

Übungen zur Vorlesung
Grundlagen der Informatik
WS 2010/11
Blatt 7

Wichtige Hinweise:

- > Kennzeichnen Sie Ihre Lösungsabgabe deutlich durch die Nummer des Übungszettels, Ihre Namen, Ihre Matrikelnummern und Ihre Übungsgruppen (Nummern und Zeiten)
- > Beachten Sie die Seite: <http://homepages.fh-regensburg.de/~vok39696/ginf201011.html>

Aufgabe 1:

(Punkte: 3+3=6)

Sei eine beliebige TM $M = (Q, \Sigma, \Gamma, \delta, q_0, \#, F)$ mit $Q = \{q_0, \dots, q_n\}$, $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, \#\}$, $F = \{q_n\}$ und einer Übergangsfunktion δ gegeben.

1. Geben Sie eine geeignete Binärcodierung $\langle M \rangle$ über $\{0, 1\}^*$ für M an, die M eindeutig beschreibt.
2. Demonstrieren Sie Ihre Codierung anhand eines einfachen Beispiels, indem Sie eine beliebige, aber konkrete TM M definieren und $\langle M \rangle$ angeben.

Aufgabe 2:

(Punkte: 3+3=6)

1. Sie haben in der Vorlesung das Halteproblem und das Äquivalenzproblem kennengelernt. Das Halteproblem ist nicht entscheidbar, d.h. es gibt kein Computer-Programm, das alle möglichen Endlosschleifen in einem Programm identifizieren kann. Beim Äquivalenzproblem soll entschieden werden, ob zwei Algorithmen die gleiche Funktion berechnen. Betrachten Sie folgenden vierten Algorithmus zur Berechnung der Summe der ersten n Zahlen:

```
ALGORITHMUS4( $n \in \mathbb{N}$ )
1   $i = 1$ 
2   $s = 0$ 
3  while  $i \leq n$ 
4  do  $s = s + i$ 
5      $i = i + 1$ 
6  Starte beliebiges Programm p mit Eingabe e
7  return s
```

Erklären Sie anhand des Algorithmus, warum auch das Äquivalenzproblem nicht entscheidbar ist. Tipp: Verwenden Sie das Resultat zum Halteproblem.

2. Erläutern Sie, was die beiden folgenden Algorithmen Alg1 und Alg2 berechnen. Bewerten Sie die beiden Algorithmen aus Sicht der Komplexitätstheorie (Zeit/Platz). Welcher Algorithmus ist besser? Begründen Sie Ihre Antworten.

```
int Alg1(int[] z,int n)
{
    int a = -32768,b = 32767;

    for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++)
            if (z[i] >= z[j])
                {
                    if (z[i] >= a) a = z[i];
                    if (z[j] <= b) b = z[j];
                }

    return (a + b);
}
```

```
int Alg2(int[] z, int n)
{
    int a = -32768,b = 32767;

    for (int i = 0; i < n; i++)
        {
            if (z[i] >= a) a = z[i];
        }

    for (int j = 0; j < n; j++)
        {
            if (z[j] <= b) b = z[j];
        }

    return (a + b);
}
```