

**Einführung in die theoretische Informatik**  
Sommersemester 2017 – Übungsblatt Lösungsskizze 9

**Übungsblatt**

Wir unterscheiden zwischen Übungs- und Abgabebblättern. Auf diesem *Übungsblatt* finden Sie eine Übersicht über die Kernaspekte, die Sie in Kalenderwoche 26 in den Tutorien diskutieren, üben und vertiefen. Die Aufgaben auf diesem Blatt dienen dem Üben und Verstehen des Vorlesungsstoffes, sowie dem *eigenständigen Erarbeiten* der Kernaspekte. Außerdem sollen Ihnen diese Aufgaben auch helfen, ein Gefühl dafür zu bekommen, was Sie inhaltlich in der Klausur erwartet. Klausuraufgaben können jedoch deutlich von den hier gestellten Aufgaben abweichen. Abschreiben und Auswendiglernen von Lösungen wird Ihnen daher keinen dauerhaften Erfolg in der Vorlesung bringen. Fragen zu den Übungsblättern können Sie montags bis donnerstags von 12 Uhr bis 14 Uhr in der *THEO-Sprechstunde* in Raum 03.11.034 stellen.

**Kernaspekte**

- K9.1 eine gegebene kontextfreie Grammatik  $G$  in einen PDA  $P$  übersetzen, so dass  $L(G) = L(P)$
- K9.2 einen gegebenen PDA  $P$  in eine Grammatik  $G$  übersetzen, so dass  $L(G) = L(P)$
- K9.3 Abschlusseigenschaften von CFL und DCFL beweisen bzw. widerlegen
- K9.4 Eigenschaften, die den Zusammenhang von  $L_\epsilon(A)$  und  $L_F(A)$  auf DPDAs beschreiben, beweisen oder widerlegen

**AUFGABE 9.1.** ( $CFG \longleftrightarrow PDA$ )

Stufe C

Wie Sie in der Vorlesung gesehen haben, 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  $L_\epsilon(M) = L(G)$ :

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

- (b) Übersetzen Sie den folgenden PDA  $M = (\{q, p\}, \{l, n, x, y, a, o, r\}, \{X, Y, Z, [YXZ], [YXZ], [XZ]\}, q, X, \delta)$  in eine CFG  $G$  mit  $L_\epsilon(M) = L(G)$ , wobei  $\delta$  definiert ist durch:

$$qX \xrightarrow{l} qX[YXZ] \quad q[YXZ] \xrightarrow{\epsilon} pY[XZ] \quad q[XZ] \xrightarrow{\epsilon} qXZ \quad qX \xrightarrow{n} qX \quad qX \xrightarrow{x,y} q\epsilon \quad pY \xrightarrow{a,o} q\epsilon \quad qZ \xrightarrow{r} q\epsilon$$

*Notation:*  $pX \xrightarrow{a} qYZ$  steht kurz für  $(q, YZ) \in \delta(p, a, X)$ .

*Lösungsskizze*

- (a) Normalform für Übersetzung:

$$S \rightarrow SS \mid aSB \mid bSA \mid c$$

$$A \rightarrow a$$

$$B \rightarrow b$$

PDA mit Startkonfiguration  $qS$ :

$$qS \rightarrow qSS$$

$$qS \xrightarrow{a} qSB$$

$$qS \xrightarrow{b} qSA$$

$$qS \xrightarrow{c} q\epsilon$$

$$qA \xrightarrow{a} q\epsilon$$

$$qB \xrightarrow{b} q\epsilon$$

(b)

$$\begin{aligned} X_{q,[YXZ],p} &\rightarrow X_{p,Y,q}X_{q,[XZ],p} \mid X_{p,Y,p}X_{p,[XZ],p} \\ X_{p,Y,q} &\rightarrow a \mid o \\ X_{q,X,p} &\rightarrow lX_{q,X,p}X_{p,[YXZ],p} \mid nX_{q,X,p} \mid lX_{q,X,q}X_{q,[YXZ],p} \\ X_{q,X,q} &\rightarrow nX_{q,X,q} \mid y \mid x \mid lX_{q,X,p}X_{p,[YXZ],q} \mid lX_{q,X,q}X_{q,[YXZ],q} \\ X_{q,Z,q} &\rightarrow r \\ X_{q,[XZ],q} &\rightarrow X_{q,X,q}X_{q,Z,q} \mid X_{q,X,p}X_{p,Z,q} \\ X_{q,[XZ],p} &\rightarrow X_{q,X,q}X_{q,Z,p} \mid X_{q,X,p}X_{p,Z,p} \\ X_{q,[YXZ],q} &\rightarrow X_{p,Y,q}X_{q,[XZ],q} \mid X_{p,Y,p}X_{p,[XZ],q} \\ S &\rightarrow X_{q,X,p} \mid X_{q,X,q} \end{aligned}$$

### AUFGABE 9.2. (Deterministische PDAs)

Stufe D

In der Vorlesung haben Sie die 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 echter Präfix von einem anderen Wort aus  $L$ .

Zeigen Sie diese Äquivalenz.

*Lösungsskizze*

- 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.
- Sei (1)  $D$  mit  $L_F(D) = L$  und (2) kein Wort aus  $L$  ist ein echter Präfix eines weiteren Wortes 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. Offensichtlich gilt dann:  $L_F(D) \supseteq L_\varepsilon(D')$ . 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.

### AUFGABE 9.3.

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  $|\Gamma'| = 2$  an, 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 wird wieder mit einer Reihe von Hilfszuständen die neuen Stacksymbole auf den Stack gepusht.

### AUFGABE 9.4.

Stufe D

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.

*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 9.5. (Komplementierung von DPDAs)

Stufe E

Erarbeiten Sie zusammen mit Ihrem Tutor ein Verfahren zur Komplementierung von DPDAs.

**Hinweis:** Auf Blatt 7 der THEO 2015 (<http://www.mayr.in.tum.de/lehre/2015SS/theo/>) finden Sie die nötigen Begriffe und Grundlagen.