Kapitel 9

Berechenbarkeit

In diesem Kapitel geben wir Antworten auf die folgenden Fragen:

1. Was ist eine berechenbare Funktion?

2. Ist die Funktion $\lambda\, A \in \mathit{Assn}\,.\,$ if $\models A$ then 0 else 1 berechenbar?

Diese Fragen wurden erstmals um 1930 von Logikern untersucht [Church, Gödel, Turing].

Um Berechenbarkeit zu definieren, benötigen wir ein mathematisch formuliertes Berechnungsmodell. Dieses muss über eine universelle Datenstruktur verfügen, mit der sich syntaktische Objekte (beispielsweise Formeln) darstellen lassen.

Eine mögliche universelle Datenstruktur sind Zeichenreihen. Das von Turing entwickelte Berechnungsmodell (sogenannte Turingmaschinen) basiert auf Zeichenreihen. Auch heutige Computer rechnen mit Zeichenreihen (mit den Zeichen 0 und 1).

Auch natürliche Zahlen eignen sich als universelle Datenstruktur. Sie sind die Grundlage von Gödels Berechnungsmodell (sogenannte μ-rekursive Funktionen).

Wir werden Berechenbarkeit mithilfe von IMP definieren. IMP eignet sich als universelles Berechnungsmodell, da es mit beliebig großen Zahlen rechnen kann.

Lesematerial [Winskel, Anhang A]

9.1 Gödelisierung

Kurt Gödel hat um 1930 eine einfache Technik entdeckt, mit der sich syntaktische Objekte als natürliche Zahlen darstellen lassen. Seine Darstellungstechnik, die heute als *Gödelisierung* bezeichnet wird, beruht auf der Eindeutigkeit der Primzahlzerlegung.

Wir zeigen zunächst, wie man mithilfe von Gödelisierung Paare von Zahlen als Zahlen darstellen kann. Dazu geben wir vier Funktionen

$$\begin{aligned} \mathit{pair} &\in \mathbb{Z} \times \mathbb{Z} \to \mathbb{N}^+ \\ \mathit{first} &\in \mathbb{Z} \to \mathbb{Z} \\ \mathit{second} &\in \mathbb{Z} \to \mathbb{Z} \\ \mathit{ispair} &\in \mathbb{Z} \to \mathbb{B} \end{aligned}$$

an, die die folgenden Eigenschaften erfüllen:

(1) $\forall n_1, n_2 \in \mathbb{Z}\colon\ \mathit{first}(\mathit{pair}(n_1, n_2)) = n_1$
(2) $\forall n_1, n_2 \in \mathbb{Z}\colon\ \mathit{second}(\mathit{pair}(n_1, n_2)) = n_2$
(3) $\forall n \in \mathbb{Z}\colon\ \mathit{ispair}(n) = 1 \iff (\exists n_1, n_2 \in \mathbb{Z}\colon\ n = \mathit{pair}(n_1, n_2))$

Wir definieren *pair* wie folgt:

$$\mathit{pair}(n_1, n_2) = 2^{sg(n_1)} \cdot 3^{|n_1|} \cdot 5^{sg(n_2)} \cdot 7^{|n_2|}$$

wobei $sg(n) = \text{if } n \geq 0 \text{ then } 1 \text{ else } 0$. Wegen der Eindeutigkeit der Primzahlzerlegung können wir jetzt die Funktionen *first*, *second* und *ispair* wie gewünscht definieren. Man überzeugt sich leicht, dass *pair* und diese Funktionen mit IMP berechnet werden können.

Mit derselben Technik können wir auch Tupel mit mehr als zwei Komponenten als Zahlen darstellen. Auch Listen lassen sich als Zahlen darstellen. Dazu ordnen wir der leeren Liste die Zahl 0 zu und erinnern uns daran, dass nichtleere Listen Paare sind.

In Kapitel 3 haben wir syntaktische Objekte als Paare definiert, deren erste Komponente (die sogenannte Variantennummer) festlegt, wie die zweite Komponente zu interpretieren ist. Wenn wir bei der Definition von syntaktischen Objekten nur darstellbare Grundmengen verwenden (z. B. $\mathit{Var} = \mathit{Loc} = \mathbb{N}$), dann sind syntaktische Objekte als Zahlen darstellbar.

Im Folgenden nehmen wir $\mathit{Var} = \mathit{Loc} = \mathbb{N}$ an.

Die einem Objekt x durch Gödelisierung zugeordnete Zahl bezeichnen wir mit $\#x$ und nennen sie die ***Gödelnummer*** von x. Wir verwenden die Bezeichnungen

$$\#Com \stackrel{\text{def}}{=} \{\#c \mid c \in Com\}$$
$$\#Assn \stackrel{\text{def}}{=} \{\#A \mid A \in Assn\}$$

und gehen davon aus, dass die Funktionen $\lambda\, c \in Com\,.\,\#c$ und $\lambda\, A \in Assn\,.\,\#A$ injektiv sind.

9.2 Berechenbare Funktionen

Universelle Berechnungsmodelle haben die Eigenschaft, dass bestimmte Programme für bestimmte Eingaben divergieren (d. h. nicht terminieren). Es ist wichtig, den Begriff der berechenbaren Funktion so zu definieren, dass die Möglichkeit der Divergenz sichtbar bleibt.

Wir wählen ein Objekt $\bot \notin \mathbb{Z}$ und definieren

$$\mathbb{Z}_\bot \stackrel{\text{def}}{=} \mathbb{Z} \cup \{\bot\}$$

Wir werden die Funktionen aus $\mathbb{Z} \to \mathbb{Z}_\bot$ als berechenbar auszeichnen, die sich mit einem IMP-Kommando berechnen lassen. Dabei bedeutet das Ergebnis $\bot$, dass die Ausführung des Kommandos nicht terminiert.

Wir wählen eine Lokation $X_0 \in Loc = \mathbb{N}$ und definieren:

$$\sigma_0 = \lambda\, X \in Loc\,.\,0$$

$$\mathcal{F} \in Com \to \mathbb{Z} \to \mathbb{Z}_\bot$$
$$\mathcal{F}(c)x \;=\; \text{let } s = \mathcal{C}(c)(\sigma_0[x/X_0]) \text{ in if } s = \bot \text{ then } \bot \text{ else } s(X_0)$$

Eine Funktion $f \in \mathbb{Z} \to \mathbb{Z}_\bot$ heißt ***berechenbar*** genau dann, wenn es ein Kommando $c \in Com$ gibt mit $f = \mathcal{F}(c)$. Wenn $f = \mathcal{F}(c)$ gilt, sagen wir, dass das Kommando c die Funktion f ***berechnet***.

Gödel, Church, Turing und andere Mathematiker haben ab 1930 verschiedene mathematische Berechnungsmodelle entwickelt. Sie konnten zeigen, dass diese Modelle alle denselben Berechenbarkeitsbegriff liefern, den man heute als ***Turing-Berechenbarkeit*** bezeichnet. Auch unser auf IMP basierendes Berechnungsmodell liefert Turing-Berechenbarkeit. Die Annahme, dass Turing-Berechenbarkeit *intuitive Berechenbarkeit* adäquat formalisiert, bezeichnet man als ***Churchschen These***.

Welche Funktionen $\mathbb{Z} \to \mathbb{Z}_\perp$ können wir mit einem Computer berechnen? Streng genommen gar keine, da ein Computer mit endlichem Speicher nur endlich viele Zahlen darstellen kann.

Welche Funktionen $\mathbb{Z} \to \mathbb{Z}_\perp$ können wir mit einem Computer berechnen, der einen unendlich Speicher hat? Genau die Turing-berechenbaren! Um die Richtigkeit dieser Behauptung beweisen zu können, benötigt man natürlich ein mathematisches Modell eines Computers mit unendlichem Speicher. Die Formulierung eines solchen Modells ist nicht schwer.

Im Folgenden werden wir öfter annehmen, dass intuitiv berechenbare Funktionen in IMP berechenbar sind. Die Stellen, an denen wir das tun, werden wir mit dem Vermerk „*Churchsche These*“ markieren. Um die Idee der intuitiven Berechenbarkeit hinreichend konkret zu machen, stellen wir uns eine idealisierte Teilsprache ***IML*** der Programmiersprache Standard ML vor, die mit beliebig großen ganzen Zahlen rechnen kann. Eine Funktion $\mathbb{Z} \to \mathbb{Z}_\perp$ nennen wir intuitiv berechenbar, wenn wir sie mit einer IML-Prozedur `int->int` berechnen können.

Satz 9.2.1 (Universelle Funktion) *Es gibt eine berechenbare Funktion $f \in \mathbb{Z} \to \mathbb{Z}_\perp$ mit:*

$$\forall c \in \mathit{Com}\ \forall x \in \mathbb{Z}\colon\ f(\#(\#c, x)) = \mathcal{F}(c)x$$

Beweis Es ist nicht schwer, in IML eine Prozedur zu schreiben, die zu einer Gödelnummer $\#(\#c, x)$ das Kommando c und die Zahl x liefert. Außerdem können wir eine Prozedur schreiben, die $\mathcal{F}$ berechnet. Mit einem kräftigen Schuss Churchsche These folgt, dass es eine berechenbare Funktion wie behauptet gibt. □

Korollar 9.2.2 *Es gibt ein $k \in \mathbb{N}$, sodass jede berechenbare Funktion mit einem Kommando berechnet werden kann, dass höchstens die Lokationen $0, \ldots, k$ verwendet.*

Beweis Sei u eine Kommando, das eine universelle Funktion wie im obigen Satz berechnet. Sei X_1 eine von X_0 verschiedene Lokation. Sei g ein Kommando mit

$$\forall \sigma \in \Sigma\colon\ \mathcal{C}(g)\sigma = \sigma[\#(\sigma X_1, \sigma X_0)/X_0]$$

Sei f eine berechenbare Funktion und c ein Kommando, das f berechnet. Dann berechnet das Kommando

$$X_1 := \#c;\ g;\ X_1 := 0;\ u$$

die Funktion f. Da die in diesem Kommando vorkommenden Lokationen nicht von f oder c abhängen, und jedes Kommando nur endlich viele Lokationen verwendet, folgt die Behauptung. □

Am Beweis dieses Korollars sieht man, dass schematisches Programmieren in IMP eine mühselige Angelegenheit ist. Kompakter und lesbarer können wir das für f konstruierte Kommando durch das ***Pseudokommando***

$$X_0 := \#(\#c, X_0);\ u$$

beschreiben. Von solchen Pseudokommandos werden wir im Folgenden gelegentlich Gebrauch machen. Für die Übersetzbarkeit von Pseudokommandos in Kommandos von IMP berufen wir uns auf die Churchsche These.

9.3 Abzählbare Mengen

Eine Menge M heißt ***abzählbar*** genau dann, wenn es eine injektive Funktion $M \to \mathbb{N}$ gibt.

Offensichtlich ist jede endliche Menge abzählbar. Auch $\mathbb{N}$ und $\mathbb{Z}$ sind abzählbar.

Informell gesprochen ist eine Menge abzählbar, wenn sie höchstens soviele Elemente enthält wie die Menge der natürlichen Zahlen.

Eine Menge heißt ***überabzählbar***, wenn sie nicht abzählbar ist. Wir sagen, dass eine Menge ***abzählbar viele*** [***überabzählbar viele***] ***Elemente*** hat, wenn sie abzählbar [überabzählbar] ist.

Proposition 9.3.1 *Eine nichtleere Menge M ist genau dann abzählbar, wenn es eine surjektive Funktion $\mathbb{N} \to M$ gibt.*

Proposition 9.3.2 *Der Schnitt, die Vereinigung und die Differenz von zwei abzählbaren Mengen ist abzählbar.*

Beweis Wir zeigen nur, dass die Vereinigung von zwei abzählbaren Mengen abzählbar ist. Die anderen Behauptungen folgen mit ähnlichen Argumenten.

Seien $f \in M \to \mathbb{N}$ und $f' \in M' \to \mathbb{N}$ injektive Funktionen. Dann ist

$$h \in M \cup M' \to \mathbb{N}$$
$$h(x) = \text{if } x \in M \text{ then } 2 \cdot f(x) \text{ else } 2 \cdot f'(x) + 1$$

eine injektive Funktion. □

Mithilfe von Gödelisierung bekommen wir injektive Funktionen $Com \to \mathbb{N}$ und $Assn \to \mathbb{N}$. Folglich sind *Com* und *Assn* abzählbar. Da *Com* abzählbar ist, ist auch die Menge der berechenbaren Funktionen abzählbar.

Proposition 9.3.3 *Die Menge der berechenbaren Funktionen ist abzählbar.*

Proposition 9.3.4 *Die Menge $\mathbb{Z} \to \mathbb{Z}_\perp$ ist überabzählbar.*

Beweis Durch Widerspruch. Sei $\alpha \in \mathbb{N} \to (\mathbb{Z} \to \mathbb{Z}_\perp)$ surjektiv. Wir definieren

$$f \in \mathbb{Z} \to \mathbb{Z}_\perp$$
$$f(n) = \text{if } (\alpha n)n = 0 \text{ then } 1 \text{ else } 0$$

Offensichtlich gilt

$$\forall n \in \mathbb{N}\colon\ (\alpha n)n \neq f(n)$$

Das ist ein Widerspruch zu der Annahme, dass α surjektiv ist. □

Die gerade verwendete Beweistechnik wird als ***Cantorsches Diagonalargument*** bezeichnet.

Wir wissen jetzt, dass es überabzählbar viele Funktionen $\mathbb{Z} \to \mathbb{Z}_\perp$ gibt, die nicht berechenbar sind.

9.4 Entscheidbare und prüfbare Mengen

Wir betrachten jetzt Teilmengen $M \subseteq \mathbb{Z}$ und stellen die folgenden Fragen:

1. Ist die Funktion $\lambda\, x \in \mathbb{Z}\,.$ if $x \in M$ then 1 else 0 berechenbar?
2. Ist die Funktion $\lambda\, x \in \mathbb{Z}\,.$ if $x \in M$ then 1 else $\perp$ berechenbar?
3. Ist die Funktion $\lambda\, x \in \mathbb{Z}\,.$ if $x \in M$ then $\perp$ else 0 berechenbar?

Wenn wir die erste Frage mit Ja beantworten können, nennen wir die Menge M ***entscheidbar***. Wenn wir die zweite Frage mit Ja beantworten können, nennen wir M ***prüfbar*** oder ***semientscheidbar***.[1] Statt *nicht entscheidbar* sagen wir auch ***unentscheidbar***. Die dritte Frage können wir genau dann mit Ja beantworten, wenn das ***Komplement***

$$\overline{M} \stackrel{\text{def}}{=} \mathbb{Z} - M$$

von M bezüglich $\mathbb{Z}$ prüfbar ist.

[1] In der Literatur findet man oft den Begriff *recursively enumerable (r. e.)*. Dieser bezeichnet eine zu Prüfbarkeit äquivalente Eigenschaft.

Für eine entscheidbare Menge M können wir eine Prozedur `int->bool` angeben, die für alle Eingaben terminiert und genau für die Elemente von M das Ergebnis 1 liefert. Für eine prüfbare Menge M können wir eine Prozedur `int->unit` angeben, die genau für die Elemente von M terminiert.

Entscheidbare Mengen sind immer prüfbar. Dasselbe gilt für die Komplemente von entscheidbaren Mengen. Wir werden eine prüfbare Menge angeben, deren Komplement nicht prüfbar ist. Das bedeutet, dass Terminierung und Divergenz keine symmetrischen Eigenschaften sind. Schließlich werden wir eine abzählbare Menge M angeben, sodass weder M noch $\overline{M}$ prüfbar ist.

Proposition 9.4.1 *Die Mengen #Com und #Assn sind entscheidbar.*

Beweis Folgt mit der Churchschen These. □

Proposition 9.4.2 *Die Menge der entscheidbaren [prüfbaren] Teilmenge von $\mathbb{Z}$ ist abzählbar, und die Menge der nicht entscheidbaren [nicht prüfbaren] Teilmenge von $\mathbb{Z}$ ist überabzählbar.*

Beweis Die erste Behauptung folgt aus der Tatsache, dass es nur abzählbar viele berechenbare Funktionen gibt. Die zweite Behauptung folgt aus der Tatsache, dass $\mathcal{P}(\mathbb{Z})$ nicht abzählbar ist. □

Proposition 9.4.3 *Der Schnitt, die Vereinigung und die Differenz von zwei entscheidbaren Mengen ist entscheidbar.*

Sei $M \subseteq \mathbb{Z}$. Eine ***Prüffunktion*** für M ist eine Funktion $f \in \mathbb{Z} \to \mathbb{Z}_\perp$ mit $M = \{ x \in \mathbb{Z} \mid f\,x \neq \perp \}$. Offensichtlich ist M genau dann prüfbar, wenn für M eine berechenbare Prüffunktion existiert. Ein ***Prüfer*** für M ist ein Kommando, das eine Prüffunktion für M berechnet.

Ein ***steuerbarer Prüfer*** für eine Menge $M \subseteq \mathbb{Z}$ ist ein Kommando $c \in \mathit{Com}$ wie folgt:

(1) $\forall n \in \mathbb{Z}\colon\ \mathcal{F}(c)n \neq \perp$

(2) $\forall n \in \mathbb{Z}\colon\ n \in M \iff \exists k \in \mathbb{N}\colon\ \mathcal{F}(c)(\#(n,k)) = 1$

(3) $\forall n \in \mathbb{Z}\ \forall k \in \mathbb{N}\colon\ \mathcal{F}(c)(\#(n,k)) = 1 \Rightarrow \forall m \geq k\colon\ \mathcal{F}(c)(\#(n,m)) = 1$

Proposition 9.4.4 *Sei $c \in \mathit{Com}$ ein Prüfer für M. Dann kann man aus c einen steuerbaren Prüfer für M konstruieren.*

Beweis Den steuerbaren Prüfer erhält man dadurch, dass man einen Zähler einbaut, der die Anzahl der Schleifendurchläufe zählt. Zusätzlich werden die Schleifenbedingungen so verstärkt, dass nach Überschreiten der durch das zweite Ar-

gument vorgegebenen Maximalanzahl keine weiteren Schleifendurchläufe mehr möglich sind.

Seien Z und S zwei Lokationen, die nicht in c vorkommen. Sei c' das Kommando, das man aus c wie folgt erhält:

1. Ersetze jede Schleifenbedingung b durch $b \wedge Z \leq S$.
2. Ersetze jeden Schleifenrumpf c'' durch $Z := Z + 1 \ ; \ c''$.

Der steuerbare Prüfer sieht jetzt wie folgt aus (als Pseudokommando):

$$\begin{array}{l} \textsf{if } \mathit{ispair}(X_0) \textsf{ then } S := \mathit{second}(X_0);\ X_0 := \mathit{first}(X_0) \textsf{ else } S := -1; \\ Z := 0; \\ c'; \\ \textsf{if } Z \leq S \textsf{ then } X_0 := 1 \textsf{ else } X_0 := 0 \end{array}$$

□

Proposition 9.4.5 *Der Schnitt und die Vereinigung von zwei prüfbaren Mengen ist prüfbar.*

Beweis Seien M_1 und M_2 prüfbar. Aus steuerbaren Prüfern für M_1 und M_2 kann man Prüfer für $M_1 \cup M_2$ und $M_1 \cap M_2$ konstruieren. Dieser führt beide steuerbaren Prüfer bis zu einer schrittweise erhöhten Maximalzahl von Schleifendurchläufen aus. □

Proposition 9.4.6 *Eine Menge $M \subseteq \mathbb{Z}$ ist genau dann entscheidbar, wenn M und $\overline{M}$ prüfbar sind.*

Beweis Die eine Richtung ist trivial. Für die andere Richtung nehmen wir an, dass $M \subseteq \mathbb{Z}$ und $\overline{M}$ prüfbar sind. Also gibt es steuerbare Prüfer für M und $\overline{M}$. Wir konstruieren einen Entscheider für M, indem wir beide steuerbaren Prüfer bis zu einer schrittweise erhöhten Maximalzahl von Schleifendurchläufen ausführen. Sobald einer der Prüfer ein Ergebnis liefert, sind wir fertig. □

Proposition 9.4.7 *Sei $M \subseteq E \subseteq \mathbb{Z}$ und sei E entscheidbar. Dann:*

$$\overline{M} \textit{ prüfbar} \iff \overline{M} \cap E \textit{ prüfbar}$$

Beweis Gilt da $\overline{M} = \overline{E} \cup (\overline{M} \cap E)$. □

9.5 Unentscheidbarkeit des Halteproblems

Wir definieren zwei Mengen:

$$S \stackrel{\text{def}}{=} \{\#c \mid c \in \mathit{Com} \wedge \mathcal{F}(c)(\#c) = \bot\}$$
$$S_0 \stackrel{\text{def}}{=} \{\#c \mid c \in \mathit{Com} \wedge \mathcal{F}(c)(0) = \bot\}$$

Wir werden zeigen, dass keine dieser Mengen prüfbar ist. Diese Tatsache bezeichnet man als die *Unentscheidbarkeit des Halteproblems.*

Proposition 9.5.1 *S ist nicht prüfbar.*

Beweis Durch Widerspruch. Sei T ein Prüfer für S. Dann haben wir den folgenden Widerspruch:

$$\begin{array}{lll} \mathcal{F}(T)(\#T) = \bot \iff \#T \in S & & \text{Definition von } S \\ \iff \mathcal{F}(T)(\#T) \neq \bot & & T \text{ Prüfer für } S \end{array}$$ □

Machen Sie sich klar, dass die Nichtprüfbarkeit der Menge S eine völlig offensichtliche Angelegenheit ist. Der Beweis benötigt keine vorher bewiesenen Resultate, sondern nur die Definitionen der Menge S und der Eigenschaft prüfbar. Aus dem Beweis sieht man sofort, dass es keine Rolle spielt, welche Kommandos IMP hat. Also kann auch das Hinzufügen beliebig mächtiger Kommandos an der Nichtprüfbarkeit von S nichts ändern. Es spielt auch keine Rolle, dass IMP mit Zahlen rechnet. Jeder andere Datenstruktur und jede Funktion von Programmen nach Eingaben (statt Gödelisierung) führt zum selben Ergebnis.

Satz 9.5.2 (Halteproblem) *S_0 ist nicht prüfbar.*

Beweis Durch Widerspruch. Sei S_0 prüfbar. Wir zeigen, dass dann S prüfbar ist. Zuerst stellen wir fest, dass für alle $x \in \mathbb{Z}$ gilt:

$$x \in S \iff x \in \#\mathit{Com} \wedge \#(X_0 := x;\ \#^{-1}x) \in S_0$$

Beachten Sie, dass $\# \in \mathit{Com} \to \#\mathit{Com}$ eine Bijektion ist und $\#^{-1}x$ das durch die Gödelnummer x dargestellte Kommando bezeichnet. Sei p eine berechenbare Prüffunktion für S_0. Aus der Äquivalenz folgt, dass $f \in \mathbb{Z} \to \mathbb{Z}_\bot$ mit

$$f\,x = \text{if } x \in \#\mathit{Com} \text{ then } p(\#(X_0 := x;\ \#^{-1}x)) \text{ else } \bot$$

eine Prüffunktion für S ist. Da f berechenbar ist, ist S prüfbar. Widerspruch. □

Proposition 9.5.3 *Die Mengen $\overline{S}$ und $\overline{S_0}$ sind prüfbar.*

Beweis Wegen der Propositionen 9.4.7 und 9.4.1 genügt es, zu zeigen, dass die Mengen

$$H \stackrel{\text{def}}{=} \{\#c \mid c \in \mathit{Com} \wedge \mathcal{F}(c)(\#c) \neq \bot\}$$
$$H_0 \stackrel{\text{def}}{=} \{\#c \mid c \in \mathit{Com} \wedge \mathcal{F}(c)(0) \neq \bot\}$$

prüfbar sind (wähle $E = \#\mathit{Com}$). Die Prüfbarkeit von H und H_0 folgt aus der Existenz einer berechenbaren universellen Funktion (Satz 9.2.1) und einer Prise Churchscher These. □

9.6 Gödelscher Unvollständigkeitssatz

Zu Anfang des zwanzigsten Jahrhunderts waren Hilbert und andere Mathematiker der Ansicht, dass die Gültigkeit von arithmetischen Formeln (so wie in ASSN) entscheidbar ist (im intuitiven Sinne, formal war Entscheidbarkeit damals noch nicht definiert). Kurt Gödel zeigte um 1930, dass dies nicht der Fall ist. Dieses fundamentale Ergebnis ist als *Gödelscher Unvollständigkeitssatz* bekannt.

Wir definieren: $G \stackrel{\text{def}}{=} \{\#A \mid A \in \mathit{Assn} \wedge \models A\}$.

Satz 9.6.1 (Gödels Unvollständigkeitssatz) *G ist nicht prüfbar.*

Beweis Durch Widerspruch. Sei G prüfbar. Wir zeigen, dass dann S_0 prüfbar ist. Sei w eine SVB-Funktion für IMP. Dann gilt

$$\begin{aligned} x \in S_0 &\iff x \in \#\mathit{Com} \wedge \models w\,(Y_1 := 0;\ \ldots\ ;\ Y_n = 0;\ \#^{-1}x)\,\mathsf{false} \\ &\iff x \in \#\mathit{Com} \wedge \#(w\,(Y_1 := 0;\ \ldots\ ;\ Y_n = 0;\ \#^{-1}x)\,\mathsf{false}) \in G \end{aligned}$$

wobei es sich bei $Y_1, \ldots, Y_n$ gerade um die im Kommando $\#^{-1}x$ vorkommenden Lokationen handeln soll. In Kapitel 8 haben wir gezeigt, dass man für jedes Kommando und jede Nachbedingung eine schwächste Vorbedingung bestimmen kann (Satz 8.7.8). Mit einem kräftigen Schuss Churchscher These und der Annahme, dass G prüfbar ist, folgt, dass S_0 prüfbar ist. □

Korollar 9.6.2 *$\overline{G}$ ist nicht prüfbar.*

Beweis Durch Widerspruch. Sei $\overline{G}$ prüfbar. Wir zeigen, dass dann G prüfbar ist. Für alle $A \in \mathit{Assn}$ gilt

$$\models A \iff \not\models \neg(\forall A)$$

wobei $\forall A$ eine Formel $\forall X_1 \ldots \forall X_n A$ mit $FV(A) = \{X_1, \ldots, X_n\}$ bezeichnen soll. Also gilt für alle $x \in \mathbb{Z}$:

$$\begin{aligned} x \in G &\iff x \in \#Assn \wedge \models \#^{-1}x \\ &\iff x \in \#Assn \wedge \not\models \neg(\forall(\#^{-1}x)) \\ &\iff x \in \#Assn \wedge \#(\neg(\forall(\#^{-1}x))) \in \overline{G} \end{aligned}$$

Beachten Sie, dass $\# \in Assn \to \#Assn$ eine Bijektion ist und $\#^{-1}x$ die durch die Gödelnummer x dargestellte Formel bezeichnet. Mit einer Prise Churchscher These und der Annahme, dass $\overline{G}$ prüfbar ist, folgt, dass G prüfbar ist. Widerspruch. □

9.7 Sätze von Presburger und Matijasevic

Wir nutzen die Gelegenheit und geben noch zwei wichtige Sätze über die Gültigkeit von arithmetischen Formeln an. Die Beweise dieser Sätze geben wir nicht an, da sie zu schwer und langwierig sind (vor allem der Beweis von Matijasevic).

Satz 9.7.1 (Presburger 1929) *Die Gültigkeit von arithmetischen Formeln ohne Multiplikation ist entscheidbar. Genauer: Die folgende Menge ist entscheidbar:*

$$\{\#A \mid A \in Assn \wedge A \text{ enthält keine Multiplikation} \wedge \models A\}$$

Satz 9.7.2 (Matijasevic 1970) *Die folgende Menge ist unentscheidbar:*

$$\{\#a \mid a \in Ter \wedge \exists \sigma \in \Sigma : \mathcal{T}(a)\sigma = 0\}$$

Auf dem internationalen Mathematikerkongress in Paris im Jahre 1900 hielt David Hilbert eine viel beachtete Rede, in der er 23 wichtige offene mathematische Probleme formulierte. Der Satz von Matijasevic löst Hilberts zehntes Problem.