Prof. Martin Hofmann
Dr. Ulrich Schöpp

Ludwig-Maximilians-Universität München
Institut für Informatik
1. Juli 2011

Formale Sprachen und Komplexität

Blatt 9

Aufgabe 9-1. (3 Punkte) Gegeben sei die Turing-Maschine $M = (Z, \Sigma, \Gamma, \delta, z_0, \Box, E)$ mit $Z = \{z_0, z_1, z_2, z_3, z_4, z_5, z_6, z_7\}$, $\Sigma = \{a, b\}$, $\Gamma = \{a, b, \Box\}$, $E = \{z_7\}$ sowie:

$$\begin{array}{lll}
\delta(z_0, a) = \{(z_1, \Box, R), (z_3, a, N)\} & \delta(z_1, a) = \{(z_2, \Box, R)\} & \delta(z_2, a) = \{(z_7, \Box, N)\} \\
\delta(z_3, a) = \{(z_4, \Box, R)\} & \delta(z_3, \Box) = \{(z_7, \Box, N)\} & \\
\delta(z_4, a) = \{(z_4, a, R)\} & \delta(z_4, b) = \{(z_4, b, R)\} & \delta(z_4, \Box) = \{(z_5, \Box, L)\} \\
\delta(z_5, b) = \{(z_6, \Box, L)\} & & \\
\delta(z_6, \Box) = \{(z_3, \Box, R)\} & \delta(z_6, a) = \{(z_6, a, L)\} & \delta(z_6, b) = \{(z_6, b, L)\}
\end{array}$$

Auf allen weiteren Argumenten soll δ den Wert $\emptyset$ haben.

a) Geben Sie einen Lauf der Turingmaschine an, welcher bezeugt, dass aaa in der akzeptierten Sprache liegt.

b) Geben Sie drei weitere Wörter an, die in der von der Maschine akzeptierten Sprache liegen. Geben Sie für jedes der Wörter einen akzeptierenden Lauf an.

c) Geben Sie drei Wörter an, die nicht in der von der Maschine akzeptierten Sprache liegen.

Aufgabe 9-2. (3 Punkte) In dieser Aufgabe wollen wir eine Turing-Maschine konstruieren, welche die Sprache $\{wcw \mid w \in \{a, b\}^*\}$ akzeptiert.

Die Maschine soll das Eingabealphabet $\Sigma = \{a, b, c\}$ haben und als Arbeitsalphabet $\{a, b, c, \underline{a}, \underline{b}\}$ benutzen. Im Arbeitsalphabet sind $\underline{a}$ und $\underline{b}$ neue Zeichen. Die Notation drückt aus, dass diese Zeichen als speziell markierte Varianten der Buchstaben a und b verstanden werden sollen.

Die Maschine soll folgendermaßen funktionieren:

a) Zuerst prüft sie, dass im Eingabewort genau ein c vorkommt. Ist dies nicht der Fall, so lehnt sie die Eingabe ab.

b) Wenn das Eingabewort ein einziges c ist, akzeptiert die Maschine sofort. Andernfalls bringt sie Markierungen an den ersten Buchstaben der beiden Wörter links und rechts vom c an. Aus $abaacabaa$ wird so zum Beispiel $\underline{a}baac\underline{a}baa$.

c) Nun geht die Maschine das Band von links nach rechts durch und prüft dabei, dass beide markierten Zeichen gleich sind. Wenn die beiden Zeichen gleich sind, wie zum Beispiel in $\underline{a}baac\underline{a}baa$, so fährt die Maschine mit dem Schritt d) fort. Andernfalls, zum Beispiel mit Bandinhalt $\underline{a}baac\underline{b}baa$, hält die Maschine an, so dass die Eingabe nicht akzeptiert wird.

d) Die Maschine geht nun das gesamte Wort durch und verschiebt die Markierungen ein Zeichen nach rechts. Aus dem Bandinhalt $\underline{a}baac\underline{a}baa$ wird so $a\underline{b}aaca\underline{b}aa$.

   Können beide Markierungen nicht mehr nach rechts verschoben werden, wie zum Beispiel in $aba\underline{a}caba\underline{a}$, so akzeptiert die Maschine die Eingabe. Kann nur eine Markierung nicht verschoben werden, z.B. mit Bandinhalt $ab\underline{a}cab\underline{a}b$ oder $ab\underline{a}bcab\underline{a}$, so lehnt die Maschine die Eingabe ab.

   Konnten die Markierungen verschoben werden, so fährt die Maschine nun wieder mit Schritt c) fort.

Dieses Verhalten der Punkte a) und b) kann man wie folgt implementieren:

a) Anfangs steht der Arbeitskopf auf dem linkesten Zeichen der Eingabe. Fahre dann nach rechts bis ein c gelesen wird. Wenn kein c im Wort vorkommt wird abgelehnt:

$$\delta(z_0, a) = (z_0, a, R) \quad \delta(z_0, b) = (z_0, b, R) \quad \delta(z_0, c) = (z_1, c, R) \quad \delta(z_0, \square) = (z_{\text{fail}}, \square, N)$$

Fahre dann weiter nach rechts bis zum Ende des Worts. Wenn ein weiteres c kommt wird abgelehnt:

$$\delta(z_1, a) = (z_1, a, R) \quad \delta(z_1, b) = (z_1, b, R) \quad \delta(z_1, c) = (z_{\text{fail}}, c, N) \quad \delta(z_1, \square) = (z_2, \square, L)$$

Fahre nun den Arbeitskopf zurück zum Anfang des Worts:

$$\delta(z_2, a) = (z_2, a, L) \quad \delta(z_2, b) = (z_2, b, L) \quad \delta(z_2, c) = (z_2, c, L) \quad \delta(z_2, \square) = (z_3, \square, R)$$

b) Behandle den Fall, dass die gesamte Eingabe ein einziges c ist:

$$\delta(z_3, c) = (z_4, c, R) \qquad \delta(z_4, \square) = (z_{\text{accept}}, \square, N)$$

Markiere sonst den Anfangsbuchstaben im ersten Wort:

$$\delta(z_3, a) = (z_5, \underline{a}, R) \qquad \delta(z_3, b) = (z_5, \underline{b}, R)$$

Fahre den Arbeitskopf zum Anfang des zweiten Worts:

$$\delta(z_5, a) = (z_5, a, R) \qquad \delta(z_5, b) = (z_5, b, R) \qquad \delta(z_5, c) = (z_6, c, R)$$

Markiere das erste Zeichen des zweiten Worts:

$$\delta(z_6, a) = (z_7, \underline{a}, L) \qquad \delta(z_6, b) = (z_7, \underline{b}, L) \qquad \delta(z_6, \square) = (z_{\text{fail}}, \square, N)$$

Fahre zurück zum Anfang des Worts:

$$\delta(z_7, a) = (z_7, a, L) \qquad \delta(z_7, b) = (z_7, b, L) \qquad \delta(z_7, c) = (z_7, c, L)$$
$$\delta(z_7, \underline{a}) = (z_7, \underline{a}, L) \qquad \delta(z_7, \underline{b}) = (z_7, \underline{b}, L) \qquad \delta(z_7, \square) = (z_8, \square, R)$$

In dieser Beschreibung soll z_{accept} ein akzeptierender Zusand sein und z_{fail} ein nichtakzeptierender. Hat die Maschine einmal diese Zustände erreicht, so verbleibt sie darin und führt keine Bewegungen, Schreib- oder Leseoperationen mehr durch. Oben nicht erwähnte Zustandsübergänge sollen in den Zustand z_{fail} gehen.

Vervollständigen Sie analog die Implementierung der noch fehlenden Schritte c) und d), entsprechend der obigen Beschreibung. Erklären Sie kurz ihre Konstruktion. Geben Sie alle Komponenten des Septupels, das Ihre Maschine definiert, an.

Aufgabe 9-3.

- Ein Kellerautomat akzeptiert ein Wort, wenn es einen Lauf gibt, der dieses Wort liest und mit einem leeren Keller endet. Man könnte Kellerautomaten auch anders definieren, so dass die Akzeptanz durch Endzustände gegeben ist. Ein *Endzustands-Kellerautomat* $M = (Z, \Sigma, \Gamma, \delta, z_0, \#, E)$ besteht aus den gleichen Komponenten wie ein Kellerautomat sowie einer Menge $E \subseteq Z$ von Endzuständen. Die Sprache von M ist definiert durch:

  $$L(M) = \{w \mid (z_0, w, \#) \vdash^* (z, \varepsilon, \gamma) \text{ für ein } z \in Z \text{ und } \gamma \in \Gamma^*\}$$

  Zeigen Sie, dass Endzustands-Kellerautomaten ebenfalls genau die kontextfreien Sprachen erfassen.

  Hinweis: Konstruieren Sie zu jedem Kellerautomaten einen Endzustands-Kellerautomaten mit der gleichen Sprache und umgekehrt.

- Die kontextfreien Sprachen sind nicht unter Durchschnitt abgeschlossen. Zeigen Sie, dass jedoch der Durchschnitt einer kontextfreien Sprachen mit einer regulären Sprache stets wieder kontextfrei ist. Zeigen Sie dies, indem sie aus einem gegebenen NFA M und einem gegebenen Kellerautomaten N einen neuen Kellerautomaten D mit $L(D) = L(M) \cap L(N)$ konstruieren.

Abgabe: Sie können ihre Lösungen bis Freitag, den 08.07., um 18:00 Uhr im Abgabekasten in der Theresienstraße oder über UniWorX abgeben. In UniWorX werden Dateien im `txt`-Format (reiner Text) oder im `pdf`-Format akzeptiert.