11. Übungsblatt (Aufgabe 1-3 für PVL) + freiwillige Wiederholungsaufgaben

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 (Turing-Reduktion)

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.

2. Aufgabe (Turing-Maschine)

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 nach rechts und 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 dort 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 dort 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:

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

3. Aufgabe (Problemreduktion und Komplexität)

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: $O \to I\!N$ 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: $O \to I\!N$ 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 $\sum_{x \in M} w(x) = m \le b$ 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 $1 \le b \le 63$ 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 eventuell sofort in einer Minute und fünf Zeilen hinschreiben. Dadurch, dass Objekte gleiche Werte haben können, geht es mathematisch jeweils um die <u>Multi</u>menge der Werte der Objekte. Mit {1, 3, 3, 9} sind die Teilsummen 1, 3, 4, 6, 7, 9, 10, 12, 13, 15 und 16 bildbar. Die maximale Summe ≤ 11 ist also 10.

Übungen Nr. 7- 12 zur Wiederholung – quer durch den Stoff

Sie zählen nicht zur PVL und werden nicht eingereicht.
Vollständigkeit bezüglich der Klausurthemen wird nicht garantiert, und einige sind umfangreicher als klausurtypisch.

W7 Aufgabe (Pumping-Eigenschaft, kontextfreie Grammatik)

Es seien $\Sigma = \{a,b,c\}$. und $L = \{a^i b^j a^i c^k \mid i, j \geq 0, k \geq 1\}$.

a) Geben Sie ein 3- und ein 5-aufpumpbares Wort in L an, und begründen Sie Ihre Wahl.

b) Geben Sie ein Wort aus L an, das in L nicht 4-aufpumpbar ist, und begründen Sie Ihre Wahl.

c) Geben Sie eine kontextfreie Grammatik für L an.

W8 Aufgabe (Chomsky-Normalform und CYK)

a) Transformieren Sie die Grammatik $G = [\Sigma, V, S, R]$ mit $\Sigma = \{a,b\}$, $V = \{A,B,C\}$ und den unten folgenden Regeln in eine Grammatik in Chomsky-Normalform mit gleicher Sprache:
S → aA | bB, A → Baa | ba | C, B → bCC | ab, C → A

b) Prüfen Sie nach, ob das Wort *bbbaba* zur Sprache von G gehört.

W9 Aufgabe (Kellerautomaten)

Gegeben sei die Sprache $L = \{0^n 1^{2n} \mid n \in I\!N\}$

a) Leiten Sie aus einer kontextfreien Grammatik für L einen Kellerautomaten K_1 mit Kelleralphabet $\Gamma = \{S, 0, 1\}$ ab, der L beschreibt und nicht deterministisch ist.

b) Geben Sie die Übergangsfunktion eines deterministischen Kellerautomaten K_2 mit Kelleralphabet $\Gamma = \{a\}$ an, der die Sprache L beschreibt.

c) Demonstrieren Sie die Berechnung in K_2 für die Eingabe 10 und 01 bis zur Ablehnung und für die Eingabe 001111 bis zum Akzeptieren (in z_e).

W10 Aufgabe (Kellerautomaten)

Gegeben sei die Sprache $L = \{a^m b^n \mid m > n \geq 1\}$.

a) Geben Sie die Übergangsfunktion eines deterministischen Kellerautomaten K mit Kelleralphabet $\Gamma = \{A,a\}$ an, der die Sprache L beschreibt.

Tipps: Der Kellerautomat kann im akzeptierenden Zustand bis zu einer gewissen Zahl weitere *b* akzeptieren, will dies aber beim untersten „a“ im Keller nicht tun, damit er nur *weniger* b’s als a’s akzeptiert. Wie können wir dieses unterste „a“ „markieren“, damit der Kellerautomat es erkennt?

b) Und wie ginge das Ganze mit Kelleralphabet $\Gamma = \{a\}$?

W11 Aufgabe (Turing-Maschinen)

Eine Turing-Maschine mit Ein-/Ausgabe-Alphabet Σ = Bandalphabet $\Gamma = \{a,b,c\}$, Startzustand z_s und Endzustand z_e soll folgendes leisten: Bei jedem eingegebenen Wort *w* der Sprache $(a+b)(a+b)^*$ (also aus *a*'s und *b*'s und mit mindestens einem Zeichen) soll sie zwar jedes Zeichen mit *a* überschreiben, aber jedes dritte Zeichen stattdessen mit *c*. Das so entstandene Wort soll sie als Ergebnis liefern. Beispiel: abababb $\mapsto$ aacaaca.

a) Schreiben Sie eine passende Zustandsüberführungsfunktion δ mit möglichst wenigen Zuweisungen für diese Turing-Maschine.
b) Geben Sie die Folge der Konfigurationen Ihrer Turing-Maschine aus (a) an, wenn sie mit der Eingabe *abb* rechnet.

W12 Aufgabe (Turing-Maschinen als „Unterprogramme")

Es sei *M* eine Turing-Maschine mit der Zustandsmenge *Z*, dem Anfangszustand z_0, dem Endzustand z_e, dem Bandalphabet Γ und der Zustandsüberführungsfunktion δ. Ferner sei bekannt, dass *M* die folgende Funktion f über den natürlichen Zahlen (in Binärdarstellung) „in gutem Stil" (nichts außer Output auf dem Band) berechnet:

$$f(x) = \begin{cases} 1, & \text{wenn } x \text{ durch 187 teilbar ist,} \\ 0 & \text{sonst.} \end{cases}$$

Geben Sie eine Turing-Maschine M_0 an, die die folgende Funktion f_0 über den natürlichen Zahlen (in Binärdarstellung) berechnet:

$$f(x) = \begin{cases} 1, & \text{wenn } x \text{ nicht durch 187 teilbar ist,} \\ 0 & \text{sonst.} \end{cases}$$

Hinweis: Verwenden Sie die Komponenten von *M*.