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

Übungsblatt 3

Vorlesung Theoretische Grundlagen der Informatik im WS 17/18

Ausgabe 21. November 2017
Abgabe 5. Dezember 2017, 11:00 Uhr (im Kasten im UG von Gebäude 50.34)

Aufgabe 1

(1 + 1 + 1 + 1 + 1 = 5 Punkte)

Betrachte das Post'sche Korrespondenzproblem (kurz: PKP) aus der Vorlesung mit Eingabe $K = ((x_1, y_1), \dots, (x_n, y_n))$, $x_i, y_i \in \Sigma^*$ für $i = 1, \dots, n$.

(a) Zeigen Sie dass PKP entscheidbar ist, wenn $|x_i| = |y_i|$ für $i = 1, \dots, n$.

(b) Zeigen Sie dass PKP entscheidbar ist, wenn $|\Sigma| = 1$.

(c) Zeigen Sie dass $K = ((10, 101), (011, 11), (101, 011))$ eine Nein-Instanz des PKP ist.

(d) Zeigen Sie dass $K = ((001, 0), (01, 011), (01, 101), (10, 001))$ eine Ja-Instanz des PKP ist.

(e) Zeigen Sie dass PKP nicht entscheidbar ist, wenn $|\Sigma| = 2$.
(Benutzen Sie dass PKP nicht entscheidbar ist für allgemeine Σ.)

Aufgabe 2

(3 Punkte)

Sei $\mathcal{M}$ eine nichtdeterministische Turingmaschine mit Eingabealphabet Σ und ausgezeichneten Endzuständen q_J, q_N. 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 worst-case Laufzeit (a.k.a. Zeitkomplexitätsfunktion) für $\mathcal{M}$ alternativ zu definieren:

$$A_{\mathcal{M}}(n) := \max\left(\left\{ m \;\middle|\; \begin{array}{l} \text{Es gibt eine Eingabe } x \in L_{\mathcal{M}} \text{ mit } |x| = n \text{, so dass} \\ \mathcal{M} \text{ in höchstens } m \text{ Schritten} \\ \text{in den Zustand } q_J \text{ überführt werden kann.} \end{array} \right\}\right)$$

$$B_{\mathcal{M}}(n) := \min\left(\left\{ m \;\middle|\; \begin{array}{l} \text{Für alle Eingaben } x \in L_{\mathcal{M}} \text{ mit } |x| = n \text{, gilt dass} \\ \mathcal{M} \text{ nicht in } m \text{ Schritten} \\ \text{in den Zustand } q_J \text{ überführt werden kann.} \end{array} \right\}\right) + 1$$

$$C_{\mathcal{M}}(n) := \max\left(\left\{ m \;\middle|\; \begin{array}{l} \text{Für alle Eingaben } x \in \Sigma^* \text{ mit } |x| = n \text{ gilt, dass} \\ \mathcal{M} \text{ in höchstens } m \text{ Schritten} \\ \text{in den Zustand } q_J \text{ oder } q_N \text{ überführt werden kann.} \end{array} \right\}\right)$$

$$D_{\mathcal{M}}(n) := \min\left(\left\{ m \;\middle|\; \begin{array}{l} \text{Für alle Eingaben } x \in \Sigma^* \text{ mit } |x| = n \text{ gilt, dass} \\ \mathcal{M} \text{ in höchstens } m \text{ Schritten} \\ \text{in den Zustand } q_J \text{ oder } q_N \text{ überführt werden kann.} \end{array} \right\}\right)$$

$$E_{\mathcal{M}}(n) := \max\left(\left\{ m \;\middle|\; \begin{array}{l} \text{Es gibt eine Eingabe } x \in \Sigma^* \text{ mit } |x| = n, \text{ so dass} \\ \mathcal{M} \text{ in höchstens } m \text{ Schritten hält.} \end{array} \right\}\right)$$

$$F_{\mathcal{M}}(n) := \max\left(\left\{ m \;\middle|\; \begin{array}{l} \text{Es gibt eine Eingabe } x \in L_{\mathcal{M}} \text{ mit } |x| = n, \text{ so dass} \\ \mathcal{M} \text{ nicht in } m \text{ Schritten} \\ \text{in den Zustand } q_J \text{ überführt werden kann.} \end{array} \right\}\right) + 1$$

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 3

(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 4

(2 + 3 = 5 Punkte)

Eine Turingmaschine M zählt eine 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. Die Wörter aus L dürfen in beliebiger Reihenfolge in der Liste vorkommen. Eine Sprache L ist aufzählbar falls eine Turingmaschine existiert, die L aufzählt.

(a) Zeigen Sie, dass L in aufsteigender Reihenfolge aufzählbar ist, wenn L entscheidbar ist.

(b) Zeigen Sie, dass L aufzählbar ist, wenn L semi-entscheidbar ist. Erklären Sie auch, wieso die Worte in beliebiger Reihenfolge vorkommen können.

Aufgabe 5

(2 + 2 = 4 Punkte)

In der Übung[1] wurde Cantors zweites Diagonalargument vorgestellt. Es wurde bewiesen, dass die Teilmenge $(0, 1)$ der reellen Zahlen überabzählbar ist. In dieser Aufgabe geht es darum, das Argument in angepasster Form zu verwenden, um ganz einfach zu zeigen, dass nicht alle Sprachen entscheidbar sind.

[1] zweite Übung, Folie 15

(a) Sei A eine nichtendliche, abzählbare Menge. Zeigen Sie, dass die Potenzmenge 2^A überabzählbar ist. Gehen Sie dabei analog zur Übung vor, d.h. geben Sie eine Funktion f und eine geeignete beispielhafte Tabelle an, und konstruieren Sie ein Element $x \in 2^A \setminus \text{Bild}(f)$.

(b) Nutzen Sie Teilaufgabe (a), um zu zeigen, dass nicht alle Sprachen entscheidbar sind.

Aufgabe 6

(2 + 2 = 4 Punkte)

In der Vorlesung hatten wir gesehen, dass die endliche Automaten *genau* die regulären Sprachen akzeptieren können. Bei Turingmaschinen ist die Situation leider nicht so einfach. Neben den entscheidbaren Sprachen gibt es noch die semi-entscheidbaren und die nicht-entscheidbaren Sprachen. Wäre es nicht schön, wenn wir ein Berechnungsmodell hätten, das *genau* die entscheidbaren Sprachen akzeptiert?

Der ebenso optimistische wie unüberlegte Superwissenschaftler Doktor Meta ist schon dabei, eine solche Maschine, die sogenannte *Metamagiemaschine* (M^3), zu konstruieren – reichlich Gelder von seriösen Investoren hat er bereits eingestrichen[2]. Sie sind dazu angestellt, die letzten Details der M^3 auszuarbeiten. Die M^3 soll folgenden Anforderungen genügen:

- Jede entscheidbare Sprache wird von einer M^3 akzeptiert, und jede M^3 akzeptiert eine entscheidbare Sprache.
- Jede M^3 stoppt bei jeder Eingabe nach endlicher Zeit.
- Zu jeder M^3 M lässt sich eine endliche Beschreibung $\langle M \rangle$ finden.
- Die M^3 ist ein effektives Berechnungsmodell. Das heißt, die Church-Turing-These besagt, dass Turingmaschinen mindestens so mächtig sind wie die M^3. Anders gesagt: es gibt eine Turingmaschine, die bei Eingabe einer kodierten M^3 $\langle M \rangle$ und einem Wort w die Berechnung von M mit Eingabe w simuliert.

So eine Maschine kann es leider nicht geben. Beweisen Sie das in folgenden zwei Schritten. Geben Sie immer explizit an, wie Sie die oben geforderten Eigenschaften nutzen.

(a) Zeigen Sie, dass die Sprache $L = \{\langle M \rangle \mid M \text{ ist eine M}^3\text{, die } \langle M \rangle \text{ ablehnt}\}$ entscheidbar ist.

(b) Zeigen Sie, dass die M^3 nicht existieren kann.

Aufgabe 7

(2+1+3+2 = 8 Punkte)

Ziel dieser Aufgabe ist es, den Satz von Rice zu beweisen. Dazu gehen Sie nach folgenden Schritten vor.

(a) Betrachten Sie das modifizierte Halteproblem $\mathcal{H}_\varepsilon = \{w \in \{0,1\}^* \mid T_w \text{ hält bei Eingabe } \varepsilon\}$ und zeigen Sie, dass $\mathcal{H}_\varepsilon$ nicht entscheidbar ist.

[2] `https://medium.com/the-mission/d7908013f8c9`

(b) Betrachten Sie die berechenbare Funktion f_∞ für die gilt dass $f_\infty(v)$ undefiniert ist für jedes $v \in \{0,1\}^*$, und zeigen Sie dass o.B.d.A. $f_\infty \notin S$.

(c) Betrachten Sie für jede berechenbare Funktion f eine Turingmaschine $\mathcal{M}_f$ die f realisiert. Zeigen Sie dass eine Turingmaschine $\mathcal{M}^*$ berechnet[3] werden kann, die als Eingabe zwei (als Gödelnummer kodierte) Turingmaschinen $\mathcal{M}_1$, $\mathcal{M}_2$, sowie ein Wort $v \in \{0,1\}^*$ erhält und $\mathcal{M}_1$ mit Eingabe ε, gefolgt von $\mathcal{M}_2$ mit Eingabe v simuliert.

(d) Reduzieren Sie schliesslich das modifizierte Halteproblem $\mathcal{H}_\varepsilon$ auf die Sprache $L(S)$ aus dem Satz von Rice. Zu diesem Zweck konstruieren Sie eine Turingmaschine die $\mathcal{H}_\varepsilon$ entscheidet aus einer hypothetischen Turingmaschine $M_{L(S)}$ die S entscheidet. Benutzen Sie dabei die Turingmaschine $\mathcal{M}^*$ mit Eingabe $\mathcal{M}_{f_\infty}$, $\mathcal{M}_{f^*}$, $v \in \{0,1\}^*$ für ein beliebig aber festes $f^* \in S$.

[3] D.h. es gibt eine Turingmaschine die (unabhängig von der Eingabe) M^* (als Gödelnummer) ausgibt.