Universität Duisburg-Essen | **WS 2013/14**
Ingenieurwissenschaften / Informatik | **2. Dezember 2013**
Dozentin: Prof. Barbara König | **Übungsblatt 7**
Übungsleitung: H. Kerstan, S. Küpper | **Abgabe: 09. Dezember 2013**

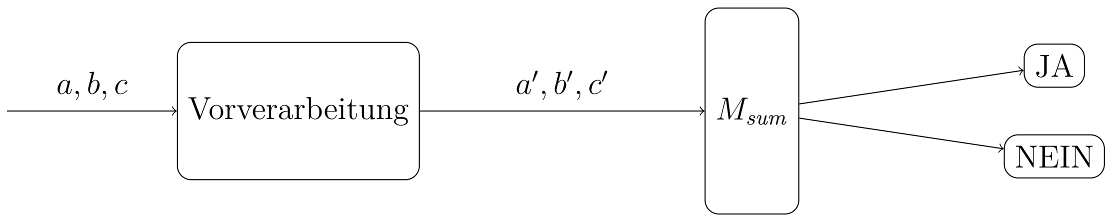
Übung zur Vorlesung BeKo 2013/14

Berechenbarkeit und Komplexität / Theoretische Informatik

Sie können für die 3 Aufgaben auf diesem Übungszettel insgesamt bis zu **20** Punkte erhalten. Genauere Angaben zur Abgabe der Übungszettel finden Sie auf der letzten Seite nach den Aufgaben.

Aufgabe 23 *Reduktionen I* (5 Punkte)

In der Vorlesung wurde das Konzept der *Reduktion* vorgestellt. Da dieses Konzept im weiteren Verlauf der Vorlesung eine wichtige Rolle spielen wird, soll es hier nochmal an einem einfachen Beispiel verdeutlicht werden.
Gegeben seien zwei Maschinen (beliebiger Art, z.B. Java-Programme) M_{sum} und M_{diff} die jeweils ein Tripel $(a, b, c) \in \mathbb{Z}^3$ ganzer Zahlen als Eingabe bekommen. Dabei soll M_{sum} entscheiden, ob $c = a + b$ gilt und M_{diff} enscheiden, ob $c = a - b$ gilt. Zeigen Sie nun (durch eine geeignete Reduktion), dass sich die durch die Maschine M_{diff} berechnete Funktion auf die durch die Maschine M_{sum} berechnete Funktion reduzieren lässt. (5 Punkte)
Dazu geht man wie folgt vor:



Man konstruiert also die Maschine M_{diff}, die die Eingabe (a, b, c) erst umbauen darf zu einem Tripel (a', b', c') und dieses dann als Eingabe an die Maschine M_{sum} weitergibt. Es soll nun eine Vorverarbeitung angegeben werden, die aus (a, b, c) das Tripel (a', b', c') konstruiert, so dass $c = a - b$ genau dann gilt, wenn $c' = a' + b'$ gilt.

Aufgabe 24 *Reduktionen II* (7 Punkte)

Gegeben seien jeweils die Probleme A und B. Zeigen Sie dass sich A auf B reduzieren lässt, d.h. dass gilt: $A \leq B$.

(a) $A = \{x \in \mathbb{Z} \mid x \text{ ist gerade}\}$ und
$B = \{y \in \mathbb{Z} \mid y \text{ ist ungerade}\}$. (1 Punkt)

(b) $A = \{x \in \mathbb{Z} \mid x \mod 11 = 0\}$ und
$B = \{y \in \mathbb{Z} \mid y \mod 121 = 0\}$. (3 Punkte)

(c) $A = \left\{ w \in \{a, b\}^* \mid \#_a(w) = 2 \cdot \#_b(w) \right\}$ und
$B = \left\{ w' \in \{c, d, e\}^* \mid \#_c(w') = \#_d(w') + \#_e(w') \right\}$. (3 Punkte)

(*Hinweis:* Sei Σ ein Alphabet, $a \in \Sigma$ und $w \in \Sigma^*$. Dann gibt $\#_a(w)$ die Anzahl der a's im Wort w an.)

Aufgabe 25 *Entscheidbarkeit* (8 Punkte)

In der Vorlesung wurde gezeigt, dass die Begriffe *rekursive Aufzählbarkeit* und *Semi-Entscheidbarkeit* für Sprachen äquivalent sind. In dieser Aufgabe soll nun gezeigt werden, dass die Begriffe *Aufzählbarkeit in längen-lexikographischer Reihenfolge* und „echte“ *Entscheidbarkeit* für Sprachen ebenfalls äquivalent sind.
Sei $n \in \mathbb{N}, n > 1$ und $\Sigma = \{a_1, \ldots, a_n\}$ ein Alphabet. Wir nehmen an, dass Σ wie folgt (lexikographisch) geordnet ist: $a_1 <_\Sigma a_2 <_\Sigma \ldots <_\Sigma a_n$. Wir erweitern die lexikographische Ordnung $<_\Sigma$ nun wie folgt zu einer Ordnung $<_*$ über Σ^*, der sogenannten *längen-lexographischen Ordnung*. Für beliebige Wörter $v, w \in \Sigma^*$ definieren wir, dass $v <_* w$ genau dann gilt, wenn genau eine der folgenden Eigenschaften[1] zutrifft:

- $|v| < |w|$,
- $|v| = |w|$ und es gibt ein $i \in \{1, ..., |v|\}$ so dass für alle $j \in \mathbb{N}$ mit $j < i$ stets $v_j = w_j$ gilt und zusätzlich $v_i <_\Sigma w_i$.

Wir setzen nun

$$N := \begin{cases} \{0, ..., n-1\}, & \text{falls } L \text{ eine endliche Sprache mit } n \text{ Wörtern ist} \\ \mathbb{N}_0, & \text{falls } L \text{ eine unendliche Sprache ist} \end{cases}$$

und definieren damit, dass eine Sprache $L \subseteq \Sigma^*$ in *längen-lexikographischer Reihenfolge aufzählbar* ist, wenn es eine surjektive, totale und berechenbare Funktion $f\colon N \to L$ gibt, so dass für alle $k, l \in N$ mit aus $k < l$ stets $f(k) <_* f(l)$ folgt. Das heißt, die Funktion f gibt die Wörter der Sprache L in längen-lexikographischer Ordnung (aufsteigend) sortiert aus, $f(0)$ liefert das erste Wort (bezüglich der Ordnung), $f(1)$ liefert das zweite Wort, usw.

(a) Geben Sie an, wie man mit Hilfe der gegebenen Funktion f eine Maschine baut, die die charakteristische Funktion der Sprache L berechnet. (4 Punkte)

(b) Geben Sie an, wie man aus einer gegebenen Maschine für die charakteristische Funktion von L eine Maschine baut, die f berechnet. (4 Punkte)

Hinweise zur Abgabe

Die Hausaufgaben zu diesem Übungsblatt müssen bis spätestens Montag, den 09. Dezember 2013 um 12:00 Uhr abgegeben werden. Eine Bearbeitung der Übungsaufgaben *zu zweit* ist möglich. In diesem Fall geben Sie bitte *nur eine Abgabe (auch bei Moodle!)*

[1] Für ein Wort $v \in \Sigma^*$ bezeichne $|v|$ dabei die Anzahl der Buchstaben des Wortes und es sei $v_i \in \Sigma$ für $i \in \{1, ..., |v|\}$ der i-te Buchstabe des Worts.

ab und schreiben Sie den *Namen* und die *Matrikelnummern beider Personen* auf Ihre Abgabe bzw. in das PDF-Dokument. Ihr Name, Ihre Matrikelnummer, Ihre Gruppennummer und das Fach (BeKo 2013/14) müssen deutlich sichtbar auf die Hausaufgabe geschrieben sein. Sie können Ihre Aufgaben an folgenden Stellen abgeben:

Campus Duisburg:	**Moodle:**
Der mit *Berechenbarkeit und Komplexität / Theoretische Informatik* beschriftete Briefkasten neben Raum LF259.	Sie können ihre Hausaufgaben auch gerne per MOODLE[2] abgeben. Achten Sie aber darauf, dass Sie ihre Hausaufgaben als eine einzelne pdf-Datei hochladen. Wenn ein Betreuer Ihre Abgabe nicht öffnen kann (Formate wie docx o.ä.), bringt das nur unnötige Verzögerungen mit sich.

---

[2] http://moodle2.uni-due.de/course/view.php?id=1338