Grundzüge der Theoretischen Informatik, WS 21/22: Musterlösung zum 8. Präsenzblatt

Julian Dörfler

Aufgabe P8.1 (Satz von Rice)
Sei Ω die Gödelisierung eines WHILE-Programms, welches die überall undefinierte Funktion berechnet und sei I eine nicht-triviale Indexmenge. Beweisen Sie den Satz von Rice indem Sie zeigen, dass

(a) $\Omega \notin I \Rightarrow H_0 \leq I$

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

Warum benötigt Ihr Beweis, dass I nicht trivial ist?

Lösung P8.1 (Satz von Rice)
Da I nicht trivial ist, gibt es $g, h \in \mathbb{N}$, so dass $g \in I$ und $h \notin I$ (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 $g \in I$ und I eine Indexmenge ist, gilt also auch, dass $f(i) = \text{göd}(P_i) \in I$.

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

(b) Wir zeigen die äquivalente Aussage $H_0 \leq \overline{I}$. 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 $h \notin I$ und I eine Indexmenge ist, gilt also auch, dass $f(i) = \text{göd}(P_i) \notin I$.

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

Aufgabe P8.2 (Indexmengen)
Welche der folgenden Sprachen sind Indexmengen? Welche der Indexmengen sind nicht-trivial? Beweisen Sie Ihre Antworten.

(a) $L_1 = \{i \in \mathbb{N} \mid \exists m \in \mathbb{N} : \varphi_i(m) = 42\}$

(b) $L_2 = \{i \in \mathbb{N} \mid \varphi_i(i) = i\}$

(c) $L_3 = \{i \in \mathbb{N} \mid \forall m \in \mathbb{N} : \varphi_i(m) = m^2\}$

(d) $L_4 = \{i \in \mathbb{N} \mid \forall m \in \mathbb{N} : \varphi_i(m) = m^2 \text{ und } i \leq 100.000.000.000\}$

(e) H

Lösung P8.2 (Indexmengen)
Für alle Sprachen von denen wir zeigen wollen, dass Sie Indexmengen sind, nehmen wir an, dass wir Gödelisierungen i und j gegeben haben, so dass i in der Sprache enthalten ist und $\varphi_i = \varphi_j$.

(a) L_1 ist eine Indexmenge: Da $i \in L_1$ existiert ein $m \in \mathbb{N}$ mit $\varphi_i(m) = 42$. Wenn $\varphi_i = \varphi_j$ gilt aber ebenfalls $\varphi_j(m) = 42$ und daher $j \in L_1$.

Weiterhin ist L_1 nicht-trivial, denn die Gödelnummer g_1 eines Programmes, das für jede Eingabe 42 ausgibt ist in L_1 und die Gödelnummer g_2 eines Programmes, das für jede Eingabe 0 ausgibt nicht in L_1 ist.

(b) L_2 ist keine Indexmenge. Sei $g \in L_2$ die Gödelisierung des Programmes, das wir in Aufgabe P8.3(b) konstruieren[1] mit

$$\varphi_g(x) = g \quad \forall x \in \mathbb{N}\,.$$

Wir können nun an g eine terminierende Anweisung, die das Ergebnis nicht verändert (z.B. $x_0 := x_0$) anhängen und erhalten eine Gödelnummer $g' \neq g$. Nun gilt $\varphi_{g'} = \varphi_g$, aber $\varphi_{g'}(g') = \varphi_g(g') = g \neq g'$, also $g' \notin L_2$. L_2 kann also keine Indexmenge sein.

[1] Dort hieß die Sprache N_2, aber es gilt $L_2 = N_2$

(c) L_3 ist eine Indexmenge: Wenn $\varphi_i = \varphi_j$, dann gilt für alle $m \in \mathbb{N}$ direkt $\varphi_j(m) = \varphi_i(m) = m^2$ und daher ist j ebenfalls in L_3.

Weiterhin ist L_3 nicht-trivial: Ω hält auf keiner Eingabe und berechnet somit insbesondere nicht die Quadratfunktion, es gilt also $\Omega \notin L_3$. Umgekehrt ist die Quadratfunktion `WHILE`-berechenbar, es gibt also ein Programm $g \in L_3$.

(d) Diese Aufgabe hängt sehr stark von der verwendeten Gödelisierung ab. Da L_4 endlich ist, gilt dass $L_4 \in \mathsf{REC}$. Wäre L_4 eine Indexmenge, müsste sie daher - nach dem Satz von Rice - trivial sein.

Mit der von uns verwendeten Gödelisierung lässt sich beweisen, dass es aber kein `WHILE`-Programm mit einer Gödelisierung $\leq 100.000.000.000$ gibt, das die Quadratfunktion berechnet. Es gilt also L_4 ist eine triviale Indexmenge.

Für andere Gödelisierungen könnte L_4 aber nicht-trivial sein, womit sie dann keine Indexmenge wäre.

(e) H ist nicht definiert als eine Menge von Gödelnummern, es ist also nicht sinnvoll bei dieser Menge zu betrachten, ob sie eine Indexmenge ist.

Aufgabe P8.3 (Rekursionstheorem)
Zeigen Sie, dass die folgenden Mengen nicht leer sind:

(a) $N_1 = \{i \in \mathbb{N} \mid \forall m \in \mathbb{N} : \varphi_i(m) = i^2\}$

(b) $N_2 = \{i \in \mathbb{N} \mid \varphi_i(i) = i\}$

Lösung P8.3 (Rekursionstheorem)
Wir benutzen für alle Aufgabenteile das Rekursionstheorem.

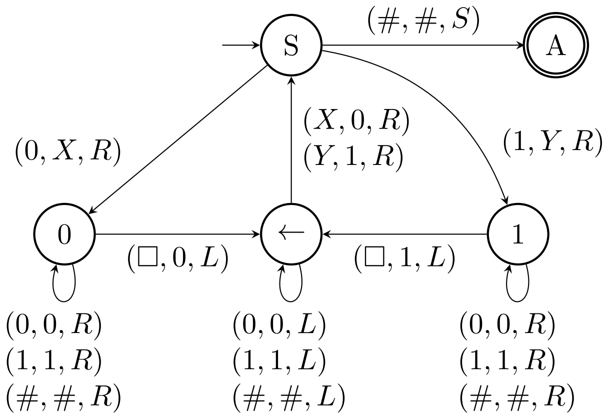
(a) Sei $f(g, m) = g^2$. f ist offensichtlich `WHILE`-berechenbar, daher gibt es ein i mit $\varphi_i(m) = f(i, m) = i^2$ für alle $m \in \mathbb{N}$. Somit ist $i \in N_1$.

(b) Sei $f(g, m) = g$. f ist offensichtlich `WHILE`-berechenbar, daher gibt es ein i mit $\varphi_i(i) = f(i, i) = i$. Somit ist $i \in N_2$.

Aufgabe P8.4 (Turing Maschinen)
Entwickeln Sie eine 1-Band-Turing-Maschine, die für alle Wörter $w \in \{0,1\}^\star$ auf Eingabe $w\#$ mit der Ausgabe $w\#w$ terminiert. (Wie sich Ihre Turing-Maschine bei anderen Eingaben verhält, ist egal.)

Es gibt eine 1-Band-Turing-Maschine mit 5 Zuständen, die obige Spezifikation erfüllt. Sie sollten nicht wesentlich mehr Zustände benutzen. Geben Sie Ihre Turing-Maschine als Zustandsübergangsdiagramm[2] an und kommentieren Sie dieses bitte ausreichend!
Hinweis: Überlegen Sie sich zunächst, wie eine Turing-Maschine das Zeichen unter der aktuellen Kopfposition auf das nächste freie Leerzeichen rechts vom Kopf kopiert und dann den Kopf wieder auf die Ausgangsposition zurück bewegt. Es könnte hierfür nützlich sein, das Bandalphabet zu erweitern!

Lösung P8.4 (Turing Maschinen) Der Start-Zustand ist durch ein „S“ markiert, der Endzustand durch ein „A“. Eine Position, an der eine 0 steht, wird durch ein „X“ markiert, eine Position mit einer 1 durch ein „Y“. In den Zuständen 0 und 1 merkt sich die Maschine das gelesene Zeichen und schreibt es dann beim Übergang nach $\leftarrow$ auf die erste freie Position. Dann läuft sie wieder nach links, ersetzt X durch 0 oder Y durch 1 und geht einen Schritt nach rechts, um dort weiter zu machen.



[2] Vgl. Skript Seite 141.