

**Einführung in die theoretische Informatik**  
Sommersemester 2019 – Übungsblatt Lösungsskizze 7

**AUFGABE 7.1.** (Wichtige Begriffe)

Stufe A

Überprüfen Sie, dass Sie die Folgenden Begriffe korrekt definieren können.

- PDA
- Kelleralphabet und -höhe
- Konversion CFG  $\leftrightarrow$  PDA
- Unterschied  $L_\varepsilon(A)$  und  $L_F(A)$  für einen PDA A

**Notation**

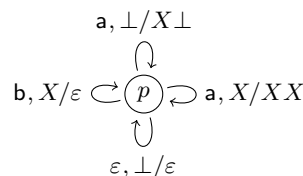
*Notation von PDA-Regeln:* Anstatt der in den Folien verwendeten Schreibweise  $(q, YZ) \in \delta(p, a, X)$  für die Ersetzungsregeln eines PDA kann man alternativ schreiben:  $pX \xrightarrow{a} qYZ$  ( $p, q \in Q, X, Y, Z \in \Gamma, a \in \Sigma$ ). Für den PDA mit  $\delta$ :

$$\delta(p, a, \perp) = \{(p, X\perp)\} \quad \delta(p, a, X) = \{(p, XX)\} \quad \delta(p, b, X) = \{(p, \varepsilon)\} \quad \delta(p, \varepsilon, \perp) = \{(p, \varepsilon)\}$$

schreibt man daher alternativ:

$$p\perp \xrightarrow{a} pX\perp \quad pX \xrightarrow{a} pXX \quad pX \xrightarrow{b} p \quad p\perp \xrightarrow{\varepsilon} p$$

oder stellt diesen als Graph mit Knotenmenge  $Q$  dar, wobei die Kante  $(p, q)$  dann mit "a, X/YZ" beschriftet ist (siehe Hopcroft et al., *Introduction to Automata Theory, Kapitel 6*):



**AUFGABE 7.2.** (Automata Tutor)

Stufe B

In Automata Tutor finden Sie neue Übungsaufgaben zu PDA's. Dort haben Sie auch die Möglichkeit Ihre PDA's auf konkreten Eingaben zu simulieren.

**AUFGABE 7.3.** (Pushdown-Automata / Kellerautomaten)

Stufe B/C

Geben Sie für die folgenden Sprachen jeweils einen Kellerautomaten  $A_i$  in allen oben aufgeführten Darstellungsarten an, so dass  $L_i = L(A_i)$ . Der Automat soll mit **leerem Stack** akzeptieren. Geben Sie dann zusätzlich für jeden Automaten jeweils ein nicht-leeres Wort  $w$  mit akzeptierendem Lauf an.

- (a)  $L_1 = \{a^n b^{3n} \mid n \geq 0\}$
- (b)  $L_2 = \{a^n b^m \in \{a, b\}^* \mid n \leq m \leq 2n\}$
- (c)  $L_3 = \{w \in \{a, b\}^* \mid 2 \cdot |w|_a = 3 \cdot |w|_b\}$

*Lösungsskizze*

Dies ist nur eine Skizze, die die Transitionsrelation angibt. Es fehlen die anderen Komponenten des 6-Tupels, um eine vollständige Antwort zu geben. Alternativ kann man die Lösungen auch bildlich darstellen.

- (a)  $qX \xrightarrow{a} qBBB \quad qX \xrightarrow{\varepsilon} q\varepsilon \quad qB \xrightarrow{a} qBBBB \quad qB \xrightarrow{b} p\varepsilon \quad pB \xrightarrow{b} p\varepsilon$   
 $(q, abbb, X) \rightarrow (q, bbb, BBB) \rightarrow (p, bb, BB) \rightarrow (p, b, B) \rightarrow (p, \varepsilon, \varepsilon)$
- (b) Idee: Für jedes a lege nichtdeterministisch entweder ein oder zwei b auf den Stack und überprüfe dann, ob die geratene Anzahl von bs mit der gegebenen übereinstimmt.

$$qX \xrightarrow{a} qB \quad qX \xrightarrow{a} qBB \quad qX \xrightarrow{\varepsilon} q\varepsilon$$

$$qB \xrightarrow{a} qBB \quad qB \xrightarrow{a} qBBB \quad qB \xrightarrow{b} p\varepsilon \quad pB \xrightarrow{b} p\varepsilon$$

$$(q, aabbb, X) \rightarrow (q, abbb, BB) \rightarrow (q, bbb, BBB) \rightarrow (p, bb, BB) \rightarrow (p, b, B) \rightarrow (p, \varepsilon, \varepsilon)$$

- (c) Idee: Verwende Stack als (unären) Zähler und benutze explizites Bottom-Symbol, um auf 0 zu testen. Für jedes  $a$  zähle um 2 (codiert als  $XX$ ) hoch und für jedes  $b$  ziehe 3 (codiert als  $YYY$ ) ab.

$$\begin{aligned}
 q\perp &\xrightarrow{a} qXX\perp & qX &\xrightarrow{a} qXXX & qY &\xrightarrow{a} p^+\varepsilon \\
 q\perp &\xrightarrow{b} qYYY\perp & qY &\xrightarrow{b} qYYYY & qX &\xrightarrow{b} p^{--}\varepsilon \\
 p^+\perp &\xrightarrow{\varepsilon} qX\perp & p^+Y &\xrightarrow{\varepsilon} q\varepsilon \\
 p^{--}\perp &\xrightarrow{\varepsilon} qYY\perp & p^{--}X &\xrightarrow{\varepsilon} p^-\varepsilon \\
 p^-\perp &\xrightarrow{\varepsilon} qY\perp & p^-X &\xrightarrow{\varepsilon} q\varepsilon \\
 q\perp &\xrightarrow{\varepsilon} q\varepsilon
 \end{aligned}$$

$$(q, \text{abbaa}, \perp) \rightarrow (q, \text{bbaa}, XX\perp) \rightarrow (p^{--}, \text{baa}, X\perp) \rightarrow (p^-, \text{baa}, \perp) \rightarrow (q, \text{baa}, Y\perp) \rightarrow (q, \text{aa}, YYYYY\perp) \rightarrow (p^+, a, YYY\perp) \rightarrow (q, a, YY\perp) \rightarrow (p^+, \varepsilon, Y\perp) \rightarrow (q, \varepsilon, \perp) \rightarrow (q, \varepsilon, \varepsilon)$$

**AUFGABE 7.4.** (CFG  $\longleftrightarrow$  PDA)

Stufe C

Wie Sie in der Vorlesung gesehen haben (Folie 214ff), können kontextfreie Grammatiken und Pushdown-Automaten sich gegenseitig simulieren. Wir üben nun diese Übersetzungen zwischen CFG und PDA.

- (a) Überführen Sie die folgende CFG  $G = (\{S\}, \{a, b, c\}, P, S)$  mithilfe von Satz 4.57 in einen PDA  $M$  mit  $L_\varepsilon(M) = L(G)$ :

$$S \rightarrow SS \mid aSb \mid bSa \mid c$$

- (b) Übersetzen Sie den folgenden PDA  $M = (\{q, p\}, \{a, b\}, \{\perp, A, B\}, q, \perp, \delta)$  in eine CFG  $G$  mit  $L_\varepsilon(M) = L(G)$ , wobei  $\delta$  definiert ist durch:

$$\delta(q, a, \perp) = \{(q, A)\} \quad \delta(q, a, A) = \{(q, AA)\} \quad \delta(q, b, A) = \{(p, \varepsilon)\} \quad \delta(p, b, A) = \{(p, \varepsilon)\} \quad \delta(q, b, \perp) = \{(q, \varepsilon)\}$$

*Lösungsskizze*

- (a) Normalform für Übersetzung:

$$\begin{aligned}
 S &\rightarrow SS \mid aSB \mid bSA \mid c \\
 A &\rightarrow a \\
 B &\rightarrow b
 \end{aligned}$$

PDA  $M = (\{q\}, \{a, b, c\}, \{S, A, B\}, q, S, \delta)$  mit  $\delta$ :

$$\begin{aligned}
 qS &\rightarrow qSS \\
 qS &\xrightarrow{a} qSB \\
 qS &\xrightarrow{b} qSA \\
 qS &\xrightarrow{c} q\varepsilon \\
 qA &\xrightarrow{a} q\varepsilon \\
 qB &\xrightarrow{b} q\varepsilon
 \end{aligned}$$

- (b)  $G = ((\{p, q\} \times \{\perp, A, B\} \times \{p, q\}) \cup \{S\}, \{a, b\}, P, S)$  mit den Produktionen  $P$ :

$$\begin{aligned}
 S &\rightarrow [q, \perp, p] \mid [q, \perp, q] \\
 [q, \perp, q] &\rightarrow a[q, A, q] \\
 [q, \perp, p] &\rightarrow a[q, A, p] \\
 [q, A, q] &\rightarrow a[q, A, q][q, A, q] \mid a[q, A, p][p, A, q] \\
 [q, A, p] &\rightarrow a[q, A, q][q, A, p] \mid a[q, A, p][p, A, p] \mid b \\
 [p, A, p] &\rightarrow b \\
 [q, \perp, q] &\rightarrow b
 \end{aligned}$$

**AUFGABE 7.5.** (Kellularalphabet beschränken)

Stufe D

Wir beschränken die Größe des Kellularalphabets  $\Gamma$  von PDAs und zeigen, dass jede kontextfreie Sprache von einem PDA mit  $|\Gamma| = 2$  erkannt werden kann. Skizzieren Sie hierzu eine allgemeine Übersetzung von einem PDA mit  $|\Gamma| > 2$  zu einem PDA mit  $|\Gamma'| = 2$ , so dass beide Automaten die gleiche Sprache erkennen.

### Lösungsskizze

Wir encodieren binär die Stacksymbole  $\Gamma$  mithilfe von  $\Gamma'$ . Sei  $k$  eine passende Konstante, so dass  $|\Gamma| \leq 2^k$ . Der neue PDA liest immer genau  $k$  Zeichen über eine Reihe von Hilfszuständen und rekonstruiert somit  $X \in \Gamma$  und führt dann die passende Transition aus. Hierauf werden wieder mit einer Reihe von Hilfszuständen die neuen Stacksymbole auf den Stack gepusht.

### AUFGABE 7.6. (Kellerhöhe beschränken)

Wir beschränken die Kellerhöhe von PDAs auf maximal  $k$  Kellerzeichen und nennen diese PDAs *k-bounded-Stack-PDA*. Insbesondere kann ein solcher PDA keine PUSH-Operationen ausführen sollten danach mehr als  $k$  Symbole auf dem Stack liegen. Zeigen Sie, dass *k-bounded-Stack-PDA* genau die regulären Sprachen erkennen, indem Sie eine allgemeine Übersetzung von PDAs zu  $\varepsilon$ -NFA angeben.

Stufe D

### Lösungsskizze

Da der Stack beschränkt ist, kann der  $\varepsilon$ -NFA den Stack in den Zuständen encodieren:  $Q' = Q \times \bigcup_{i=0}^k \Gamma^i$  und dementsprechend die Transitionen simulieren. Falls wir  $L_\varepsilon(M)$  betrachten, definieren wir  $F' = \{(q, \varepsilon) \mid q \in Q\}$ . Falls wir  $L_F(M)$  betrachten, definieren wir  $F' = \{(q, \alpha) \in Q' \mid q \in F\}$ .

### AUFGABE 7.7. (Deterministische PDAs)

In der Vorlesung haben Sie Lemma 4.68 ohne Beweis gesehen:

Sei  $L \subseteq \Sigma^*$ . Dann sind äquivalent:

- Es gibt einen DPDA  $D$  mit  $L_\varepsilon(D) = L$
- Es gibt einen DPDA  $D'$  mit  $L_F(D') = L$  **und** kein Wort aus  $L$  ist ein echtes Präfix von einem anderen Wort aus  $L$ .

Zeigen Sie diese Äquivalenz.

Stufe D

### Lösungsskizze

- (a)  $\implies$  (b):

Sei  $D$  mit  $L_\varepsilon(D) = L$ . Erweitere  $D$  um explizites Bottom-Symbol  $\perp$  mit  $q\perp \xrightarrow{\varepsilon} q_F$  für alle  $q \in Q$  und  $q_F$  neuer und einziger Endzustand. Der so erhaltene PDA ist noch deterministisch mit  $L_F(D') = L_\varepsilon(D)$ .

Seien  $u, uv \in L$ . Da  $D$  deterministisch, muss  $D$  nach Lesen von  $u$  stets in derselben Konfiguration sein, insbesondere der Stack somit leer, womit  $uv$  nur für  $v = \varepsilon$  akzeptiert werden kann.

- (b)  $\implies$  (a):

Sei (1)  $D$  mit  $L_F(D) = L$  und (2) kein Wort aus  $L$  ist ein echtes Präfix eines weiteren Worts aus  $L$ .

Wieder analog zu allgemeinen PDAs: Erweitere  $D$  so zu  $D'$ , dass beim ersten Erreichen einer Konfiguration mit Endzustand einfach der Stack deterministisch geleert wird, der Automat somit keine weitere Rechnung ausführen kann. Hier ist es wichtig, mit einem expliziten Bottom-Symbol zu verhindern, dass der Stack *außer* im Fall des deterministischen Leerens leer wird.

Übergänge, die vorher von Endzuständen aus möglich waren, werden ersetzt durch einen  $\varepsilon$ -Übergang zu dem Zustand, der deterministisch den Stack leert. Das ist möglich, da diese Übergänge keinen akzeptierenden Lauf mehr erreichen können. Beweis:

Sei  $w \in L_F(D) \setminus L_\varepsilon(D')$ . Dann muss die eindeutige akzeptierende Berechnung von  $D$  auf  $w$  einen Endzustand mindestens zweimal besuchen und schließlich in einer solchen Konfiguration enden. Der Fall, dass man nach dem ersten Besuch einer Endkonfiguration nur noch  $\varepsilon$  liest, kann nicht sein, da dann das gelesene Wort  $w$  auch von  $D'$  noch akzeptiert wird. Daher gibt es aber auch einen echten Präfix von  $w$ , der von  $D$  akzeptiert und damit in  $L$  liegt. Widerspruch zu (2). Es kann somit kein solches  $w$  geben.

Offensichtlich gilt dann:  $L_F(D) = L_\varepsilon(D')$ .

### AUFGABE 7.8. (Zählerautomaten)

Wir beschränken in dieser Aufgabe das Kellularalphabet von PDAs auf nur ein einziges Symbol und untersuchen die Ausdrucksmächtigkeit dieser Automaten etwas genauer.

- Geben Sie eine äquivalente Formulierung als Automaten, die statt des Kellers einen Zähler verwenden, an.
- Geben Sie einen solchen Automaten an, der eine nicht reguläre Sprache akzeptiert.
- Nun betrachten wir eine Variante, die zusätzlich noch ein Kellersymbol erlaubt, um den Anfang des Kellers zu markieren. Geben Sie wieder eine Charakterisierung mit Zählern an und einen Automaten, der eine kontextfreie Sprache akzeptiert, die von keinem Automaten im vorherigen Modell akzeptiert wird.
- Geben Sie eine kontextfreie Sprache an, die von keiner der betrachteten Automatenklassen akzeptiert wird.

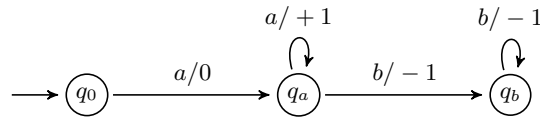
*Knobelaufgaben:* Die folgenden Aufgaben können alle unabhängig voneinander gelöst werden:

- Geben Sie einen solchen Automaten an, der die Sprache  $\bar{L}$  für  $L = \{ww^R \mid w \in \{a, b\}^*\}$  akzeptiert.
- Nun betrachten wir deterministische Varianten dieser Automaten. Zeigen Sie zunächst, dass die Klasse dieser deterministischen Automaten unter dem Sprachkomplement abgeschlossen ist.
- Zeigen Sie, dass kein deterministischer Zählerautomat die Sprache  $L$  akzeptiert. *Hinweis:* Wie viele verschiedene Zustände kann ein Automat beim Lesen von Worten der Länge  $n$  erreichen?
- Zeigen Sie aus den vorherigen Aussagen, dass nichtdeterministische Automaten mit einem Zähler strikt mächtiger sind als deterministische Automaten mit beliebig vielen Zählern.

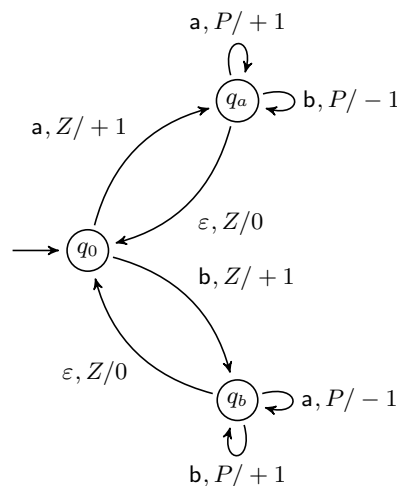
Stufe E

Lösungsskizze

- (a) Die Automaten sind ähnlich zu NFAs, aber besitzen einen Zähler der am Anfang auf eins steht. Beim Lesen eines Zeichens kann man sich entscheiden, den Zähler zu inkrementieren, zu dekrementieren, oder gleich zu lassen. Falls der Zähler 0 erreicht, bleibt der Automat stecken. Solche Automaten nennt man manchmal auch halbblinde Zählerautomaten. Wir werden im folgenden ein Akzeptanzkriterium mit Zählerwert 0 verwenden.
- (b) Der folgende Automat akzeptiert die Sprache  $\{a^n b^n \mid n \in \mathbb{N}\}$ .



- (c) In dieser Variante steht der Zähler am Anfang auf 0 und beim Lesen eines Zeichens kann der Zähler auf Gleichheit mit 0 überprüft werden. Akzeptiert wird bei Zählerwert 0. Der folgende Automat akzeptiert die Sprache  $\{w \mid |w|_a = |w|_b \wedge w \in \{a, b\}^*\}$ .

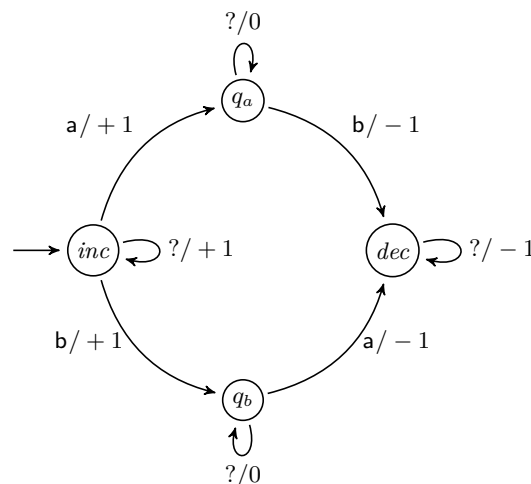


Hier bezeichnet  $Z$ , dass der Zähler gleich null ist, und  $P$ , dass der Zähler ungleich null ist. Es ist auch möglich diesen Automaten zu determinisieren.

- (d) Beispiele:  $\{a^n b^{2m} a^n \mid n, m \in \mathbb{N}\}$ ,  $L$

Knobelaufgaben:

- (a) Idee: Wir raten eine Stelle  $k$ , an der der  $k$ -te Buchstabe nicht mit dem  $k$ -letzten Buchstaben übereinstimmt.



- (b) Wir betrachten nun statt Akzeptanz mit Zählerwert 1/0 wieder Endzustände als Akzeptanzkriterium. Für die nichtdeterministische Variante ist eine entsprechende Konversion zwischen den Akzeptanzkriterien analog zu den Verfahren für PDA aus der Vorlesung möglich. Durch Vertauschen der Endzustände erhalten wir nun wieder ein Komplementierungsverfahren für deterministische Zählerautomaten analog zu DFA's.
- (c) Bei  $k$  Zählern können beim Lesen von Worten der Länge  $n$  maximal  $\mathcal{O}(n^k)$  verschiedene Konfigurationen erreicht werden.

---

*Beweis.* Die Anzahl der Zustände im Automaten ist konstant. Nach  $n$  Schritten kann jeder Zähler nur einen Wert im Zahlenbereich von 0 bis  $n$  annehmen. Damit erhalten wir  $|Q|(n+1)^k \in \mathcal{O}(n^k)$  mögliche Konfigurationen.  $\square$

Damit fallen alle Worte der Länge  $n$  in  $\mathcal{O}(n^k)$  viele Äquivalenzklassen. Für die Sprache  $L$  befindet sich aber jedes Wort in einer eigenen Äquivalenzklasse. Damit fallen die Worte der Länge  $n$  in  $\mathcal{O}(2^n)$  verschiedene Äquivalenzklassen. Also kann kein deterministischer Zählerautomat die Sprache  $L$  akzeptieren.

- (d) Angenommen es gäbe einen deterministischen Zählerautomaten  $A$ , der  $\bar{L}$  akzeptiert. Dann gäbe es auch einen deterministischen Zählerautomaten  $A'$ , der  $L$  akzeptiert. Widerspruch. Da wir aber bereits einen nichtdeterministischen Automaten mit einem Zähler konstruiert haben, der  $\bar{L}$  akzeptiert, ist diese Klasse strikt mächtiger.