Arbeitsheft 1: Berechenbarkeit

(BuK / WS 2018 / RWTH Aachen)

Gerhard J. Woeginger

Dieses Arbeitsheft enthält einige Übungsaufgaben zur Berechenbarkeit. Jede Aufgabe besteht im Wesentlichen aus einem langen Beweis, der in viele kleine Stücke zerbrochen wurde. Wenn man sich Schritt für Schritt durch diese kleinen Stücke durcharbeitet, entdeckt man die volle Beweiskette.

Die Aufgaben sind für alle Informatik-Studenten im zweiten Studienjahr lösbar, die die Vorlesungen über Berechenbarkeit und Komplexität (BuK) besucht haben. **Für die Aufgaben in diesem Heft werden keine Musterlösungen bereitgestellt. Die Aufgaben werden weder im Tutorium noch in der Globalübung diskutiert.** Zehn Minuten eigenständiges Denken sind nützlicher, als wenn man sich fünf Stunden lang Lösungen durchliest, die von anderen erstellt wurden.

---

1 Der Rice Trick

Sehen wir uns den Beweis des Satzes von Rice noch einmal an: Das zentrale Argument besteht aus einer Reduktion, die Instanzen $\langle M \rangle$ des Epsilon-Halteproblems H_ϵ in eine entsprechende Bildinstanz $\langle M^* \rangle$ mit nur zwei möglichen Verhaltensweisen übersetzt:

- Falls $\langle M \rangle \in H_\epsilon$ gilt, so berechnet M^* die Funktion f.
- Falls $\langle M \rangle \notin H_\epsilon$ gilt, so berechnet M^* die Funktion u.

Wenn wir nun irgendwie entscheiden könnten, welche der beiden Funktionen f und u von M^* berechnet wird, so könnten wir davon ablesen, ob $\langle M \rangle \in H_\epsilon$ gilt, und auf diese Art das unentscheidbare Epsilon-Halteproblem lösen. Wir wollen diesen Rice'schen Übersetzungstrick nun ein wenig abändern und weiter entwickeln.

— * * * — * * * — * * * —

Dazu betrachten wir zwei beliebige Turingmaschinen M_1 und M_2 mit $L(M_1) = L_1$ und $L(M_2) = L_2$. Unser erstes Ziel ist es, eine gegebene Instanz $\langle M \rangle$ des Epsilon-Halteproblems H_ϵ in eine neue Turingmaschine M^+ zu übersetzen. Für ein Eingabewort x führt diese neue Maschine M^+ zwei parallele Berechnungen durch. Die erste Berechnung überprüft ob $x \in L_1$ gilt. Die zweite Berechnung überprüft zuerst ob $\langle M \rangle \in H_\epsilon$ gilt und (falls diese Überprüfung terminiert) danach ob $x \in L_2$ gilt. Sobald/falls eine dieser beiden Berechnungen mit Akzeptanz von x terminiert, terminiert auch die Maschine M^+ und akzeptiert das Eingabewort x.

(a) Erklären Sie, wie man die Turingmaschine M^+ aus den Maschinen M_1, M_2 und M zusammenbauen kann. Wie implementiert man die parallelen Berechnungen? An welchen Stellen wird die universelle Turingmaschine eingesetzt?

(b) Angenommen, es gilt $\langle M \rangle \in H_\epsilon$. Welche Sprache wird in diesem Fall von M^+ akzeptiert?

(c) Angenommen, es gilt $\langle M \rangle \notin H_\epsilon$. Welche Sprache wird in diesem Fall von M^+ akzeptiert?

Wir betrachten (ähnlich zum Satz von Rice) eine gewisse gute Eigenschaft $\mathcal{E}$, die von gewissen rekursiv aufzählbaren Sprachen erfüllt wird. Eine Turingmaschine nennen wir gut, wenn sie eine Sprache mit Eigenschaft $\mathcal{E}$ akzeptiert, und eine Gödelnummer nennen wir gut, wenn sie eine gute Turingmaschine kodiert.

Szenario #1: Die leere Sprache $L_1 = \emptyset$ ist gut, und es existiert eine rekursiv aufzählbare Sprache L_2, die schlecht ist.

Wir wollen nun zwecks Widerspruchs annehmen, dass es eine Turingmaschine $T(\mathcal{E})$ gibt, die alle guten Gödelnummern akzeptiert und alle schlechten Gödelnummern verwirft, und somit die folgende Menge $L_\mathcal{E}$ entscheidet:

$$L_\mathcal{E} \;=\; \{\langle M \rangle \mid L(M) \text{ hat Eigenschaft } \mathcal{E}\} \tag{1}$$

Diese Maschine $T(\mathcal{E})$ kann dann insbesondere Gödelnummern von Turingmaschinen, die die gute leere Sprache L_1 akzeptieren, von Gödelnummern von Turingmaschinen unterscheiden, die die schlechte Sprache L_2 akzeptieren.

(d) Was macht die Maschine $T(\mathcal{E})$, wenn wir sie mit der Gödelnummer $\langle M^+ \rangle$ füttern? Und was können wir aus ihrer Antwort über das Problem "Gilt $\langle M \rangle \in H_\epsilon$" lernen?

(e) Was folgt aus all dem für die Entscheidbarkeit des Epsilon-Halteproblems?

Die Maschine $T(\mathcal{E})$ kann es also gar nicht geben. Wir schlussfolgern, dass unter Szenario #1 die Menge $L_\mathcal{E}$ automatisch **unentscheidbar** ist. Bis zu diesem Punkt haben wir uns ausschliesslich auf Szenario #1 konzentriert, in dem die leere Sprache die Eigenschaft $\mathcal{E}$ besitzt.

(f) Wie kann man unsere Argumentation anpassen, wenn sich das Szenario ändert und wenn die leere Sprache schlecht ist und die Eigenschaft $\mathcal{E}$ **nicht** besitzt?

Falls die Eigenschaft $\mathcal{E}$ nicht trivial ist (das heisst: falls nicht alle rekursiv aufzählbaren Sprachen gut sind und auch nicht alle schlecht sind), so implizieren unsere Resultate die Unentscheidbarkeit der Menge $L_\mathcal{E}$. Diese Unentscheidbarkeit folgt natürlich schon aus dem Satz von Rice.

— * * * — * * * — * * * —

Die obige Konstruktion der Maschine M^+ liefert uns auch ein weiteres nützliches Werkzeug, mit dem man zeigen kann, dass gewisse Mengen von Gödelnummern nicht **rekursiv aufzählbar** sind. (In den folgenden Absätzen sind wir also nicht an **Entscheidbarkeit**, sondern an **rekursiver Aufzählbarkeit** interessiert.) Dazu betrachten wir wieder eine gute Eigenschaft $\mathcal{E}$ von gewissen rekursiv aufzählbaren Sprachen, und von den entsprechenden Turingmaschinen und Gödelnummern.

Szenario #2: Es gibt zwei rekursiv aufzählbare Sprachen $L_1 \subset L_2$, wobei die Untermenge L_1 gut ist, während die Obermenge L_2 schlecht ist.

Es seien M_1 und M_2 zwei Turingmaschinen mit $L(M_1) = L_1$ und $L(M_2) = L_2$, und es sei $\langle M \rangle$ eine beliebige Instanz des Epsilon-Halteproblems. Wir bauen aus den drei Maschinen M_1, M_2, M die unter Punkt (a) konstruierte Turingmaschine M^+ zusammen. Wir nehmen zwecks Widerspruchs an, dass die Menge $L_{\mathcal{E}}$ in (1) rekursiv aufzählbar ist. Dann gibt es also eine Turingmaschine $T(\mathcal{E})$, die die Menge $L_{\mathcal{E}}$ akzeptiert (und die daher alle guten Gödelnummern, aber keine einzige schlechte Gödelnummer akzeptiert). Wir füttern die Maschine $T(\mathcal{E})$ mit der Gödelnummer $\langle M^+ \rangle$.

(g) Zeigen Sie: Falls $\langle M \rangle \notin H_\epsilon$, so akzeptiert $T(\mathcal{E})$ die Gödelnummer $\langle M^+ \rangle$.

(h) Zeigen Sie: Falls $\langle M \rangle \in H_\epsilon$, so akzeptiert $T(\mathcal{E})$ die Gödelnummer $\langle M^+ \rangle$ nicht (und das geschieht entweder durch Verwerfen oder durch Nicht-anhalten).

(i) Folgt nun weiter, dass H_ϵ entscheidbar ist? Oder folgt nun weiter, dass H_ϵ rekursiv aufzählbar ist? Wo ist denn da der Widerspruch?

Der folgende Satz fasst unsere bisherigen Erkenntnisse zusammen. Der Satz wurde 1953 von Henry Rice bewiesen:

Falls für eine Eigenschaft $\mathcal{E}$ Sprachen L_1 und L_2 wie im Szenario #2 existieren, so ist die Menge $L_{\mathcal{E}}$ in (1) **nicht** rekursiv aufzählbar.

Die folgende Liste enthält fünfzehn Mengen von Gödelnummern, von denen nur vier rekursiv aufzählbar sind:

- $\{\langle M \rangle \mid L(M) = \emptyset\}$
- $\{\langle M \rangle \mid L(M) \neq \emptyset\}$
- $\{\langle M \rangle \mid L(M) = \{0,1\}^*\}$
- $\{\langle M \rangle \mid \varepsilon \in L(M)\}$
- $\{\langle M \rangle \mid \varepsilon \notin L(M)\}$
- $\{\langle M \rangle \mid 11101 \in L(M)\}$

- $\{\langle M\rangle \mid L(M)$ enthält alle Worte in $\{0,1\}^*$ mit gerader Länge$\}$
- $\{\langle M\rangle \mid L(M)$ ist regulär$\}$
- $\{\langle M\rangle \mid L(M)$ ist nicht regulär$\}$
- $\{\langle M\rangle \mid L(M)$ ist rekursiv$\}$
- $\{\langle M\rangle \mid L(M)$ ist nicht rekursiv$\}$
- $\{\langle M\rangle \mid |L(M)| = 1\}$
- $\{\langle M\rangle \mid |L(M)| \leq 3\}$
- $\{\langle M\rangle \mid |L(M)| \geq 3\}$
- $\{\langle M\rangle \mid |L(M)| = \infty\}$

Unter den elf nicht rekursiv aufzählbaren Mengen auf der Liste gibt es nur drei, für die das Werkzeug in Szenario #2 nicht geeignet ist. Für die restlichen acht Mengen kann man durch ein geeignetes Szenario #2 zeigen, dass sie nicht rekursiv aufzählbar sind.

(j) Welche acht Mengen sind das?

— ∗ ∗ ∗ — ∗ ∗ ∗ — ∗ ∗ ∗ —

Bisher haben wir nur das Szenario #2 betrachtet, in dem L_1 gut und L_2 schlecht ist. Wie sieht es im symmetrischen Fall aus, wenn stattdessen L_1 schlecht und L_2 gut ist?

Szenario #3: Es gibt zwei rekursiv aufzählbare Sprachen $L_1 \subset L_2$, wobei die Untermenge L_1 schlecht ist, während die Obermenge L_2 gut ist.

Wir können wieder (genau wie zuvor) aus M_1, M_2, M die Maschine M^+ bauen. Wir können wieder (genau wie zuvor) zwecks Widerspruchs annehmen, dass es eine Turingmaschine $T(\mathcal{E})$ gibt, die die Sprache $L_{\mathcal{E}}$ akzeptiert. Und wir können wieder (genau wie zuvor) diese Maschine $T(\mathcal{E})$ mit der Gödelnummer $\langle M^+\rangle$ füttern.

(k) Zeigen Sie: Falls $\langle M\rangle \in H_\epsilon$, so akzeptiert $T(\mathcal{E})$ die Gödelnummer $\langle M^+\rangle$. Falls $\langle M\rangle \notin H_\epsilon$, so akzeptiert $T(\mathcal{E})$ die Gödelnummer $\langle M^+\rangle$ nicht.

(l) Was können wir aus dem Verhalten von $T(\mathcal{E})$ über die Sprache H_ϵ folgern? Wieso erhalten wir in diesem Fall überhaupt keinen Widerspruch?

Zusammengefasst: Es ist uns nicht gelungen, aus Szenario #3 einen Widerspruch zur rekursiven Aufzählbarkeit der in (1) definierten Menge $L_{\mathcal{E}}$ herauszuarbeiten. Das sollte uns aber nicht weiter überraschen, da Szenario #3 tatsächlich mit rekursiv aufzählbaren Mengen $L_{\mathcal{E}}$ kompatibel ist:

(m) Wir betrachten die Eigenschaft $\mathcal{E}$ =“nicht leer”. Zeigen Sie, dass für diese Eigenschaft die entsprechende Menge $L_{\mathcal{E}}$ rekursiv aufzählbar ist. Finden Sie zwei Sprachen L_1 und L_2, die Szenario #3 für diese Eigenschaft $\mathcal{E}$ erfüllen.

— ∗ ∗ ∗ — ∗ ∗ ∗ — ∗ ∗ ∗ —

2 Ein weiterer Rice Trick

Genau wie im letzten Kapitel wollen wir eine gute Eigenschaft $\mathcal{E}$ von gewissen rekursiv aufzählbaren Sprachen betrachten, und parallel dazu gute Turingmaschinen und gute Gödelnummern definieren.

> Szenario #4: Es gibt eine rekursiv aufzählbare Sprache L_4, die gut und unendlich ist. Keine endliche Teilmenge von L_4 ist gut.

Es sei M_4 eine Turingmaschine mit $L(M_4) = L_4$, und es sei $\langle M \rangle$ eine beliebige Instanz des Epsilon-Halteproblems H_ϵ. Ähnlich wie im letzten Kapitel ist unser erstes Ziel, die Instanz $\langle M \rangle$ in eine neue Turingmaschine M^{++} zu übersetzen.

Für ein Eingabewort x führt diese Maschine M^{++} zwei parallele Berechnungen durch. Die erste Berechnung überprüft, ob $x \in L_4$ gilt. Die zweite Berechnung simuliert die ersten $|x|$ Schritte der Turingmaschine M auf dem Eingabewort ϵ. Die Maschine M^{++} terminiert und akzeptiert das Wort x, falls die erste Berechnung mit dem Ergebnis $x \in L_4$ terminiert und falls die Simulation in der zweiten Berechnung **nicht** den Endzustand von M erreicht.

(a) Erklären Sie, wie man die Turingmaschine M^{++} aus den Maschinen M_4 und M zusammenbauen kann. Wie implementiert man die parallelen Berechnungen? An welchen Stellen wird die universelle Turingmaschine eingesetzt?

(b) Angenommen, es gilt $\langle M \rangle \notin H_\epsilon$. Welche Sprache wird in diesem Fall von M^{++} akzeptiert? Ist diese Sprache gut?

(c) Angenommen, es gilt $\langle M \rangle \in H_\epsilon$. Welche Sprache wird in diesem Fall von M^{++} akzeptiert? (Hinweis: Diese Sprache ist eine gewisse Teilmenge von L_4.) Ist die von M^{++} akzeptierte Sprache gut?

Wir nehmen nun zwecks Widerspruchs an, dass die Menge $L_\mathcal{E}$ rekursiv aufzählbar ist und dass $L_\mathcal{E}$ von der Turingmaschine $T(\mathcal{E})$ akzeptiert wird.

(d) Zeigen Sie: Falls $\langle M \rangle \notin H_\epsilon$, so akzeptiert $T(\mathcal{E})$ die Gödelnummer $\langle M^{++} \rangle$.

(e) Zeigen Sie: Falls $\langle M \rangle \in H_\epsilon$, so akzeptiert $T(\mathcal{E})$ die Gödelnummer $\langle M^{++} \rangle$ nicht.

(f) Daraus folgt nun, dass eine gewisse Menge rekursiv aufzählbar ist. Wie lautet diese Menge? Ist diese Menge wirklich rekursiv aufzählbar? Wo ist der Widerspruch?

Wir fassen all unsere Beobachtungen im folgenden Satz (von Henry Rice, 1953) zusammen:

> Falls für eine Eigenschaft $\mathcal{E}$ eine Sprache L_4 wie im Szenario #4 existiert, so ist die Menge $L_\mathcal{E}$ in (1) **nicht** rekursiv aufzählbar.

Nun kehren wir zu unserer Liste auf Seite 3 zurück, in der fünfzehn Mengen von Gödelnummern definiert werden.

(g) Für welche dieser fünfzehn Mengen kann durch ein geeignetes Szenario #4 gezeigt werden, dass sie nicht rekursiv aufzählbar sind?

(h) Und wie sieht es mit jenen Mengen auf unserer Liste aus, die weder mit Szenario #2 noch mit Szenario #4 erledigt werden können? Sind alle überlebenden Mengen wirklich rekursiv aufzählbar?

— * * * — * * * — * * * —

3 Unentscheidbarkeit für context-freie Grammatiken

Wir betrachten eine Instanz des PCPs mit k Dominosteinen, die mit den oberen Worten $x_1, \ldots, x_k \in \{0,1\}^*$ und den unteren Worten $y_1, \ldots, y_k \in \{0,1\}^*$ beschriftet sind. Zu dieser PCP Instanz konstruieren wir zwei context-freien Grammatiken $G_1 = (N_1, \Sigma, P_1, S_1)$ und $G_2 = (N_2, \Sigma, P_2, S_2)$. Das Alphabet Σ besteht aus den Symbolen 0 und 1, und aus k weiteren Symbolen $d_1, \ldots, d_k$, die den k Dominosteinen entsprechen. Die Regeln in P_1 sind

$$S_1 \;\rightarrow\; d_1 S_1 x_1 \mid d_2 S_1 x_2 \mid d_3 S_1 x_3 \mid \;\cdots\; \mid d_k S_1 x_k \mid \epsilon,$$

und die Regeln in P_2 in der zweiten Grammatik sind

$$S_2 \;\rightarrow\; d_1 S_2 x_1 \mid d_2 S_2 x_2 \mid d_3 S_2 x_3 \mid \;\cdots\; \mid d_k S_2 x_k \mid \epsilon.$$

Die von den Grammatiken erzeugten Sprachen $L(G_1)$ und $L(G_2)$ sind natürlich context-frei. Mit den Methoden der FOSAP Vorlesung kann man noch mehr dazu sagen:

(a) Zeigen Sie, dass die Sprachen $L(G_1)$ und $L(G_2)$ sogar **deterministisch** context-frei sind.

(b) Folgern Sie aus den Abschlusseigenschaften der deterministisch context-freien Sprachen: Es gibt einen Algorithmus, der aus G_1 und G_2 zwei neue context-freie Grammatiken G_1' und G_2' berechnet, sodass $L(G_1') = \Sigma^* - L(G_1)$ und $L(G_2') = \Sigma^* - L(G_2)$ gilt.

(c) Folgern Sie aus den Abschlusseigenschaften der context-freien Sprachen: Es gibt einen Algorithmus, der aus G_1, G_2, G_1', G_2' zwei neue context-freie Grammatiken G_3 und G_4 berechnet, sodass $L(G_3) = L(G_1) \cup L(G_2')$ und $L(G_4) = L(G_1') \cup L(G_2')$ gilt.

— * * * — * * * — * * * —

Die Konstruktion dieser sechs Grammatiken G_1, G_2, G_1', G_2', G_3 und G_4 liefert uns Unmengen an Unentscheidbarkeitsresultaten:

(d) Zeigen Sie, dass die ursprüngliche PCP Instanz genau dann eine correspondierende Folge erlaubt, wenn $L(G_1) \cap L(G_2) \neq \emptyset$ gilt. Folgern Sie daraus: Es ist unentscheidbar, ob zwei gegebene context-freie Grammatiken ein gemeinsames Wort erzeugen.

(e) Zeigen Sie, dass $L(G_1) \cap L(G_2) \neq \emptyset$ zu $|L(G_1) \cap L(G_2)| = \infty$ äquivalent ist. Folgern Sie daraus: Es ist unentscheidbar, ob zwei gegebene context-freie Grammatiken unendlich viele gemeinsame Worte erzeugen.

(f) Zeigen Sie, dass $L(G_1) \cap L(G_2) = \emptyset$ zu $L(G_1) \subseteq L(G_2')$ äquivalent ist. Folgern Sie daraus: Es ist unentscheidbar, ob die von einer gegebenen context-freien Grammatik erzeugte Sprache eine Teilmenge der von einer zweiten gegebenen context-freien Grammatik erzeugten Sprache ist.

(g) Zeigen Sie, dass $L(G_1) \cap L(G_2) = \emptyset$ zu $L(G_3) = L(G_2')$ äquivalent ist. Folgern Sie daraus: Es ist unentscheidbar, ob zwei gegebene context-freie Grammatiken die selbe Sprache erzeugen.

(h) Zeigen Sie, dass $L(G_1) \cap L(G_2) = \emptyset$ zu $L(G_4) = \Sigma^*$ äquivalent ist. Folgern Sie daraus: Es ist unentscheidbar, ob eine gegebene context-freie Grammatik ganz Σ^* erzeugt.

— * * * — * * * — * * * —

Zum Abschluss wollen wir uns noch eine allerletzte context-freie Grammatik definieren. Als Zutaten verwenden wir die kontext-freie Sprache $L_4 = L(G_4)$, die nicht-reguläre kontext-freie Sprache $L_0 = \{0^n 1^n \mid n \geq 1\}$, und das Trennsymbol $\$ \notin \Sigma$.

(i) Folgern Sie aus den Abschlusseigenschaften der context-freien Sprachen: Es gibt einen Algorithmus, der aus der gegebenen PCP Instanz eine context-freie Grammatik G_5 mit $L(G_5) = \Sigma^*\$L_0 \cup L_4\$\Sigma^*$ berechnet.

(j) Zeigen Sie: Falls $L(G_4) = \Sigma^*$, so ist $L(G_5)$ regulär. Falls $L(G_4) \neq \Sigma^*$, so ist $L(G_5)$ nicht regulär.

(k) Folgern Sie: Es ist unentscheidbar, ob eine gegebene context-freie Grammatik eine reguläre Sprache erzeugt.

— * * * — * * * — * * * —

4 Das zehnte Hilbert'sche Problem

In diesem Abschnitt haben alle betrachteten Polynome ganzzahlige Koeffizienten. In der Vorlesung haben wir gesehen, dass das zehnte Hilbert'sche Problem unentscheidbar ist:

Problem Dioph($\mathbb{Z}$) Eingabe: Ein multivariates Polynom $p(x_1, \ldots, x_n)$. Frage: Existieren Werte $x_1, \ldots, x_n \in \mathbb{Z}$, sodass $p(x_1, \ldots, x_n) = 0$ gilt?

Eine nahe verwandte Fragestellung sucht die Lösung nicht über den ganzen Zahlen, sondern nur über den natürlichen Zahlen:

Problem Dioph($\mathbb{N}$) Eingabe: Ein multivariates Polynom $p(x_1, \ldots, x_n)$. Frage: Existieren Werte $x_1, \ldots, x_n \in \mathbb{N}$, sodass $p(x_1, \ldots, x_n) = 0$ gilt?

Analog dazu definieren wir das Problem Dioph($\mathbb{N}_g$), in dem die Variablen nur **gerade** natürliche Werte annehmen dürfen, und das Problem Dioph($\mathbb{N}_u$), in dem die Variablen nur **ungerade** natürliche Werte annehmen dürfen.

(a) Zeigen Sie, dass Dioph($\mathbb{Z}$) $\leq$ Dioph($\mathbb{N}$) gilt. (Hinweis: Jede ganze Zahl kann als Differenz von zwei natürlichen Zahlen geschrieben werden.) Folgern Sie, dass Dioph($\mathbb{N}$) unentscheidbar ist.

(b) Zeigen Sie, dass Dioph($\mathbb{N}_g$) unentscheidbar ist.

(c) Zeigen Sie, dass Dioph($\mathbb{N}_u$) unentscheidbar ist.

Ein berühmter Satz aus der Zahlentheorie (der "Vier-Quadrate-Satz" von Lagrange) besagt, dass jede natürliche Zahl als Summe von vier Quadraten geschrieben werden kann. Zum Beispiel gilt $2019 = 43^2 + 13^2 + 1^2 + 0^2$ und $3719 = 53^2 + 30^2 + 3^2 + 1^2$.

(d) Zeigen Sie mit Hilfe des Vier-Quadrate-Satzes, dass Dioph($\mathbb{N}$) $\leq$ Dioph($\mathbb{Z}$) gilt.

— * * * — * * * — * * * —

Ein auf den ersten Blick noch schwieriger aussehendes Problem besteht darin, ein ganzes Diophantisches System von Polynomgleichungen über den ganzen Zahlen zu lösen:

$$\begin{aligned} q_1(x_1, \ldots, x_n) &= 0 \quad \text{und} \\ q_2(x_1, \ldots, x_n) &= 0 \quad \text{und} \\ q_3(x_1, \ldots, x_n) &= 0 \quad \text{und} \\ \vdots \qquad &\vdots \\ q_k(x_1, \ldots, x_n) &= 0 \end{aligned}$$

(e) Zeigen Sie, dass die Lösung eines Diophantischen Gleichungssystems über $\mathbb{Z}$ auf Dioph($\mathbb{Z}$) reduziert werden kann.

(f) Zeigen Sie, dass Dioph($\mathbb{Z}$) auf die Lösung eines Diophantischen Gleichungssystems über $\mathbb{Z}$ reduziert werden kann, in dem alle Gleichungen höchstens Grad 2 haben.

— * * * — * * * — * * * —

Zum Schluss wollen wir noch folgendes Problem mit zwei Polynomen betrachten: Für zwei gegebene multivariate Polynome $p_1(x_1, \ldots, x_n)$ und $p_2(x_1, \ldots, x_n)$ mit lauter **positiven** ganzzahligen Koeffizienten soll entschieden werden, ob alle $x_1, \ldots, x_n \in \mathbb{Z}$ die strikte Ungleichung $p_1(x_1, \ldots, x_n) < p_2(x_1, \ldots, x_n)$ erfüllen.

(g) Zeigen Sie, dass die Unentscheidbarkeit von Dioph($\mathbb{Z}$) die Unentscheidbarkeit des Problems mit den zwei Polynomen impliziert.