

Lösungsvorschlag zur Übung 8 zur Vorlesung
Theoretische Informatik für Studierende der Medieninformatik

TIMI8-1 LBAs

(0 Punkte)

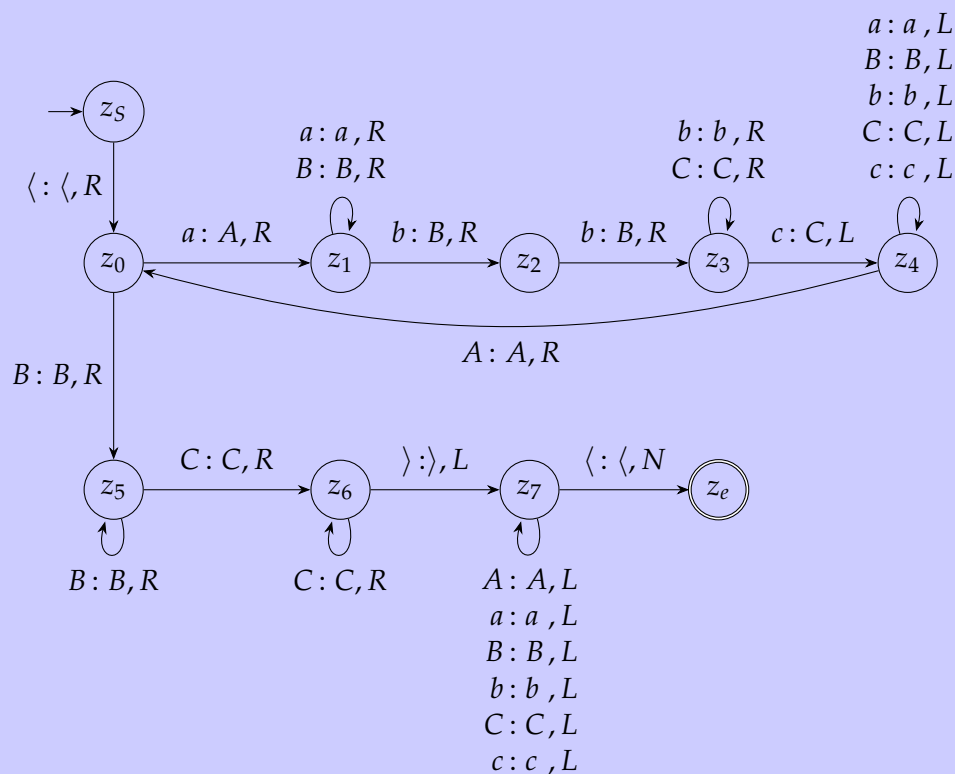
Wir betrachten die Sprache $L = \{a^n b^{2^n} c^n \mid n \in \mathbb{N}_{>0}\}$ über dem Alphabet $\{a, b, c\}$.

- a) Geben Sie einen Zustandsgraphen für einen LBA an, der L erkennt.

LÖSUNGSVORSCHLAG:

LBA: <http://turingmachinesimulator.com/shared/odgebpeprk>

Hierbei müssen Sie manuell $\langle$ anstelle $\langle$ und $\rangle$ anstelle $\rangle$ um das Eingabewort schreiben.



- b) Geben Sie eine Definition dafür an, dass ein LBA "deterministisch" ist in dem Sinne wie wir es für die anderen Konstrukte dieser Vorlesung getan haben.

LÖSUNGSVORSCHLAG:

Ein LBA $M = (Z, \Sigma, \Gamma, \delta, z_0, \langle, \rangle, E)$ ist deterministisch g.d.w. $|\delta(z, a)| = 1$ für alle $z \in Z, a \in \Gamma$.

- c) Geben Sie eine Definition dafür an, dass ein nach Ihrer Definition aus b) deterministischer LBA eine Funktion berechnet. Orientieren Sie sich dabei an der entsprechenden Definition für deterministische Turingmaschinen.

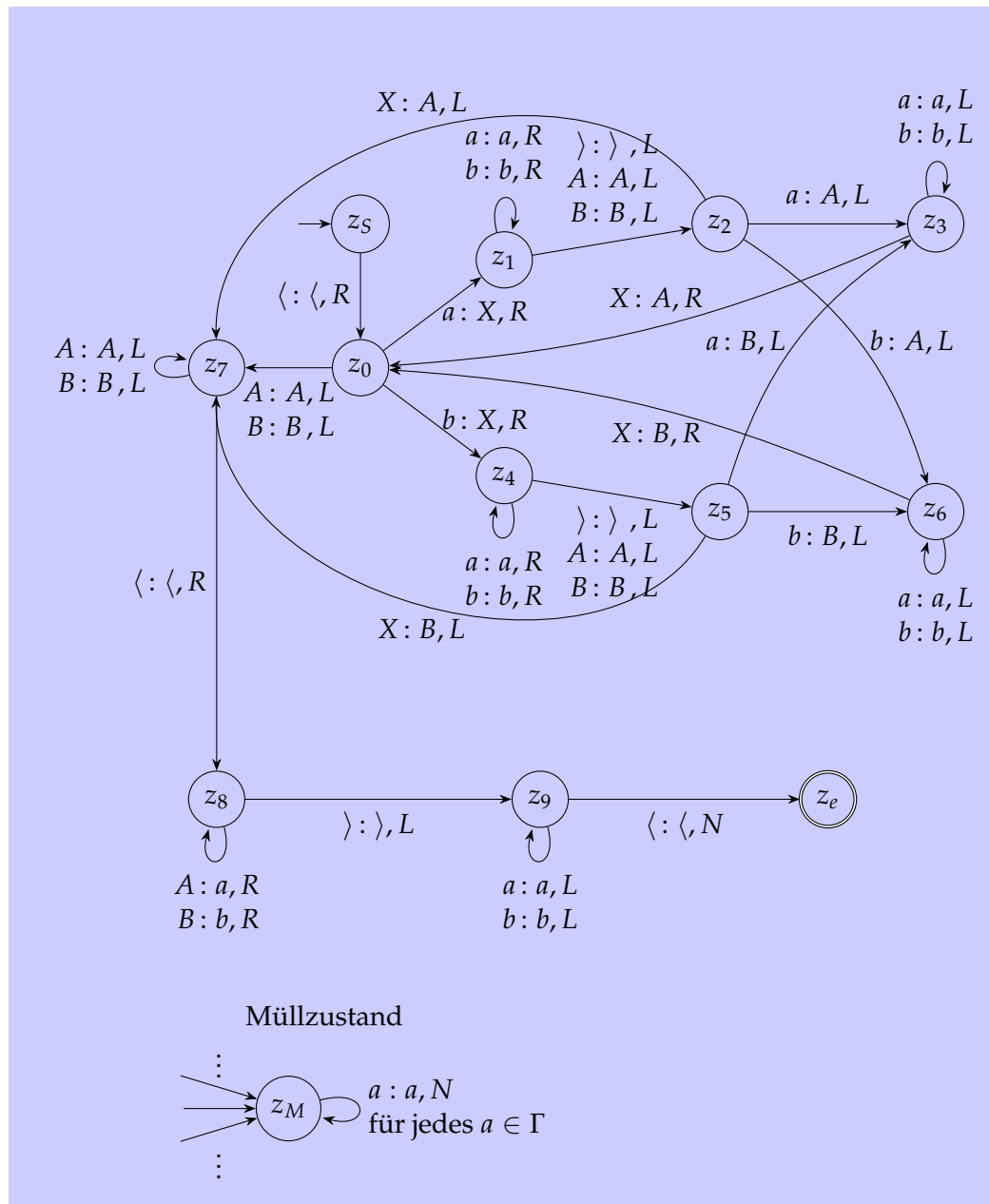
LÖSUNGSVORSCHLAG:

Ein LBA $M = (Z, \Sigma, \Gamma, \delta, z_0, \langle, \rangle, E)$ berechnet eine Funktion $f : \Sigma^* \rightarrow \Sigma^*$, wenn für alle $u, v \in \Sigma^*$ gilt: $f(u) = v$ g.d.w.

$$z_0 \langle u \rangle \vdash^* \square \cdots \square z_e \langle v \rangle \square \cdots \square \text{ mit } z_e \in E.$$

- d) Geben Sie den Zustandsgraphen eines nach Ihrer Definition aus b) deterministischen LBA an, der nach Ihrer Definition aus c) die Funktion $f(w) = \bar{w}$ über dem Alphabet $\Sigma = \{a, b\}$ berechnet.

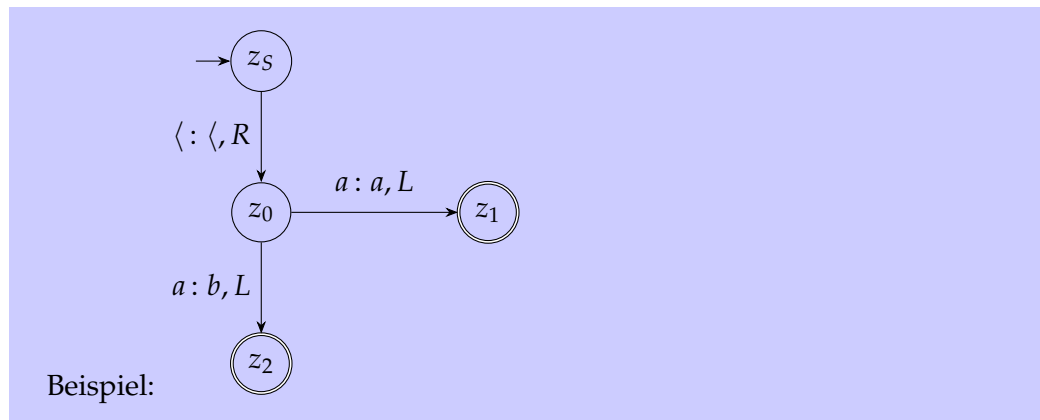
LÖSUNGSVORSCHLAG:



- e) Warum ist es für die Definition aus c) notwendig, dass der LBA deterministisch ist?

LÖSUNGSVORSCHLAG:

Ein nichtdeterministischer LBA könnte eine "Funktion" definieren, welche für denselben Input mehrere Outputs hat. Dies ist dann aber keine Funktion im mathematischen Sinne.



TIMI8-2 Abschlusseigenschaften von Kontextfreien Sprachen

(2 Punkte)

Die kontextfreien Sprachen sind unter kleeneschem Abschluss, Produkt und Vereinigung mit kontextfreien Sprachen, sowie unter Schnitt mit regulären Sprachen abgeschlossen.

Zeigen Sie mithilfe dieser Abschlusseigenschaften, dass die Sprache

$$L_1 = \{w \mid \#_a(w) = \#_b(w) = \#_c(w) + 1\}$$

über dem Alphabet $\Sigma = \{a, b, c\}$ nicht kontextfrei ist. Sie dürfen annehmen, dass die Sprache $L_2 = \{a^n b^n c^n \mid n \in \mathbb{N}_{>0}\}$ nicht kontextfrei ist.

LÖSUNGSVORSCHLAG:

Widerspruchsbeweis: Nimm an, dass L_1 kontextfrei ist. Wir definieren die Sprache

$$L'_1 = L(a^* b^* c^*) \cap L_1 = \{a^n b^n c^{n-1} \mid n \in \mathbb{N}_{>0}\}$$

L'_1 ist kontextfrei, denn $L(a^* b^* c^*)$ ist eine reguläre Sprache, da sie von einem regulären Ausdruck erzeugt wird, und die kontextfreien Sprachen sind unter Schnitt mit regulären Sprachen abgeschlossen.

Nun definieren wir

$$L''_1 = L'_1 \cdot \{c\} = \{a^n b^n c^n \mid n \in \mathbb{N}_{>0}\}$$

L''_1 ist kontextfrei, denn $\{c\}$ ist regulär und damit kontextfrei und die kontextfreien Sprachen sind unter Produkt abgeschlossen.

Allerdings wissen wir, dass $L''_1 = L_2$ nicht kontextfrei ist. Die Annahme, dass L_1 kontextfrei sei, führt somit zu einem Widerspruch.

TIMI8-3 Turingmaschinen

(0 Punkte)

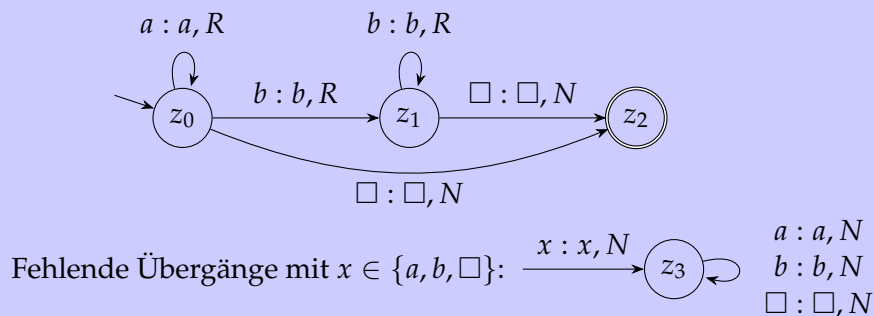
- a) Seien $M = \{w \mid \#_a(w) = \#_b(w)\}$ und $N = L(a^*b^*)$ Sprachen über $\Sigma = \{a, b\}$. Zeigen Sie, dass es eine Funktion f gibt mit $\forall x \in \Sigma^* . x \in M \iff f(x) \in N$.

LÖSUNGSVORSCHLAG:

$$f(x) = \begin{cases} ab & \text{wenn } \#_a(x) = \#_b(x) \\ ba & \text{sonst} \end{cases}$$

- b) Geben Sie eine deterministische Turingmaschine an, welche N erkennt.

LÖSUNGSVORSCHLAG:



- c) Was muss für f gelten, damit man daraus folgern kann, dass M von einer Turingmaschine erkannt werden kann (ohne die Turingberechenbarkeit von M direkt zu zeigen)?

LÖSUNGSVORSCHLAG: f muss turingberechenbar sein. Sei F die passende Turingmaschine. Dann kann man M erkennen, indem man zunächst F ausführt und danach die Turingmaschine für N ausführt. Wie wir im Laufe der Vorlesung sehen werden, nennt man f in diesem Fall eine Reduktionsfunktion, für eine Reduktion von M auf N .