Grundzüge der Theoretischen Informatik, WS 21/22: Musterlösung zum 9. Präsenzblatt

Julian Dörfler

Aufgabe P9.1 (Sandwich Satz von Rice)
Wir zeigen eine verallgemeinerte Version des Satzes von Rice. Seien hierzu $L \subseteq \mathbb{N}$ eine Sprache und I_1, I_2 **nicht-triviale Indexmengen** mit $I_1 \subseteq L \subseteq I_2$. Sei Ω wieder die Gödelisierung eines `WHILE`-Programms, welches die überall undefinierte Funktion berechnet. Zeigen Sie nun, dass aus $\Omega \in I_1 \Leftrightarrow \Omega \in I_2$ nun schon $L \notin \mathsf{REC}$ folgt indem Sie die folgenden Aussagen zeigen:

(a) $\Omega \notin I_2 \Rightarrow H_0 \leq L$

(b) $\Omega \in I_1 \Rightarrow \overline{H}_0 \leq L$

Benutzen Sie diesen Satz nun um zu zeigen, dass

$$A = \{i \in \mathbb{N} \mid \varphi_i(i) = i\}$$

nicht entscheidbar ist.

Lösung P9.1 (Sandwich Satz von Rice)
Da I_1 und I_2 nicht-trivial sind, gibt es $g, h \in \mathbb{N}$, so dass $g \in I_1$ und $h \notin I_2$ (hier benötigen wir Nicht-Trivialität).

(a) Wir definieren

$$f(i) := \text{göd}(P_i)$$

wobei P_i das folgende `WHILE`-Programm ist:

Gegeben m, führen wir i auf i aus. Danach geben wir $\varphi_g(m)$ aus.

Beachten Sie, dass φ_g `WHILE`-berechenbar ist, und wir es daher in einem `WHILE`-Programm wie P_i einfach berechnen und ausgeben können. Nun gilt:

Sei $i \in H_0$. Dann hält i auf i. Es folgt, dass $\varphi_{\text{göd}(P_i)} = \varphi_g$. Da $g \in I_1$ und I_1 eine Indexmenge ist, gilt also auch, dass $f(i) = \text{göd}(P_i) \in I_1 \subseteq L$.

Sei $i \notin H_0$. Dann hält i nicht auf i. Daher gilt, dass $\varphi_{\text{göd}(P_i)} = \varphi_\Omega$. Es folgt, dass $f(i) = \text{göd}(P_i) \notin I_2 \supseteq L$, da ansonsten - I_2 ist eine Indexmenge - auch $\Omega \in I_2$, was der Premisse widerspricht.

(b) Wir zeigen die äquivalente Aussage $H_0 \leq \overline{L}$. Dazu definieren wir

$$f(i) := \text{göd}(P_i)$$

wobei P_i das folgende `WHILE`-Programm ist:

Gegeben m, führen wir i auf i aus. Danach geben wir $\varphi_h(m)$ aus.

Beachten Sie, dass φ_h `WHILE`-berechenbar ist, und wir es daher in einem `WHILE`-Programm wie P_i einfach berechnen und ausgeben können. Nun gilt:

Sei $i \in H_0$. Dann hält i auf i. Es folgt, dass $\varphi_{\text{göd}(P_i)} = \varphi_h$. Da $h \notin I_2$ und I_2 eine Indexmenge ist, gilt also auch, dass $f(i) = \text{göd}(P_i) \notin I_2 \supseteq L$.

Sei $i \notin H_0$. Dann hält i nicht auf i. Daher gilt, dass $\varphi_{\text{göd}(P_i)} = \varphi_\Omega$. Es folgt, dass $f(i) = \text{göd}(P_i) \in I_1$, da I_1 eine Indexmenge ist und $\Omega \in I_1$. Da $f(i) \in L$ gilt auch $f(i) \notin \overline{L}$.

Um zu zeigen, dass A unentscheidbar ist, betrachten wir die Mengen

$$I_1 = \{i \in \mathbb{N} \mid \forall x \in \mathbb{N} : \varphi_i(x) = x\}$$
$$I_2 = \{i \in \mathbb{N} \mid \operatorname{dom} \varphi_i(x) \neq \emptyset\}$$

Es gilt nun $I_1 \subseteq A$, da für $i \in I_1$ gilt $\varphi_i(x) = x$ für alle $x \in \mathbb{N}$, also insbesondere für $x = i$ auch gilt $\varphi_i(i) = i$ und damit $i \in A$. Weiterhin gilt $A \subseteq I_2$, da für $i \in A$ gilt $\varphi_i(i) = i$ und somit $\operatorname{dom} \varphi_i \supseteq \{i\} \neq \emptyset$ und damit $i \in I_2$.

Weiterhin sind I_1 und I_2 Indexmengen: Für $i \in I_1$ und $j \in \mathbb{N}$ mit $\varphi_i = \varphi_j$ gilt für alle $x \in \mathbb{N}$ direkt $\varphi_j(x) = \varphi_i(x) = x$, also $j \in I_1$.
Für $i \in I_2$ und $j \in \mathbb{N}$ mit $\varphi_i = \varphi_j$ gilt $\operatorname{dom} \varphi_j = \operatorname{dom} \varphi_i \neq \emptyset$, also $j \in I_2$.

Weiterhin sind beide Indexmengen nicht-trivial, da Ω weder in I_1 noch in I_2 ist und ein Programm, dass die Identitätsfunktion angibt sowohl in I_1, als auch in I_2 ist.

Nun folgt aus $\Omega \notin I_1$ und $\Omega \notin I_2$, dass $H_0 \leq A$. Es folgt aus $H_0 \notin \mathsf{REC}$ also $L \notin \mathsf{REC}$, bzw aus $H_0 \notin \mathsf{co\text{-}RE}$ folgt $A \notin \mathsf{co\text{-}RE}$.

Aufgabe P9.2 (Entscheidbarkeit revisited)
Entscheiden Sie für jede der folgenden Sprachen ob diese (1) eine Indexmenge, (2) rekursiv aufzählbar, (3) co-rekursiv aufzählbar und/oder (4) entscheidbar ist. Beweisen Sie Ihre Antworten.

(a) $A = \{i \in \mathbb{N} \mid \varphi_i(i^2) \text{ ist definiert}\}$

(b) $B = \{i \in \mathbb{N} \mid \varphi_i \text{ ist injektiv}\}$[1]

(c) $C = \{i \in \mathbb{N} \mid \varphi_i \text{ ist total } \implies \exists e \in \mathbb{N} : \varphi_e = \varphi_{\varphi_i(e)}\}$

(d) $D = \{\langle i, j\rangle \in \mathbb{N} \mid \operatorname{im} \varphi_i = \operatorname{dom} \varphi_j\}$

(e) $E = \{\langle i, m\rangle \in \mathbb{N} \mid m \in \operatorname{im} \varphi_i\}$

[1] Zur Erinnerung: injektiv impliziert total

Lösung P9.2 (Entscheidbarkeit revisited)

(a) A ist keine Indexmenge (1). Wir betrachten dazu

$$f(g,x) = \begin{cases} 1 & \text{falls } x = g^2 \\ \text{undef.} & \text{sonst} \end{cases}$$

Durch das Rekursionstheorem existiert also ein $i \in \mathbb{N}$, so dass $\varphi_i(x) = f(i,x)$. Es gilt nun $\varphi_i(i^2) = 1$, also $i \in A$. Wenn wir nun aber eine Gödelnummer j konstruieren, indem wir an das Programm $\text{göd}^{-1}(i)$ eine Anweisung anfügen, die das Ergebnis nicht verändert, so ist $j \neq i$ und $\varphi_j = \varphi_i$, aber $\varphi_j(j^2) = \varphi_i(j^2) = \text{undef.}$, also $j \notin A$.

Es gilt $A \in \mathsf{RE}$ (2), wir können bei Eingabe i einfach i mit Eingabe i^2 simulieren, und falls dies terminiert 1 ausgeben.

Es gilt $A \notin \text{co-}\mathsf{RE}$ (3). Wir reduzieren von H_0 auf A. Gegeben g, geben wir die Gödelisierung des folgenden Programms P_g aus:

Gegeben m, simuliere g auf g.

Sei nun $g \in H_0$. Dann hält g auf Eingabe g. Daher hält P_g auf jeder Eingabe, also insbesondere bei p^2, also $\text{göd}(P_g) \in A$.
Sei nun $g \notin H_0$. Dann hält g nicht bei Eingabe g. In diesem Fall hält P_g bei keiner Eingabe, insbesondere nicht bei p^2, also $\text{göd}(P_g) \notin A$.

Da $H_0 \notin \text{co-}\mathsf{RE}$ ist und unsere Reduktion offensichtlich `WHILE`-berechenbar ist, ist also auch $A \notin \text{co-}\mathsf{RE}$.

Da $A \notin \text{co-}\mathsf{RE}$, gilt ebenfalls $A \notin \mathsf{REC}$ (4).

(b) B ist eine Indexmenge (1). Denn wenn $i \in B$ und $j \in \mathbb{N}$ mit $\varphi_i = \varphi_j$, dann ist φ_j aber direkt auch schon injektiv, also $j \in B$.

Es gilt $B \notin \mathsf{RE}$ (2). Wir reduzieren von $\overline{H_0}$ auf B. Gegeben g, geben wir die Gödelisierung des folgenden Programms P_g aus:

Gegeben m, simuliere g auf g für m Schritte. Falls g innerhalb dieser m Schritte nicht terminiert gib m aus, ansonsten divergiere.

Sei nun $g \in \overline{H_0}$. Dann hält g auf Eingabe g nicht. Daher hält P_g auf jeder Eingabe und berechnet die Identitätsfunktion, p ist also injektiv, also $\text{göd}(P_g) \in B$.
Sei nun $g \notin \overline{H_0}$. Dann existiert eine Schrittzahl t, so dass g auf Eingabe g nach t Schritten hält. In diesem Fall hält P_g bei keiner Eingabe $\geq t$, insbesondere ist p also nicht total und damit auch nicht injektiv, also $\text{göd}(P_g) \notin B$.

Da $\overline{H_0} \notin \mathsf{RE}$ ist und unsere Reduktion offensichtlich `WHILE`-berechenbar ist, ist also auch $B \notin \mathsf{RE}$.

Es gilt $B \notin \text{co-}\mathsf{RE}$ (3). Wir reduzieren von H_0 auf B. Gegeben g, geben wir die Gödelisierung des folgenden Programms P_g aus:

Gegeben m, simuliere g auf g. Danach gib m aus.

Sei nun $g \in H_0$. Dann hält g auf Eingabe g. Daher hält P_g auf jeder Eingabe und berechnet die Identitätsfunktion, p ist also injektiv, also $\text{göd}(P_g) \in B$.
Sei nun $g \notin H_0$. Dann hält g nicht bei Eingabe g. In diesem Fall hält P_g bei keiner Eingabe, insbesondere ist p also nicht total und damit auch nicht injektiv, also $\text{göd}(P_g) \notin B$.

Da $H_0 \notin \mathsf{co\text{-}RE}$ ist und unsere Reduktion offensichtlich `WHILE`-berechenbar ist, ist also auch $B \notin \mathsf{co\text{-}RE}$.

Dass $B \notin \mathsf{REC}$ (4) kann man nun daran sehen, dass B weder in RE, noch in $\mathsf{co\text{-}RE}$ liegt. Alternativ kann man auch verwenden, dass B eine Indexmenge ist und den Satz von Rice verwenden. Hierbei müsste man jedoch noch zeigen, dass B nicht trivial ist.

(c) Es gilt $C \in \mathsf{REC}$ (4), da $C = \mathbb{N}$. Die Eigenschaft φ_i ist total $\implies \exists e \in \mathbb{N} : \varphi_e = \varphi_{\varphi_i(e)}$ ist genau die Aussage des Fixpunktsatzes, somit also für alle $i \in \mathbb{N}$ erfüllt.

Somit ist C sowohl eine (triviale) Indexmenge (1), als auch in RE (2) und $\mathsf{co\text{-}RE}$ (3).

(d) Alle Elemente aus D sind zwar natürliche Zahlen, jedoch interpretieren wir diese nicht als (einzelne) Gödelnummern. Daher macht die Frage ob D eine Indexmenge ist keinen Sinn und hängt sehr stark von der verwendeten Gödelisierung ab.

Es gilt $D \notin \mathsf{RE}$ (3). Wir reduzieren von $\overline{H_0}$ auf D. Gegeben g, geben wir $\langle \text{göd}(P_g), 0\rangle$ aus, wobei P_g das folgenden Programm ist:

Gegeben m, simuliere g auf g für m Schritte. Falls die Simulation hält, divergiere, ansonsten gib m aus.

In jedem Fall hält das Programm mit Gödelnummer 0 ($x_0 := x_0 + x_0$) auf jeder Eingabe, es gilt also $\operatorname{dom} \varphi_0 = \mathbb{N}$.

Sei nun $g \in \overline{H_0}$. Dann Dann hält g auf Eingabe g nicht. Daher hält P_g auf allen Eingaben und berechnet die Identitätsfunktion, also $\operatorname{im} \varphi_{P_g} = \mathbb{N}$, also $\operatorname{im} \varphi_{P_g} = \operatorname{dom} \varphi_0$, also $\langle \text{göd}(P_g), 0\rangle \in D$.
Sei nun $g \notin H_0$. Dann existiert eine Schrittzahl t, so dass g bei Eingabe g hält nach t Schritten hält. In diesem Fall hält P_g bei keiner Eingabe $\geq t$, also $\operatorname{dom} \varphi_{P_g} = \{0, 1, \ldots, t-1\}$, also $\langle \text{göd}(P_g), 0\rangle \notin D$.

Da $\overline{H_0} \notin \mathsf{RE}$ ist und unsere Reduktion offensichtlich `WHILE`-berechenbar ist, ist also auch $D \notin \mathsf{RE}$.

Es gilt $D \notin \mathsf{co\text{-}RE}$ (3). Wir reduzieren von H_0 auf D. Gegeben g, geben wir $\langle \text{göd}(P_g), 0\rangle$ aus, wobei P_g das folgenden Programm ist:

Gegeben m, simuliere g auf g. Danach gib m aus.

In jedem Fall hält das Programm mit Gödelnummer 0 ($x_0 := x_0 + x_0$) auf jeder Eingabe, es gilt also $\operatorname{dom} \varphi_0 = \mathbb{N}$.

Sei nun $g \in H_0$. Dann hält g auf Eingabe g. Daher hält P_g auf allen Eingaben und berechnet die Identitätsfunktion, also $\operatorname{im} \varphi_{P_g} = \mathbb{N}$, also $\operatorname{im} \varphi_{P_g} = \operatorname{dom} \varphi_0$, also $\langle \text{göd}(P_g), 0 \rangle \in D$.
Sei nun $g \notin H_0$. Dann hält g nicht bei Eingabe g. In diesem Fall hält P_g bei keiner Eingabe, also $\operatorname{dom} \varphi_{P_g} = \emptyset$, also $\langle \text{göd}(P_g), 0 \rangle \notin D$.

Da $H_0 \notin \mathsf{co\text{-}RE}$ ist und unsere Reduktion offensichtlich `WHILE`-berechenbar ist, ist also auch $D \notin \mathsf{co\text{-}RE}$. Da D weder in RE, noch $\mathsf{co\text{-}RE}$ ist, gilt ebenfalls $E \notin \mathsf{REC}$ (4).

(e) Alle Elemente aus E sind zwar natürliche Zahlen, jedoch interpretieren wir diese nicht als (einzelne) Gödelnummern. Daher macht die Frage ob E eine Indexmenge ist keinen Sinn und hängt sehr stark von der verwendeten Gödelisierung ab.

Es gilt $E \in \mathsf{RE}$ (2), denn ein Semi-Entscheider kann gegeben $\langle g, m \rangle$, für alle Paare $\langle m', t \rangle$ einfach g mit Eingabe m' für t Schritte simulieren. Gibt eine dieser Simulationen m aus, so gib 1 aus.

Es gilt $E \notin \mathsf{co\text{-}RE}$ (3). Wir reduzieren von H_0 auf E. Gegeben g, geben wir $\langle \text{göd}(P_g), 0 \rangle$ aus, wobei P_g das folgenden Programmes ist:

Gegeben m, simuliere g auf g. Danach gib m aus.

Sei nun $g \in H_0$. Dann hält g auf Eingabe g. Daher hält P_g auf jeder Eingabe und berechnet die Identitätsfunktion, also $\operatorname{im} \varphi_{P_g} = \mathbb{N}$. Insbesondere gilt $0 \in \operatorname{im} \varphi_{P_g}$, also $\langle \text{göd}(P_g), 0 \rangle \in E$.
Sei nun $g \notin H_0$. Dann hält g nicht bei Eingabe g. In diesem Fall hält P_g bei keiner Eingabe, somit gilt $\operatorname{im} \varphi_{P_g} = \emptyset$, also $\langle \text{göd}(P_g), 0 \rangle \notin E$.

Da $H_0 \notin \mathsf{co\text{-}RE}$ ist und unsere Reduktion offensichtlich `WHILE`-berechenbar ist, ist also auch $E \notin \mathsf{co\text{-}RE}$. Da $E \notin \mathsf{co\text{-}RE}$, gilt ebenfalls $E \notin \mathsf{REC}$ (4).