TECHNISCHE UNIVERSITÄT DRESDEN

Fakultät Informatik, Institut für Künstliche Intelligenz, Professur Computational Logic

Dr. habil. Hannes Straß | Dr. Dörthe Arndt

Übungen zur Lehrveranstaltung

# Theoretische Informatik und Logik

Sommersemester 2022

## 2. Übungsblatt

Woche vom 25. bis 29. April 2022

Die folgenden Aufgaben werden nicht in den Übungen besprochen und dienen der Selbstkontrolle.

**Aufgabe C**

Zeigen Sie, dass $\{1\}^*$ unentscheidbare Teilmengen besitzt.

**Aufgabe D**

Welche der folgenden Aussagen sind wahr? Begründen Sie Ihre Antwort.

a) Jedes LOOP-Programm terminiert.

b) Zu jedem WHILE-Programm gibt es ein äquivalentes LOOP-Programm.

c) Die Anzahl der Ausführungen von $P$ in der LOOP-Schleife

$$\texttt{LOOP } x_i \texttt{ DO } P \texttt{ END}$$

kann beeinflusst werden, indem $x_i$ in $P$ entsprechend modifiziert wird.

d) Die Ackermannfunktion ist total und damit LOOP-berechenbar.

**Aufgabe 1**

Zeigen Sie, dass folgende Funktionen $f\colon \mathbb{N}^2 \to \mathbb{N}$ LOOP-berechenbar sind:

a) $f(x, y) := \max(x - y, 0)$

b) $f(x, y) := x \cdot y$

c) $f(x, y) := \max(x, y)$

d) $f(x, y) := \mathrm{ggT}(x, y)$, wobei $\mathrm{ggT}(x, y)$ den größten gemeinsamen Teiler von $x$ und $y$ bezeichnet.

Implementieren Sie einen Interpreter für LOOP-Programme, der in der Lage ist beliebige LOOP-Programme auszuführen. Verwenden Sie Ihren Interpreter um Ihre LOOP-Programme aus den Aufgabenteilen a–d zu testen. Achten Sie hierbei erneut auf mögliche Randfälle.

**Aufgabe 2**

Mit $\mathrm{kgV}(x_1, x_2)$ bezeichnen wir das kleinste gemeinsame Vielfache zweier natürlicher Zahlen $x_1$ und $x_2$.

a) Geben Sie ein WHILE-Programm an, das die Funktion $f\colon \mathbb{N}^2 \to \mathbb{N}, (x_1, x_2) \mapsto \mathrm{kgV}(x_1, x_2)$ berechnet und erklären Sie seine Arbeitsweise.

b) Erweitern Sie Ihren Interpreter für LOOP-Programme aus Aufgabe 1 um WHILE-Schleifen und testen Ihr WHILE-Programm für kgV mithilfe Ihres Interpreters. Achten Sie erneut auf mögliche Randfälle.

**Aufgabe 3**

Zeigen Sie, dass es keine Many-One-Reduktion vom Halteproblem $\mathbf{P}_{\mathsf{halt}}$ von Turing-Maschinen auf das Leerheitsproblem

$$\mathbf{P}_{\mathsf{leer}} := \{\mathsf{enc}(\mathcal{M}) \mid \mathcal{L}(\mathcal{M}) = \emptyset\}$$

von Turing-Maschinen gibt.

**Aufgabe 4**

Zeigen Sie, dass jede semi-entscheidbare Sprache $L$ auf das Halteproblem $\mathbf{P}_{\mathsf{halt}}$ many-one-reduziert werden kann.