

## Funktionale Programmierung

### 14. Übungsblatt (Abgabe: Mi., den 10. Februar, um 10:10 Uhr)

Prof. Dr. Margarita Esponda

#### 1. Aufgabe (4 Punkte)

Definieren Sie eine Turing-Maschine, die aus einer beliebigen Binärzahl  $n$  den entsprechenden negativen Zahl ( $-n$ ) berechnet. In der Start-Konfiguration soll der Kopf am Anfang der Binärzahl sich befinden und am Ende der Berechnung soll der Kopf am Anfang des Ergebnisses stehen bleiben.

Beispiel einer mögliche Start- und End-Konfiguration:

" 00101000 "	" 11011000 "
^	^
S <sub>start</sub>	S <sub>end</sub>

#### 2. Aufgabe (6 Punkte)

Definieren Sie eine Turing-Maschine, die als Eingabe zwei Binärzahlen bekommt, die als positive Zahlen interpretiert und miteinander verglichen. Die Vergleichsoperation soll die Semantik folgender Haskell-Funktion entsprechen. Sie können annehmen, dass beide Zahlen die gleiche Bitlänge haben.

```
compare n m | n < m   = "01"
              | n == m = "00"
              | n > m   = "10"
```

Beispiel einer mögliche Start- und End-Konfiguration:

" 00101000<01000111 "	" 00101000<01000111=01 "
^	^
S <sub>start</sub>	S <sub>end</sub>

#### 3. Aufgabe (8 Punkte)

Definieren Sie eine Turing-Maschine, die zwei Binärzahlen addiert und das Ergebnis in das zweite Binärzahl überschreibt. Das bedeutet, dass keinen zusätzlichen Platz außer den Bandbereich, indem die Eingabe steht, verwendet wird. Eine Zweierkomplement-Darstellung der Zahlen wird vorausgesetzt (positive und negative Zahlen). In der Start- und End-Konfiguration soll der Kopf der Maschine am Anfang der ersten Zahl sich befinden.

Beispiel:

" 1011+0111 "	" 1011+0010 "
^	^
S <sub>start</sub>	S <sub>end</sub>

#### 4. Aufgabe (8 Punkte)

Definieren Sie zwei Turing-Maschinen, die als Eingabe für die Minsky-UTM (Universal Turing Maschine) geeignet sein sollten.

- a) Die erste Maschine soll eine Zahl in Unär-Darstellung verdoppeln.
- b) Die zweite soll eine Variante der TM aus der Vorlesung, die die Balancierung von Klammern überprüft.

**Wichtige Hinweise:**

- 1) Für jeder TM soll die formale Beschreibung mit Hilfe des  $(Q, A, B, F, z_0, \text{Leerzeichen}, Z)$ -Tupels (siehe Vorlesungsfolien) abgegeben werden.
- 2) Für alle Turing-Maschinen sollen lauffähige Testfunktionen für den Haskell-Simulator definiert werden.
- 3) Geeignete globale und lokalen konstante Funktionen sollen jeweils für den Zustandstabellen und die dazugehörigen Zuständen definiert werden.
- 4) Ein Zustandsübergangsdiagramm soll für jeder Turing-Maschine abgegeben werden. Bei Bedarf sollen Kommentare über die Aktionen, die innerhalb der verschiedenen Zustände stattfinden mitgeliefert werden.
- 5) Die Abgabe soll elektronisch und im Papierform erfolgen.
- 6) Denken Sie daran, dass die Korrektur von Turing-Maschinen sehr Zeitaufwendig. Aus diesem Grund ist eine möglichst klare Beschreibung Ihrer TM erforderlich. Schlecht beschriebenen TM mit z.B. unübersichtlichen Zustandsdiagrammen oder unübersichtliche Benennung der Zuständen ohne Erläuterungen oder nicht getestete TM werden nicht korrigiert.

Alle TM müssen mit der Haskell-Simulator ausführlich getestet werden. Die Turing-Maschinen für die UTM können dazu noch mit der ins Netz gestelltten Java-Simulator zusätzlich getestet werden.