Übungsblatt 14

Aufgabe 1. Die Softwarefirma HALTING & CO. KG bietet folgende Produkte zur Programmverifikation an.

(a) Produkt A überprüft, ob ein gegebenes Programm auf allen Eingaben terminiert.

(b) Produkt B überprüft, ob ein gegebenes Programm auf einer gegebenen Eingabe höchstens 1,000 Rechenschritte durchführt.

(c) Produkt C überprüft, ob ein gegebenes Programm auf einer gegebenen Eingabe höchstens 1 GB Speicher benötigt.

(d) Produkt D überprüft, ob ein gegebenes Programm niemals die Ausgabe 123 produziert.

Welche Produkte kann es tatsächlich geben?

Lösung.

(a) Falsch. Das Halteproblem (Folie 272) kann auf das Problem a) reduziert werden. Dazu wird die gegebene Turing-Maschine so modifiziert, dass sie zu Beginn die eigentliche Eingabe gegen eine spezifische Eingabe austauscht. Die modifizierte Turing-Maschine terminiert nun bei beliebiger Eingabe genau dann, wenn die originale auf der spezifischen Eingabe terminiert. Da das Halteproblem unentscheidbar ist, muss auch das Problem a) unentscheidbar sein (Folie 422).

(b) Wahr. Das Programm wird unter der gegebenen Eingabe simuliert und die Rechenschritte mitgezählt. Falls das simulierte Programm bis zum 1000 Rechenschritt terminiert, meldet der Simulator ≪terminiert≫. Andernfalls wird die Simulation abgebrochen und der Simulator meldet ≪terminiert nicht bei bis zu 1000 Rechenschritten".

(c) Wahr. Das Programm wird unter der gegebenen Eingabe simuliert und alle bereits erreichten Konfigurationen gespeichert. Sollte eine Konfiguration mehr als 1 GB Speicher benötigen, bricht der Simulator mit ≪braucht mehr Speicher» ab. Wird eine bereits bekannte Konfiguration erneut besucht, bricht der Simulator mit «terminiert nicht» ab. Sollte das simulierte Programm terminieren, meldet der Simulator «terminiert». Da es in 1 GB Speicher nur endlich viele Konfigurationen gibt, muss eine der drei Varianten irgendwann eintreten.

(d) Falsch. Es gibt ein Programm, das ein anderes Programm ausführt und falls dieses terminiert, 123 ausgibt. Das heißt, genau dann, wenn das andere Programm nie terminiert, gibt es nie die Ausgabe 123. Dies entspricht einer Reduktion des komplementierten Problem a) auf das Problem d).

Aufgabe 2. Zeigen Sie, dass die Menge der Primzahlen und die Menge der Quadratzahlen entscheidbar ist.

Lösung. Mögliche Entsprechung: Schreiben Sie ein Loop-Programm, das die Ausgabe 1 oder 0 erzeugt, je nachdem, ob die Eingabe (x_1) eine Primzahl beziehungsweise Quadratzahl ist. Konstruktionen aus Folien und Übungen werden als gegebenen angenommen.

Quadratzahl:

$$
\begin{aligned}
& x_2 := x_1; \\
& x_1 := 0; \\
& x_3 := 0; \\
& \mathit{LOOP}\ x_2\ \mathit{DO} \\
& x_4 := x_3 \cdot x_3; \\
& x_3 := x_3 + 1; \\
& \mathit{IF}\ x_4 = x_2\ \mathit{DO}\ x_1 := 1\ \mathit{END} \\
& \mathit{END}
\end{aligned}
$$

Primzahl:

$$
\begin{aligned}
&x_2 := x_1;\\
&x_3 := x_1 - 2;\\
&x_4 := 1;\\
&x_1 := 1;\\
&LOOP\ x_3\ DO\\
&x_4 := x_4 + 1;\\
&x_5 := x_2\ MOD\ x_4;\\
&IF\ x_5 = 0\ DO\ x_1 := 0\ END\\
&END
\end{aligned}
$$

Aufgabe 3. Ein bekanntes Problem aus der Mathematik ist Hilberts zehntes Problem: Gegeben ein Polynom $p(x_1, \ldots, x_n)$ mit ganzzahligen Koeffizienten in n Variablen ($n \geq 1$ beliebig), existieren $x_1, \ldots, x_n \in \mathbb{Z}$ mit $p(x_1, \ldots, x_n) = 0$? Erst 1970 wurde bewiesen, dass dieses Problem unentscheidbar ist.

(a) Ist Hilberts zehntes Problem semi-entscheidbar?

(b) Ist das Komplement semi-entscheidbar?

Lösung.

(a) Wahr. Die Menge der möglichen Belegungen ($\mathbb{Z}^n$) ist abzählbar unendlich (Diagonalisierung). Die möglichen Belegungen können systematisch durchprobiert werden. Das Verfahren terminiert nur, wenn es eine passende Belegung gibt.

(b) Falsch. Aus a) und der Unentscheidbarkeit des Problems folgt, dass b) nicht semi-entscheidbar ist (Folie 402).

Aufgabe 4 ($\star$). Der ebenso geniale wie unberechenbare Wissenschaftler und Superbösewicht Doktor Meta ist unentschlossen. Ein neuer Held ist in Siegen aufgetaucht. Theorie-Man streift durch Siegen und bekämpft das Verbrechen. Doktor Meta weiß nicht, wo Theorie-Man seine Basis hat. Das Rennmotorrad von Theorie-Man, eine *Turing 3000*, hat keine Höchstgeschwindigkeit. Doktor Meta möchte Theorie-Man stellen. Er weiß aber nicht, wie er ihn finden kann.

Zur Vereinfachung des Problems überlegt er sich eine Abstraktion. Er modelliert den Ort von Theorie-Man als natürliche Zahl auf einem eindimensionalen Zahlenstrahl. Die Zeit diskretisiert er ebenfalls als natürliche Zahl. Dann kann er mit einer Funktion $f : \mathbb{N} \to \mathbb{N}$ den Aufenthaltsort von

Theorie-Man beschreiben. Er sucht nun ein System, mit dem er garantiert irgendwann zur richtigen Zeit am richtigen Ort ist.

Gegeben sei eine Funktion $f(t) = v \cdot t + s_0$. Die Parameter $v \in \mathbb{N}$ und $s_0 \in \mathbb{N}$ sind unbekannt, aber fest.

Beschreiben Sie einen Algorithmus, der eine Folge $(s_i)_{i \in \mathbb{N}}$ von natürlichen Zahlen ausgibt, so dass ein $j \in \mathbb{N}$ mit $s_j = f(j)$ existiert. (Der Algorithmus muss nicht terminieren, er muss nur irgendwann ein korrektes Folgenglied ausgeben.)

Lösung. $\mathbb{N}^2$ ist abzählbar unendlich (Diagonalisierung). Folglich existiert eine Bijektion $g \colon \mathbb{N} \to \mathbb{N}^2$. Sei

$$(s_j)_{j \in \mathbb{N}} = (\pi_1(g(j)) \cdot j + \pi_2(g(j)))_{j \in \mathbb{N}}.$$

Zu $v, s_0 \in \mathbb{N}$ existiert ein $j \in \mathbb{N}$ mit $s_j = f(j)$ nämlich $j = g^{-1}(v, s_0)$:

$$\begin{aligned}
s_j &= s_{g^{-1}(v,s_0)} \\
&= \pi_1(g(g^{-1}(v, s_0))) \cdot g^{-1}(v, s_0) + \pi_2(g(g^{-1}(v, s_0))) \\
&= \pi_1(v, s_0) \cdot g^{-1}(v, s_0) + \pi_2(v, s_0) \\
&= v \cdot g^{-1}(v, s_0) + s_0 \\
&= f(g^{-1}(v, s_0)) \\
&= f(j)
\end{aligned}$$

g und somit $j \to s_j$ sind totale berechenbare Funktionen, also gibt es ein Programm, das $s_1, s_2, \ldots$ ausgibt.