

# Übungsblatt 7

Abgabe: 18.12.2006

---

## Turing Maschinen

Turing Maschinen (TM) sind fiktive Systeme zur Realisierung von Algorithmen. Aus der theoretischen Informatik wissen wir, dass TM *rekursiv aufzählbare Sprachen* erkennen. Daraus lässt sich folgern, dass – salopp gesprochen – die Lösungen der schwierigsten Probleme, die man gerade noch algorithmisch bewältigen kann, mit TM codiert werden können. Insofern ist es beruhigend sicherzustellen, dass sich eine TM – bis auf die unendliche Länge des Eingabebandes (s.u.) – durch einen konventionellen Computer mit von-Neumann-Organisation realisieren lässt. Definieren wir hierzu formal, was eine TM ist:

**Definition 1** Eine Turing Maschine (TM) ist ein Tupel  $TM = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$  mit folgenden Eigenschaften:

1.  $Q$  ist eine endliche Menge von Kontrollzuständen.
2.  $q_0$  ist der Initialzustand von TM.
3.  $\Sigma$  ist eine endliche Menge von Eingabesymbolen.
4.  $\Gamma$  ist eine endliche Menge von Bandsymbolen, so dass  $\Sigma \subseteq \Gamma$ .
5.  $B \in \Gamma - \Sigma$  ist das Leerzeichensymbol (“BLANK”).
6.  $F \subseteq Q$  ist die Menge der akzeptierenden Zustände.
7. TM liest und schreibt auf ein unendlich langes Band von Symbolen. Lesen und Schreiben erfolgt an der aktuellen Bandposition. Mit Befehlen  $L$  und  $R$  kann die Bandposition um 1 Zeichen nach links oder rechts verschoben werden. Zu Beginn einer Ausführung steht eine endliche Folge von Symbolen ungleich BLANK auf dem Band, die links und rechts von unendlich vielen BLANKs umgeben sind.
8.  $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$  ist die partielle Transitionsfunktion mit folgender Interpretation: Wenn im Kontrollzustand  $q$  an der aktuellen Lese- und Schreibposition des Eingabebandes Symbol  $s \in \Gamma$  gelesen wird und  $\delta(q, s) = (q', s', d)$  gilt, dann
  - (a) erfolgt ein Zustandsübergang in Kontrollzustand  $q'$ ,
  - (b) wird an die aktuelle Bandposition das Symbol  $s'$  geschrieben,
  - (c) rückt die neue Bandposition bei  $d = L$  um 1 nach links, bei  $d = R$  um 1 nach rechts.
9. TM terminiert erfolgreich, wenn im aktuellen Kontrollzustand  $q$  zum gelesenen Symbol  $s$  keine Transition definiert ist ( $(q, s) \notin \text{dom } \delta$ ) und  $q \in F$  gilt.
10. TM terminiert mit Fehler (“die Turing Machine stirbt”), wenn im aktuellen Kontrollzustand  $q$  zum gelesenen Symbol  $s$  keine Transition definiert ist und  $q \notin F$  gilt.

**Aufgabe 1: TM goes Java****(50%)**

Schreiben Sie einen Interpreter für Turing-Maschinen in Java nach folgendem Grundkonzept:

1. Kontrollzustände werden als int-Werte codiert.
2. Eingabeband und *Accepting States* werden durch int-Arrays realisiert.
3. Die Transitionsfunktion wird als 2-Dimensionaler Array realisiert, der Paare (*Kontrollzustand, Eingabewert*) auf Tripel (*neuer Kontrollzustand, an der Bandposition zu schreibender Wert, Leserichtung links/rechts*) abbildet. Die Tripel werden als Instanzen von Java Klassen mit entsprechenden Attributen und eine Konstruktor gebildet, der die Attribute aus Eingangsparametern vorbelegt.
4. Der Interpreter starten mit der Auswertung des Initialzustands 0 und terminiert, sobald der ERROR Kontrollzustand oder ein *Accepting State* erreicht wurde.

**Aufgabe 2: Gerechtigkeit für Nullen und Einsen!****(50%)**

“Programmieren” Sie Ihre TM durch entsprechende Belegung der Programmspeicher-Arrays, um folgendes Problem zu lösen:

1. Auf dem Eingabeband steht eine endliche Folge von Eingabesymbolen 0 oder 1.
2. Die Folge wird auf dem Band durch Leerzeichen abgeschlossen.
3. Die TM soll genau dann erfolgreich terminieren, wenn die Eingabefolge für beliebiges  $n > 0$ 
  - zuerst  $n$  mal 0,
  - dann  $n$  mal 1,
  - dann Leerzeichen

enthält. In allen anderen Fällen “stirbt” die TM in einem Fehlzustand.

Testen Sie Ihr Programm mit folgenden Eingabebändern:

1. { BLANK }
2. { 0,0,0,0,0,1,1,1,1,1,BLANK }
3. { 1,1,1,1,0,0,0,0,BLANK }
4. { 0,0,0,0,1,1,1,1,0,BLANK }
5. { 0,1,0,0,0,0,1,1,1,1,BLANK }

**Hinweise:** Codieren Sie Ihre TM nach folgender informellen Programmbeschreibung:

1. Die TM beginnt mit dem Lesen des Bandes an der ersten Position (Index 0) und wechselt die erste 0 gegen ein X aus. Falls etwas anderes als 0 dort steht, terminiert der Algorithmus mit Fehler. Andernfalls wird bei Schritt 2 fortgesetzt.
2. Nach rechts fortschreitend werden alle 0, X, Y übergangen, bis eine 1 gefunden wird. Diese wird gegen Y ausgetauscht und bei Schritt 3 fortgefahren. Wenn die 1 nicht gefunden wird, sondern man bis BLANK nach rechts fortschreitet, schlägt der Algorithmus fehl.
3. Es wird nach links fortgeschritten, wobei Y und 0 unverändert gelassen wird, bis ein X gefunden wird.
4. Wenn rechts neben dem X eine 0 gefunden wird, wird diese mit X überschrieben und bei 2. fortgefahren. Wenn rechts neben dem X ein Y gefunden wird, erfolgt eine abschließende Prüfung mit Schritt 5. vorgenommen.
5. Es wird nach rechts fortgeschritten, solange man ein Y findet. Endet dieser Vorgang bei dem Wert BLANK, terminiert das Verfahren erfolgreich. Endet dieser Vorgang bei einem anderen Wert als BLANK, schlägt das Verfahren fehl.