Theoretische Informatik II
Aufgabenblatt 4

Prof. Dr. Roland Meyer
Sebastian Muskalla, Peter Chini
TU Braunschweig
Sommersemester 2017

Ausgabe: 17. Mai — Abgabe: 26. Mai

Werfen Sie Ihre Lösungen bis Freitag, 26. Mai, 12:00 Uhr in die Abgabekästen im Informatikzentrum neben Büro 343. Geben Sie zu dritt oder zu viert ab.

Auf diesem Aufgabenblatt genügt es, das Verhalten von Turing-Maschinen informell zu beschreiben. Sie können sich hierbei an den Vorlesungsnotizen zu Unentscheidbarkeit orientieren.

Aufgabe 1: Messen von Zeit und Platz

Sei M eine gegebene TM über dem Eingabealphabet Σ. Wir definieren die Funktion

$$\begin{aligned} \mathrm{time}_M \quad &: \quad \Sigma^* \quad \to_p \quad \mathbb{N} \\ & \qquad x \quad \mapsto \quad \begin{cases} n, & \text{falls } M \text{ auf Eingabe } x \text{ nach } n \text{ Schritten hält,} \\ \text{undefiniert}, & \text{sonst.} \end{cases} \end{aligned}$$

Hiermit ist gemeint, dass $\mathrm{time}_M(x)$ die kleinste Zahl n ist, so dass M nach n Schritten hält.

Zu einer Konfiguration $c = u\,q\,v$ definieren wir $\mathrm{usedspace}(c) = |u| + |v|$ als die Menge der Zellen auf dem Band, die in der Konfiguration belegt sind. Man beachte, dass wir davon ausgehen, dass der letzte Buchstabe von v nicht das Blank-Symbol $\sqcup$ ist, d.h. wir lassen die unendlich vielen Blanks am Ende des Bandinhaltes aus.

$$\begin{aligned} \mathrm{space}_M \quad &: \quad \Sigma^* \quad \to_p \quad \mathbb{N} \\ & \qquad x \quad \mapsto \quad \begin{cases} \max_{i \in \mathbb{N}} \mathrm{usedspace}(c_i), & \text{falls } M \text{ auf Eingabe } x \text{ hält,} \\ \text{undefiniert}, & \text{sonst.} \end{cases} \end{aligned}$$

Hierbei ist $c_0 \to c_1 \to c_2 \to \ldots$ die Folge der Konfigurationen, die bei der Berechnung zur Eingabe x auftreten.

a) Beweisen Sie, dass für jede TM M die Funktion time_M berechenbar ist.

   Überlegen Sie sich zunächst, was es bedeutet, dass eine Funktion mit Signatur $\Sigma^* \to_p \mathbb{N}$ berechenbar ist.

b) Beweisen Sie, dass für jede TM M die Funktion space_M berechenbar ist.

   Zeigen Sie zunächst, dass die Funktion wohldefiniert ist, also tatsächlich jeder Eingabe w, auf welcher die Maschine hält, eine eindeutige natürliche Zahl zuordnet.

c) Wir definieren nun die Funktion $\mathrm{time}\colon \{0,1\}^* \times \{0,1\}^* \to_p \mathbb{N}$. die eine Kodierung w einer TM und eine Eingabe x nimmt, und den Zeitverbrauch von M_w für Eingabe x zurückgibt, also $\mathrm{time}(w,x) = \mathrm{time}_{M_w}(x)$.

   Zeigen Sie, dass auch diese Funktion berechenbar ist.

Aufgabe 2: Ein nicht-semi-entscheidbares Problem

In der Vorlesung haben Sie gesehen, dass das Halteproblem unentscheidbar ist. Nun wollen wir ein Problem kennenlernen, das nicht einmal semi-entscheidbar ist.

Non-Self-Acceptance	
Gegeben:	Turing-Maschine M mit Eingabealphabet $\{0, 1\}$
Entscheide:	Akzeptiert M ihre eigene Kodierung $\langle M \rangle$ nicht?

Als Sprache lässt sich das Problem wie folgt auffassen:

$$\mathcal{L}_{NSA} = \left\{ w \in \{0,1\}^* \mid \text{Die von } w \text{ codierte Turing-Maschine akzeptiert w nicht, d.h. } w \notin \mathcal{L}(M_w) \right\}.$$

a) Beweisen Sie unter Verwendung von Diagonalisierung, dass $\mathcal{L}_{NSA}$ nicht semi-entscheidbar ist.

   Nehmen Sie also an, dass es eine TM M mit $\mathcal{L}(M) = \mathcal{L}_{NSA}$ gibt, betrachten Sie die Kodierung $\langle M \rangle$ und leiten Sie einen Widerspruch her.

b) Zeigen Sie, dass $\mathcal{L}_{NSA}$ co-semi-entscheidbar ist, d.h. beweisen Sie, dass das Komplementproblem semi-entscheidbar ist.

Aufgabe 3: Reduktionen

Beweisen Sie, dass die folgenden Sprachen unentscheidbar sind, in dem Sie ein bereits als unentscheidbar bekanntes Problem auf sie reduzieren.

a)

$$\mathcal{L}_{\text{Accept}} = \left\{ w\#x \in \{0,1,\#\}^* \mid w, x \in \{0,1\}^*, x \in \mathcal{L}(M_w) \right\}$$

b)

$$\mathcal{L}_{\text{Empty}} = \left\{ w \in \{0,1\}^* \mid \mathcal{L}(M_w) = \varnothing \right\}$$

(Verwenden Sie nicht den Satz von Rice!)

Aufgabe 4: Von-Neumann-Architektur

Informieren Sie sich über die Von-Neumann-Architektur.

In dieser Aufgabe sollen sollen Sie ausarbeiten, in wie fern sich die Konzepte der universellen Turing-Maschine in diesem Modell wiederfinden lassen.

Welche der Komponenten haben eine direkte Entsprechung in der universelle Turing-Maschine, und wie ist diese Entsprechung? Was ist mit den sonstigen Komponenten?

In wie fern entspricht der Ablauf eines Programms in der Von-Neumann-Architektur einem Lauf einer universellen Turing-Maschine?