{s} g_j(\underline{u}, \underline{x}) > 0 \wedge \bigwedge_{k=1}^{t} h_k(\underline{u}, \underline{x}) \neq 0 \Big) \Big)$$

eliminieren.

Nach der Elimination aller Äste des Gröbnersystems müssen die einzelnen Resultate zu einer Ergebnisformel zusammengesetzt werden. Seien (R_ℓ, G_ℓ) die Bedingungen der Fälle des Gröbnersystems und ψ_ℓ die Ergebnisse der jeweiligen Elimination. Dann hat das Gesamtergebnis folgende Form

$$\Big(\bigvee_{\ell=1}^{q} (p_\ell \; \varrho_\ell \; 0) \Big) \wedge \Big(\big(\bigvee_{p \in R_\ell} p \neq 0 \big) \wedge \big(\bigvee_{p \in G_\ell} p = 0 \big) \wedge \psi_\ell \Big).$$

In den einzelnen Bedingungen des Gröbnersystems können Polynome enthalten sein, die auch in den atomaren Formeln $(p_\ell \ \varrho_\ell \ 0)$ vorkommen, da aus diesen die initiale Fallunterscheidung abgeleitet wurde. Atomare Formeln, die aus der Bedingung des einzelnen Astes des Gröbnersystems abgeleitet wurden, die solche Polynome enthalten, können gestrichen werden.

Algorithmus 4.17.
Der in Abbildung 11 *gezeigte Algorithmus eliminiert einen Existenzquantorenblock.*

Beweis. Wir zeigen, daß die Anzahl der gebundenen Variablen antiton ist und niemals länger als über zwei rekursive Aufrufe gleich bleibt.

Zu den Fällen $r \neq 0$ und $-1 \leqslant \dim(I) < n$: Die `FOR`-Schleife über die Äste des Gröbnersystems ist endlich, da ein Gröbnersystem nach Definition nur endlich viele Äste besitzt. In jedem rekursiven Aufruf von `qenf` bzw. von `qe` nimmt die Anzahl der gebundenen Variablen echt ab. Da diese durch 0 nach unten begrenzt ist, können die Fälle $r \neq 0$ und $-1 \leqslant \dim(I) < n$ nicht unendlich oft hintereinander auftreten.

Im Fall $r = 0$ werden zunächst endlich viele Konjunktionen mit jeweils einer gebundenen Variablen in der `FOR EACH`-Schleife eliminiert. Nach Satz 4.6 gibt es in jeder zu eliminierenden Konjunktion eine nicht-triviale Gleichung bezüglich der gebundenen Variablen. Im rekursiven Aufruf hat also das von den Polynomen in den Gleichungen erzeugte Ideal nicht die Dimension 1. Somit kann dieses Ideal nur die Dimension 0 oder -1 haben. In beiden Fällen werden aber keine erneuten rekursive Aufrufe gemacht und der Algorithmus terminiert. Im rekursiven Aufruf zur Elimination von $x_1, \ldots, x_{n-1}$ ist eine Variable weniger zu eliminieren.

Im Fall $\dim(I) = n$ wird der Algorithmus `qe` erneut aufgerufen, jedoch werden die – allesamt trivialen – Gleichungen weggelassen. Im rekursiven Aufruf tritt somit der Fall $r = 0$ ein. Wie bereits gezeigt wurde, terminiert dieser Aufruf aber, da weder der Fall $r = 0$ noch der Fall $\dim(I) = n$ erneut auftreten kann.

Der Aufrufbaum der Algorithmen ist somit endlich verzweigt und jeder Ast ist endlich. Also terminiert der Aufruf von `qe` nach Königs Lemma.

Zur Korrektheit: Beweis über Induktion nach der Rekursionstiefe t. Für $t = 0$ kommt nur der Fall $\dim(I) = 0$ in Frage, der nach Kapitel 2 und Kapitel 3 korrekt behandelt wird. Sei also $t > 0$. Der Fall $r = 0$ ist nach Satz 4.6 korrekt. Ist $r \neq 0$ so wird eine Schleife über alle Fälle des berechneten Gröbnersystems durchlaufen. Aus den Eliminationsergebnissen der einzelnen Fällen wird eine Disjunktion gebildet. Dies ist korrekt, da die Bedingungen der einzelnen Äste eine Fallunterscheidung bilden. Da φ unter der Prämisse der Gültigkeit von ξ eliminiert wird, ist die Fallunterscheidung des Gröbnersystems vollständig.

Im Fall $\dim(I) = -1$ gilt $1 \in I$ und I hat weder reelle noch komplexe Nullstellen. Somit kann auch φ nicht erfüllt sein und das Ergebnis ist `FALSE`.

Im Fall $0 < \dim(I) < n$ wird der Quantorenblock in 2 Quantorenblöcke aufgespaltet, die dann getrennt eliminiert werden. Aus der Korrektheit der einzelnen Aufrufe, folgt die Korrektheit dieses Falls.

Im Fall $\dim(I) = n$ folgt $I = \{0\}$. Somit sind alle Gleichungen trivial und können vernachlässigt werden. Genau dies wird im rekursiven Aufruf gemacht. Wiederum folgt aus der Korrektheit des rekursiven Aufrufs auch die Korrektheit dieses Falls. □

Eingabe: Eine Formel φ in Eliminationsnormalform, eine Konjunktion ξ atomarer Formeln, die unabhängig von den gebundenen Variablen sind.

Ausgabe: Eine quantorenfreie Formel ψ, so daß $\xi \wedge \varphi \iff \xi \wedge \psi$.

```
1  PROCEDURE qenf(ξ, φ)
2  BEGIN
3      IF r = 0 THEN
4          Forme die Matrix von φ nach Satz 4.6 bezüglich x_n in
           ( ⋀_{k=1}^{t} h_k ≠ 0 ) ∧ ( φ_1 ∨ φ_2 ∨ φ'_3 ∨ φ'_4 ) um.
5          ψ := ( ⋀_{k=1}^{t} h_k ≠ 0 ) ∧ ( φ_1 ∨ φ_2 )
6          FOR EACH Konjunktion φ' in φ'_3 bzw φ'_4 DO
7              ψ := ψ ∨ qenf( ξ ∧ ( ⋀_{k=1}^{t} h_k ≠ 0 ), ∃x_n(φ') )
8          END
9          ψ := qe( ξ ∧ ∃x_1, ..., ∃x_{n-1}(ψ) )
10         RETURN ψ
11     END
12     S := GSYS({f_i}) über der von ξ bestimmten initialen Bedingung.
13     FOR EACH Ast ((R, G), P) in S DO
14         ζ := ( ⋀_{p∈G} g(u) ≠ 0 ) ∧ ( ⋀_{p∈R} p(u) = 0 )
15         SELECT dim(P) OF
16         CASE −1
17             ψ := FALSE
18         CASE 0
19             χ := ∏_{e∈E_s} χ_e nach Korollar 2.18 und sei χ = Y^c + Σ_{ℓ=0}^{c-1} a_ℓ Y^ℓ
20             ψ := ¬T_c(a_0, ..., a_{c-1})
21             ψ := „Äquivalente Formel in der Sprache der geordneten Ringe."
22         CASE 1, ..., n − 1
23             Wähle eine maximale Menge unabhängiger Variablen.
24             Numeriere die Variablen x_i so um, daß {x_{n-d+1}, ..., x_n} die
                   gewählte maximale Menge unabhängiger Variablen ist.
25             ψ := „Matrix von φ"
26             ψ := qenf( ζ ∧ ξ, ∃x_1, ..., ∃x_{n-d}(ψ) )
27             ψ := qe( ∃x_{n-d+1}, ..., ∃x_n(ψ) )
28         CASE n
29             ψ := qenf( ζ ∧ ξ ∧ ∃x( ⋀_{j=1}^{s} g_j(u, x) > 0 ∧ ⋀_{k=1}^{t} h_k(u, x) ≠ 0 ) )
30         ENDSELECT
31     END
32     RETURN ψ
33 END qenf
```

ABBILDUNG 11. Algorithmus zur Elimination eines Existenzquantorenblocks

KAPITEL 5

Allgemeine Bemerkungen

5.1. Typformeln

Unabhängigkeit der Typformeln. Typformeln hängen nur vom Grad d der Polynome ab. Aus ihnen werden durch Substitutionen die Teilergebnisse der Elimination gebildet. Somit können Typformeln vor dem Ablauf des eigentlichen Eliminationsverfahren unabhängig vom eigentlichen Eliminationsproblem berechnet und abgespeichert werden. Insbesondere kann dies dazu genutzt werden, die Typformeln entweder von Hand oder mittels automatischer Verfahren zu optimieren.

Direkte Bestimmung von Typformeln. Verzichtet man auf eine Vorausberechnung der Typformeln und bestimmt sie erst, wenn die Koeffizienten des charakteristischen Polynoms bekannt sind, so kann die Berechnung der Typformel weiter optimiert werden. Kommt in keinem Koeffizienten des charakteristischen Polynoms ein Parameter vor, so braucht keine Typformel bestimmt werden, sondern der Typ kann mittels der Vorzeichenregel von Descartes bestimmt werden. Sind einige Koeffizienten des charakteristischen Polynoms nicht parametrisch aber andere parametrisch, so kann dennoch die Berechnung effizienter durchgeführt werden. Bei der Berechnung der Koeffiziententupeln mittels des Algorithmus ctj werden in diesem Fall an einer Stelle des Koeffiziententupels nur solche charakteristischen Koeffizienten eingesetzt, die möglich sind. Sei z. B. der Koeffizient c_i nicht parametrisch und $c_i > 0$. Dann wird beim Erzeugen der Koeffiziententupel an der Stelle i nur die charakteristischen Koeffizienten J und 1 generiert. Für $c_i = 0$ wird dementsprechend nur der Koeffizient J und für $c_i < 0$ werden die Koeffizienten J und -1 generiert.

Struktur der Typformeln. Typformeln, wie sie mittels des Algorithmus ctj bestimmt werden, haben eine ungünstige Form für das iterierte Anwenden des Elimination. Bei der Bestimmung der strikten Typformeln werden nur atomare Formeln mit den Relation $<$ und $>$ erzeugt, aber es werden keine Gleichungen gebildet, die für das Verfahren notwendig sind.

5.2. Das Eliminationsverfahren

Elimination mehrerer Quantorenblöcke. Im einfachsten Fall der Elimination eines Existenzquantorenblocks ist das Ergebnis des Verfahrens die Negation einer Typformel. Bei der Berechnung von Typformeln, wie wir sie durchführen, erhält man eine Disjunktion von Konjunktionen der Form

$$c_0 = 0 \wedge \cdots \wedge c_i = 0 \wedge T'_d(c_{i+1}, \ldots, c_{d-1}).$$

Zieht man die Konjunktion $(\bigwedge_j c_j = 0)$ in die einzelnen Disjunktionsglieder der strikten Typformel T'_d, so erhält man eine disjunktive Normalform der Typformel. Ist der nächste zu eliminierende Quantorenblock ein Allquantorenblock, so kann der Allquantor direkt in die Konjunktion gezogen werden. In diesem Fall tritt somit durch die notwendige Berechnung der Normalform beim zweiten Quantorenblock noch kein exponentielles Wachstum der Anzahl der atomaren Formeln auf.

Umformen der Eingabeformel. Gleichungen spielen in dem Eliminationsverfahren eine tragende Rolle. Mittels folgender Äquivalenzen können Ungleichungen in der Eingabeformel in Gleichungen umgeformt werden.

$$f > 0 \Longleftrightarrow \exists Z(f \cdot Z^2 - 1 = 0) \quad \text{reeller Rabinowitsch Trick}$$
$$f \neq 0 \Longleftrightarrow \exists Z(f \cdot Z - 1 = 0) \quad \text{Rabinowitsch Trick}$$

Jedoch wird eine neue Variable eingeführt werden, was an anderen Stellen des Eliminationsverfahren wie z. B. bei der Berechnung des Gröbnersystems störend ist.

Das Produkt der charakteristischen Polynome. Beim parametrischen Zählen reeller Nullstellen wird immer wieder das Produkt der charakteristischen Polynome X_e gebildet. Für Zeitoptimierungen kann man in Erwägung ziehen, Produkte von Polynomen parametrisch zu berechnen und abzuspeichern. Dieses Produkt hängt von d der Dimension des Restklassenrings, s der Anzahl der Nebenbedingungen der Form $g_j > 0$ und t der Anzahl der Nebenbedingungen der Form $h_k \neq 0$ ab. In diesem Fall berechnet man für $1 \leqslant j \leqslant s$ und $1 \leqslant k \leqslant t$ und für Polynome

$$g_j := \sum_{i=0}^{d} a_{ij} \quad \text{und} \quad h_k := \sum_{i=0}^{d} b_{ik}$$

das Produkt

$$\prod_{e \in \{1,2\}^s} g_i^{e_i} h^2.$$

Vermeidung der Produktbildung. Nach Proposition 2.14 muß zum parametrischen Zählen von Nullstellen charakterisiert werden, daß die Summe der Signaturen gewisser Matrizen gleich Null ist. Anderes gesagt, wir müssen charakterisieren, daß die Summe der Typen charakteristischer Polynome gleich Null ist. Dies wird im Verfahren, wie es hier vorgestell wurde, auf die Typbestimmung des Produktes dieser Polynome zurückgeführt. Mit der Definition der Typformel bezüglich des Typs t kann auch direkt eine Formel konstruiert werden, die genau dann erfüllt ist, wenn die Summe der Typen der entsprechenden Polynome gleich Null ist. Dies erfordert bei der Konstruktion der Formeln mehr Aufwand, hält aber die Komplexität der Terme z. B. gemessen am Totalgrad niedriger, da nicht das Produkt gebildet werden muß.

Auffinden nicht trivialer Gleichungen. Sind keine Gleichungen in der Eingabeformel enthalten, müssen nicht-triviale Gleichungen zur Eingabe hinzugefügt werden. Ist jedoch nur eine Formel der Form

$$\exists x_1 \ldots \exists x_n (\bigwedge_{k=0}^{t} h_k \neq 0)$$

zu eliminieren, so kann eine äquivalente quantorenfreie Formel direkt gebildet werden. Bezeichne $h \neq 0$ die Bedingung, daß h nicht das Nullpolynom ist. Zu jedem multivariaten Polynom $0 \neq f$ über $\mathbb{R}$ gibt es unendlich viele Stellen $\underline{x} \in \mathbb{R}^n$ mit

$f(\underline{x}) \neq 0$ (vgl. [BW93b] Lemma 7.50). Seien $h_1, \ldots, h_k \in \mathbb{R}[\underline{U}, \underline{X}]$ parametrisierte Polynome und $\underline{u}$ eine Spezialisierung. Dann gilt

$$\exists \underline{x}(\bigwedge_{k=1}^{t} h_k(\underline{u}, \underline{x}) \neq 0) \Longleftrightarrow \exists \underline{x}(\prod_{k=1}^{t} h_k(\underline{u}, \underline{x}) \neq 0) \Longleftrightarrow (\prod_{k=1}^{t} h_k \neq 0) \Longleftrightarrow (\bigwedge_{k=1}^{t} h_k \neq 0)$$

Sind a_{ik} die Koeffizienten von h_k, so ist die obige Formel äquivalent zu

$$\bigwedge_k \bigvee_i a_{ik} \neq 0.$$

Komplexität der Ergebnisformeln. Ein Maß für die Komplexität von Formeln ist die Anzahl ihrer atomaren Formeln. Betrachtet man die Ergebnisformeln der Quantorenelimination, so fällt auf, daß in den atomaren Formeln große Terme enthalten sind. So ist die Anzahl der atomaren Formeln nicht immer repräsentativ für die Komplexität der Formel. Als Maß für die Komplexität der Terme bietet sich der Totalgrad an. Die Komplexität einer Formel kann dann durch die Anzahl der atomaren Formeln und dem Maximum der Totalgrade der vorkommenden Terme angegeben werden. Es ist festzustellen, daß auch bei Eingabeformeln mit einfachen Termen, in den Ergebnisformeln komplexe Terme auftreten können. In einem Beispiel, das 2 atomaren Formeln und einem maximalen Totalgrad von 4 in der Eingabeformel hatte, waren Polynome bis zu einem Totalgrad von 15 enthalten.

Partielle Quantorenelimination. Das Verfahren zur Quantorenelimination wie es hier vorgestellt wurde gliedert sich semantisch in zwei Teile. Zum einen wird die Existenz reeller Nullstellen nulldimensionaler Ideale unter Nebenbedingungen ermittelt. Zum anderen werden die Fälle, die nicht durch das Zählen der Nullstellen behandelt werden können, weil die entsprechenden Ideal nicht nulldimensional sind, auf diesen Fall zurückgeführt. Bei der Elimination möchte man sich eventuell auf die Fälle der nulldimensionale Fälle beschränken. Die nicht nulldimensionalen Fälle, die während der Elimination auftreten, codiert man dann in quantifizierte Formeln. Wir bezeichnen dies als *partielle* Quantorenelimination der nulldimensionalen Fälle.

5.3. Zusammenfassung

Das hier beschriebene Verfahren zur reellen Quantorenelimination mittels parametrischen Zählens reeller Nullstellen wurde von V. Weispfenning in [Wei93a] bzw. [Wei93b] veröffentlicht. In der vorliegenden Arbeit wird das Verfahren insbesondere in der Hinsicht auf die praktische Implementierung beschrieben.

Besonderen Wert wurde dabei auf die Bestimmung der Typformeln gelegt. Ergebnis ist hier der in der Arbeit gezeigte Algorithmus `ctj`. Aus der von ihm berechneten Menge von charakteristischen Koeffiziententupeln lassen sich direkt relativ kurze Typformeln bestimmen. Im Gegensatz dazu bestimmt der naive Algorithmus zeitaufwendig lange Typformeln, die ebenfalls zeitaufwendig zeitaufwendig optimiert werden müßten.

Ein weiteres wichtiges Resultat für die Implementierung ist die Verallgemeinerung des parametrischen Zählens reeller Nullstellen auch auf Nebenbedingungen der Form $f \neq 0$. Eine Ungleichung $f \neq 0$ muß somit weder durch die Formel $f > 0 \vee f < 0$ noch durch die Formel $f^2 > 0$ simuliert werden. Dieses hätte entweder die Anzahl der Konjunktionen in der zu bildenden disjunktiven Normalform erhöht oder die Grade der Polynome in den Nebenbedingungen vergrößert. Bei unserem Ansatz werden bei einer zusätzlichen Ungleichung nur jeweils die charakteristischen Polynome von $Q_{h^e \cdot f^2}$ anstatt von Q_{h^e} berechnet.

In der Praxis hat sich gezeigt, daß die Berücksichtigung der initialen Bedingung bei der Berechnung von Gröbnersystemen sich immer wieder positiv bemerkbar macht. Insbesondere ist dies der Fall, wenn die Bedingungen bei der Berechnung des Gröbnersystems mittels optimierter Verfahren ausgewertet werden.

Ein kritischer Punkt des Verfahrens sind Formeln, in denen keine Gleichungen enthalten sind. Das Verfahren zur Erzeugung nicht-trivialer Gleichungen konnte auch auf Ungleichungen erweitert werden. Auch hier sind Ungleichungen $f \neq 0$ direkt wesentlich einfacher zu behandeln als durch Simulation. Wichtiger ist es jedoch in diesem Zusammenhang, daß keine Rekursion, wie sie in der Veröffentlichung beschrieben wurde, notwendig ist.

Die vorliegende Implementierung des Verfahrens dient vorallem dazu, erste Erkenntnisse zu gewinnen, ob und wann das Verfahren praktikabel durchführbar ist. Generell limitierende Faktoren sind eine hohe Dimensionen des Restklassenrings $\mathbb{R}[\underline{X}]/I$, eine große Anzahl von Nebenbedingungen insbesondere der Form $g > 0$ sowie die Berechnung eines Gröbnersystems.

Innerhalb der von diesen Parametern gegebenen Rahmen zeigen sich folgenden Tendenzen: Bei global nulldimensionalen Problemen, d. h., alle auftretenden Ideale sind nulldimensional, ist das Verfahren gut anwendbar. Je mehr Nebenbedingungen auftreten, desto komplexer werden die Ergebnisformeln. Hier ist insbesondere das rapide Ansteigen der Totalgrade der Terme zu erwähnen.

Sind auftretende Ideale nicht nulldimensional, so stößt man sehr schnell an die Grenzen des Verfahrens. Insbesondere gibt es Probleme, wenn keine Gleichungen in der Eingabe vorhanden sind und nicht-triviale Gleichungen erzeugt werden müssen. Einige Beispiele konnten trotz solcher Fälle eliminiert werden.

Die parametrische Bestimmung der Existenz reeller Nullstellen nulldimensionaler Ideale unter Nebenbedingungen ist praktisch gut durchführbar. Die Behandlung der Fälle, die nicht nulldimensionale Ideale enthalten sind eher vom theoretischen Interesse. Sie zeigen jedoch, daß man ein allgemeines Verfahren zur reellen Quantorenelimination beliebiger Formeln erster Stufe in der Sprache der angeordneten Ringe hat.

KAPITEL 6

Die Implementierung des Verfahrens

6.1. Das System *MAS*

Das in den vorherigen Kapiteln dargestellte Verfahren zur reellen Quantorenelimination wurde im Computeralgebrasystem *MAS* implementiert. Näheres zu *MAS* findet man in [Kre93b] und [Kre93a]. Zur Darstellung der Formeln wird das *MASLOG*-System [Dol93] und zur Darstellung der atomaren Formeln das darauf aufbauende *PQ*-System verwendet. Dieses stellt die Terme in der Sprache der Ringe als distributive Polynome dar. Genauer verwendet es das in *MAS* implementierte *DIP*-Paket in Verbindung mit dem *Arbitrary-Domain*-Paket. Die Koeffizienten der Polynome könnten also prinzipiell beliebige Elemente einer der implementierten Domains sein. Wir lassen jedoch ausschließlich Koeffizienten des Domains `INT` zu.

Das im Kapitel 2 angesprochene Verfahren zum Zählen reeller Nullstellen nulldimensionaler Ideale wurde von F. Lippold [Lip93] im *MAS* implementiert. Diese Implementierung wird in der Implementierung des Verfahrens zur reellen Quantorenelimination zur Berechnung der entsprecheden charakteristischen Polynome benutzt.

Die Algorithmen zur Berechnung von Gröbnersystemen, zur parametrischen Bestimmung der Dimension von Idealen sowie zur Bestimmung maximal unabhängiger Variablenmengen wurden von E. Schönfeld [Sch91] im System *SAC-2/ALDES* implementiert. Das auf einer Portierung dieser Algorithmen basierende *CGB*-Paket von *MAS* wird von M. Pesch [Pes94b] weiterentwickelt

6.2. Die Syntax von Formeln

Die Terme in den Formeln des *PQ*-Systems werden als distributive Polynome dargestellt. Der entsprechende Polynomring wird über globale Variablen festgelegt. Der Benutzer kann den Polynomring entweder über einen Aufruf der Prozedur `PQPRING` oder bei der Eingabe der Formel festlegen. Die Prozedur `PQPRING` hat eine maximal 3 elementige Liste als Parameter:

- Der erste Eintrag ist der Domaindescriptor des Grundringes. Hier ist für die Quantorenelimination nur `INT` zulässig.
- Der zweite Eintrag ist die Variablenliste, eine Liste aller verwendeten Variablen als Strings.
- Der dritte Eintrag ist die benutzte Termordnung.

Termordnungen werden in *MAS* mittels β-Integer für vordefinierte Termordnungen oder mittels Linearformen, die als Listen repräsentiert werden, eingegeben. Die wichtigsten Termordnungen sind

2: invers-lexikographische-Termordnung
4: Totalgrad-Termordnung
7: Buchbergs-Totalgrad-Termordnung.

Ein Beispiel:

```
d:=ADDDREAD().␣INT␣
PQPRING(LIST(d,LIST("x","a","b"),4)).
```

Wird in der Parameterliste von `PQPRING` an einer Stelle -1 angegeben, so wird die entsprechende Spezifikation nicht geändert. Bei der Angabe der Termordnung ist zu beachten, daß sie mit der Termordnung, die bei der Berechnung der Gröbnersysteme benutzt wird, übereinstimmen muß.

Formeln können nach dem Aufruf von `PQIREAD()` eingegeben werden. Ihre Syntax ist in Abbildung 12 beschrieben. Wir verwenden hierbei die erweiterte BNF Notation. Einige Bemerkungen zur Syntax von Formeln: Die Groß-Klein-Schreibung von Symbolen ist nicht relevant. So bezeichnen `AND`, `and` und `AnD` die konjunktive Verknüpfung zweier Formeln. Im Gegensatz dazu müssen die Schlüsselworte `VALIS`, `EVORD` und `DOMAIN` immer groß geschrieben werden.

Leerzeichen sind innerhalb lexikalischer Token wie Symbolen, Atomen und Namen nicht erlaubt. Sie müssen jedoch zwischen zwei benachbarten Symbolen stehen, wie z. B. bei `T␣AND␣F`. Ansonsten werden sie beim Lesen der Formeln ignoriert.

Nichtangegebene Nicht-Terminale wie „Ident“ oder „Atom“ leiten sich aus der allgemeinen Syntax von *MAS* ab. Eine Sonderstellung nimmt „#MasVar“ ein. Dabei bezeichnet „MasVar“ eine Variable des Interpreters, in der eine Formel gespeichert ist. Diese Formel wird beim Lesen an der betreffenden Stelle eingesetzt. Dies erlaubt Zwischenergebnisse bei der Konstruktion von Formeln direkt zu verwenden.

Generelle Nebenbedingung bei der Eingabe von Formeln ist es, daß alle benutzten Variablen in der Variablenliste des Polynomringes deklariert sein müssen.

Die Operatoren haben verschiedene Prioritäten, so daß Formeln nicht vollständig geklammert werden müssen. In absteigender Reihenfolge haben die einzelnen Operatoren folgende Prioritäten untereinander.

- Negation
- Konjunktion
- Disjunktion
- Implikation und Replikation
- Äquivalenz und Antivalenz
- Quantoren

Ist eine andere Reihenfolge der Auswertung beabsichtigt, muß die Formel entsprechend geklammert werden.

6.3. Die Benutzung der Implementierung

In Abbildung 13 ist ein Beispielprogramm zur Benutzung der Implementierung gezeigt. Wie bereits erwähnt, wird das *CGB*-Paket benutzt. Mittels `CGBOPT` werden die Optionen dieses Paketes gesetzt. Näheres über die einzelnen Optionen findet man in [Pes94a]. Mit `RQEOPTSET` werden die Optionen des Eliminationsverfahrens gesetzt. Argument ist eine maximal zweielementige Liste.

Der erste Eintag gibt den „Trace-Level“ an. Er steuert ob und wieviel Angaben über den Programmablauf ausgegeben werden. Bei 0 werden keinerlei Ausgaben gemacht. Ein Trace-Level von 1 dokumentiert den Ablauf durch kurze, codierte Ausgaben. In

```
                Input = [ PRing ] Formula .

                PRing = '{' PRingParam {',' PRingParam} '}'
           PRingParam = 'VALIS='VarList | 'EVORD='TermOrd | 'DOMAIN='Ring
              VarList = '(' Ident {',' Ident} ')'
              TermOrd = Atom
                 Ring = SymbolicDomainDescriptor

              Formula = '(' Formula ')' | AtomicFormula | UnaryOp Formula |
                        Formula BinaryOp Formula | QuantifiedFormula |
                        TruthVal | '#'MasVar
    QuantifiedFormula = Quantifier BoundVars '(' Formula ')' |
                        Quantifier BoundVars ':' Formula
            BoundVars = Ident {[','] Ident }
             TruthVal = TrueSym | FalseSym
              UnaryOp = NotSym
             BinaryOp = AndSym | OrSym | ImplSym | ReplSym |
                        EquivSym | XorSym
           Quantifier = ForallSym | ExistsSym
        AtomicFormula = '[' DipPolynomial Relation DipPolynomial ']'
             Relation = LessSym | LessOrEqualSym |
                        EqualSym | NotEqualSym |
                        GreaterOrEqualSym | GreaterSym

       DipPolynomial = 0 | '(' Term {('+'|'-') Term}
                Term = Power { Power }
               Power = Factor [ '**' Atom ]
              Factor = Ident | Atom

             TrueSym = 't' | 'true' | 'verum'
            FalseSym = 'f' | 'false' | 'falsum'
              NotSym = '--' | '~' | 'not' | 'non'
              AndSym = '/\' | 'and' | 'et'
               OrSym = '\/' | 'or' | 'vel'
             ImplSym = '=>' | '==>' | 'impl'
             ReplSym = '<=' | '<==' | 'repl'
            EquivSym = '<=>' | 'equiv'
              XorSym = '<#>' | 'xor'
           ForallSym = 'a' | 'all' | 'fa' | 'forall'
           ExistsSym = 'e' | 'ex' | 'exists'
             LessSym = '<' | 'les'
      LessOrEqualSym = '<=' | 'leq' | 'lsq'
            EqualSym = '=' | 'equ'
         NotEqualSym = '<>' | '#' | '!=' | 'neq'
   GreaterOrEqualSym = '>=' | 'geq' | 'grq'
          GreaterSym = '>' | 'gre'
```

ABBILDUNG 12. Syntax der Formeln

```
PRAGMA(TIME).
CGBOPT(LIST(0,1,0,2,0,4,4)).
RQEOPTSET(LIST(0,0)).
TfComputeTf().
phi:=PQIREAD().
{VALIS=(a,x), EVORD=4, DOMAIN=INT}
(  ex x ( [ ( a x ) > 0] ) ) .
ws:=RQEQE(phi).
PQPPRT(ws).
FORCOUNTAF(ws).
```

ABBILDUNG 13. Beispiel zur Quantorenelimination

den Trace-Level 2 und 3 werden ausführlichere Ausgaben in Klartext gemacht. Im Unterschied zu 2 werden bei 3 auch Zwischenergebnisse ausgegeben.

Das zweite Element der Liste steuert die Elimination selber. Ist es 0, so wird der komplette Algorithmus durchlaufen. Ist es jedoch 1, so wird nur eine partielle Elimination der nulldimensionalen Ideale durchgeführt. In diesem Fall kann das Ergebnis noch Quantoren enthalten.

Mittels `TfComputeTf()` bzw. `TfUseDb()` wird die Erzeugung von Typformeln gesteuert. Im ersten Fall werden alle Typformeln immer wieder neu berechnet. Im zweiten Fall wird versucht, bereits berechnete Typformeln zu benutzen. Nur wenn eine Typformel für einen Grad d noch nicht berechnet wurde, wird sie neu berechnet und abgespeichert.

Der Aufruf `RQEQE` eliminiert schließlich die übergeben Formel. Mittels `PQPPRT` wird das Ergebnis in der Syntax der Eingabe ausgegeben. `FORCOUNTAF` gibt die Anzahl der atomaren Formeln im Ergebnis aus.

In der Formel, die man mit der Prozedur `RQEQE` eliminiert, darf nach dem Auflösen der erweiterten boolschen Operatoren wie Implikation, Replikation etc. keine Variable von 2 Quantoren gebunden werden bzw. gebunden und frei auftreten. Man beachte, daß somit z. B. die Formel $(\exists x(\varphi(x))) \Leftrightarrow \psi$ nicht zugelassen ist. Will man die Quantoren einer solchen Formel eliminieren, so muß man die betroffenen Variablen mittels der Prozedur `PQMKVD` umbenennen, nachdem die erweiterten Operatoren aufgelöst wurden. Beim Umbenennen der Variablen wird die globale Liste der Variablen des benutzen Polynomrings modifiziert.

6.4. TYPFORMELN

Das Modul `TFORM` stellt Prozeduren zur Bestimmung von Typformeln zur Verfügung. Typformeln sind formal Formeln über der Sprache der multiplikativ geschriebenen Halbgruppen und den ordnungstheoretischen Relationen. Variablen haben dabei die Form `a`i mit $i \in \mathbb{N}$. Typformeln werden auch mittels des *MASLOG*-Systems repräsentiert. Dabei wird jedoch nicht auf das *PQ*-System zurückgegriffen, sondern es wird eine eigenständige Repräsentation der atomaren Formeln gewählt. Erst bei der Substitution der Variablen der Typformeln durch die Koeffizienten eines charakteristischen Polynoms wird die Formel in das *PQ*-System umgewandelt. Multiplikative Halbgruppen wurden gewählt um Optimierungen der Form

$$a < 0 \wedge b > 0 \vee a > 0 \wedge b < 0 \Longleftrightarrow a \cdot b < 0$$

zu codieren. Im jetzigen Ansatz zur Bestimmung von Typformeln gibt es keinen Hinweis darauf, daß auch Summen zur Optimierung von Typformeln verwendet werden können.

Typformeln haben eine ähnliche Syntax wie Formeln des *PQ*-Systems (vgl. Abbildung 12). Unterschiedlich ist nur die Syntax der atomaren Formeln. Atomare Formeln von Typformeln haben folgende Syntax:

‘`[`’ [‘`(`’] ‘`a`’atom { [‘`*`’] ‘`a`’atom } [‘`)`’] rel ‘`0`’ ‘`]`’

Die Implementierung unterstützt auch den Ansatz, Typformeln im voraus zu berechnen und abzuspeichern. In diesem Fall wird beim Aufruf der Prozedur zur Berechnung einer Typformel zunächst geschaut, ob sie bereits berechnet wurde. In diesem Fall wird die gespeicherte Typformel zurückgegeben. Ansonsten wird die betreffende Typformel berechnet, abgespeichert und zurückgegeben. Eine Typformel für Polynome des Grades d wird dabei in einem File mit dem Namen `TF.`d`.db` gespeichert.

6.5. Top-Level Prozeduren der Implementierung

Nur wenige Prozeduren der Implementierung sind für den Anwender interessant. Diese sollen hier kurz vorgestellt werden.

$\psi :=$ `RQEQE(`φ`)`: Diese Prozedur ist die Implementierung des Algorithmus `qe` aus Kapitel 4. Die Variablen φ und ψ sind Formeln im Sinne des *PQ*-Systems. Nach dem Auflösen der erweiterten boolschen Operationen darf keine Variable mehrfach gebunden sein oder frei und gebunden auftreten.

`OldOpt:=RQEOPTSET(Opt)`: Die Variablen `Opt` und `OldOpt` sind Listen der Form `(TraceLevel,DimensionZeroOnly)`. Dabei ist `TraceLevel` eine positive Integer Zahl, die angibt wieviel Testausgaben gemacht werden sollen. Ist die Variable `DimensionZeroOnly` $= 1$ wird nur die partielle Quantoreneliminationen der nulldimensionalen Fälle durchgeführt Ansonsten werden alle Quantoren eliminiert und es wird eine quantorenfreie Formel zurückgegeben. Die Variable `OldOpt` enthält nach dem Aufruf die Liste der alten Optionen. Wird `RQEOPTSET` mit der leeren Liste als Argument aufgerufen, so wird nur die aktuelle Liste der Optionen zurückgegeben. Standardmäßig ist ein Trace-Level von 3 gesetzt, und es wird eine vollständige Quantorenelimination durchgeführt.

`RQEOPTWR()`: Die aktuell gesetzten Optionen werden in Klartext in den Output-Stream geschrieben.

φ`:=TfTypeFormula(d)`: Diese Prozedur berechnet eine Typformel für Polynome des Grades `d`.

`TfUseDb()`: Nach dem Aufruf dieser Prozedur wird bei der Berechnung von Typformeln die Datenbank der bereits berechneten Typformeln benutzt. Standardmäßig werden Typformeln auf Anforderung berechnet.

`TfComputeTf()`: Nach dem Aufruf dieser Prozedur werden Typformeln immer wieder neu berechnet. Die Datenbank wird nicht benutzt. Dies ist das Standardverhalten.

ψ`:=TFIREAD()`: Eine Formel – in der Syntax von Typformeln – wird aus dem Input-Stream gelesen.

`TFPPRT()`: Eine Formel – in der Syntax von Typformeln – wird in den Output-Stream geschrieben.

APPENDIX A

Beispiele zur Quantorenelimination

A.1. Vorbemerkungen

Sämtliche Beispiele wurden auf einer *IBM* RS6000/220 Anlage mit 64 MByte Hauptspeicher gerechnet. Das System *MAS* lief mit 32 MByte Speicher für die Listenverwaltung. Das System wurde mit dem Compiler *Mocka* der GMD Karlsruhe übersetzt. Alle Zeitangaben sind auf Sekunden gerundet.

Bei der Berechnung der Beispiele wurden Typformeln jeweils neu berechnet und nicht abgespeichert. Während der Quantorenelimination wurden keine Testausgaben gemacht. Die Gröbnersysteme wurden jeweils bezüglich der Totalgrad-Termordnung berechnet. Bei der Berechnung der Gröbnersysteme wurden die Koeffizientenpolynome faktorisiert und die Bedingungen mit Hilfe von Gröbnerbasen ausgewertet.

A.2. Einfache Testbeispiele

In diesem Abschnitt werden einige einfache Testbeispiele präsentiert.

Existenz des Inversen. Die Eingabeformel ist

$$\exists a(ax = 1).$$

Das Ergebnis $x \neq 0$ erhalten wir in ca. 0.1 Sekunden.

Reelle Nullstellen einer normierten Parabel. Es wird der Quantor der Eingabeformel

$$\exists x(x^2 + px + q = 0)$$

eliminiert. Das Ergebnis $\neg(p^2 - 2q + 2 = 0 \vee p^2 - 4q < 0)$ erhalten wir in ca. 0.3 Sekunden.

Reelle Nullstellen einer Parabel. Im Gegensatz zum letzten Beispiel betrachten wir eine nicht normierte Parabel. Ergebnis der Elimination von $\exists x(ax^2 + px + q = 0)$ ist

$$(a = 0 \wedge p = 0 \wedge q = 0) \vee (p \neq 0 \wedge a = 0) \vee$$
$$(a \neq 0 \wedge (\neg(2aq - p^2 - 2a^2 = 0 \vee 4a^5q - a^4p^2 > 0))).$$

Dieses Ergebnis mit 8 atomaren Formeln erhielten wir in 39 Sekunden.

Reelle Nullstellen eines normierten kubischen Polynoms. Ebenso wie Parabeln betrachten wir auch ein normiertes, kubisches Polynom.

$$\exists x(x^3 + px^2 + qx + r = 0).$$

Das folgende Ergebnis der Elimination hat 3 atomare Formeln und wurde in 1 Sekunde berechnet.

$$\begin{array}{l}\neg(4p^3r - p^2q^2 - 18pqr + 4q^3 + 27r^2 = 0 \wedge \\ (p^4 - 4p^2q + 4pr + 2q^2 + p^2 - 2q + 3 = \\ 0 \vee 2p^3r - p^2q^2 - 2p^4 - 10pqr + 4q^3 + 8p^2q + 9r^2 - 12pr - 2q^2 - 2p^2 + 6q > 0))\end{array}$$

Reelle Nullstellen eines kubischen Polynoms. Als letztes Beispiel in dieser Reihe betrachten wir die Existenz reeller Nullstellen eines kubischen Polynoms. Die zu eliminierende Formel ist

$$\exists x(x^3 + px^2 + qx + r = 0).$$

Das folgende Ergebnis der Elimination hat 11 atomare Formeln und wurde in 79 Sekunden berechnet.

$$\begin{array}{l}(x \neq 0 \wedge z \neq 0 \wedge x^4 - y^2z = 0) \vee \\ (z \neq 0 \wedge x = 0 \wedge y = 0 \wedge (\neg(z + 1 = 0 \vee z < 0))) \vee (x = 0 \wedge y = 0 \wedge z = 0)\end{array}$$

Die Formel von Binomi. Es ist allgemein bekannt, daß für alle a und b

$$a^2 + 2ab + b^2 = (a + b)^2 \geqslant 0.$$

Wir eliminieren die Quantoren der geschlossenen Formel $\forall a \forall b(a^2 + 2ab + b^2 \geqslant 0)$. Das Ergebnis `TRUE` erhalten wir in ca. 0.26 Sekunden.

Faktorisierung eines kubischen Polynoms. Wir betrachten die geschlossene Formel

$$\begin{array}{l}\forall a_3 \forall a_2 \forall a_1 \forall a_0 (\exists b_1 \exists b_0 \exists c_2 \exists c_1 \exists c_0 (\\ \quad a_3x^3 + a_2x^2 + a_1x + a_0 = b_0c_0 - b_0c_1x - b_0c_2x^2 - b_1c_0x - b_1c_1x^2 - b_1c_2x^3)).\end{array}$$

Sie formuliert die Aussage, daß jedes kubische Polynom eine reelle Faktorisierung in einen quadratischen $(c_2x^2+c_1x+c_0)$ und einen linearen Faktor (b_1x+b_o) besitzt. Das Eliminationsverfahren stellt die Allgemeingültigkeit dieser Formel in 639 Sekunden fest.

A.3. Einige algebraische Kurven

Im folgenden suchen wir Formeln, die die Bilder von Funktionen $f : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ beschreiben, d. h. es sind quantorenfreie Formeln gesucht, die genau dann erfüllt sind, wenn ein Punkt im Bild der jeweiligen Funktion liegt. Die Beispiel wurden aus [CLO92] Seite 132ff. entnommen.

Der Whitney Umbrella. Gegeben ist die Abbildung

$$f : \mathbb{R}^2 \longrightarrow \mathbb{R}^3 \quad \text{mit} \quad (u, v) \longmapsto (uv, v^2, u^2).$$

Gesucht ist eine quantorenfreie Formel $\varphi(x, y, z)$, die genau dann erfüllt ist, wenn (x, y, z) im Bild von f enthalten ist. Dazu eliminieren wir die Formel

$$\exists u \exists v((x = uv \wedge y = u^2 \wedge z = v^2)).$$

Mittels der Quantorenelimination erhalten wir in 20 Sekunden das folgende Ergebnis mit 6 atomaren Formeln.

$$(y \neq 0 \wedge y^2 z - x^2 = 0) \vee (x = 0 \wedge y = 0 \wedge (\neg(z + 1 = 0 \vee z < 0)))$$

Das Descartsche Blatt. Die selbe Problemstellung betrachten wir für die Abbildung

$$g : \mathbb{R} \longrightarrow \mathbb{R}2 \quad \text{mit} \quad t \longmapsto (\frac{3t}{1 + t^3}, \frac{3t^2}{1 + t^3}).$$

Dazu eliminieren wir die Formel

$$\exists t(xt^3 + x = 3t \wedge yt^3 + y = 3t^2 \wedge t \neq -1).$$

Die folgende quantorenfreie Formel mit 7 atomaren Formel wurde in 20 Sekunden berechnet.

$(y^2 - 3x \neq 0 \wedge x \neq 0 \wedge y \neq 0 \wedge y^3 + x^3 - 3xy = 0 \wedge$
$y^4 - 2x^2y^2 + x^4 - 6xy^2 + 6x^3 + 9x^2 \neq 0) \vee (x = 0 \wedge y = 0)$

Zwei weitere Kurven. Das Verfahren eliminiert die Quantoren von

$$\exists u \exists v(x = uv \wedge y = u^2 \wedge z = v^2)$$

in 40 Sekunden. Das Ergebnis mit 12 atomaren Formeln ist

$(x = 0 \wedge y = 0 \wedge z = 0) \vee (z \neq 0 \wedge x = 0 \wedge y = 0 \wedge (\neg(z + 1 = 0 \vee z <$
$0))) \vee (y \neq 0 \wedge yz - x^2 = 0 \wedge (\neg(y + 1 = 0 \vee y < 0))).$

Die zweite Eingabeformel ist

$$\exists u \exists v(x = uv \wedge y = uv^2 \wedge z = u^2).$$

Das Ergebnis

$(x \neq 0 \wedge z \neq 0 \wedge x^4 - y^2 z = 0) \vee$
$(z \neq 0 \wedge x = 0 \wedge y = 0 \wedge (\neg(z + 1 = 0 \vee z < 0))) \vee (x = 0 \wedge y = 0 \wedge z = 0)$

enthält 11 atomare Formeln und wurde in 79 Sekunden berechnet.

A.4. Bekannte Testbeispiele

Das Davenport–Heintz Problem. Dies ist ein bekanntes Testproblem für Verfahren zur reellen Quantorenelimination. Gegeben ist die Formel

$$\exists c \forall b \forall a(((a = d \wedge b = c) \vee (a = c \wedge b = 1)) \Longrightarrow a^2 = b).$$

Bekannte äquivalente quantorenfreie Formeln sind $d^4 = 1$ und $d = 1 \vee d = -1$ [DH88]. Das Ergebnis der Elimination ist

$(d + 1 = 0 \wedge d^4 + 1 \neq 0) \vee$
$(d \neq 0 \wedge d - 1 \neq 0 \wedge d + 1 \neq 0 \wedge d^2 + 1 = 0 \wedge d^4 + 1 \neq 0) \vee$
$(d - 1 = 0 \wedge d^4 + 1 \neq 0).$

Dieses Resultat mit 9 atomaren Formeln wurde in 82 Sekunden berechnet. Das Ergebnis konnte mittels eines vom Autor implementierten Simplifikationsverfahren zu den optimalen Lösungen $d^4 - 1 = 0$ bzw. $d - 1 = 0 \vee d + 1 = 0$ vereinfacht werden.

Das Quartik Problem. Beim Quartik Problem wird eine quantorenfreie Formel gesucht, ob eine Parabel vierten Grades positiv definit ist. Eine von Hand bestimmte Lösung findet sich in [Laz88]. Die Eingabeformel ist

$$\forall x(x^4 + px^2 + qx + r \geqslant 0).$$

Die Formel kann in 216 Sekunden eliminiert werden. Wir verzichten auf einen Abdruck der Ausgabe des Programms. Sei $T_6(\underline{c})$ die folgende Formel

$$\neg((c0 = 0 \wedge c1 = 0 \wedge ((c2 = 0 \wedge c3 = 0 \wedge (c5 = 0 \vee c4 < 0)) \vee (c2 > 0 \wedge (c4 < 0 \vee c3 * c5 < 0)))) \vee (c0 < 0 \wedge (((c2 > 0 \vee c1 * c3 < 0) \wedge (c4 < 0 \vee c3 * c5 < 0)) \vee (c1 * c5 > 0 \wedge (c2 * c4) < 0)))).$$

Ergebnis der Elimination ist die $T_6(\underline{c})$ mit 16 atomaren Formeln, wobei die c_i durch die folgenden Terme substituiert werden.

$$c_0 := -(12230590464q^4r^9 + 7247757312p^3q^2r^9 + 1073741824p^6r^9 - 18345885696p^2q^4r^8 - 10871635968p^5q^2r^8 - 1610612736p^8r^8 + 20639121408pq^6r^7 + 23696769024p^4q^4r^7 + 8606711808p^7q^2r^7 + 1006632960p^{10}r^7 - 3869835264q^8r^6 - 23505666048p^3q^6r^6 - 16732127232p^6q^4r^6 - 4127195136p^9q^2r^6 - 335544320p^{12}r^6 + 15479341056p^2q^8r^5 + 17485922304p^5q^6r^5 + 7001800704p^8q^4r^5 + 1154482176p^{11}q^2r^5 + 62914560p^{14}r^5 - 4353564672pq^{10}r^4 - 10480803840p^4q^8r^4 - 6569164800p^7q^6r^4 - 1657110528p^{10}q^4r^4 - 174587904p^{13}q^2r^4 - 6291456p^{16}r^4 + 408146688q^{12}r^3 + 4716361728p^3q^{10}r^3 + 3986288640p^6q^8r^3 + 1278996480p^9q^6r^3 + 186028032p^{12}q^4r^3 + 11993088p^{15}q^2r^3 + 262144p^{18}r^3 - 1428513408p^2q^{12}r^2 - 1360488960p^5q^{10}r^2 - 489888000p^8q^8r^2 - 82861056p^{11}q^6r^2 - 6586368p^{14}q^4r^2 - 196608p^{17}q^2r^2 + 229582512pq^{14}r + 229582512p^4q^{12}r + 88179840p^7q^{10}r + 16236288p^{10}q^8r + 1437696p^{13}q^6r + 49152p^{16}q^4r - 14348907q^{16} - 14880348p^3q^{14} - 5983632p^6q^{12} - 1166400p^9q^{10} - 110592p^{12}q^8 - 4096p^{15}q^6)/4294967296$$

$$c_1 := -(254803968q^4r^8 + 301989888p^3q^2r^8 - 226492416p^2q^2r^8 + 1358954496pq^2r^8 + 67108864p^6r^8 - 67108864p^5r^8 + 402653184p^4r^8 - 339738624p^2q^4r^7 - 254803968pq^4r^7 + 509607936q^4r^7 - 440401920p^5q^2r^7 + 207618048p^4q^2r^7 - 2000683008p^3q^2r^7 + 226492416p^2q^2r^7 - 1358954496pq^2r^7 - 100663296p^8r^7 + 83886080p^7r^7 - 637534208p^6r^7 + 67108864p^5r^7 - 402653184p^4r^7 + 429981696pq^6r^6 + 143327232q^6r^6 + 681246720p^4q^4r^6 + 42467328p^3q^4r^6 + 1507590144p^2q^4r^6 + 509607936pq^4r^6 - 1528823808q^4r^6 + 376438784p^7q^2r^6 - 141557760p^6q^2r^6 + 1918894080p^5q^2r^6 - 75497472p^4q^2r^6 + 1358954496p^3q^2r^6 + 62914560p^{10}r^6 - 41943040p^9r^6 + 436207616p^8r^6 - 67108864p^7r^6 + 536870912p^6r^6 - 53747712q^8r^5 - 453869568p^3q^6r^5 - 382205952p^2q^6r^5 + 143327232pq^6r^5 - 429981696q^6r^5 - 557973504p^6q^4r^5 - 10616832p^5q^4r^5 - 2043740160p^4q^4r^5 - 509607936p^3q^4r^5 - 191102976p^2q^4r^5 - 197656576p^9q^2r^5 + 65798144p^8q^2r^5 - 1184366592p^7q^2r^5 - 28311552p^6q^2r^5 - 1075838976p^5q^2r^5 - 20971520p^{12}r^5 + 10485760p^{11}r^5 - 167772160p^{10}r^5 + 25165824p^9r^5 - 301989888p^8r^5 + 286654464p^2q^8r^4 + 214990848pq^8r^4 - 188116992q^8r^4 + 463822848p^5q^6r^4 + 161243136p^4q^6r^4 + 1039122432p^3q^6r^4 + 668860416p^2q^6r^4 - 1003290624pq^6r^4 + 265654272p^8q^4r^4 - 8257536p^7q^4r^4 + 1340964864p^6q^4r^4 + 297271296p^5q^4r^4 + 764411904p^4q^4r^4 + 58720256p^{11}q^2r^4 - 15433728p^{10}q^2r^4 + 433913856p^9q^2r^4 + 15204352p^8q^2r^4 + 625999872p^7q^2r^4 + 3932160p^{14}r^4 - 1310720p^{13}r^4 + 39321600p^{12}r^4 - 4194304p^{11}r^4 + 92274688p^{10}r^4 - 60466176pq^{10}r^3 - 30233088q^{10}r^3 - 198567936p^4q^8r^3 - 174680064p^3q^8r^3 - 156764160p^2q^8r^3 - 349360128pq^8r^3 + 80621568q^8r^3 - 198402048p^7q^6r^3 - 49102848p^6q^6r^3 - 832094208p^5q^6r^3 - 376233984p^4q^6r^3 - 238878720p^3q^6r^3 - 69246976p^{10}q^4r^3 + 4313088p^9q^4r^3 - 479182848p^8q^4r^3 - 91422720p^7q^4r^3 - 567115776p^6q^4r^3 - 9125888p^{13}q^2r^3 + 1499136p^{12}q^2r^3 - 92028928p^{11}q^2r^3 - 2260992p^{10}q^2r^3 - 199819264p^9q^2r^3 - 393216p^{16}r^3 +$$

$$65536p^{15}r^3 - 5636096p^{14}r^3 + 262144p^{13}r^3 - 16252928p^{12}r^3 + 2834352q^{12}r^2 + 74742912p^3q^{10}r^2 + 88179840p^2q^{10}r^2 - 75582720pq^{10}r^2 + 45349632q^{10}r^2 + 101088000p^6q^8r^2 + 38817792p^5q^8r^2 + 326965248p^4q^8r^2 + 181398528p^3q^8r^2 - 100776960p^2q^8r^2 + 43888896p^9q^6r^2 + 3020544p^8q^6r^2 + 294285312p^7q^6r^2 + 73654272p^6q^6r^2 + 322486272p^5q^6r^2 + 7952640p^{12}q^4r^2 - 329728p^{11}q^4r^2 + 87297024p^{10}q^4r^2 + 7741440p^9q^4r^2 + 192651264p^8q^4r^2 + 618496p^{15}q^2r^2 - 32768p^{14}q^2r^2 + 10665984p^{13}q^2r^2 + 196608p^{12}q^2r^2 + 31457280p^{11}q^2r^2 + 16384p^{18}r^2 + 458752p^{16}r^2 + 1572864p^{14}r^2 - 17950896p^2q^{12}r - 19840464pq^{12}r + 14171760q^{12}r - 25334208p^5q^{10}r - 10497600p^4q^{10}r - 101196864p^3q^{10}r - 50388480p^2q^{10}r - 15116544pq^{10}r - 11930976p^8q^8r - 1524096p^7q^8r - 106033536p^6q^8r - 28180224p^5q^8r - 161803008p^4q^8r - 2472576p^{11}q^6r - 32256p^{10}q^6r - 39700224p^9q^6r - 5218560p^8q^6r - 100196352p^7q^6r - 233472p^{14}q^4r + 4096p^{13}q^4r - 6875392p^{12}q^4r - 438272p^{11}q^4r - 21931008p^{10}q^4r - 8192p^{17}q^2r - 552960p^{15}q^2r - 16384p^{14}q^2r - 2015232p^{13}q^2r - 16384p^{18}r - 65536p^{16}r + 1594323pq^{14} + 1594323q^{14} + 2263545p^4q^{12} + 944784p^3q^{12} + 12518388p^2q^{12} + 5668704pq^{12} + 8503056q^{12} + 1113912p^7q^{10} + 174960p^6q^{10} + 13506912p^5q^{10} + 3674160p^4q^{10} + 25194240p^3q^{10} + 248400p^{10}q^8 + 10368p^9q^8 + 5607792p^8q^8 + 855360p^7q^8 + 15256512p^6q^8 + 25856p^{13}q^6 + 1121088p^{11}q^6 + 92160p^{10}q^6 + 3697920p^9q^6 + 1024p^{16}q^4 + 108544p^{14}q^4 + 4096p^{13}q^4 + 401408p^{12}q^4 + 4096p^{17}q^2 + 16384p^{15}q^2)/268435456$$

$$c_2 := (14155776p^2q^2r^7 - 28311552pq^2r^7 + 84934656q^2r^7 + 4194304p^5r^7 - 8388608p^4r^7 + 25165824p^3r^7 + 15925248pq^4r^6 - 10616832q^4r^6 - 12976128p^4q^2r^6 + 40108032p^3q^2r^6 - 108527616p^2q^2r^6 + 56623104pq^2r^6 - 5242880p^7r^6 + 14680064p^6r^6 - 37748736p^5r^6 + 54525952p^4r^6 - 50331648p^3r^6 + 150994944p^2r^6 - 21233664p^2q^4r^5 + 79626240pq^4r^5 + 47775744q^4r^5 + 8847360p^6q^2r^5 - 34209792p^5q^2r^5 + 68419584p^4q^2r^5 - 7077888p^3q^2r^5 - 113246208p^2q^2r^5 + 226492416pq^2r^5 + 84934656q^2r^5 + 2621440p^9r^5 - 11534336p^8r^5 + 23068672p^7r^5 - 72351744p^6r^5 + 54525952p^5r^5 - 197132288p^4r^5 + 25165824p^3r^5 + 14929920p^2q^6r4 - 2985984pq^6r^4 - 5971968q^6r^4 - 1990656p^5q^4r^4 + 32182272p4q^4r^4 - 56623104p^3q^4r^4 + 39813120p^2q^4r^4 + 31850496pq^4r^4 + 63700992q^4r^4 - 4112384p^8q^2r^4 + 22118400p^7q^2r^4 - 24903680p^6q^2r^4 + 40108032p^5q^2r^4 + 73138176p^4q^2r^4 - 51904512p^3q^2r^4 - 70778880p^2q^2r^4 - 655360p^{11}r^4 + 5242880p^{10}r^4 - 7340032p^9r^4 + 40632320p^8r^4 - 24117248p^7r^4 + 108003328p^6r4 - 20971520p^5r^4 - 3359232pq^8r^3 + 4478976q^8r^3 - 3732480p^4q6r^3 - 7796736p^3q^6r^3 + 16920576p^2q^6r^3 + 32845824pq^6r^3 + 47775744q^6r^3 + 1511424p^7q^4r^3 - 19488768p^6q^4r^3 + 9142272p^5q^4r^3 - 14598144p^4q^4r^3 - 88473600p^3q^4r^3 + 122093568p^2q^4r^3 + 15925248pq^4r^3 + 964608p^{10}q^2r^3 - 9924608p^9q^2r^3 + 5459968p^8q^2r^3 - 48496640p^7q^2r^3 - 12517376p^6q^2r^3 - 68812800p^5q^2r^3 + 27721728p^4q^2r^3 + 81920p^{13}r^3 - 1474560p^{12}r^3 + 1277952p^{11}r^3 - 12386304p^{10}r^3 + 5505024p^9r^3 - 31719424p^8r^3 + 6815744p^7r^3 + 5038848p^3q^8r^2 + 1119744p^2q^8r^2 - 1679616pq^8r^2 + 10077696q^8r^2 + 933120p^6q^6r^2 + 9994752p^5q^6r^2 + 1866240p^4q^6r^2 + 22394880p^3q^6r^2 + 19408896p^2q^6r^2 + 11943936pq^6r^2 - 8957952q^6r^2 - 435456p^9q^4r^2 + 8159232p^8q4r^2 + 2672640p^7q^4r^2 + 24662016p^6q^4r^2 + 40919040p^5q^4r2 - 18247680p^4q^4r^2 - 3981312p^3q^4r^2 - 93696p^{12}q^2r^2 + 2724864p^{11}q^2r^2 - 698368p^{10}q^2r^2 + 18948096p^9q^2r^2 + 147456p^8q^2r^2 + 35422208p^7q^2r^2 - 3932160p^6q^2r^2 - 4096p^{15}r^2 + 253952p^{14}r^2 - 114688p^{13}r^2 + 2162688p^{12}r^2 - 655360p^{11}r^2 + 5242880p^{10}r^2 - 1048576p^9r^2 - 1732104p^2q^{10}r + 2204496pq^{10}r + 1574640q^{10}r - 536544p^5q^8r - 979776p^4q^8r - 2892672p^3q^8r + 5878656p^2q^8r + 3359232pq^8r + 8398080q^8r + 74304p^8q^6r - 3514752p^7q^6r - 3746304p^6q^6r - 6531840p^5q^6r - 25878528p^4q^6r + 32099328p^3q^6r - 14929920p^2q^6r + 30976p^{11}q^4r - 2091008p^{10}q^4r - 395008p^9q^4r - 12950784p^8q^4r - 5087232p^7q^4r - 15943680p^6q^4r - 3981312p^5q^4r + 2048p^{14}q^2r - 402432p^{13}q^2r + 36352p^{12}q^2r -$$

$$3113984p^{11}q^2r + 45056p^{10}q^2r - 6119424p^9q^2r + 352256p^8q^2r - 24576p^{16}r + 4096p^{15}r - 204800p^{14}r + 32768p^{13}r - 458752p^{12}r + 65536p^{11}r + 177147pq^{12} - 177147q^{12} + 65610p^4q^{10} + 813564p^3q^{10} + 1312200p^2q^{10} + 1574640pq^{10} + 629856q^{10} - 3888p^7q^8 + 1452816p^6q^8 + 1286928p^5q^8 + 5901984p^4q^8 + 5785344p^3q^8 + 559872p^2q^8 + 3359232pq^8 - 3168p^{10}q^6 + 854448p^9q^6 + 320832p^8q^6 + 5273856p^7q^6 + 1842048p^6q^6 + 6905088p^5q^6 + 1119744p^4q^6 - 256p^{13}q^4 + 214928p^{12}q^4 + 30592p^{11}q^4 + 1554304p^{10}q4 + 152576p^9q^4 + 2773248p^8q^4 - 18432p^7q^4 + 24320p^{15}q^2 + 1024p^{14}q^2 + 188416p^{13}q^2 + 372736p^{11}q^2 - 16384p^{10}q^2 + 1024p^{18} + 8192p^{16} + 16384p^{14})/16777216$$

$$c_3 := (1769472q^2r^6 + 524288p^3r^6 - 2064384p^2q^2r^5 + 589824pq^2r^5 - 1769472q^2r^5 - 786432p^5r^5 + 786432p^4r^5 - 3670016p^3r^5 + 4718592p^2r^5 - 9437184pr^5 + 1658880pq^4r^4 + 331776q^4r^4 + 1228800p^4q^2r^4 + 638976p^3q^2r^4 - 3244032p^2q^2r^4 + 8257536pq^2r^4 - 8847360q^2r^4 + 524288p^7r^4 - 851968p^6r^4 + 4194304p^5r^4 - 5242880p^4r^4 + 13107200p^3r^4 - 4718592p^2r^4 + 9437184pr^4 - 373248q^6r^3 - 1271808p^3q^4r^3 + 608256p^2q^4r^3 - 2985984pq^4r^3 - 1658880q^4r^3 - 471040p^6q^2r^3 - 196608p^5q^2r^3 + 1105920p^4q^2r^3 - 5259264p^3q^2r^3 + 4866048p^2q^2r^3 - 7077888pq^2r^3 + 3538944q^2r^3 - 196608p^9r^3 + 376832p^8r^3 - 1966080p^7r^3 + 2457600p^6r^3 - 7012352p^5r^3 + 3801088p^4r^3 - 7602176p^3r^3 + 311040p^2q^6r^2 - 62208pq^6r^2 + 248832q^6r^2 + 225792p^5q^4r^2 - 23040p^4q^4r^2 - 313344p^3q^4r^2 + 2543616p^2q^4r^2 + 995328pq^4r^2 - 995328q^4r^2 + 144128p^8q^2r^2 - 194560p^7q^2r^2 + 561152p^6q^2r^2 + 655360p^5q^2r^2 - 405504p^4q^2r^2 + 5013504p^3q^2r^2 - 1327104p^2q^2r^2 + 43008p^{11}r^2 - 86016p^{10}r^2 + 475136p^9r^2 - 606208p^8r^2 + 1835008p^7r^2 - 1179648p^6r^2 + 2359296p^5r^2 - 279936pq^8r + 69984q^8r - 69984p^4q^6r + 207360p^3q^6r - 673920p^2q^6r + 124416pq^6r - 746496q^6r + 28032p^7q^4r - 8832p^6q^4r + 268032p^5q^4r - 573696p^4q^4r - 1410048p^3q^4r + 580608p^2q^4r - 2985984pq^4r - 30592p^{10}q^2r + 74496p^9q^2r - 217856p^8q^2r + 78848p^7q^2r - 41984p^6q^2r - 1007616p^5q^2r + 258048p^4q^2r - 5120p^{13}r + 10240p^{12}r - 59392p^{11}r + 77824p^{10}r - 237568p^9r + 163840p^8r - 327680p^7r + 19683q^{10} - 81648p^3q^8 + 23328p^2q^8 + 139968q^8 - 58320p^6q^6 + 41040p^5q^6 - 322272p^4q^6 + 368064p^3q^6 + 93312p^2q^6 - 373248pq^6 + 186624q^6 + 464p^9q^4 - 16368p^8q^4 - 72448p^7q^4 + 169728p^6q^4 - 587520p^5q^4 + 864000p^4q^4 - 1078272p^3q^4 + 2784p^{12}q^2 - 6464p^{11}q^2 + 16512p^{10}q^2 - 14848p^9q^2 + 18944p^8q^2 + 39936p^7q^2 - 10240p^6q^2 + 256p^{15} - 512p^{14} + 3072p^{13} - 4096p^{12} + 12288p^{11} - 8192p^{10} + 16384p^9)/1048576$$

$$c_4 := -(2304q^2r^4 + 2048p^3r^4 - 2048p^2r^4 + 12288pr^4 - 1536p^2q^2r^3 - 4608pq^2r^3 + 13824q^2r^3 - 2048p^5r^3 + 2048p^4r^3 - 14336p^3r^3 + 16384p^2r^3 - 24576pr^3 + 36864r^3 + 2592pq^4r^2 + 3888q^4r^2 + 2304p^4q^2r^2 + 5760p^3q^2r^2 - 8640p^2q^2r^2 + 27648pq^2r^2 - 18432q^2r^2 + 768p^7r^2 - 896p^6r^2 + 6400p^5r^2 - 10752p^4r^2 + 17408p^3r^2 - 20480p^2r^2 + 12288pr^2 + 486q^6r - 792p^3q^4r + 1080p^2q^4r - 3024pq^4r + 3888q^4r - 1008p^6q^2r - 2304p^5q^2r + 3616p^4q^2r - 19008p^3q^2r + 18240p^2q^2r - 16128pq^2r + 6912q^2r - 128p^9r + 192p^8r - 1280p^7r + 2432p^6r - 4352p^5r + 4096p^4r - 5120p^3r + 648p^2q^6 + 729pq^6 + 486q^6 + 564p^5q^4 - 1179p^4q^4 + 5004p^3q^4 - 4644p^2q^4 + 3888pq^4 + 1296q^4 + 123p^8q^2 + 352p^7q^2 - 708p^6q^2 + 2912p^5q^2 - 2000p^4q^2 + 2304p^3q^2 + 4032p^2q^2 + 8p^{11} - 16p^{10} + 96p^9 - 192p^8 + 384p^7 - 256p^6 + 512p^5)/4096$$

$$c_5 = -(128p^2r^2 - 256pr^2 + 768r^2 + 288pq^2r - 288q^2r - 64p^4r + 128p^3r - 384p^2r + 256pr - 768r + 27q^4 - 136p^3q^2 + 204p^2q^2 - 432pq^2 + 144q^2 + 8p^6 - 16p^5 + 64p^4 - 64p^3 + 128p^2)/256$$

Für unser Eliminationsverfahren ist es geschickter, eine äquivalente Formulierung des Problems von Gonzales-Vega [GV93] zu benutzen. Äquivalent zum Quartik Problem ist die Frage, ob eine Parabel vierten Grades keine einfache Nullstelle besitzt. Wir eliminieren die Negation

$$\exists x(x^4 + px^2 + qx + r = 0 \wedge 4x^3 + 2px + q \neq 0).$$

dieser Aussage. Sei $T_4(\underline{c})$ die folgende vom Programm berechnete Typformel für Polynome des Grades 4.

$$(c0 = 0 \wedge c1 = 0 \wedge (c3 = 0 \vee c2 < 0)) \vee (c0 > 0 \wedge (c2 < 0 \vee c1 * c3 < 0)).$$

Das Ergebnis der Quantorenelimination ist $T_4(c_0, \ldots, c_3)$, wobei die c_i durch die folgenden Terme substituiert werden.

$c_0 := (16777216r^9 - 25165824p^2r^8 + 28311552pq^2r^7 + 15728640p^4r^7 - 5308416q^4r^6 - 29097984p^3q^2r^6 - 5242880p^6r^6 + 21233664p^2q^4r^5 + 11403264p^5q^2r^5 + 983040p^8r^5 - 5971968pq^6r^4 - 10838016p^4q^4r^4 - 2064384p^7q^2r^4 - 98304p^{10}r^4 + 559872q^8r^3 + 6137856p^3q^6r^3 + 1781760p^6q^4r^3 + 159744p^9q^2r^3 + 4096p^{12}r^3 - 1959552p^2q^8r^2 - 705024p^5q^6r^2 - 82176p^8q^4r^2 - 3072p^{11}q^2r^2 + 314928pq^{10}r + 128304p^4q^8r + 17280p^7q^6r + 768p^{10}q^4r - 19683q^{12} - 8748p^3q^{10} - 1296p^6q^8 - 64p^9q^6)$

$c_1 := -(393216pr^7 + 262144r^7 - 884736q^2r^6 - 524288p^3r^6 - 786432p^2r^6 - 131072pr^6 - 262144r^6 + 1622016p^2q^2r^5 + 688128pq^2r^5 + 294912q^2r^5 + 278528p^5r^5 + 753664p^4r^5 + 131072p^3r^5 + 393216p^2r^5 - 1410048pq^4r^4 - 909312p^4q^2r^4 - 1155072p^3q^2r^4 - 245760p^2q^2r^4 - 589824pq^2r^4 - 73728p^7r^4 - 344064p^6r^4 - 49152p^5r^4 - 229376p^4r^4 + 279936q^6r^3 + 1078272p^3q^4r^3 + 585216p^2q^4r^3 + 193536pq^4r^3 + 221184q^4r^3 + 211968p^6q^2r^3 + 616448p^5q^2r^3 + 73728p^4q^2r^3 + 450560p^3q^2r^3 + 9728p^9r^3 + 82944p^8r^3 + 8192p^7r^3 + 65536p^6r^3 - 684288p^2q^6r^2 - 41472pq^6r^2 - 31104q^6r^2 - 210240p^5q^4r^2 - 450816p^4q^4r^2 - 32256p^3q^4r^2 - 373248p^2q^4r^2 - 19584p^8q^2r^2 - 136704p^7q^2r^2 - 9216p^6q^2r^2 - 116736p^5q^2r^2 - 512p^{11}r^2 - 10240p^{10}r^2 - 512p^9r^2 - 9216p^8r^2 + 196830pq^8r - 2916q^8r + 75816p^4q^6r + 180576p^3q^6r + 10368p^2q^6r + 155520pq^6r + 9504p^7q^4r + 80864p^6q^4r + 4416p^5q^4r + 71424p^4q^4r + 384p^{10}q^2r + 11520p^9q^2r + 384p^8q^2r + 10752p^7q^2r + 512p^{12}r + 512p^{10}r - 19683q^{10} - 8748p^3q^8 - 23814p^2q^8 - 1458pq^8 - 20412q^8 - 1296p^6q^6 - 12600p^5q^6 - 648p^4q^6 - 11232p^3q^6 - 64p^9q^4 - 2208p^8q^4 - 64p^7q^4 - 2080p^6q^4 - 128p^{11}q^2 - 128p^9q^2)$

$c_2 := (6144pr^5 - 4096r^5 - 13824q^2r^4 - 17408p^3r^4 + 3072p^2r^4 - 8192pr^4 - 4096r^4 + 14976p^2q^2r^3 + 2304pq^2r^3 - 8704q^2r^3 + 14720p^5r^3 - 768p^4r^3 + 13312p^3r^3 + 3072p^2r^3 + 2048pr^3 - 12312pq^4r^2 - 9936q^4r^2 - 17088p^4q^2r^2 - 1664p^3q^2r^2 - 11520p^2q^2r^2 + 768pq^2r^2 - 4608q^2r^2 - 5504p^7r^2 + 64p^6r^2 - 7168p^5r^2 - 768p^4r^2 - 2048p^3r^2 - 2916q^6r + 6588p^3q^4r + 1620p^2q^4r - 288pq^4r - 3024q^4r + 7136p^6q^2r + 528p^5q^2r + 8608p^4q^2r + 128p^3q^2r + 2688p^2q^2r + 960p^9r + 1600p^7r + 64p^6r + 640p^5r - 4131p^2q^6 - 1458pq^6 - 702q^6 - 3852p^5q^4 - 648p^4q^4 - 4076p^3q^4 - 396p^2q^4 - 1512pq^4 - 912p^8q^2 - 64p^7q^2 - 1424p^6q^2 - 80p^5q^2 - 640p^4q^2 - 64p^{11} - 128p^9 - 64p^7)$

$c_3 := (64r^3 - 144p^2r^2 + 96pr^2 - 64r^2 - 306pq^2r + 108q^2r + 64p^4r - 56p^3r + 48p^2r - 32pr - 27q^4 + 134p^3q^2 - 90p^2q^2 + 54pq^2 - 28q^2 - 8p^6 + 8p^5 - 8p^4 + 8p^3)$

A.5. Der reelle Rabinowitsch Trick

Die Anwendung des reellen Rabinowitsch Tricks demonstrieren wir an einem für unser Verfahren schlecht geeignetem Beispiel. Gegeben ist die Formel

$$\exists x(ax + b > 0 \wedge cx + d > 0).$$

Das Ergebnis der Elimination hat die folgende Struktur

$((a < 0 \vee (a = 0 \wedge b > 0)) \wedge (c < 0 \vee (c = 0 \wedge d > 0))) \vee ((a > 0 \vee (a = 0 \wedge b > 0)) \wedge (c > 0 \vee (c = 0 \wedge d > 0))) \vee (c - a \neq 0 \wedge (\neg T_4(c_0, c_1, c_2, c_3))).$

Dabei sind die c_i die folgenden Terme.

$c_0 := (a^{12}d^{12} - 12a^{11}bcd^{11} + 66a^{10}b^2c^2d^{10} - 220a^9b^3c^3d^9 + 495a^8b^4c^4d^8 - 792a^7b^5c^5d^7 + 924a^6b^6c^6d^6 - 792a^5b^7c^7d^5 + 495a^4b^8c^8d^4 - 220a^3b^9c^9d^3 +$

$$66a^2b^{10}c^{10}d^2 - 12ab^{11}c^{11}d + b^{12}c^{12})/$$
$$(c^{12} - 12ac^{11} + 66a^2c^{10} - 220a^3c^9 + 495a^4c^8 - 792a^5c^7 + 924a^6c^6 - 792a^7c^5 + 495a^8c^4 - 220a^9c^3 + 66a^{10}c^2 - 12a^{11}c + a^{12})$$

$$c_1 := -(a^{10}d^{10} - 10a^9bcd^9 - 2a^9cd^9 + 2a^{10}d^9 + 45a^8b^2c^2d^8 + 18a^8bc^2d^8 + a^8c^2d^8 - 18a^9bcd^8 - 2a^9cd^8 + a^{10}d^8 - 120a^7b^3c^3d^7 - 72a^7b^2c^3d^7 - 8a^7bc^3d^7 + 72a^8b^2c^2d^7 + 16a^8bc^2d^7 - 8a^9bcd^7 + 210a^6b^4c^4d^6 + 168a^6b^3c^4d^6 + 28a^6b^2c^4d^6 - 168a^7b^3c^3d^6 - 56a^7b^2c^3d^6 + 28a^8b^2c^2d^6 - 252a^5b^5c^5d^5 - 252a^5b^4c^5d^5 - 56a^5b^3c^5d^5 + 252a^6b^4c^4d^5 + 112a^6b^3c^4d^5 - 56a^7b^3c^3d^5 + 210a^4b^6c^6d^4 + 252a^4b^5c^6d^4 + 70a^4b^4c^6d^4 - 252a^5b^5c^5d^4 - 140a^5b^4c^5d^4 + 70a^6b^4c^4d^4 - 120a^3b^7c^7d^3 - 168a^3b^6c^7d^3 - 56a^3b^5c^7d^3 + 168a^4b^6c^6d^3 + 112a^4b^5c^6d^3 - 56a^5b^5c^5d^3 + 45a^2b^8c^8d^2 + 72a^2b^7c^8d^2 + 28a^2b^6c^8d^2 - 72a^3b^7c^7d^2 - 56a^3b^6c^7d^2 + 28a^4b^6c^6d^2 - 10ab^9c^9d - 18ab^8c^9d - 8ab^7c^9d + 18a^2b^8c^8d + 16a^2b^7c^8d - 8a^3b^7c^7d + b^{10}c^{10} + 2b^9c^{10} + b^8c^{10} - 2ab^9c^9 - 2ab^8c^9 + a^2b^8c^8)/$$
$$(c^{10} - 10ac^9 + 45a^2c^8 - 120a^3c^7 + 210a^4c^6 - 252a^5c^5 + 210a^6c^4 - 120a^7c^3 + 45a^8c^2 - 10a^9c + a^{10})$$

$$c2 = -(2a^7d^7 - 14a^6bcd^6 - 2a^6cd^6 + 2a^7d^6 + 42a^5b^2c^2d^5 + 12a^5bc^2d^5 + 2a^5c^2d^5 - 12a^6bcd^5 - 4a^6cd^5 + 2a^7d^5 - 70a^4b^3c^3d^4 - 30a^4b^2c^3d^4 - 10a^4bc^3d^4 + 30a^5b^2c^2d^4 + 20a^5bc^2d^4 - 10a^6bcd^4 + 70a^3b^4c^4d^3 + 40a^3b^3c^4d^3 + 20a^3b^2c^4d^3 - 40a^4b^3c^3d^3 - 40a^4b^2c^3d^3 + 20a^5b^2c^2d^3 - 42a^2b^5c^5d^2 - 30a^2b^4c^5d^2 - 20a^2b^3c^5d^2 + 30a^3b^4c^4d^2 + 40a^3b^3c^4d^2 - 20a^4b^3c^3d^2 + 14ab^6c^6d + 12ab^5c^6d + 10ab^4c^6d - 12a^2b^5c^5d - 20a^2b^4c^5d + 10a^3b^4c^4d - 2b^7c^7 - 2b^6c^7 - 2b^5c^7 + 2ab^6c^6 + 4ab^5c^6 - 2a^2b^5c^5)/$$
$$(c^7 - 7ac^6 + 21a^2c^5 - 35a^3c^4 + 35a^4c^3 - 21a^5c^2 + 7a^6c - a^7)$$

$$c3 = -(a^4d^4 - 4a^3bcd^3 - 2a^3cd^3 + 2a^4d^3 + 6a^2b^2c^2d^2 + 6a^2bc^2d^2 + a^2c^2d^2 - 6a^3bcd^2 - 2a^3cd^2 + a^4d^2 - 4ab^3c^3d - 6ab^2c^3d - 2abc^3d + 6a^2b^2c^2d + 4a^2bc^2d - 2a^3bcd + b^4c^4 + 2b^3c^4 + b^2c^4 - 2ab^3c^3 - 2ab^2c^3 + a^2b^2c^2)/$$
$$(c^4 - 4ac^3 + 6a^2c^2 - 4a^3c + a^4)$$

Die Ergebnisformel hat 20 atomare Formeln und wurde in 229 Sekunden berechnet.

Wendet man den reellen Rabinowitsch Trick an, so erhält man die Eingabeformel

$$\exists x \exists y \exists z (axz^2 + bz^2 - 1 = 0 \wedge cxy^2 + dy^2 - 1 = 0).$$

Das folgende Ergebnis mit 175 atomaren Formeln wurde in 840 Sekunden berechnet. Im Gegensatz zum vorherigen Ergebnis enthält das letzte Ergebnis nur 2 mäßig komplexe Terme besitzt aber dafür aber fast 9 mal soviele atomare Formeln.

$$((ad - bc \neq 0 \wedge a \neq 0 \wedge c \neq 0 \wedge$$
$$(\neg(a^8d^8 - 8a^7bcd^7 + 28a^6b^2c^2d^6 - 56a^5b^3c^3d^5 + 70a^4b^4c^4d^4 - 56a^3b^5c^5d^3 + 28a^2b^6c^6d^2 - 8ab^7c^7d + b^8c^8 - 2a^7cd^7 + 14a^6bc^2d^6 - 42a^5b^2c^3d^5 + 70a^4b^3c^4d^4 - 70a^3b^4c^5d^3 + 42a^2b^5c^6d^2 - 14ab^6c^7d + 2b^7c^8 + 2a^6c^2d^6 - 12a^5bc^3d^5 + 30a^4b^2c^4d^4 - 40a^3b^3c^5d^3 + 30a^2b^4c^6d^2 - 12ab^5c^7d + 2b^6c^8 - 2a^5c^3d^5 + 10a^4bc^4d^4 - 20a^3b^2c^5d^3 + 20a^2b^3c^6d^2 - 10ab^4c^7d + 2b^5c^8 + a^4c^4d^4 - 4a^3bc^5d^3 + 6a^2b^2c^6d^2 - 4ab^3c^7d + b^4c^8 =$$
$$0 \vee$$
$$a^{14}c^{28}d^{14} - 14a^{13}bc^{29}d^{13} + 91a^{12}b^2c^{30}d^{12} - 364a^{11}b^3c^{31}d^{11} + 1001a^{10}b^4c^{32}d^{10} - 2002a^9b^5c^{33}d^9 + 3003a^8b^6c^{34}d^8 - 3432a^7b^7c^{35}d^7 + 3003a^6b^8c^{36}d^6 - 2002a^5b^9c^{37}d^5 + 1001a^4b^{10}c^{38}d^4 - 364a^3b^{11}c^{39}d^3 + 91a^2b^{12}c^{40}d^2 - 14ab^{13}c^{41}d + b^{14}c^{42} - 4a^{13}c^{29}d^{13} + 52a^{12}bc^{30}d^{12} - 312a^{11}b^2c^{31}d^{11} + 1144a^{10}b^3c^{32}d^{10} - 2860a^9b^4c^{33}d^9 + 5148a^8b^5c^{34}d^8 - 6864a^7b^6c^{35}d^7 + 6864a^6b^7c^{36}d^6 - 5148a^5b^8c^{37}d^5 + 2860a^4b^9c^{38}d^4 - 1144a^3b^{10}c^{39}d^3 + 312a^2b^{11}c^{40}d^2 - 52ab^{12}c^{41}d + 4b^{13}c^{42} + 6a^{12}c^{30}d^{12} - 72a^{11}bc^{31}d^{11} + 396a^{10}b^2c^{32}d^{10} - 1320a^9b^3c^{33}d^9 + 2970a^8b^4c^{34}d^8 - 4752a^7b^5c^{35}d^7 + 5544a^6b^6c^{36}d^6 - 4752a^5b^7c^{37}d^5 + 2970a^4b^8c^{38}d^4 - 1320a^3b^9c^{39}d^3 + 396a^2b^{10}c^{40}d^2 - 72ab^{11}c^{41}d + 6b^{12}c^{42} - 4a^{11}c^{31}d^{11} + 44a^{10}bc^{32}d^{10} - 220a^9b^2c^{33}d^9 + 660a^8b^3c^{34}d^8 - 1320a^7b^4c^{35}d^7 + 1848a^6b^5c^{36}d^6 - 1848a^5b^6c^{37}d^5 + 1320a^4b^7c^{38}d^4 -$$

$$660a^3b^8c^{39}d^3 + 220a^2b^9c^{40}d^2 - 44ab^{10}c^{41}d + 4b^{11}c^{42} + a^{10}c^{32}d^{10} - 10a^9bc^{33}d^9 + 45a^8b^2c^{34}d^8 - 120a^7b^3c^{35}d^7 + 210a^6b^4c^{36}d^6 - 252a^5b^5c^{37}d^5 + 210a^4b^6c^{38}d^4 - 120a^3b^7c^{39}d^3 + 45a^2b^8c^{40}d^2 - 10ab^9c^{41}d + b^{10}c^{42} < 0))) \vee (ad - bc \neq 0 \wedge a \neq 0 \wedge c \neq 0 \wedge (a^2 > 0 \vee (a^2 = 0 \wedge (ab < 0 \vee (ab = 0 \wedge b^2 > 0))) \vee (a^2 = 0 \wedge (ab > 0 \vee (ab = 0 \wedge b^2 > 0))))) \wedge (a \neq 0 \vee b \neq 0) \wedge (c \neq 0 \vee d \neq 0) \wedge (a \neq 0 \vee b+1 \neq 0)) \vee (a = 0 \wedge c = 0 \wedge d = 0 \wedge b \neq 0 \wedge b+1 \neq 0 \wedge b^2 \geqslant 0) \vee (c = 0 \wedge d = 0 \wedge a \neq 0 \wedge a^2 = 0 \wedge ab > 0 \wedge b+1 \neq 0) \vee (c = 0 \wedge d = 0 \wedge a \neq 0 \wedge a^2 = 0 \wedge ab = 0 \wedge b^2 > 0 \wedge b+1 \neq 0) \vee (c = 0 \wedge d = 0 \wedge a \neq 0 \wedge a^2 = 0 \wedge ab < 0 \wedge b+1 \neq 0) \vee (c = 0 \wedge d = 0 \wedge a \neq 0 \wedge a^2 > 0 \wedge b+1 \neq 0) \vee (c = 0 \wedge d = 0 \wedge a \neq 0 \wedge a^2 = 0 \wedge ab > 0 \wedge b \neq 0 \wedge b+1 \neq 0) \vee (c = 0 \wedge d = 0 \wedge a \neq 0 \wedge a^2 = 0 \wedge ab = 0 \wedge b^2 > 0 \wedge b \neq 0 \wedge b+1 \neq 0) \vee (c = 0 \wedge d = 0 \wedge a \neq 0 \wedge a^2 > 0 \wedge b \neq 0 \wedge b+1 \neq 0) \vee (c = 0 \wedge d = 0 \wedge a \neq 0 \wedge a^2 = 0 \wedge ab > 0) \vee (c = 0 \wedge d = 0 \wedge a \neq 0 \wedge a^2 = 0 \wedge ab = 0 \wedge b^2 > 0) \vee (c = 0 \wedge d = 0 \wedge a \neq 0 \wedge a^2 = 0 \wedge ab < 0) \vee (c = 0 \wedge d = 0 \wedge a \neq 0 \wedge a^2 > 0) \vee (c = 0 \wedge d = 0 \wedge a \neq 0 \wedge a^2 = 0 \wedge ab > 0 \wedge b \neq 0) \vee (c = 0 \wedge d = 0 \wedge a \neq 0 \wedge a^2 = 0 \wedge ab = 0 \wedge b^2 > 0 \wedge b \neq 0) \vee (c = 0 \wedge d = 0 \wedge a \neq 0 \wedge a^2 = 0 \wedge ab < 0 \wedge b \neq 0) \vee (c = 0 \wedge d = 0 \wedge a \neq 0 \wedge a^2 > 0 \wedge b \neq 0) \vee (ad - bc = 0 \wedge a \neq 0 \wedge c \neq 0 \wedge (a^2 > 0 \vee (a^2 = 0 \wedge (ab < 0 \vee (ab = 0 \wedge b^2 > 0))) \vee (a^2 = 0 \wedge (ab > 0 \vee (ab = 0 \wedge b^2 > 0)))) \wedge (a \neq 0 \vee b \neq 0) \wedge (a \neq 0 \vee b+1 \neq 0)) \vee (a = 0 \wedge b \neq 0 \wedge c \neq 0 \wedge b+1 \neq 0 \wedge b^2 \geqslant 0 \wedge (c \neq 0 \vee d \neq 0)) \vee (a = 0 \wedge b \neq 0 \wedge c \neq 0 \wedge b+1 \neq 0 \wedge b^2 \geqslant 0) \vee (c = 0 \wedge a \neq 0 \wedge d \neq 0 \wedge (a^2 > 0 \vee (a^2 = 0 \wedge (ab < 0 \vee (ab = 0 \wedge b^2 > 0))) \vee (a^2 = 0 \wedge (ab > 0 \vee (ab = 0 \wedge b^2 > 0)))) \wedge (a \neq 0 \vee b \neq 0) \wedge (a \neq 0 \vee b+1 \neq 0)) \vee (a = 0 \wedge c = 0 \wedge b \neq 0 \wedge d \neq 0 \wedge b+1 \neq 0 \wedge b^2 \geqslant 0))$$

Literaturverzeichnis

[BKR86] Michael Ben-Or, Dieter Kozen, and John Reif. The complexity of elementary algebra and geometry. *Journal of Computer and System Sciences*, 32:251–264, 1986.

[BW91] Eberhard Becker and Thorsten Woermann. On the trace formula for quadratic forms and some applications. In *Proc. of RAGSQUAD*, Berkeley, 1991. (to appear).

[BW93a] Thomas Becker and Volker Weispfenning. *Gröbner Bases*. Springer-Verlag, New York, 1993.

[BW93b] Thomas Becker and Volker Weispfenning. *Gröbner Bases, A Computational Approach to Commutative Algebra*, volume 141 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, Berlin, Heidelberg, 1993.

[CLO92] David Cox, John Little, and Donald O'Shea. *Ideals, Varieties and Algorithms*. Undergraduate Texts in Mathematics. Springer-Verlag, New York, Berlin, Heidelberg, 1992.

[DH88] James H. Davenport and Joos Heintz. Real Quantifier Elimination is Doubly exponential. *Journal of Symbolic Computation*, 5(1&2):29–35, February 1988.

[Dol93] Andreas Dolzmann. Maslog. Übersicht über die Maslog-Bibliothek für MAS, August 1993.

[GV93] Laureno Gonzales-Vega. A combinatorical algorithm solving some quantifier elimination problems. In *Proc. of the Collins Symposium*, Linz, Austria, October 1993. (to appear).

[Kre93a] Heinz Kredel. MAS modula-2 algebra system, interactive usage. Technical report, Universität Passau, Passau, February 1993. Available for anonymous ftp from alice.fmi.uni-passau.de.

[Kre93b] Heinz Kredel. MAS modula-2 algebra system, specifications, definition modules, indexes. Technical report, Universität Passau, Passau, February 1993. Available for anonymous ftp from alice.fmi.uni-passau.de.

[Laz88] Daniel Lazard. Quantifier Elimination: Optimal Solution for Two Classical Examples. *Journal of Symbolic Computation*, 5(1&2):261–266, February 1988.

[Lip93] Frank Lippold. Implementierung eines Verfahrens zum Zählen reeller Nullstellen multivariater Polynome. Diplomarbeit, Universität Passau, Passau, 1993.

[LT85] P. Lancaster and Tismenetsky. *The Theory of Matrices*. Academic Press, Orlando, 2 edition, 1985.

[Pes94a] Michael Pesch. Computing Comprehensive Gröbner Bases using MAS. User Manual, September 1994.

[Pes94b] Michael Pesch. Die MAS-Implementierung des Algorithmus zur Berechnung umfassender Gröbnerbasen. Anlage zu einem DFG-Antrag, April 1994.

[PRS93] P. Pedersen, M.-F. Roy, and A. Szpirglas. Counting real zeroes in the multivariate case. In *Computational Algebraic Geometry*, pages 203–224, Boston, 1993. MEGA '92, Eysette, F. and Galigo, A.

[Sch91] Elke Schönfeld. Parametrische Gröbnerbasen im Computeralgebrasystem ALDES/SAC-2. Diplomarbeit, Universität Passau, Passau, 1991.

[SS88] Günter Scheja and Uwe Storch. *Lehrbuch der Algebra*, volume 2. Teubner, Stuttgart, 1988.

[Wei92] Volker Weispfenning. Comprehensive Gröbner Bases. *Journal of Symbolic Computation*, 14:1–29, July 1992.

[Wei93a] Volker Weispfenning. A New Approach to Quantifier Elimination for Real Algebra. Technical Report MIP-9305, University of Passau, Passau, July 1993.

[Wei93b] Volker Weispfenning. A New Approach to Quantifier Elimination for Real Algebra. In *Proc. of the Collins Symposium*, Linz, Austria, October 1993. (to appear).