

# Aufgaben zum Tut am 28.11.2005

Thomas Pajor

27. November 2005

## Aufgabe 1)

Sei  $\Sigma := \{0, 1\}$  und  $L$  die Sprache aller Palindrome gerader Länge.

- (a) Konstruieren Sie eine deterministische Turingmaschine, die das Wortproblem zu  $L$  entscheidet.
- (b) Geben Sie zu der konstruierten Turingmaschine eine Folge der durchlaufenen Konfigurationen bei der Eingabe  $w := 0110$  an.

## Lösung

- (a) Die Sprache  $L$  wird definiert durch

$$L := \{ww^R \mid w \in \Sigma^*\}$$

Die Turingmaschine soll nun das erste Zeichen  $a$  des Eingabewortes  $w$  einlesen, dieses durch ein  $\sqcup$  ersetzen und bis zum Ende des Wortes fahren. Dort schaut sie ob das letzte Zeichen dem Zeichen  $a$  entspricht. Falls nein soll sie in einem Fehlerzustand halten. Andernfalls löscht sie das letzte Zeichen und fährt an den Anfang des Wortes, wo sie den ganzen Vorgang wiederholt. Ist das komplette Wort durch  $\sqcup$  ersetzt worden, soll die Maschine in einem akzeptierenden Zustand halten.

Um beim „Nach hinten Fahren“ zu merken, welches Zeichen wir am Anfang des Wortes gelesen haben, kann man jeweils zwei verschiedene Zustandsgruppen (eine falls das Zeichen eine 0 und eine falls das Zeichen eine 1 war) verwenden.

Der Graph in Abbildung (1) definiert eine Turingmaschine  $\mathcal{T} := (Q, \Sigma, \Gamma, \delta, s, F)$  bei der gilt  $\mathcal{L}(\mathcal{T}) = L$ .

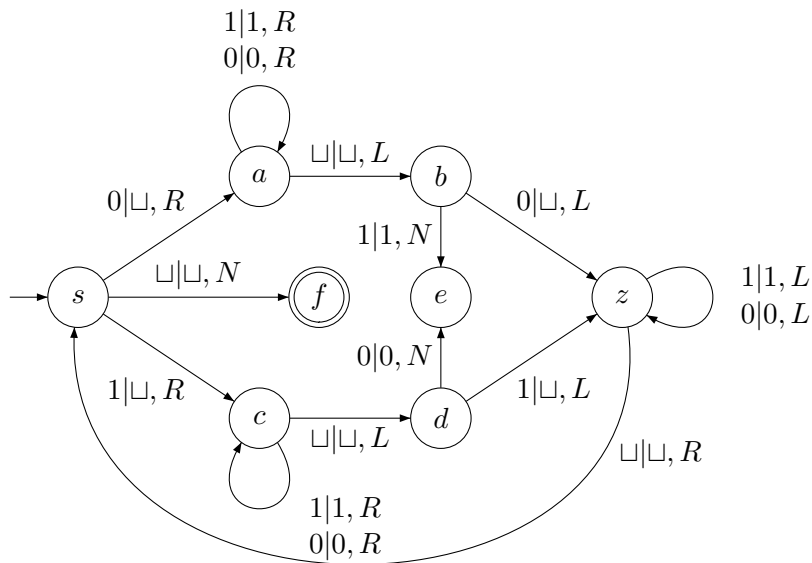


Abbildung 1: Graph der Turingmaschine aus Aufgabe 1

(b) Folgende Konfigurationen werden bei der Eingabe  $w = 0110$  durchlaufen:

$(s)0110 \rightarrow (a)110 \rightarrow 1(a)10 \rightarrow 11(a)0 \rightarrow 110(a)\sqcup \rightarrow 11(b)0 \rightarrow 1(z)1 \rightarrow (z)11 \rightarrow (z)\sqcup11$   
 $\rightarrow (s)11 \rightarrow (c)1 \rightarrow 1(c)\sqcup \rightarrow (d)1 \rightarrow (z)\sqcup \rightarrow (s)\sqcup \rightarrow (f)\sqcup$

## Aufgabe 2)

Seien  $L_1$  und  $L_2$  Sprachen über  $\Sigma$ . Beweisen Sie:

- Die semientscheidbaren Sprachen sind unter Komplementbildung nicht abgeschlossen
- Die semientscheidbaren Sprachen sind unter Vereinigung und Durchschnitt abgeschlossen
- Sind  $L_1$  und  $L_2$  entscheidbar, so ist auch  $L_1 \setminus L_2$  entscheidbar.

## Lösung

- Sei  $L_1$  semientscheidbar. Das heißt, es existiert eine Turingmaschine, die bei Eingabe  $w \in L_1$  hält.

**Annahme:**  $L_1^c$  ebenfalls semientscheidbar. Also gibt es auch eine Turingmaschine, die bei Eingabe  $w \notin L_1$  hält.

$\Rightarrow$  Es gibt eine Turingmaschine die sowohl für  $w \in L_1$  als auch für  $w \notin L_1$  hält.

$\Rightarrow L_1$  ist entscheidbar. Dies ist ein Widerspruch zur Voraussetzung. Also sind die semientscheidbaren Sprachen unter Komplementbildung nicht abgeschlossen.

(b) Seien  $L_1$  und  $L_2$  semientscheidbar und

$$\mathcal{T}_1 := (Q_1, \Sigma_1, \Gamma_1, \delta_1, s_1, F_1) \quad \text{und} \quad \mathcal{T}_2 := (Q_2, \Sigma_2, \Gamma_2, \delta_2, s_2, F_2)$$

die TMs so, dass  $\mathcal{T}_1$  akzeptiert falls  $w \in L_1$  und  $\mathcal{T}_2$  akzeptiert falls  $w \in L_2$ .

Konstruiere nun eine neue 2-Band-, 2-Kopf-Turingmaschine  $\mathcal{T} := (Q, \Sigma, \Gamma, \delta, s, F)$  wie folgt:

- $Q := Q_1 \times Q_2$
- $\Sigma := \Sigma_1 \cup \Sigma_2$
- $\Gamma := \Gamma_1 \cup \Gamma_2$
- $s := (s_1, s_2)$
- Hier unterscheiden wir zwischen „Vereinigung“ und „Durchschnitt“:

$$F := \begin{cases} \{(q_1, q_2) \in Q \mid q_1 \in F_1 \wedge q_2 \in F_2\} & \text{falls } L_1 \cap L_2 \\ \{(q_1, q_2) \in Q \mid q_1 \in F_1 \vee q_2 \in F_2\} & \text{falls } L_1 \cup L_2 \end{cases}$$

- $\delta((q_1, q_2), a, b) := ((p_1, p_2), c, d, X, Y)$  wobei  $\delta_1(q_1, a) = (p_1, c, X)$  und  $\delta_2(q_2, b) = (p_2, d, Y)$ .

Da jede  $n$ -Band-,  $k$ -Kopf Turingmaschine durch eine normale Turingmaschine simuliert werden kann, leistet  $\mathcal{T}$  das Gewünschte.

(c) Seien  $L_1$  und  $L_2$  entscheidbar. Dann sind ihre charakteristischen Funktionen  $\chi_1$  und  $\chi_2$ , gegeben durch

$$\chi_i := \begin{cases} 1 & \text{falls } w \in L_i \\ 0 & \text{sonst} \end{cases} \quad i = 1..2$$

berechenbar.

Also lässt sich eine Turingmaschine  $\mathcal{T}$  bauen die

$$\chi := \begin{cases} 1 & \text{falls } w \in L_1 \wedge w \notin L_2 \\ 0 & \text{sonst} \end{cases}$$

berechnet.

$\Rightarrow L_1 \setminus L_2$  ist entscheidbar.