

Theoretische Informatik

Prof. Dr. Peter Thiemann
Marius Weidner
Simon Dorer

Universität Freiburg
Institut für Informatik
Sommersemester 2026

Übungsblatt 8 – Lösungen

Abgabe: Dienstag, 23.06.2026, 16:00 Uhr

Aufgabe 8.1 (Pumping Lemma über einelementigem Alphabet; 5 Punkte)

Sei L eine Sprache über $\Sigma = \{a\}$. Angenommen, es gibt einen Beweis mit dem Pumping Lemma für reguläre Sprachen (PLR), der beweist, dass L nicht regulär ist.

Zeigen Sie mit dem Pumping Lemma für kontextfreie Sprachen (PLK), dass L auch nicht kontextfrei ist.

Lösung:

Angenommen, L ist kontextfrei. Dann gelten die Bedingungen aus dem PLK. Sei $p \in \mathbb{N}$ die zugehörige Pumping-Länge.

Nach Annahme gibt es einen PLR-Beweis, der beweist, dass L nicht regulär ist. Wir wenden diesen Beweis mit $n := p$ an. Dann gibt es ein Wort $z \in L$ mit $|z| \geq p$, sodass es für alle Zerlegungen $z = uvw$ mit $|v| > 0$ und $|uv| \leq p$ ein $i \in \mathbb{N}$ gibt mit $uv^i w \notin L$.

Sei nun $z = UVWXY$ eine beliebige PLK-Zerlegung mit

$$|VX| > 0 \quad \text{und} \quad |VWX| \leq p.$$

Da alle Wörter über dem einelementigen Alphabet $\Sigma = \{a\}$ liegen, ist jedes Wort eindeutig durch seine Länge bestimmt. Deshalb dürfen wir die Faktoren in anderer Reihenfolge schreiben, ohne das entstehende Wort zu ändern.

Wir setzen

$$u := W, \quad v := VX, \quad w := UY.$$

Dann gilt $uvw = W(VX)(UY) = z$. Außerdem gilt

$$|v| = |VX| > 0 \quad \text{und} \quad |uv| = |WVX| = |VWX| \leq p.$$

Also ist $z = uvw$ eine gültige PLR-Zerlegung. Nach Wahl von z gibt es daher ein $i \in \mathbb{N}$ mit $uv^i w \notin L$. Für dieses i gilt wiederum wegen $\Sigma = \{a\}$:

$$UV^iWX^iY = WV^iX^iUY = W(VX)^iUY = uv^i w \notin L.$$

Damit gibt es für jede PLK-Zerlegung ein $i \in \mathbb{N}$, sodass $UV^iWX^iY \notin L$. Dies widerspricht dem PLK. Also ist L nicht kontextfrei.

□

Aufgabe 8.2 (Kellerautomaten; 5 Punkte)

Konstruieren Sie für jede der folgenden Sprachen einen NPDA, der genau diese Sprache erkennt.

Erklären Sie dabei *kurz*, welche Information auf dem Keller gespeichert wird und warum der Automat genau die angegebene Sprache erkennt.

- (a) $L_1 = \{a^n b^m \mid m \leq n \leq 2m\}$ über dem Alphabet $\Sigma = \{a, b\}$.



- (b) $L_2 = \{w \in \Sigma^* \mid w \text{ ist ein korrekt geklammerter Ausdruck}\}$ über dem Alphabet $\Sigma = \{\langle, \rangle, [,]\}$ (das Alphabet der öffnenden und schließenden spitzen und eckigen Klammern). Zum Beispiel gilt $\varepsilon \in L_2$ und $\langle [] \rangle \in L_2$, aber $\langle \rangle \notin L_2$ und $\langle [] \rangle \notin L_2$. Ihr NPDA für L_2 soll dabei eine **minimale Anzahl an Zuständen** haben.

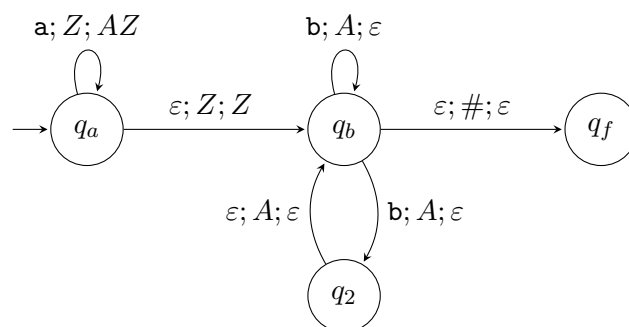


Hinweis: Aus der graphischen Darstellung von Kellerautomaten geht nicht direkt hervor, welches Kellersymbol als Kellerbodensymbol verwendet werden soll. Geben Sie ihr gewähltes Symbol also explizit an, um einen vollständig definierten Kellerautomaten zu erhalten.

Lösung:

In den folgenden NPDAs verwenden wir $\#$ als Kellerbodensymbol.

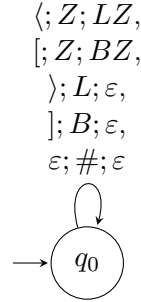
- (a)



für beliebige $Z \in \Gamma = \{\#, A\}$.

Für jedes gelesene **a** wird ein Symbol A auf den Keller gelegt. Beim Lesen der **a**-Phase rät der Automat nichtdeterministisch, dass alle **a**-Symbole gelesen wurden. Danach entscheidet er für jedes **b**, ob ein oder zwei A -Symbole entfernt werden. Am Ende wird das Kellerbodensymbol entfernt, falls alle A -Symbole entfernt wurden.

(b)

für beliebige $Z \in \Gamma = \{\#, L, B\}$.

Auf dem Keller liegen genau die noch nicht geschlossenen öffnenden Klammern: L steht für $\langle$, und B steht für $[$. Eine schließende Klammer darf nur das passende oberste Kellersymbol entfernen.

Aufgabe 8.3 (Kontextinvarianz der Schrittrelation; 5 Punkte)

In der Vorlesung haben Sie gesehen, dass es zu jeder CFG $\mathcal{G}$ ein äquivalenter NPDA $\mathcal{K}$ existiert, der dieselbe Sprache erkennt. Hierbei haben wir folgendes Lemma verwendet:

Lemma. Für alle $q, q' \in Q$, $w \in \Sigma^*$, $Z \in \Gamma$ und $n \in \mathbb{N}$ gilt:

Wenn $(q, w, Z) \triangleright^n (q', \varepsilon, \varepsilon)$, dann $\forall v \in \Sigma^*, \gamma \in \Gamma^* : (q, wv, Z\gamma) \triangleright^n (q', v, \gamma)$.

Zeigen Sie, dass das Lemma korrekt ist.

Lösung:

Wir beweisen durch Induktion über die Anzahl der Relationsschritte $n \in \mathbb{N}$ die folgende stärkere Aussage: Für alle $p, p' \in Q$, $v \in \Sigma^*$, $\alpha \in \Gamma^*$, $u \in \Sigma^*$ und $\gamma \in \Gamma^*$ gilt:

$$(p, v, \alpha) \triangleright^n (p', \varepsilon, \varepsilon) \implies (p, vu, \alpha\gamma) \triangleright^n (p', u, \gamma).$$

Diese Aussage ist stärker als das Lemma, da dort nur der Spezialfall $\alpha = Z$ mit $Z \in \Gamma$ benötigt wird. Wir brauchen die stärkere Form, weil nach dem ersten Schritt einer Rechnung der Kellerinhalt im Allgemeinen ein Wort $\alpha \in \Gamma^*$ und nicht nur ein einzelnes Kellersymbol ist.

IA: Sei $n = 0$. Angenommen,

$$(p, v, \alpha) \triangleright^0 (p', \varepsilon, \varepsilon).$$

Dann gilt direkt $(p, v, \alpha) = (p', \varepsilon, \varepsilon)$. Also ist $p = p'$, $v = \varepsilon$ und $\alpha = \varepsilon$. Damit gilt für alle $u \in \Sigma^*$ und $\gamma \in \Gamma^*$

$$(p, vu, \alpha\gamma) = (p', u, \gamma),$$

und somit

$$(p, vu, \alpha\gamma) \triangleright^0 (p', u, \gamma).$$

IV: Für ein festes $n \in \mathbb{N}$ gelte: Für alle Zustände $p, p' \in Q$, Wörter $v \in \Sigma^*$, Kellersymbole $\alpha \in \Gamma^*$ sowie alle $u \in \Sigma^*$ und $\gamma \in \Gamma^*$ gilt:

$$(p, v, \alpha) \triangleright^n (p', \varepsilon, \varepsilon) \implies (p, vu, \alpha\gamma) \triangleright^n (p', u, \gamma).$$

IS: Wir zeigen die Aussage für $n + 1$. Sei also

$$(p, v, \alpha) \triangleright^{n+1} (p', \varepsilon, \varepsilon).$$

Dann gibt es eine Konfiguration (r, x, β) mit

$$(p, v, \alpha) \triangleright (r, x, \beta) \triangleright^n (p', \varepsilon, \varepsilon).$$

Nach Definition der Schrittrelation wird in einem Schritt höchstens ein Eingabezeichen gelesen und nur das oberste Kellersymbol ersetzt. Der zusätzliche Suffix u der Eingabe und der zusätzliche Suffix γ des Kellers bleiben daher unverändert. Also gilt:

$$(p, vu, \alpha\gamma) \triangleright (r, xu, \beta\gamma).$$

Aus $(r, x, \beta) \triangleright^n (p', \varepsilon, \varepsilon)$ folgt mit der Induktionsvoraussetzung

$$(r, xu, \beta\gamma) \triangleright^n (p', u, \gamma).$$

Zusammen erhalten wir

$$(p, vu, \alpha\gamma) \triangleright (r, xu, \beta\gamma) \triangleright^n (p', u, \gamma),$$

also

$$(p, vu, \alpha\gamma) \triangleright^{n+1} (p', u, \gamma).$$

Damit gilt die stärkere Aussage für alle $n \in \mathbb{N}$. Setzen wir darin $\alpha = Z$ für ein beliebiges $Z \in \Gamma$, erhalten wir genau die Behauptung des Lemmas. □

Aufgabe 8.4 (Kellerautomaten mit akzeptierenden Zuständen; 5 Punkte)

In der Vorlesung haben wir die Akzeptanz eines Kellerautomaten $\mathcal{K}$ über den leeren Keller wie folgt definiert:

$$w \in L(\mathcal{K}) \iff \exists q \in Q : (q^{\text{init}}, w, Z^{\text{init}}) \triangleright^* (q, \varepsilon, \varepsilon)$$

Alternativ dazu können wir die Akzeptanz von Kellerautomaten über akzeptierende Zustände definieren. Ein acc-NPDA ist ein 7-Tupel

$$\mathcal{K}_{\text{acc}} = (\Sigma, Q, \Gamma, q^{\text{init}}, Z^{\text{init}}, \delta, F)$$

mit $F \subseteq Q$. Für einen solchen Automaten gilt:

$$w \in L(\mathcal{K}_{\text{acc}}) \iff \exists q_f \in F, \gamma \in \Gamma^* : (q^{\text{init}}, w, Z^{\text{init}}) \triangleright^* (q_f, \varepsilon, \gamma).$$

Beide Akzeptanzarten beschreiben dieselbe Klasse von Sprachen. In dieser Aufgabe sollen Sie diese Äquivalenz zeigen, indem Sie Konstruktionen in beide Richtungen angeben.

- (a) Sei $\mathcal{K}$ ein NPDA. Konstruieren Sie einen acc-NPDA $\mathcal{K}_{\text{acc}}$ mit $L(\mathcal{K}) = L(\mathcal{K}_{\text{acc}})$.
 (b) Sei $\mathcal{K}_{\text{acc}}$ ein acc-NPDA. Konstruieren Sie einen NPDA $\mathcal{K}$ mit $L(\mathcal{K}) = L(\mathcal{K}_{\text{acc}})$.

Hinweis: Es genügt, die jeweilige Konstruktion anzugeben. Sie müssen *nicht* formal beweisen, dass die Konstruktionen korrekt sind.

Lösung:

- (a) Sei $\mathcal{K} = (\Sigma, Q, \Gamma, q^{\text{init}}, Z^{\text{init}}, \delta)$ ein NPDA, der über den leeren Keller akzeptiert.

Wir verwenden in unserer Konstruktion folgende Idee: Unter den Keller von $\mathcal{K}$ legen wir ein neues Kellerbodensymbol und simulieren anschließend den Automaten $\mathcal{K}$. Sobald dieses neue Kellerbodensymbol oben auf dem Keller liegt, ist der ursprüngliche Keller leer und der Automat kann in einen akzeptierenden Zustand wechseln.

Sei $\# \notin \Gamma$ ein neues Kellersymbol und seien $q_{\text{new}}^{\text{init}}, q_f \notin Q$ neue Zustände. Wir setzen

$$Q' = Q \cup \{q_{\text{new}}^{\text{init}}, q_f\}, \quad \Gamma' = \Gamma \cup \{\#\}, \quad F = \{q_f\}.$$

Der gesuchte acc-NPDA ist

$$\mathcal{K}_{\text{acc}} = (\Sigma, Q', \Gamma', q_{\text{new}}^{\text{init}}, \#, \delta', F).$$

Die Übergangsfunktion δ' definieren wir wie folgt. Zunächst startet der neue Automat die Simulation von $\mathcal{K}$ durch

$$\delta'(q_{\text{new}}^{\text{init}}, \varepsilon, \#) = \{(q^{\text{init}}, Z^{\text{init}}\#)\}.$$

Für alle $q \in Q$, $a \in \Sigma \cup \{\varepsilon\}$ und $A \in \Gamma$ übernehmen wir die alten Übergänge, also

$$\delta'(q, a, A) = \delta(q, a, A).$$

Zusätzlich setzen wir für alle $q \in Q$

$$\delta'(q, \varepsilon, \#) = \{(q_f, \#)\}.$$

Für alle nicht genannten Argumente ist δ' leer.

- (b) Sei $\mathcal{K}_{\text{acc}} = (\Sigma, Q, \Gamma, q^{\text{init}}, Z^{\text{init}}, \delta, F)$ ein acc-NPDA.

Wir verwenden in unserer Konstruktion eine ähnliche Idee wie oben: Der neue Automat simuliert zunächst den Automaten $\mathcal{K}_{\text{acc}}$. Sobald ein akzeptierender Zustand erreicht ist, wechselt er in einen neuen Zustand, in dem er nur noch den Keller leert.

Sei $\# \notin \Gamma$ ein neues Kellersymbol und seien $q_{\text{new}}^{\text{init}}, q_{\text{clear}} \notin Q$ neue Zustände. Wir setzen

$$Q' = Q \cup \{q_{\text{new}}^{\text{init}}, q_{\text{clear}}\}, \quad \Gamma' = \Gamma \cup \{\#\}.$$

Der gesuchte NPDA ist

$$\mathcal{K} = (\Sigma, Q', \Gamma', q_{new}^{\text{init}}, \#, \delta').$$

Die Übergangsfunktion δ' definieren wir wie folgt. Zunächst startet der neue Automat die Simulation von $\mathcal{K}_{\text{acc}}$ durch

$$\delta'(q_{new}^{\text{init}}, \varepsilon, \#) = \{(q^{\text{init}}, Z^{\text{init}}\#)\}.$$

Für alle $q \in Q$, $a \in \Sigma \cup \{\varepsilon\}$ und $A \in \Gamma$ übernehmen wir die alten Übergänge, außer dass wir bei akzeptierenden Zuständen zusätzlich den Wechsel in den Löschzustand erlauben:

$$\delta'(q, a, A) = \delta(q, a, A) \cup \begin{cases} \{(q_{\text{clear}}, \varepsilon)\}, & \text{falls } q \in F \text{ und } a = \varepsilon, \\ \emptyset, & \text{sonst.} \end{cases}$$

Für alle $q_f \in F$ ergänzen wir außerdem

$$\delta'(q_f, \varepsilon, \#) = \{(q_{\text{clear}}, \varepsilon)\}.$$

Im Zustand q_{clear} wird der Keller geleert, das heißt für alle $A \in \Gamma'$ gilt

$$\delta'(q_{\text{clear}}, \varepsilon, A) = \{(q_{\text{clear}}, \varepsilon)\}.$$

Für alle nicht genannten Argumente ist δ' leer.