

PD Dr. Jan Johannsen
Elisabeth Lempa
Luca Maio

Ludwig-Maximilians-Universität München
Institut für Informatik
Besprechung 18.06.2026 bis 22.06.2026
Abgabe bis 30.06.2026, 14:00 Uhr

Lösungsvorschlag zur Übung 8 zur Vorlesung
Formale Sprachen und Komplexität

FSK8-1 Entscheiden des Wortproblems für Sprachen von Typ 1

Sei $G = (V, \{a, b, c\}, P, S)$ eine Grammatik mit Produktionen

$$\begin{aligned} P = \{ & S \rightarrow SabS \mid Ac \mid cB, \\ & Ac \rightarrow ccA, \\ & cA \rightarrow ca, \\ & AA \rightarrow aA, \\ & Aa \rightarrow ab, \\ & cB \rightarrow Bcc, \\ & Bc \rightarrow bc, \\ & BB \rightarrow Bb, \\ & bB \rightarrow ab \} \end{aligned}$$

- a) Berechnen Sie gemäß der Konstruktion aus der Vorlesung L_i^6 für alle $i \in \mathbb{N}$ und begründen Sie kurz.
- b) Geben Sie zwei Wörter an, die auf Grundlage der Berechnung in der Sprache $L(G)$ sind.
- c) Geben Sie drei Wörter an, die auf Grundlage der Berechnung nicht in der Sprache $L(G)$ sind.

LÖSUNGSVORSCHLAG:

a)

$$L_0^6 := \{S\}$$

$$L_1^6 := L_0^6 \cup \{SabS, Ac, cB\},$$

da $S \rightarrow SabS \mid Ac \mid cB$

$$L_2^6 := L_1^6 \cup \{AcabS, cBabS, SabAc, SabcB, ccA, Bcc\},$$

da $S \rightarrow Ac \mid cB, Ac \rightarrow ccA, cB \rightarrow Bcc$

$$L_3^6 := L_2^6 \cup \{AcabAc, AcabcB, ccAabS, cBabAc, SabccA, cBabcB, BccabS, SabBcc, cca, bcc\},$$

da $S \rightarrow Ac \mid cB, Ac \rightarrow ccA, cA \rightarrow ca, Bc \rightarrow bc$

$$L_4^6 := L_3^6 \cup \{ccaabS, ccabbS, Sabcca, bccabS, Sabbcc, Saabcc\},$$

da $Aa \rightarrow ab, bB \rightarrow ab, cA \rightarrow ca, Bc \rightarrow bc$

$$L_i^6 := L_4^6$$

für $i \geq 4$, da alle anwendbaren Regeln zu Satzformen mit Länge größer 6 führen.

b) cca, bcc

c) a, b, c

FSK8-2 LBAs

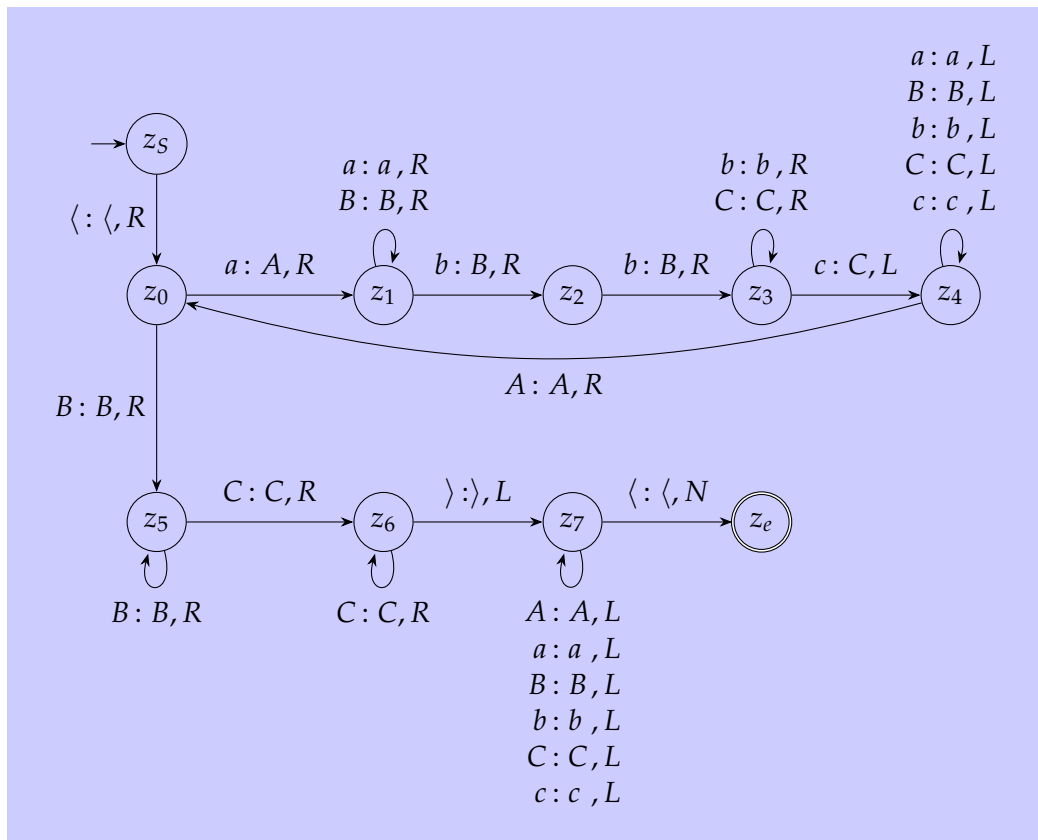
Wir betrachten die Sprache $L = \{a^n b^{2^n} c^n \mid n \in \mathbb{N}_{>0}\}$ über dem Alphabet $\{a, b, c\}$.

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

LÖSUNGSVORSCHLAG:

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

Hierbei müssen Sie manuell $<$ anstelle $\langle$ und $>$ 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:

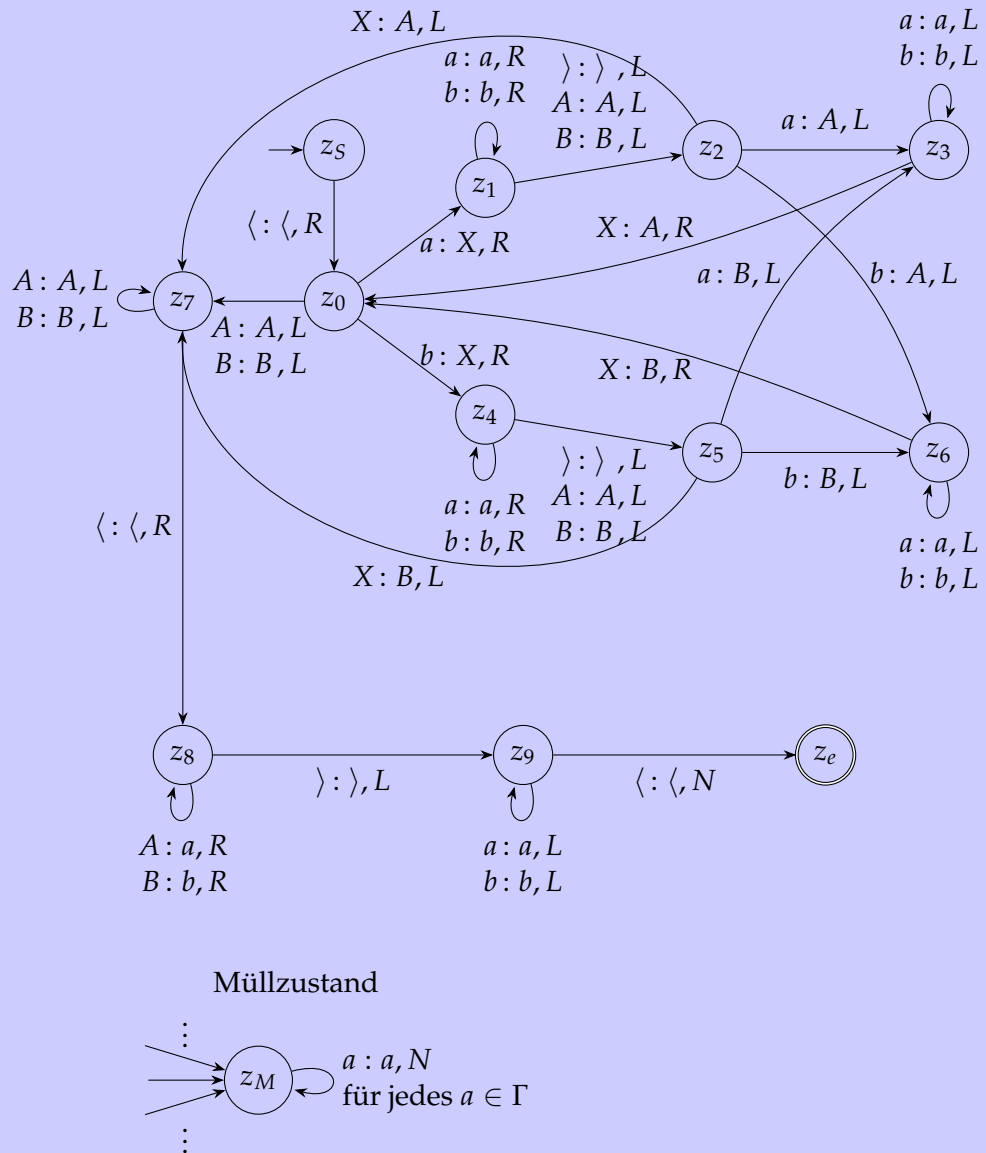
Ein LBA $M = (Z, \Sigma, \Gamma, \delta, z_0, \langle, \rangle, E)$ berechnet eine Funktion $f: \Sigma^* \rightarrow \Sigma^*$, wenn für alle $u, v \in \Sigma^*$ gilt: $f(u) = v$ g.d.w.

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

- 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.

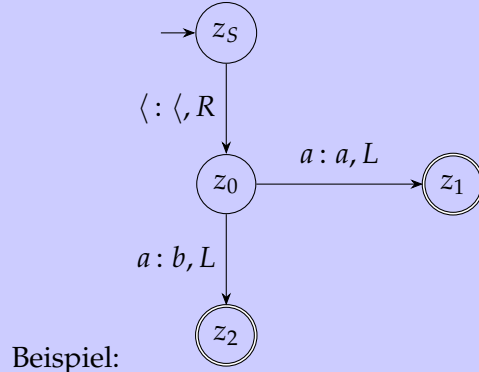
LÖSUNGSVORSCHLAG:



- 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.

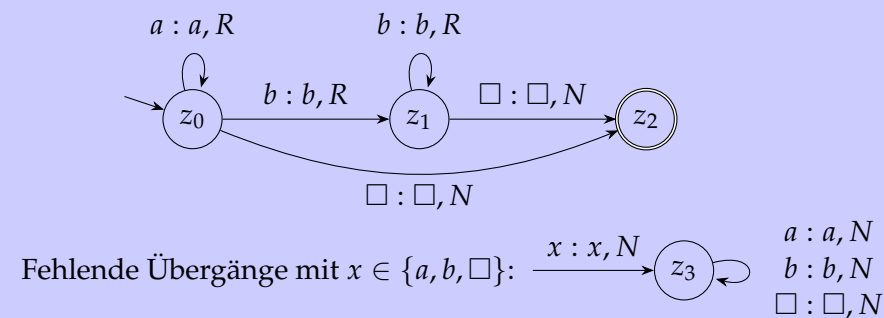
**FSK8-3 Turingmaschinen**

- 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 .

FSK8-4 LOOP-Programme

Für LOOP-Programme sind ausschließlich folgende Konstrukte erlaubt:

$$\begin{aligned}x_i &:= x_j + c \\x_i &:= x_j - c \\x_i &:= c \\P_1; P_2 \\ \mathbf{LOOP} \ x_i \ \mathbf{DO} \ P \ \mathbf{END}\end{aligned}$$

- a) Schreiben Sie ein LOOP-Programm, das die folgende Funktion berechnet:

$$f : \mathbb{N}^2 \rightarrow \mathbb{N}, \quad f(x, y) = 3 * (x - y)$$

LÖSUNGSVORSCHLAG:

Gemäß der Definition von LOOP-Berechenbarkeit erhalten wir die Eingangs-

ben x in x_1 und y in x_2 und speichern die Ausgabe in x_0 .

```
// Initialisiere  $x_0$  als  $x$ .  
 $x_0 := x_1 + 0$ ;  
// Subtrahiere  $y$ :  
LOOP  $x_2$ DO  
     $x_0 := x_0 - 1$   
END;  
 $x_4 := x_0 + 0$ ;  
 $x_5 := 2$ ;  
// 2-mal, da bereits einmal in  $x_0$ :  
LOOP  $x_5$ DO  
    // Addiere alten Wert von  $x_0$  zum Ergebnis:  
    // Initialisiere  $x_3 = \text{Altes } x_0$   
     $x_3 := x_4 + 0$ ;  
    // Altes  $x_0$ -mal:  
    LOOP  $x_3$ DO  
        // Inkrementiere das Ergebnis  
         $x_0 := x_0 + 1$   
    END  
END
```

- b) Schreiben Sie ein LOOP-Programm, das den folgenden Pseudocode implementiert:

```
FOR  $x_i := 0$  TO  $x_1$  DO  $x_0 := x_0 + x_i$  END
```

LÖSUNGSVORSCHLAG:

```
// Initialisiere  $x_i$  mit 0:  
 $x_i := 0$ ;  
//  $x_1$  mal:  
LOOP  $x_1$  DO  
  // Addiere  $x_i$  zu  $x_0$ :  
  LOOP  $x_i$  DO  
     $x_0 := x_0 + 1$   
  END;  
  // Inkrementiere  $x_i$ :  
   $x_i := x_i + 1$   
END;  
// Addiere ein letztes mal  $x_i$  zu  $x_0$ , da wir es zuletzt  
// mit dem Wert  $x_1 - 1$  addiert hatten und noch nicht mit dem Wert  $x_1$ :  
LOOP  $x_i$  DO  
   $x_0 := x_0 + 1$   
END
```