

## Einführung in die Theoretische Informatik

### Sommersemester 2024 – Übungsblatt 8

- Das Übungsblatt ist in zwei Teile gegliedert: den Vorbereitungsteil, den Sie vor der Übung selbstständig bearbeiten sollen, und den Übungs-/Nachbereitungsteil, der Aufgaben enthält, die in der Übung besprochen werden und von Ihnen anschließend zur Nachbereitung verwendet werden können.

*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  $pX \xrightarrow{a} qYZ$  schreiben wobei  $p, q \in Q$ ,  $X \in \Gamma$ ,  $YZ \in \Gamma^*$ ,  $a \in \Sigma \cup \{\varepsilon\}$ .

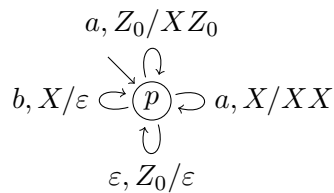
*Beispiel:* Den PDA mit  $\delta$ :

$$\begin{aligned} \delta(p, a, Z_0) &= \{(p, XZ_0)\} & \delta(p, a, X) &= \{(p, XX)\} \\ \delta(p, b, X) &= \{(p, \varepsilon)\} & \delta(p, \varepsilon, Z_0) &= \{(p, \varepsilon)\} \end{aligned}$$

schreibt man alternativ:

$$pZ_0 \xrightarrow{a} pXZ_0 \quad pX \xrightarrow{a} pXX \quad pX \xrightarrow{b} p \quad pZ_0 \xrightarrow{\varepsilon} p$$

oder man stellt diesen als Graph mit Knotenmenge  $Q$  dar, wobei die Kante  $(p, q)$  dann mit " $a, X/YZ$ " beschriftet ist:



Für einige häufige Fälle führen wir außerdem weitere Kurzschreibweisen ein. Sei  $M = (Q, \Sigma, \Gamma, q_0, Z_0, \delta, F)$  ein PDA mit  $\Gamma = \{X_1, X_2, \dots, X_k\}$  und  $a \in \Sigma \cup \{\varepsilon\}$ . Wir schreiben

- $a, \Gamma/\varepsilon$  statt  $a, X_1/\varepsilon, \dots, a, X_k/\varepsilon$  und
- $a, \Gamma/\alpha\Gamma$  statt  $a, X_1/\alpha X_1, \dots, a, X_k/\alpha X_k$ , mit  $\alpha \in \Gamma^*$

### Vorbereitung (→ vor der Übung selbstständig zu bearbeiten)

#### Vorbereitungsaufgabe Ü8.1. (Wichtige Begriffe)

Überprüfen Sie, dass Sie die folgenden Begriffe oder Notationen korrekt definieren können.

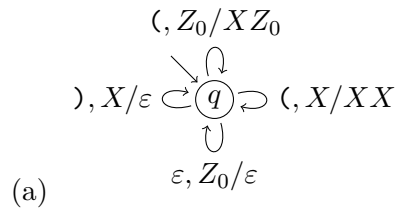
- Kellerautomat (PDA)
- Unterschied zwischen  $L_\varepsilon(A)$  und  $L_F(A)$  für einen PDA  $A$

- deterministischer Kellerautomat (DPDA)
- deterministische kontextfreie Sprache (DCFL)
- Abschlusseigenschaften von DCFL
- Abschlusseigenschaften von CFL

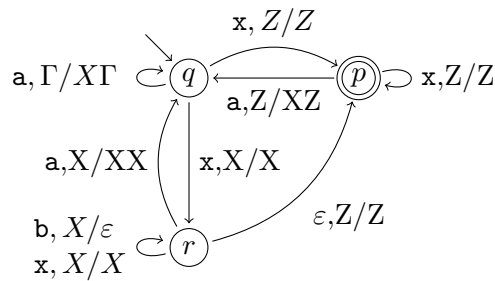
**Vorbereitungsaufgabe Ü8.2.** (*Automata Tutor: "DPDAs"*)

Lösen Sie die Aufgaben **Ü8.2** (a–c) auf Automata Tutor.

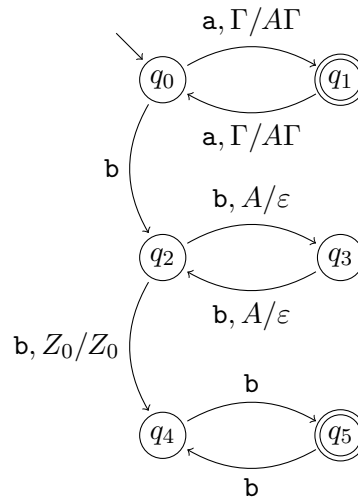
*Lösungsskizze.*



(b) Zu dieser Aufgabe gibt es ein Lösungsvideo. Mögliche Lösung:



(c) Mögliche Lösung:



## Übung und Nachbereitung

### Übungsaufgabe Ü8.3. (PDAs)

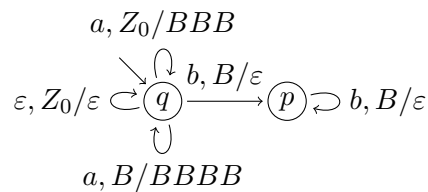
Geben Sie für die folgenden Sprachen jeweils einen Kellerautomaten  $A_i$  in einer der oben aufgeführten Darstellungsarten an, sodass  $L_i = L(A_i)$ . Der Automat soll mit *leerem Stack* akzeptieren. Geben Sie dann zusätzlich für jeden Automaten jeweils ein nicht-leeres Wort  $w$  mit akzeptierendem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

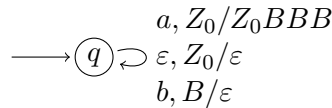
Wir geben die PDAs in Form einer Skizze des Automaten an. Für jeden der folgenden PDAs initialisieren wir den Stack mit dem Symbol  $Z_0$ .

(a)

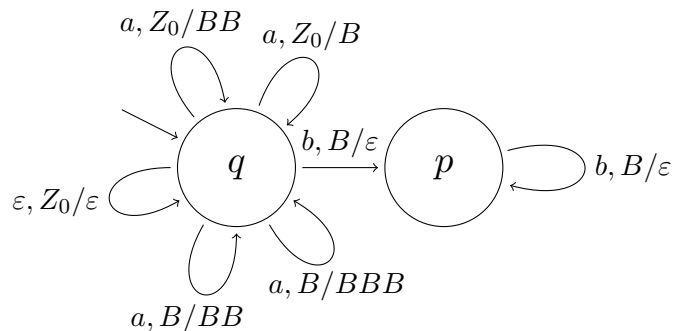


Akzeptierender Lauf:  $(q, abbb, Z_0) \rightarrow (q, bbb, BBB) \rightarrow (p, bb, BB) \rightarrow (p, b, B) \rightarrow (p, \epsilon, \epsilon)$

Lösung mit nur einem Zustand:

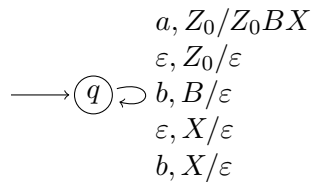


(b) Idee: Für jedes  $a$  lege nichtdeterministisch entweder ein oder zwei  $b$  auf den Stack und überprüfe dann, ob die geratene Anzahl von  $bs$  mit der gegebenen übereinstimmt.

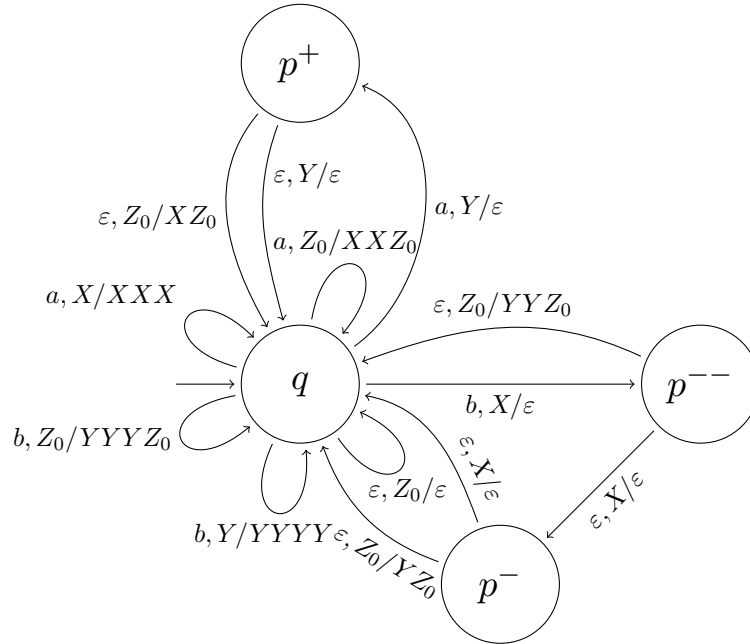


Akzeptierender Lauf:  $(q, aabbb, Z_0) \rightarrow (q, abbb, BB) \rightarrow (q, bbb, BBB) \rightarrow (p, bb, BB) \rightarrow (p, b, B) \rightarrow (p, \epsilon, \epsilon)$

Lösung mit nur einem Zustand:



- (c) Idee: Verwende Stack als (unären) Zähler und benutze explizites Bottom-Symbol (hier  $Z_0$ ), 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.



Akzeptierender Lauf:  $(q, abbaa, Z_0) \rightarrow (q, bbaa, XXZ_0) \rightarrow (p^{--}, baa, XZ_0) \rightarrow (p^-, baa, Z_0) \rightarrow (q, baa, YZ_0) \rightarrow (q, aa, YYYYYZ_0) \rightarrow (p^+, a, YYYZ_0) \rightarrow (q, a, YYZ_0) \rightarrow (p^+, \varepsilon, YZ_0) \rightarrow (q, \varepsilon, Z_0) \rightarrow (q, \varepsilon, \varepsilon)$

#### Übungsaufgabe Ü8.4. (PDA Einschränkungen)

- (a) 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  $|\Gamma'| = 2$ , sodass beide Automaten die gleiche Sprache erkennen.
- (b) 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 zu  $\varepsilon$ -NFAs angeben.

*Lösungsskizze.*

- (a) Wir kodieren binär die Stacksymbole  $\Gamma$  mithilfe von  $\Gamma' = \{0, 1\}$ . Sei  $k$  eine passende Konstante, sodass  $|\Gamma| \leq 2^k$ . Der neue PDA entfernt zur Durchführung jeder ursprünglichen Transition, ausgehend von einem Zustand  $q$ , zunächst genau  $k$  Stackzeichen (z.B. 01101) und merkt sich die entfernte Zeichenkette über eine Rei-

he von Hilfszuständen. Beispielsweise befindet sich der PDA nach diesen Schritten im Zustand  $q_{01101}$ . Der Zustand  $q_{01101}$  kodiert also, dass der ursprüngliche Automat im Zustand  $q$  jenes Stacksymbol  $X \in \Gamma$  liest, welches binär kodiert 01101 ergibt. Nun kann der PDA die ursprüngliche Transition für  $q, X$  und dem aktuellen Terminal ausführen. Hierauf werden mit einer Reihe von Hilfszuständen die neuen Stacksymbole binär kodiert auf den Stack gepusht.

Beim Start des PDAs wird das Startsymbol zunächst binär kodiert auf den Keller gepusht.

- (b) Da der Stack beschränkt ist, kann der  $\varepsilon$ -NFA den Stack in den Zuständen kodieren:  $Q' = Q \times \bigcup_{i=0}^k \Gamma^i$  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  $F' = \{(q, \alpha) \mid q \in F\}$ .

### Übungsaufgabe Ü8.5. (Kodiergott)

Sei  $n \in \mathbb{N}_+$ . Für ein Wort  $w \in \{0,1\}^*$  bezeichnen wir mit  $(w)_2$  den Dezimalwert des Binärwortes in most-significant-bit-first-Darstellung. Beispielsweise  $(1100)_2 = 12$ .

Betrachte die Sprache  $L := \{wa^{(w)_2} \mid w \in \{0,1\}^n\}$ . Beispielsweise gilt  $1100a^{12} \in L$ . Konstruieren Sie einen deterministischen PDA für  $L$  mit  $\mathcal{O}(n)$  Stacksymbolen und Zuständen. Außerdem soll die Kellerhöhe des Automaten stets  $\mathcal{O}(n)$  beschränkt sein.

*Lösungsskizze.* Wir setzen  $(Q, \Sigma, \Gamma, q_n, Z_0, \delta)$  mit

$$\begin{aligned} Q &:= \{q_0, \dots, q_n\} \\ \Sigma &:= \{0, 1, a\} \\ \Gamma &:= \{Z_0, A_1, \dots, A_n\} \\ \delta(q_i, 0, X) &:= \{(q_{i-1}, X)\}, & \text{für } i \in \{1, \dots, n\}, X \in \Gamma \\ \delta(q_i, 1, X) &:= \{(q_{i-1}, A_i X)\}, & \text{für } i \in \{1, \dots, n\}, X \in \Gamma \\ \delta(q_0, \epsilon, A_i) &:= \{(q_0, A_{i-1} A_{i-1})\}, & \text{für } i \in \{2, \dots, n\} \\ \delta(q_0, a, A_1) &:= \{(q_0, \epsilon)\} \\ \delta(q_0, \epsilon, Z_0) &:= \{(q_0, \epsilon)\} \end{aligned}$$

Idee: Für jede Zweierpotenz  $2^i$  wird ein Stacksymbol eingeführt. Falls wir das Stacksymbol für  $2^0 = 1$  sehen, lesen wir ein  $a$ . Ansonsten zerkleinern wir das Stacksymbol für  $2^{i+1}$  in zwei Stacksymbole für  $2^i$ .

### Übungsaufgabe Ü8.6. (CFG $\rightarrow$ PDA)

Wie Sie in der Vorlesung gesehen haben (ab Folie 203), können kontextfreie Grammatiken und Pushdown-Automaten sich gegenseitig simulieren. Wir üben nun die Übersetzungen von CFGs zu PDAs.

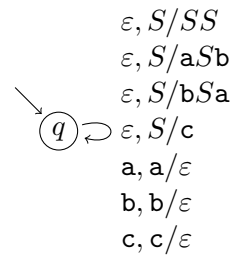
Überführen Sie die folgende CFG  $G = (\{S\}, \{a, b, c\}, P, S)$  mit Hilfe des Satzes der Vorlesung in einen PDA  $M$  mit  $L_\varepsilon(M) = L(G)$ :

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

*Lösungsskizze.* PDA  $M = (\{q\}, \{a, b, c\}, \{S, a, b, c\}, q, S, \delta)$  mit  $\delta$ :

$$\begin{aligned}
qS &\xrightarrow{\varepsilon} qSS \\
qS &\xrightarrow{\varepsilon} qaSb \\
qS &\xrightarrow{\varepsilon} qbSa \\
qS &\xrightarrow{\varepsilon} qc \\
qa &\xrightarrow{a} q\varepsilon \\
qb &\xrightarrow{b} q\varepsilon \\
qb &\xrightarrow{b} q\varepsilon
\end{aligned}$$

Alternativ die graphische Darstellung:



### Zusätzliche Übungsaufgabe Ü8.7. (Zählerautomaten)

**Hinweis:** In dieser Aufgabe werden keine neuen Vorlesungsinhalte besprochen. Sie können die Aufgabe zur zusätzlichen Übung nutzen.

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.

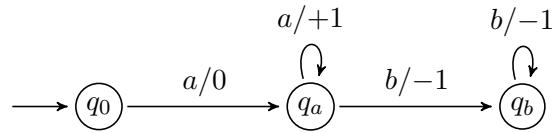
Die folgenden Aufgaben können alle unabhängig voneinander gelöst werden:

- Geben Sie einen solchen Automaten an, der die Sprache  $\bar{L}$  für  $L = \{w \in \{a, b\}^* \mid w = w^R\}$  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.

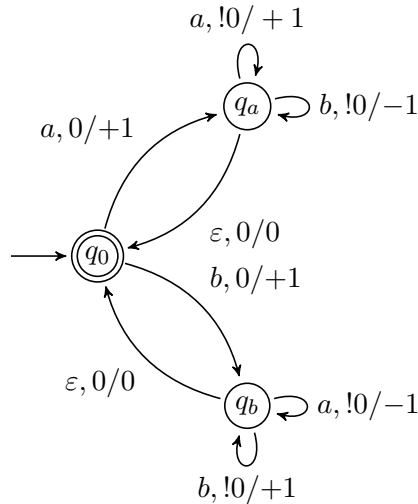
*Lösungsskizze.*

- 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  $\{a^n b^n \mid n \in \mathbb{N}_{>0}\}$ .



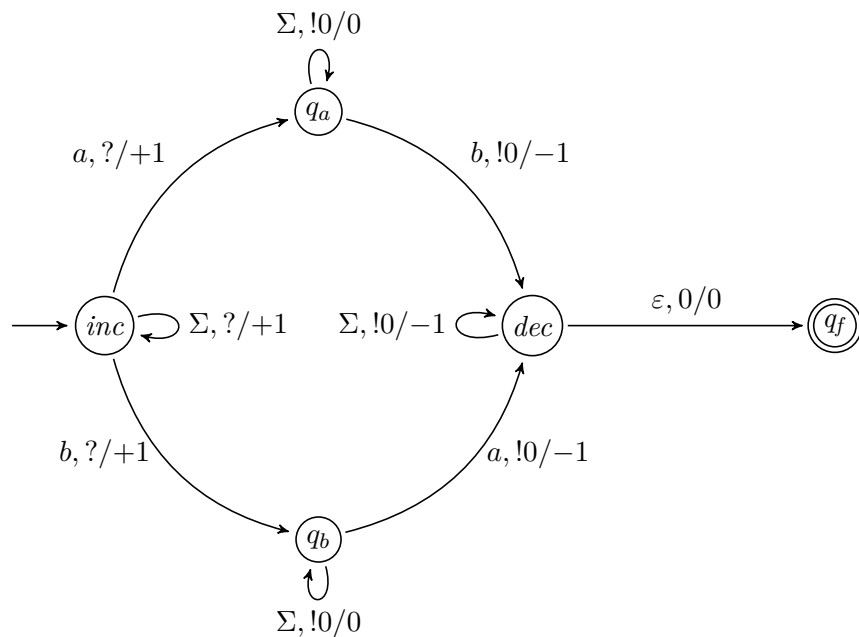
(c) In dieser Variante steht der Zähler am Anfang auf 0 und beim Lesen eines Zeichens kann der Zähler auf Gleichheit mit 0 überprüft werden. Akzeptiert wird mit Endzustand. Der folgende Automat akzeptiert die Sprache  $\{w \in \{a, b\}^* \mid |w|_a = |w|_b\}$ .



Hier bezeichnet 0 (bzw !0) an der zweiten Stelle einer Transition, dass der Zähler (nicht) gleich null ist. Es ist auch möglich diesen Automaten zu determinisieren.

(d) Beispiel:  $\{ww^R \mid w \in \{a, b\}^*\}$

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



- (f) Durch Vertauschen der Endzustände erhalten wir ein Komplementierungsverfahren für deterministische Zählerautomaten analog zu DFA's.
- (g) 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.

- (h) 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.