Name: ....................................

Matr.Nr.: ................................

Stud.Kennz.: .............................

**Klausur "Formale Grundlagen 2"** (326.933)
4.5.2002

---

*Bitte Folgendes beachten:*

- *Es dürfen keine Unterlagen zur Klausur verwendet werden.*
- *Tragen Sie – noch bevor Sie zu arbeiten beginnen — auf dem Angabenblatt Ihren Namen, Matrikelnummer und Studienkennzahl ein. Schreiben Sie auf jedes Blatt, das Sie verwenden, links oben Ihren Namen.*
- *Geben Sie das ausgefüllte Angabenblatt zusammen mit Ihren Lösungen ab. Sie finden das Angabenblatt demnächst im Netz.*
- *Verwenden Sie für jede Aufgabe ein eigenes Blatt (oder mehrere), und geben Sie Ihre Lösungen nach Aufgabennummern geordnet ab, also beginnend mit Aufgabe 1 und endend mit Aufgabe 5.*

---

(1) Sei $L_1$ folgende Sprache über dem Alphabet $\{1,2\}$:

$L_1 = \{w \in \{1,2\}^* \mid$ jedes Anfangsstueck $z$ von $w$ (also $w = zu$ fuer ein $u \in \{1,2\}^*$) enthaelt mindestens so oft das Symbol 1 wie das Symbol 2 $\}$.

Ist die Sprache $L_1$ regulär, also durch einen endlichen Automaten akzeptierbar? Begünden Sie die Antwort.
<u>Lösung:</u> $L_1$ ist nicht regulär. Wäre $L_1$ regulär, dann auch sein Komplement $\overline{L_1}$. Mit Pumping Lemma kann man aber sehen, dass $\overline{L_1}$ nicht regulär ist: $1^n 2^{n+1}$ nicht pumpbar.

(2) Konstruieren Sie eine Turingmaschine, welche die Sprache $L_1$ (siehe (1) ) entscheidet. Die Turingmaschine kann angegeben werden entweder formal durch eine Überführungstabelle, oder durch eine exakte verbale Beschreibung (muss ausführbar sein).
<u>Lösung:</u> Eine zweibändige Turingmaschine liest die Eingabe, und erhöht bei "1" einen Zähler auf Band 2 und verringert bei "2" den Zähler auf Band 2. Sollte dieser Zähler jemals negativ werden, so ist die Eingabe nicht in $L_1$.

(3) Zeigen Sie, dass die folgenden Funktionen primitiv rekursiv sind. Dabei dürfen nur die Grundfunktionen, sowie Komposition und Rekursion verwendet werden.
(a) $f(x, y) = x + y$

(b) $g(x, y) = x \cdot y$
Lösung: Siehe Beispiel 1.5.1 im Skriptum.

(4) Sei $L_4$ die Menge aller Kodes von Turingmaschinen (mit Standardeingabealphabet $\{0, 1\}$), welche das Wort 11011 akzeptieren. Also

$$L_4 \;=\; \{\langle M\rangle \mid 11011 \in L(M)\}.$$

(a) Ist $L_4$ rekursiv?
(b) Ist $L_4$ rekursiv aufzählbar?
(c) Ist $\overline{L_4}$ rekursiv aufzählbar?
Begründen Sie die Antworten.
Lösung:
(a) nicht rekursiv, wegen Satz von Rice
(b) r.a., Verfahren wie folgt: $(i, j) \in \mathbb{N}^2$ aufzählen, $i$-te Maschine $j$ Schritte auf 11011 laufen lassen.
(c) nicht r.a., da sonst $L_4$ rekursiv wäre

(5) Erklären Sie die folgenden Begriffe:
(a) die Komplexitätsklassen DTIME$(n^i)$, NTIME$(n^i)$,
(b) die Komplexitätsklassen $\mathcal{P}$ und $\mathcal{NP}$,
(c) Polynom-Zeit-Reduzierbarkeit und Log-Raum-Reduzierbarkeit,
(d) $\mathcal{NP}$–Vollständigkeit.
Was würde sich für das Problem "$\mathcal{P} = \mathcal{NP}$ ?" ergeben, wenn ein $\mathcal{NP}$–vollständiges Problem in $\mathcal{P}$ wäre?
Lösung: Zu (a,b,c,d) vergleiche Skriptum. Wenn ein $\mathcal{NP}$–vollständiges Problem in $\mathcal{P}$ wäre, dann wäre $\mathcal{P} = \mathcal{NP}$.