Algorithmen & Datenstrukturen 2

Wintersemester 2022/23

Prof. Dr. Martin Hoefer
Tim Koglin, Marco Schmalhofer

GOETHE UNIVERSITÄT FRANKFURT AM MAIN
Institut für Informatik
Algorithmen und Komplexität

Übungsblatt 6

Ausgabe: 06.12.2022
Abgabe: 13.12.2022, **08:00**

Mit * markierte Aufgaben sind Bonusaufgaben, d.h. sie werden nicht zur Summe der erreichbaren Punkte hinzugezählt.

Aufgabe 6.1 *Eine einfache Reduktion* (6 Punkte)

Betrachten Sie die folgende Sprache:

$$L = \{\, \langle M \rangle : M \text{ ist eine Turingmaschine und akzeptiert das leere Wort als Eingabe} \,\}$$

Zeigen Sie die Reduktion $H_\varepsilon \leq L$. Was bedeutet das für die Sprache L?

Aufgabe 6.2 *Entscheidbar oder nicht?* (6 + 6 Punkte)

Geben Sie für jede der folgenden Sprachen an, ob sie entscheidbar ist oder nicht. Beweisen Sie Ihre jeweilige Antwort, d.h. geben Sie entweder einen Algorithmus an, der die Sprache entscheidet, oder zeigen Sie deren Unentscheidbarkeit (z.B. durch eine geeignete Reduktion).

a) $L_A = \{\, w \in \{0,1\}^* : \text{die Länge von } w \text{ ist eine Primzahl} \,\}$

b) $L_B = \{\, \langle M \rangle w : M \text{ ist eine Turingmaschine und } M \text{ durchläuft auf Eingabe } w \text{ alle Zustände ihrer Zustandsmenge } Q \text{ mindestens einmal} \,\}$

Aufgabe 6.3 *Satz von Rice* (4 + 4 + 4 Punkte)

Zeigen Sie unter Anwendung des Satzes von Rice, dass die folgenden Sprachen unentscheidbar sind. Begründen Sie jeweils auch die Nicht-Trivialität der Menge S.

a) $L_1 = \{\, \langle M \rangle : M \text{ ist eine Turingmaschine und } M \text{ hält auf allen Eingaben und schreibt 110 als Ausgabe} \,\}$

b) $L_2 = \{\, \langle M \rangle : M \text{ ist eine Turingmaschine und immer wenn } M \text{ hält, ist das Band leer} \,\}$

c) $L_3 = \{\, \langle M \rangle : M \text{ ist eine Turingmaschine und } M \text{ hält für mindestens 2022 Eingabeworte} \,\}$

Aufgabe 6.4 *Theo am Lagerfeuer* (4 + 4 + 2 Punkte)

Auf seinem Survival-Trip ist Theo bedingt durch die Jahreszeit gezwungen, die kalten Nächte am Lagerfeuer zu verbringen, ohne dabei viel schlafen zu können. In diesen langen Nächten denkt er oft über all die Algorithmen nach, die er in seinem Leben bereits entworfen hat, und darüber, wie mühsam doch der Nachweis von Laufzeitschranken häufig war. Dabei kommt in ihm immer wieder die Frage auf, ob nicht der Nachweis von Laufzeitschranken für Algorithmen automatisiert werden kann.

Um dieser Frage nachzugehen, betrachtet er zunächst nur den Fall konstanter Laufzeiten, und definiert die folgende Sprache:

$$\begin{aligned} L = \{\, \langle M\rangle : M &\text{ ist eine Turingmaschine und es existiert eine} \\ &\text{Konstante } c > 0 \text{, so dass } \mathtt{Zeit}_M(n) \le c \text{ für alle } n \in \mathbb{N} \,\} \end{aligned}$$

Theo fragt sich: Gibt es einen Algorithmus zur Erkennung der Sprache L?

Helfen Sie Theo, indem Sie die folgenden Aussagen beweisen:

a) L ist nicht entscheidbar.

b) L ist rekursiv aufzählbar.

c) Das Komplement $\bar{L}$ von L ist nicht rekursiv aufzählbar.

Aufgabe 6.5 *Nicht rekursiv aufzählbare Sprachen* (9* Bonuspunkte)

Betrachten Sie die folgende Sprache:

$$L_{\text{bonus}} = \{\, \langle M_1\rangle\langle M_2\rangle : M_1, M_2 \text{ sind Turingmaschinen und } L(M_1) = L(M_2) \,\}$$

Zeigen Sie, dass weder L_{bonus} noch dessen Komplement $\overline{L_{\text{bonus}}}$ rekursiv aufzählbar ist.

Hinweis: Sie können beispielsweise zeigen, dass $\overline{H_\varepsilon} \le L_{\text{bonus}}$ und $\overline{H_\varepsilon} \le \overline{L_{\text{bonus}}}$. Was bedeutet das für die Sprachen L_{bonus} und $\overline{L_{\text{bonus}}}$?

Bei allgemeinen Anmerkungen zu den Übungsaufgaben oder Fragen zum Übungsbetrieb erreichen Sie uns unter der folgenden E-Mail-Adresse: algo222@cs.uni-frankfurt.de.