9. Übungsblatt

Die sechs Aufgaben zählen als vier zu erreichende PVL-Punkte und zwei Wiederholungsaufgaben. Alle brauchbar gelösten Aufgaben erhöhen also Ihre Punktequote.
Bei allen Turing-Maschinen soll der Anfangszustand z_0 und der Endzustand z_e heißen. Für zunehmend viele Befehle gibt es abnehmend viele Punkte, sonst überprüfe ich wochenlang. Und überhaupt: Überprüfen Sie mal etwas, das (Satz von Rice!) gar nicht entscheidbar ist … ☺

1. Aufgabe

Schreiben Sie das Programm (Zustandsüberführungsfunktion) einer Turing-Maschine *M*, die

- für jedes $n \in \mathbb{N}$ bei der Eingabe einer Folge von *n* Einsen
- als Ausgabe eine Folge von *n* Einsen gefolgt von *n* Nullen berechnet,

und zwar nach folgender Vorgehensweise: Solange möglich, ersetzt *M* die vorderste 1 durch eine 2 und hängt hinten eine 0 an. Wenn keine 1 mehr da ist, ersetzt *M* jede 2 durch eine 1 und geht zum Anfang.

Das Programm soll beginnen mit

$\delta(z_0,1) = (z_1,2,R)$
$\delta(z_1,1) = (z_1,1,R)$
$\delta(z_1,0) = (z_1,0,R)$
$\delta(z_1,B) = (z_2,0,L)$

2. Aufgabe

a) Beschreiben Sie eine Vorgehensweise für eine Turing-Maschine *M'*, die die in Aufgabe 1 beschriebene Funktion berechnet, auf das Band aber nur 0, 1 oder B schreibt, also insbesondere keine 2 benutzt.

b) Schreiben Sie ein Programm (Zustandsüberführungsfunktion) für *M'*.

3. Aufgabe

a) Wenn wir die 0-1-Folgen in Aufgabe (1) als binär geschriebene natürliche Zahlen interpretieren, welcher partiellen Funktion auf den natürlichen Zahlen entspricht die Aufgabenstellung für die gesuchte Turing-Maschine, d.h.
   - Auf welchen Zahlen soll sie mindestens definiert sein?
   - Was wird für jede dieser Zahlen berechnet – möglichst einfach ausgedrückt?

b) Geben Sie die Folge der Konfigurationen der Maschine *M* aus Aufgabe 1 bei der Eingabe 1 an.

4. Aufgabe

Es seien $\Sigma = \{0,1\}$ und

(i) $L_1 = \{000\}$ und $L_2 = \{111\}$.

(ii) $L_1 = \{0^n \mid n \in \mathbb{N}\}$ und $L_2 = \{1^n \mid n \in \mathbb{N}\}$.

Zeigen Sie jeweils, dass das Wortproblem (L_1,Σ^*) auf das Wortproblem (L_2,Σ^*) Turing-reduzierbar ist.

Tipps: Sie müssen jeweils das Programm einer Turing-Maschine angeben, die die Reduktion r berechnet, d.h. Σ^* in Σ^* so abbildet, dass $w \in L_1$ genau dann, wenn $r(w) \in L_2$. Ähnlichkeiten zwischen (i) und (ii) liegen nahe.

5. Aufgabe

Eine Turing-Maschine M mit $\Sigma = \{0,1\}$, $\Gamma = \{1,0,B\}$, Startzustand z_0 und Endzustand z_E macht folgendes:

z_0/z_1: Sie liest 1, löscht die 1 (d.h. überschreibt sie durch B) und geht über eventuelle weitere 1'en nach rechts, bis sie ein B liest. Dann geht sie 1 Zelle nach links.
z_2: Wenn dort B steht, schreibt sie 0 und geht in den Endzustand. Wenn dort 1 steht, löscht sie diese und geht 1 Zelle nach links.
z_3/z_4: Wenn hier nun B steht, schreibt sie 1 und geht in den Endzustand. Wenn hier aber 1 steht, geht sie über eventuelle weitere 1'en nach links, bis sie ein B liest. Dann geht sie 1 Zelle nach rechts und macht weiter bei z_0.
z_e: Im Endzustand tut sie nichts mehr.

a) Auf welchen Eingabeworten liefert M ein Ergebnis? (**Tipp**: (z_0/z_1) beachten)

b) Beschreiben Sie die Funktion, die M bei solchen Eingabeworten berechnet.
Tipp: Mit kurzen Wörtern und der Programm-Beschreibung testen.

c) Ergänzen Sie passend die folgende lückenhaft angegebene Zustandsüberführungsfunktion für M:

$$\delta(z_0,1) = (\quad,\quad,\quad)$$
$$\delta(z_1,1) = (\quad,1,\quad)$$
$$\delta(z_1,B) = (\quad,\quad,\quad)$$
$$\delta(z_2,B) = (\quad,0,\quad)$$
$$\delta(z_2,1) = (\quad,\quad,\quad)$$
$$\delta(z_3,B) = (\quad,1,\quad)$$
$$\delta(z_3,1) = (\quad,1,\quad)$$
$$\delta(z_4,1) = (\quad,1,\quad)$$
$$\delta(z_4,B) = (\quad,\quad,\quad)$$

6. Aufgabe

Gegeben sei ein Programm P_1, welches das Teilsummenproblem für eine Klasse K von Objektmengen konstruktiv löst, d.h. für

- jede geeignet eingegebene endliche Objektemenge $O \in K$,
- jede sog. Wertefunktion w: $O \rightarrow I\!N$ und
- jeden Betrag $b \in I\!N$

entscheidet P_1,

- ob eine Teilmenge $M \subseteq O$ mit $\sum_{x \in M} w(x) = b$ existiert, und
- gibt im Falle JA zusätzlich eine solche Teilmenge in geeigneter Form aus.

Gesucht ist ein Programm P_2, welches das Maximalsummenproblem für eine Klasse K von Objektmengen konstruktiv löst, d.h. für

- jede geeignet eingegebene endliche Objektemenge $O \in K$,
- jede sog. Wertefunktion w: $O \rightarrow I\!N$ und
- jeden Betrag $b \in I\!N$

berechnet P_2

- die maximalen Summe $m \in I\!N$, für die eine Teilmenge $M \subseteq O$ mit $\sum_{x \in M} w(x) = m \leq b$ existiert, (schlimmstenfalls $M = \varnothing$ und $m = 0$ (Wann passiert das?))
- und gibt zusätzlich eine Teilmenge mit maximaler Summe in geeigneter Form aus.

a) Zeigen Sie durch ein grobes Schema für ein P_1 verwendendes Programm P_2, wie das Maximalsummenproblem mit einem linearen Aufwandsfaktor auf das Teilsummenproblem reduziert werden kann.
b) Mit welcher Konstanten c braucht P_2 höchstens c mal so viele Schritte wie P_1?
c) Geben Sie eine möglichst kleine Teilmenge O' von $O = \{1,2,...,62,63\}$ an (d.h. mit möglichst wenigen Elementen), so dass mit $w(o) = o$ für jede natürliche Zahl b mit $1 \leq b \leq 63$ eine Teilmenge M von O' mit Teilsumme $\sum_{x \in M} w(x) = b$ existiert.

Tipp: Nicht erschrecken! Wenn Sie erst einmal die Aufgabenstellung(en) verstanden haben, können Sie die Lösung wahrscheinlich sofort in einer Minute und fünf Zeilen hinschreiben.