Berechenbarkeitstheorie

Prof. Dr. Christoph Walther / Nathan Wasser
Technische Universität Darmstadt — Sommersemester 2011

Lösungsvorschlag zu Hausübung 2

Lösungsvorschlag

Aufgabe 2.1 (Entscheidbarkeit) (8 Punkte)

Zeigen Sie mit dem Satz von Rice, dass die folgenden Probleme nicht entscheidbar sind.

1. (2 Punkte)

   $$M := \{n \in \mathbb{N} \mid \exists i \in \mathbb{N}.\ \varphi_n(i) = 2i + 1\}$$

   ### Lösungsvorschlag

   Für $\Phi = \{f \in [\![\mathcal{P}]\!] \mid \exists i \in \mathbb{N}.\ f(i) = 2i + 1\}$ gilt offensichtlich $M = \natural\Phi$. Außerdem gilt $\mathrm{C}_3^1 \in \Phi$, denn $\mathrm{C}_3^1(1) = 3$. Weiter ist $\mathrm{C}_0^1 \notin \Phi$, denn $\mathrm{C}_0^1(i) = 0 < 2i + 1$ für alle $i \in \mathbb{N}$. Damit ist also $\emptyset \neq \Phi$ und $\Phi \neq [\![\mathcal{P}]\!]$, so dass nach dem Satz von Rice $M = \natural\Phi$ nicht entscheidbar ist.

2. (2 Punkte)

   $$M := \{n \in \mathbb{N} \mid \varphi_n(2) = 4\}$$

   ### Lösungsvorschlag

   Für $\Phi = \{f \in [\![\mathcal{P}]\!] \mid f(2) = 4\}$ gilt offensichtlich $M = \natural\Phi$. Außerdem gilt $\mathrm{C}_4^1 \in \Phi$, denn $\mathrm{C}_4^1(2) = 4$. Weiter ist $\mathrm{S} \notin \Phi$, denn $\mathrm{S}(2) = 3$. Damit ist also $\emptyset \neq \Phi$ und $\Phi \neq [\![\mathcal{P}]\!]$, so dass nach dem Satz von Rice $M = \natural\Phi$ nicht entscheidbar ist.

3. (4 Punkte)

   $$M := \{n \in \mathbb{N} \mid \mathit{Bild}(\varphi_n) \text{ ist entscheidbar}\}$$

   ### Lösungsvorschlag

   Für $\Phi = \{f \in [\![\mathcal{P}]\!] \mid \mathit{Bild}(f) \text{ ist entscheidbar}\}$ gilt offensichtlich $M = \natural\Phi$. Außerdem gilt $\omega \in \Phi$, denn $\mathit{Bild}(\omega) = \emptyset$ ist entscheidbar. Da das Halteproblem H semi-entscheidbar und damit rekursiv aufzählbar ist, gibt es eine berechenbare Funktion a_H mit $\mathit{Bild}(a_H) = H$, also ist $a_H \notin \Phi$, denn H ist nicht entscheidbar. Damit ist also $\emptyset \neq \Phi$ und $\Phi \neq [\![\mathcal{P}]\!]$, so dass nach dem Satz von Rice $M = \natural\Phi$ nicht entscheidbar ist.

Aufgabe 2.2 (Semi-Entscheidbarkeit) (7 Punkte)

Betrachten Sie folgende Menge:

$$M := \{n \in \mathbb{N} \mid \varphi_n \text{ ist primitiv-rekursiv } \}$$

Beweisen Sie, dass M nicht semi-entscheidbar ist. Reduzieren Sie dazu das Totalitätsproblem TOT auf M.

Lösungsvorschlag

Für das folgende Programm `P` gilt $[\![\texttt{P}]\!](n,x) = \begin{cases} \bot & \text{, falls } \varphi_n(x) = \bot \\ n & \text{, sonst} \end{cases}$

```
procedure P(n, x) <=
begin var y, r;
 y := APPLY(PAIR^2(n, x));
 r := n;
 return(r)
end
```

Nach dem *s-m-n*-Theorem gibt es eine totale und berechenbare Funktion $s_1^1 : \mathbb{N}^2 \to \mathbb{N}$ mit

$$\varphi_{s_1^1(\natural \texttt{P},n)}(x) = \begin{cases} \bot & \text{, falls } \varphi_n(x) = \bot \\ n & \text{, sonst} \end{cases}$$

Damit ist die Funktion $\varrho : \mathbb{N} \to \mathbb{N}$ mit $\varrho(n) = s_1^1(\natural P, n)$ ebenfalls total und berechenbar. Es gilt

$$\begin{aligned} n \in TOT &\Rightarrow \forall x \in \mathbb{N}.\ \varphi_n(x) \neq \bot \\ &\Rightarrow \forall x \in \mathbb{N}.\ [\![\texttt{P}]\!](n,x) = n \\ &\Rightarrow \forall x \in \mathbb{N}.\ \varphi_{s_1^1(\natural P,n)}(x) = n \\ &\Rightarrow \varphi_{s_1^1(\natural P,n)} \text{ ist primitiv-rekursiv} \\ &\Rightarrow \varphi_{\varrho(n)} \text{ ist primitiv-rekursiv} \\ &\Rightarrow \varrho(n) \in M, \end{aligned}$$

denn für $n \in TOT$ ist $\varphi_{s_1^1(\natural P,n)} = \mathrm{C}_n^1$, also primitiv rekursiv.

Andernfalls gilt

$$\begin{aligned} n \notin TOT &\Rightarrow \exists i \in \mathbb{N}.\ \varphi_n(i) = \bot \\ &\Rightarrow \exists i \in \mathbb{N}.\ [\![\texttt{P}]\!](n,i) = \bot \\ &\Rightarrow \exists i \in \mathbb{N}.\ \varphi_{s_1^1(\natural P,n)}(i) = \bot \\ &\Rightarrow \varphi_{s_1^1(\natural P,n)} \text{ ist nicht primitiv-rekursiv} \\ &\Rightarrow \varphi_{\varrho(n)} \text{ ist nicht primitiv-rekursiv} \\ &\Rightarrow \varrho(n) \notin M. \end{aligned}$$

Damit ist $TOT \preceq_\varrho M$. Da TOT nicht semi-entscheidbar ist, kann also auch M nicht semi-entscheidbar sein.

Aufgabe 2.3 (μ-rekursive und primitiv-rekursive Funktionen) (15 Punkte)

Beweisen Sie, dass die folgenden Funktionen μ-rekursiv sind, indem Sie jeweils eine Definition durch Einsetzung, strukturelle Rekursion oder Minimierung angeben. Geben Sie außerdem an, ob die Funktion primitiv-rekursiv ist und belegen Sie Ihre Behauptung.

Sie können in Ihren Beweisen sämtliche aus der Vorlesung als primitiv-rekursiv oder μ-rekursiv bekannten Funktionen unter Angabe der Quelle als primitiv-rekursiv beziehungsweise μ-rekursiv annehmen.

1. (3 Punkte)

$$f_1(x) = \begin{cases} \sqrt[3]{x} & \text{, falls } \sqrt[3]{x} \in \mathbb{N} \\ \bot & \text{, sonst} \end{cases}$$

Lösungsvorschlag

$$\begin{aligned} \text{cube}(x) &:= \text{times}(\text{P}_1^1(x), \text{square}(x)) \\ h(x,y) &:= \text{cube}(\text{P}_1^2(x,y)) \\ g(x,y) &:= \text{equals}(h(x,y), \text{P}_2^2(x,y)) \\ f_1(x) &:= \mu g(x) \end{aligned}$$

Die Funktion ist nicht total und damit nicht primitiv-rekursiv.

2. (3 Punkte)

$$f_2(x,y) = \left\lceil \frac{x}{y} \right\rceil$$

Lösungsvorschlag

$$\begin{aligned} h_1(z,x,y) &:= \text{times}(\text{P}_1^3(z,x,y), \text{P}_3^3(z,x,y) \\ h_2(z,x,y) &:= \text{pred}(\text{P}_2^3(z,x,y)) \\ g(z,x,y) &:= \text{greater}(h_1(z,x,y), h_2(z,x,y)) \\ f_2(x,y) &:= \mu g(\text{P}_1^2(x,y), \text{P}_2^2(x,y)) \end{aligned}$$

Die Funktion ist nicht total ($y = 0$) und damit nicht primitiv-rekursiv. (Da $\left\lceil \frac{0}{0} \right\rceil$ undefiniert ist, prüfen wir $z * y > \text{pred}(x)$ statt $z * y \geq x$. So kann die Ungleichung für keine x, z gelten, wenn $y = 0$ ist.)

3. (3 Punkte)

$$f_3(y,x) = \begin{cases} x^y & \text{, falls } y > 0 \\ 1 & \text{, sonst} \end{cases}$$

Lösungsvorschlag

$f_3(0,x) := 1 \quad f_3(y+1,x) := \text{times}(f_3(y,x),x)$

Damit ist die Funktion primitiv-rekursiv.

4. (3 Punkte)

$$f_4(x) = \perp$$

Lösungsvorschlag

$f_4(x) := \mu \mathrm{C}_1^1()$

Die Funktion ist nicht total und damit nicht primitiv-rekursiv.

5. (3 Punkte)

$$f_5(x,y) = \begin{cases} 0 & \text{, falls } \varphi_x(y) = \varphi_y(x) \neq \perp \\ \perp & \text{, sonst} \end{cases}$$

Lösungsvorschlag

$$\begin{aligned} g(0,x,y) &:= \text{equals}(\mathsf{u}_{\mathcal{P}}(\pi^2(x,y)), \mathsf{u}_{\mathcal{P}}(\pi^2(y,x))) \\ g(z+1,x,y) &:= 1 \\ f_5(x,y) &:= \mu g(x,y) \end{aligned}$$

Die Funktion ist nicht total und damit nicht primitiv-rekursiv.

Aufgabe 2.4 (Turing Maschinen) (10 Punkte)

Geben Sie eine Turingmaschine $TM_{palindrom} = (Z, z_{start}, z_{stop}, \{\mathsf{O}, \mathsf{I}, \square\}, \square, \{\mathsf{O}, \mathsf{I}\}, \delta)$ an, die berechnet, ob das Eingabewort ein Palindrom ist, d. h.

$$\phi_{TM_{palindrom}}(x) = \begin{cases} \mathsf{I} & \text{, falls } x \in \{\mathsf{O}, \mathsf{I}\}^* \text{ ein Palindrom ist} \\ \mathsf{O} & \text{, sonst} \end{cases}$$

Lösungsvorschlag

Die Turingmaschine mit $Z := \{z_{start}, z_{null-rechts}, z_{eins-rechts}, z_{null-links}, z_{eins-links}, z_{back}, z_{stop}\}$ und δ definiert durch das folgende Turingprogramm berechnet diese Funktion.

z_{start}	O	$z_{null-rechts}$	$\square$	$\hookrightarrow$		$z_{null-rechts}$	O	$z_{null-rechts}$	O	$\hookrightarrow$
z_{start}	I	$z_{eins-rechts}$	$\square$	$\hookrightarrow$		$z_{null-rechts}$	I	$z_{null-rechts}$	I	$\hookrightarrow$
z_{start}	$\square$	z_{stop}	I	$\circlearrowleft$		$z_{null-rechts}$	$\square$	$z_{null-links}$	$\square$	$\hookleftarrow$
$z_{eins-rechts}$	O	$z_{eins-rechts}$	O	$\hookrightarrow$		$z_{null-links}$	O	z_{back}	$\square$	$\hookleftarrow$
$z_{eins-rechts}$	I	$z_{eins-rechts}$	I	$\hookrightarrow$		$z_{null-links}$	I	z_{stop}	O	$\circlearrowleft$
$z_{eins-rechts}$	$\square$	$z_{eins-links}$	$\square$	$\hookleftarrow$		$z_{null-links}$	$\square$	z_{stop}	I	$\circlearrowleft$
$z_{eins-links}$	O	z_{stop}	O	$\circlearrowleft$		z_{back}	O	z_{back}	O	$\hookleftarrow$
$z_{eins-links}$	I	z_{back}	$\square$	$\hookleftarrow$		z_{back}	I	z_{back}	I	$\hookleftarrow$
$z_{eins-links}$	$\square$	z_{stop}	I	$\circlearrowleft$		z_{back}	$\square$	z_{start}	$\square$	$\hookrightarrow$

Im Startzustand akzeptiert die TM die leere Eingabe, sonst merkt und löscht sie das erste Element und bewegt sich in einem der Zustände $z_{null-rechts}$, $z_{eins-rechts}$ bis zum Ende des Wortes. Dort wird dann in den Zustand $z_{null-links}$, bzw $z_{eins-links}$ gewechselt. Diese Zustände akzeptieren das leere Wort (damit Palindrome ungerader Länge akzeptiert werden), lehnen das Wort bei falscher Endung ab und löschen bei richtiger Endung das letzte Zeichen bevor sie in den Zustand z_{back} wechseln. Im Zustand z_{back} wird wieder zum Anfang des Wortes bewegt um dann wieder in den Startzustand zu wechseln. So akzeptieren wir auch die Palindrome gerader Länge.