Institut für Theoretische Informatik
Lehrstuhl Prof. Dr. D. Wagner

Übungsblatt 3

Vorlesung Theoretische Grundlagen der Informatik im WS 18/19

Ausgabe 20. November 2018
Abgabe 4. Dezember 2018, 11:00 Uhr (im Kasten im UG von Gebäude 50.34)

Aufgabe 1

(4 + 2 = 6 Punkte)

Gegeben sei das Alphabet $\Sigma = \{\texttt{0}, \texttt{1}\}$.

(a) Geben Sie den Zustandsgraphen einer deterministischen Turingmaschine an, die für eine Eingabe $w \in \Sigma^\star$ alle Vorkommen von `0` in w löscht. Die Eingabe `1101001` wird also z.B. zu `1111` transformiert.

(b) Geben Sie die Konfigurationsfolge an, die bei Eingabe des Wortes `01101` durchlaufen wird.

Aufgabe 2

(1 + 1 + 1 = 3 Punkte)

Betrachten Sie die folgenden Entscheidungsprobleme:

(a) Gegeben eine natürliche Zahl n, ist n *keine* Primzahl?

(b) Gegeben eine Menge $M = \{m_1, \dots, m_k\}$ von natürlichen Zahlen, gibt es eine Teilmenge $M' \subseteq M$, deren Summe genau 100 ist?

(c) Gegeben ein ungerichteter Graph $G = (V, E)$, gibt es einen Pfad, der jeden Knoten in V genau einmal besucht?

Geben Sie für jedes dieser Probleme eine Ja-Instanz und eine Nein-Instanz an. Beschreiben Sie außerdem, wie eine NTM arbeitet, die das Problem in polynomieller Zeit löst. Geben Sie dazu insbesondere an, wie die NTM den Lösungsvorschlag des Orakelmoduls interpretiert und was der deterministische Teil der NTM tun muss, um den Lösungsvorschlag zu überprüfen.

Aufgabe 3

(3 Punkte)

Sei $\mathcal{M}$ eine nichtdeterministische Turingmaschine mit Eingabealphabet Σ. Nehmen Sie an, dass für jede natürliche Zahl n eine Eingabe x der Länge n existiert, die von $\mathcal{M}$ akzeptiert wird. Betrachten Sie die folgenden Ideen, die Zeitkomplexitätsfunktion für $\mathcal{M}$ alternativ zu definieren:

(a)

$$A_{\mathcal{M}}(n) := \min\left(\left\{ m \;\middle|\; \begin{array}{l} \text{Es gibt ein } x \in L_{\mathcal{M}} \text{ mit } |x| = n, \\ \text{sodass alle akzeptierenden Berechnungen} \\ \text{höchstens } m \text{ Schritte brauchen.} \end{array} \right\}\right)$$

(b)

$$B_{\mathcal{M}}(n) := \max\left(\left\{ m \;\middle|\; \begin{array}{l} \text{Es gibt ein } x \in L_{\mathcal{M}} \text{ mit } |x| = n, \\ \text{sodass alle akzeptierenden Berechnungen} \\ \text{mindestens } m \text{ Schritte brauchen.} \end{array} \right\}\right)$$

(c)

$$C_{\mathcal{M}}(n) := \max\left(\left\{ m \;\middle|\; \begin{array}{l} \text{Für alle } x \in L_{\mathcal{M}} \text{ mit } |x| = n \\ \text{gibt es eine akzeptierende Berechnung,} \\ \text{die höchstens } m \text{ Schritte braucht.} \end{array} \right\}\right)$$

(d)

$$D_{\mathcal{M}}(n) := \min\left(\left\{ m \;\middle|\; \begin{array}{l} \text{Für alle } x \in L_{\mathcal{M}} \text{ mit } |x| = n \\ \text{gibt es eine akzeptierende Berechnung,} \\ \text{die höchstens } m \text{ Schritte braucht.} \end{array} \right\}\right)$$

(e)

$$E_{\mathcal{M}}(n) := \max\left(\left\{ m \;\middle|\; \begin{array}{l} \text{Es gibt ein } x \in L_{\mathcal{M}} \text{ mit } |x| = n, \\ \text{für das es eine akzeptierende Berechnung gibt,} \\ \text{die genau } m \text{ Schritte braucht.} \end{array} \right\}\right)$$

(f)

$$F_{\mathcal{M}}(n) := \max\left(\left\{ m \;\middle|\; \begin{array}{l} \text{Es gibt ein } x \in \Sigma^\star \text{ mit } |x| = n, \\ \text{sodass alle akzeptierenden Berechnungen} \\ \text{mindestens } m \text{ Schritte brauchen.} \end{array} \right\}\right)$$

Welche dieser Funktionen $A_{\mathcal{M}}, B_{\mathcal{M}}, C_{\mathcal{M}}, D_{\mathcal{M}}, E_{\mathcal{M}}, F_{\mathcal{M}}$ sind identisch mit der Zeitkomplexitätsfunktion $T_{\mathcal{M}}$ von $\mathcal{M}$ aus der Vorlesung? Begründen Sie!

Aufgabe 4 (1 + 1 + 1 + 1 = 4 Punkte)

Zeigen Sie:

(a) Das Komplement des Halteproblems ist nicht semi-entscheidbar.

(b) Das Komplement der Diagonalsprache ist semi-entscheidbar.

(c) Seien L_1 und L_2 semi-entscheidbare Sprachen. Dann ist auch $L_1 \cap L_2$ semi-entscheidbar.

(d) Seien L_1 und L_2 semi-entscheidbare Sprachen. Dann ist auch $L_1 \cup L_2$ semi-entscheidbar.

Aufgabe 5 (2 + 3 = 5 Punkte)

Eine Turingmaschine M zählt eine unendliche Sprache L auf, wenn M niemals stoppt und eine Liste $w_1, w_2, \ldots$ genau der Wörter aus L ausgibt. Dabei ignoriert M die Eingabe und die Wörter der ausgegebenen Liste sind eindeutig voneinander getrennt. Für die Reihenfolge der Wörter in der Liste muss gelten, dass jedes Wort aus L nach endlich vielen Schritten ausgegeben wird. Eine unendliche Sprache L ist aufzählbar, falls eine Turingmaschine existiert, die L aufzählt.

(a) Zeigen Sie, dass L genau dann entscheidbar ist, wenn L in kanonischer Reihenfolge[1] aufzählbar ist.

(b) Zeigen Sie, dass L genau dann semi-entscheidbar ist, wenn L aufzählbar ist. Erklären Sie auch, warum sich hier im Gegensatz zu Aufgabenteil (a) nicht fordern lässt, dass die Reihenfolge der Aufzählung kanonisch ist.

Aufgabe 6 (2 Punkte)

Zeigen Sie, dass die Sprache $\mathcal{H}_{\text{all}} = \{w \mid T_w \text{ hält auf allen Eingaben}\}$ nicht entscheidbar ist. Benutzen Sie dafür nicht den Satz von Rice!

Aufgabe 7 (1 + 2 + 1 = 4 Punkte)

In der Übung wurde das Äquivalenzproblem für Turingmaschinen vorgestellt:

$$L_{\text{äq}} = \{w\#v \in \{0, 1, \#\}^\star \mid L(T_w) = L(T_v)\}$$

In dieser Aufgabe sollen Sie zeigen, dass weder $L_{\text{äq}}$ noch $L_{\text{äq}}^c$ semi-entscheidbar ist. Gehen Sie dazu wie folgt vor:

(a) Zeigen Sie, dass das Komplement der universellen Sprache L_u^c nicht semi-entscheidbar ist.

(b) Zeigen Sie, dass $L_{\text{äq}}$ nicht semi-entscheidbar ist. Nehmen Sie dazu an, es gäbe eine Turingmaschine $M_{\text{äq}}$, die $L_{\text{äq}}$ akzeptiert. Konstruieren Sie daraus eine Turingmaschine M_u^c, die L_u^c akzeptiert.

(c) Zeigen Sie mit der gleichen Herangehensweise wie in Aufgabenteil (b), dass $L_{\text{äq}}^c$ nicht semi-entscheidbar ist.

[1] Siehe Vorlesung 6, Folie 19. Die kanonische Reihenfolge sortiert Wörter nach aufsteigender Länge und Wörter der gleichen Länge lexikographisch.