

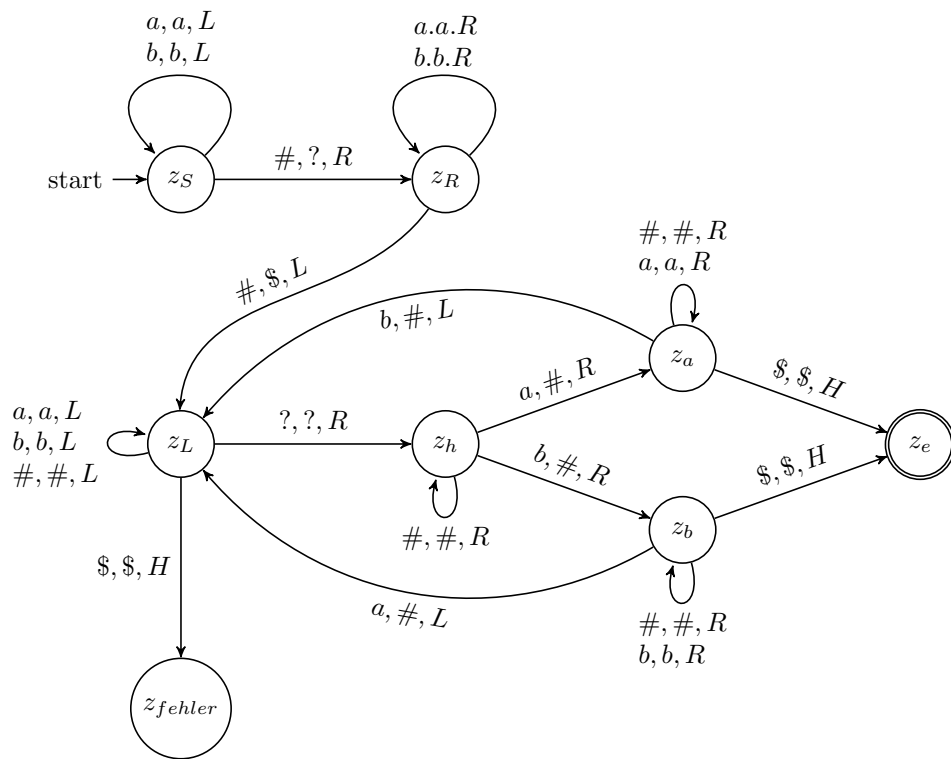
FGI I Übung, Aufgabenblatt 5, Übungsaufgaben
 von Hauke Stieler am 12. Mai 2015

Aufgabe 5.4

Angabe der Turingmaschine T :

$$\begin{aligned}
 T &= (Z, \Sigma, \Gamma, \delta, z_S, Z_{end}) \\
 Z &= \{z_S, z_R, z_L, z_h, z_a, z_b, z_e, z_{fehler}\} \\
 \Sigma &= \{a, b\} \\
 \Gamma &= \{a, b, \$, ?\} \\
 \delta &= [Aus Diagramm zu entnehmen] \\
 Z_{end} &= \{z_e\}
 \end{aligned}$$

Zustandsdiagramm von T :



Begründung

Die Turingmaschine T markiert zunächst den Anfang des Wortes mit einem $?$ und das Ende mit einem $\$$. Danach bewegt sich der LSK zurück zum anfang und markiert in z_h das erste Zeichen von nach dem $?$.

Nun werden so lange $\#$ -Symbole übersprungen, bis das erste zeichen $u \in \Sigma$ auftaucht. Es wird dann mit a in z_a und mit b in z_b gewechselt und das jeweilige zeichen vom Band gelöscht.

Nun wird in z_a nach dem ersten b gesucht und in z_b nach dem ersten a in w . Wird ein passendes Zeichen gefunden wird dieses ebenfalls gelöscht und es wird in z_L von vorne angefangen.

Sollte ein a oder b gelesen und gelöscht werden, jedoch kein geeignetes "Gegenzeichen" gefunden werden und somit ein $\$$ gelesen werden, so endet T un z_e , da nur noch a -s oder b -s auf dem Band sind und die Bedingung $|w|_a \neq |w|_b$ gilt.

Als Algorithmus:

- (1)/ z_S Gehe so lange nach links, bis das Wort zuende ist. Markiere den Anfang mit $?$.
- (2)/ z_R Gehe so lange nach rechts, bis das Wort zuende ist. Markiere das Ende mit $\$$.
- (3)/ z_L Bewege den LSK so lange nach links, bis er auf dem $?$ steht und bewege ihn einen Schritt wieder nach rechts. Nun ist steht er auf dem ersten Zeichen von w oder einem $\#$.
Wenn ein $\$$ gelesen wird heißt das, dass das Wort vollständig gelöscht wurde. Ergo gilt $|w|_a \neq |w|_b$, die Maschine, endet in einem Fehlerzustand.
- (4)/ z_h Überspringe alle $\#$ -Symbole. Wenn ein a vorkommt, gehe in (5.a), bei einem b in (5.b). Lösche das jeweilige a oder b .
- (5.a)/ z_a Ein a wurde gelesen, nun suchen wir das nächste b . Überspringe alle a -s und $\#$ -Symbole. Wenn ein b vorkommen, lösche dieses und mache in (3) weiter.
Wenn ein $\$$ vorkommt, heißt das, dass es mehr a -s als b -s gibt, die Maschine hält also an.
- (5.b)/ z_b Ein b wurde gelesen, nun suchen wir das nächste a . Überspringe alle b -s und $\#$ -Symbole. Wenn ein a vorkommen, lösche dieses und mache in (3) weiter.
Wenn ein $\$$ vorkommt, heißt das, dass es mehr a -s als b -s gibt, die Maschine hält also an.

Aufgabe 5.5

$$\begin{aligned}
 G &= \{V_N, V_T, P, C\} \\
 V_N &= \{A, B, C\} \\
 V_T &= \{a, b, c\} \\
 P &= \{C \rightarrow BCC, c, \lambda \\
 &\quad B \rightarrow ABB, b, \lambda \\
 &\quad A \rightarrow Aa, a, \lambda\}
 \end{aligned}$$

Beweis L_1 wird von G erzeugt:

Sei $w \in L_1$, so hat w die Form $w = a^{i+j}b^{j+k}c^k$.

Man kann w nun unterteilen in $w = w_1 \cdot \dots \cdot w_5$ mit den Unterworten $w_1 = a^i, w_2 = a^j, w_3 = b^i, w_4 = b^k, w_5 = c^k$.

Das Wort $w_5 = c^k$ kann durch beliebig häufiges hintereinander ausführen von $C \rightarrow BCC$ und anschließendes $C \rightarrow bc$ produziert werden. Gleichzeitig kann man dann auch $w_4 = b^k$ erzeugen, da man $B \rightarrow b$ anwenden kann.

Analog kann man obige Argumentation für $w_3 = b^i$ und $w_1 = a^i$ anwenden. Beide kann man aus den Produktionen von B und A erzeugen.

Mit $w_2 = a^j$, ist es ganz einfach, denn hier kann man $A \rightarrow Aa, a$ j -mal anwenden und erhält das gewünschte Wort.

Somit ist L_1 die von G generierte Sprache. □

Beweis G erzeugt L_1 :

Sei w ein von G erzeugtes Wort.

Wenn man $C \rightarrow CBB$ k -mal hintereinander ausführt, so erhält man $B^k C^{k+1}$, aus dem man $b^k c^k$ erzeugen kann. Zudem kann man $Bb^k c^k$ erzeugen.

Aus dem B kann man dann $A^{i-1} B b^{i-1}$ und somit $A^i b^i$ erzeugen.

Mit A^i kann man nun entweder a^i oder $a^j a^i$ erzeugen, sprich a^i und noch mehr.

Somit haben wir die Teilworte $b^k c^k, b^i, a^i$ und a^j , die zusammen $a^{i+j} b^{i+k} c^k$ ergeben und somit in L_1 liegen. □

Aufgabe 5.6

5.6.1 Zeigen, dass $L(M) = L(A)$ gilt:

Nehmen wir an, dass die TM M eine Maschine mit zwei Schreib- & Lesebändern ist.

Man kann nun auf einem Band das Wort lesen, wobei nur das rechts gegangen und nicht geschrieben werden darf

Auf dem zweiten Band kann der Keller implementiert werden. Dabei darf nur nach den bekannten Regeln verfahren werden: Nur ein element lesen/löschen, LSK kann sich nicht frei bewegen.

5.6.2 TM für L beschreiben:

Man kann eine TM der Einfachheits halber mit zwei Bändern bilden. Auf einem Band steht das Wort, auf dem anderen wird geschrieben.

Die TM liest nun alle a -s und schreibt ein Symbol (z.B. x) auf das Band. Wenn alle a -s gelesen wurden, stehen nun genau so viele x -e auf dem Band, wie a -s gelesen wurden. Der LSK vom Wort-Band steht nun auf dem ersten b , während der LSK vom Schreib-Band auf dem letzten x steht.

Nun werden alle b -s gelesen. Für jedes b - wird nun auch ein x gelesen und der Kopf dementsprechend nach links weiter bewegt.

Kommt das erste c auf dem Wort-Band, so muss der LSK vom Schreib-Band auf einem $\#$ stehen. Wenn er sich dann einen Schritt nach Rechts bewegt, muss er folglich auf einem x stehen. Ist beides nicht der Fall, kann bereits abgebrochen werden, da entweder mehr oder weniger b -s als a -s auf dem Band stehen.

Das gleiche wird nun mit allen c -s gemacht, auch hier wird für jedes c ein x gelesen. Am Ende muss der LSK auf zwei $\#$ Symbolen stehen.

Wenn der LSK vom Schreib-Band einen nach Links geht und auf einem x steht, bedeutet das, dass das Wort $a^n b^n c^n$ gelesen wurde.