Prof. T. Nipkow
K. Kappelmann, J. Rädle, L. Stevens

KEINE ABGABE

Technische Universität München
Lehrstuhl für Logik und Verifikation

Einführung in die Theoretische Informatik

Sommersemester 2020 – Übungsblatt Lösungsskizze 6

AUFGABE 6.1. (*Wichtige Begriffe*) Stufe A

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

- PDA
- Kelleralphabet und -höhe
- Unterschied $\mathsf{L}_\varepsilon(A)$ und $\mathsf{L}_\mathsf{F}(A)$ für einen PDA A
- DCFL und DPDA

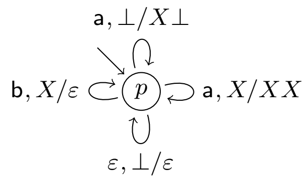
Notation von PDA-Regeln: Anstatt der in den Folien verwendeten Schreibweise $(q, YZ) \in \delta(p, \mathsf{a}, X)$ für die Ersetzungsregeln eines PDA kann man alternativ $pX \xrightarrow{\mathsf{a}} qYZ$ schreiben wobei $p, q \in \mathsf{Q}$, $X, Y, Z \in \Gamma$, $\mathsf{a} \in \Sigma$.
Beispiel: Den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

$$\delta(p, \mathsf{a}, \bot) = \{(p, X\bot)\} \qquad \delta(p, \mathsf{a}, X) = \{(p, XX)\} \qquad \delta(p, \mathsf{b}, X) = \{(p, \varepsilon)\} \qquad \delta(p, \varepsilon, \bot) = \{(p, \varepsilon)\}$$

schreibt man alternativ:

$$p\bot \xrightarrow{\mathsf{a}} pX\bot \qquad pX \xrightarrow{\mathsf{a}} pXX \qquad pX \xrightarrow{\mathsf{b}} p \qquad p\bot \xrightarrow{\varepsilon} p$$

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



AUFGABE 6.2. (*Automata Tutor*) Stufe B

In Automata Tutor finden Sie unter der Kategorie *PDA Construction* neue Übungsaufgaben zu PDA's. Dort haben Sie auch die Möglichkeit Ihre PDA's auf konkreten Eingaben zu simulieren. Beachten Sie, dass die Aufgaben *Skylines* und *Logical formulae* Hausaufgaben sind.

AUFGABE 6.3. (*Pushdown-Automata / Kellerautomaten*) Stufe B/C

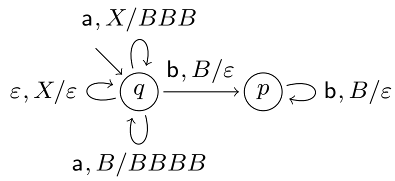
Geben Sie für die folgenden Sprachen jeweils einen Kellerautomaten A_i in allen oben aufgeführten Darstellungsarten an, so dass $\mathsf{L}_i = \mathsf{L}(\mathsf{A}_i)$. Der Automat soll mit **leerem Stack** akzeptieren. Geben Sie dann zusätzlich für jeden Automaten jeweils ein nicht-leeres Wort w mit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

(a) $\mathsf{L}_1 = \{\mathsf{a}^n\mathsf{b}^{3n} \mid n \geq 0\}$
(b) $\mathsf{L}_2 = \{\mathsf{a}^n\mathsf{b}^m \in \{\mathsf{a}, \mathsf{b}\}^* \mid n \leq m \leq 2n\}$
(c) $\mathsf{L}_3 = \{w \in \{\mathsf{a}, \mathsf{b}\}^* \mid 2 \cdot |w|_\mathsf{a} = 3 \cdot |w|_\mathsf{b}\}$

Lösungsskizze

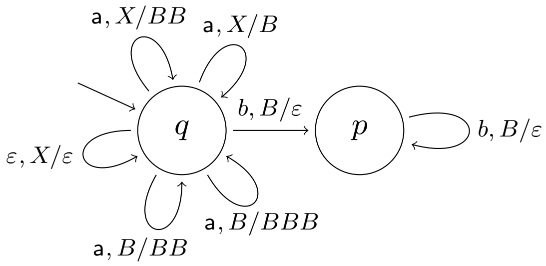
Wir geben die PDAs in Form einer Skizze des Automatens an. Für jeden der folgenden PDAs initialisieren wir den Stack mit dem Symbol X.

(a)



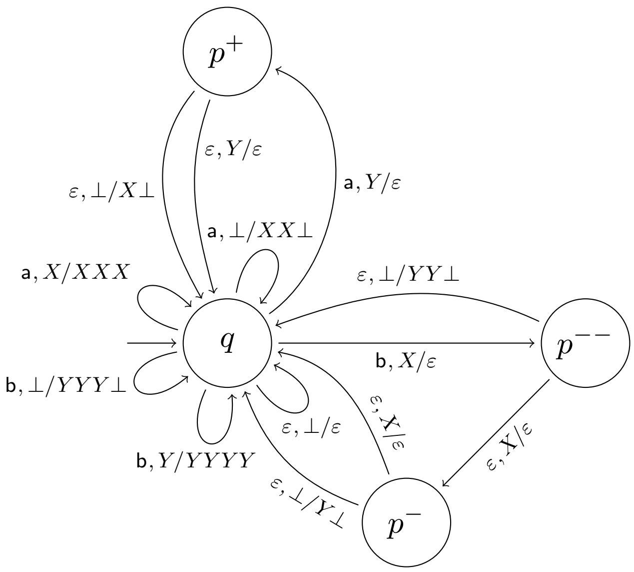
Akzeptierender Lauf: $(q, \mathsf{abbb}, X) \to (q, \mathsf{bbb}, BBB) \to (p, \mathsf{bb}, BB) \to (p, \mathsf{b}, B) \to (p, \varepsilon, \varepsilon)$

(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.



Akzeptierender Lauf: $(q, \mathsf{aabbb}, X) \to (q, \mathsf{abbb}, BB) \to (q, \mathsf{bbb}, BBB) \to (p, \mathsf{bb}, BB) \to (p, \mathsf{b}, B) \to (p, \varepsilon, \varepsilon)$

(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.



Akzeptierender Lauf: $(q, \mathsf{abbaa}, \perp) \to (q, \mathsf{bbaa}, XX\perp) \to (p^{--}, \mathsf{baa}, X\perp) \to (p^-, \mathsf{baa}, \perp) \to (q, \mathsf{baa}, Y\perp) \to (q, \mathsf{aa}, YYYY\perp) \to (p^+, \mathsf{a}, YYY\perp) \to (q, \mathsf{a}, YY\perp) \to (p^+, \varepsilon, Y\perp) \to (q, \varepsilon, \perp) \to (q, \varepsilon, \varepsilon)$

AUFGABE 6.4. (*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 $\mathsf{G} = (\{\mathsf{S}\}, \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}, \mathsf{P}, \mathsf{S})$ mit Hilfe von Satz 4.57 in einen PDA M mit $\mathsf{L}_\varepsilon(M) = \mathsf{L}(\mathsf{G})$:

$$\mathsf{S} \to \mathsf{SS} \mid \mathsf{aSb} \mid \mathsf{bSa} \mid \mathsf{c}$$

(b) Übersetzen Sie den folgenden PDA $M = (\{q, p\}, \{a, b\}, \{\perp, A, B\}, q, \perp, \delta)$ in eine CFG G mit $\mathsf{L}_\varepsilon(M) = \mathsf{L}(\mathsf{G})$, wobei δ 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:

$$\begin{aligned}\mathsf{S} &\to \mathsf{SS} \mid \mathsf{aS}B \mid \mathsf{bS}A \mid \mathsf{c}\\ A &\to \mathsf{a}\\ B &\to \mathsf{b}\end{aligned}$$

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

$$\begin{aligned} q\mathsf{S} &\xrightarrow{\varepsilon} q\mathsf{SS}\\ q\mathsf{S} &\xrightarrow{a} q\mathsf{S}B\\ q\mathsf{S} &\xrightarrow{b} q\mathsf{S}A\\ q\mathsf{S} &\xrightarrow{c} q\varepsilon\\ qA &\xrightarrow{a} q\varepsilon\\ qB &\xrightarrow{b} q\varepsilon\end{aligned}$$

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

$$\begin{array}{lcl} \mathsf{S} & \to & [q, \perp, p] \mid [q, \perp, q]\\ [q, \perp, q] & \to & a[q, A, q]\\ [q, \perp, p] & \to & a[q, A, p]\\ [q, A, q] & \to & a[q, A, q][q, A, q] \mid a[q, A, p][p, A, q]\\ [q, A, p] & \to & a[q, A, q][q, A, p] \mid a[q, A, p][p, A, p] \mid b\\ [p, A, p] & \to & b\\ [q, \perp, q] & \to & b \end{array}$$

AUFGABE 6.5. (*PDA Einschränkungen*) Stufe D

(a) Wir beschränken die Größe des Kelleralphabets Γ 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.
(b) Wir beschränken die Kellerhöhe von PDAs auf maximal k Kellerzeichen und nennen diese PDAs *k-bounded-Stack-PDA*. Insbesonders 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

(a) Wir encodieren binär die Stacksymbole Γ mit Hilfe von Γ'. 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.
(b) 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 $\mathsf{L}_\varepsilon(M)$ betrachten, definieren wir $F' = \{(q, \varepsilon) \mid q \in Q\}$. Falls wir $\mathsf{L}_F(M)$ betrachten, definieren wir $F' = \{(q, \alpha) \in Q' \mid q \in F\}$.

AUFGABE 6.6. (*Deterministische PDAs*) Stufe D

In der Vorlesung haben Sie Lemma 4.68 ohne Beweis gesehen:
Sei $\mathsf{L} \subseteq \Sigma^*$. Dann sind äquivalent:
(a) Es gibt einen DPDA D mit $\mathsf{L}_\varepsilon(D) = \mathsf{L}$
(b) Es gibt einen DPDA D' mit $\mathsf{L}_F(D') = \mathsf{L}$ **und** kein Wort aus L ist ein echtes Präfix von einem anderen Wort aus L.

Zeigen Sie diese Äquivalenz.
Bemerkung: Zu dieser Aufgabe gibt es keine Videolösung. Eine textuelle Lösung finden Sie vorab auf Moodle/der Vorlesungswebsite.
Tipp: Einen sehr ähnlichen Beweis haben Sie bereits in der Vorlesung gesehen.

Lösungsskizze
Die Konstruktionen verlaufen analog zu Satz 4.53 und 4.54:

- (a) $\Longrightarrow$ (b): Verwende die Konstruktion von Satz 4.54. Erinnerung: Sei D ein DPDA mit $\mathsf{L}_\varepsilon(D) = \mathsf{L}$, Zustandsmenge Q, Startzustand q_0 und initialem Kellerinhalt Z_0. Wir erweitern D um ein explizites Bottom-Symbol $\perp$ und neuen Startzustand q_0'. Wir fügen Transitionen $q_0'\perp \xrightarrow{\varepsilon} q_0 Z_0 \perp$ und $q\perp \xrightarrow{\varepsilon} q_F$ für alle $q \in \mathsf{Q}$ ein und setzen q_F als neuen und einzigen Endzustand. Der so erhaltene PDA ist weiterhin deterministisch mit $\mathsf{L}_F(D') = \mathsf{L}_\varepsilon(D)$ (nach VL).
  Seien nun $u, uv \in \mathsf{L}$. Da D deterministisch ist, muss D nach Lesen von u stets in derselben Konfiguration sein. Insbesondere musst der Stack leer sein, womit uv nur für $v = \varepsilon$ akzeptiert werden kann.
- (a) $\Longleftarrow$ (b):
  Sei (1) D ein DPDA mit $\mathsf{L}_F(D) = \mathsf{L}$ und (2) kein Wort aus L ein echtes Präfix eines weiteren Wortes aus L. Wir erweiteren D zu D' analog zu Satz 4.53. Um allerdings deterministisch zu bleiben, löschen wir dabei alle Transitionen ausgehend von Endzuständen aus D und fügen stattdessen nur den Epsilon-Übergang in den neuen Zustand $\bar{q}$ ein, in dem der Keller geleert wird. Formell setzen wir also $\delta'(f, a, Z) := \emptyset$ und den Fall $\delta'(f, \varepsilon, Z) = \delta(f, \varepsilon, Z) \cup \{(\bar{q}, \varepsilon)\}$ ersetzen wir durch $\delta'(f, \varepsilon, Z) := \{(\bar{q}, \varepsilon)\}$. Das heißt, beim ersten Erreichen einer Konfiguration mit Endzustand wird einfach der Stack deterministisch geleert wird.
  Das Löschen der Übergänge aus vorherigen Endzuständen verkleinert dabei die erkannte Sprache nicht. Wir beweisen dies per Widerspruch:
  Annahme: Sei $w \in \mathsf{L}_F(D) \setminus \mathsf{L}_\varepsilon(D')$. Dann muss die eindeutige akzeptierende Berechnung von D auf w mindestens einen zweiten Endzustand vor der Akzeptanz besuchen. Falls nach dem ersten Besuch eines Endzustands nur noch Epsilons gelesen werden, wird w auch von D' akzeptiert (durch Leeren des Kellers nach Besuch des ersten Endzustands). Widerspruch zur Annahme. Falls nicht nur Epsilons gelesen werden, gibt es ein echtes Präfix von w, das von D akzeptiert wird und damit in L liegt. Widerspruch zu (2). Es kann somit kein solches w geben.
  Mit Beweis des Satzes 4.53 folgt nun $\mathsf{L}_F(D) = \mathsf{L}_\varepsilon(D')$.

AUFGABE 6.7. (*Zählerautomaten*) Stufe E

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

(a) Geben Sie eine äquivalente Formulierung als Automaten, die statt des Kellers einen Zähler verwenden, an.
(b) Geben Sie einen solchen Automaten an, der eine nicht reguläre Sprache akzeptiert.
(c) 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.
(d) Geben Sie eine kontextfreie Sprache an, die von keiner der betrachteten Automatenklassen akzeptiert wird.

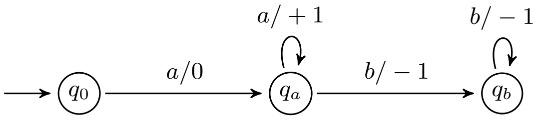
Die folgenden Aufgaben sind optional und werden nicht besprochen: Alle Aufgaben können unabhängig voneinander gelöst werden:

(a) Geben Sie einen solchen Automaten an, der die Sprache $\overline{L}$ für $L = \{ww^R \mid w \in \{\mathsf{a}, \mathsf{b}\}^*\}$ akzeptiert.
(b) Nun betrachten wir deterministische Varianten dieser Automaten. Zeigen Sie zunächst, dass die Klasse dieser deterministischen Automaten unter dem Sprachkomplement abgeschlossen ist.
(c) 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?
(d) Zeigen Sie aus den vorherigen Aussagen, dass nichtdeterministische Automaten mit einem Zähler strikt mächtiger sind als deterministische Automaten mit beliebig vielen Zählern.

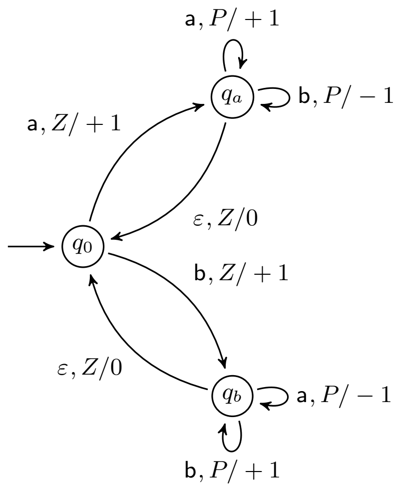
Lösungsskizze

(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 folgeden ein Akzeptanzkriterium mit Zählerwert 0 verwenden.

(b) Der folgende Automat akzeptiert die Sprache $\{\mathsf{a}^n\mathsf{b}^n \mid n \in \mathsf{N}\}$.



(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 $\{w \mid |w|_\mathsf{a} = |w|_\mathsf{b} \wedge w \in \{\mathsf{a},\mathsf{b}\}^*\}$.

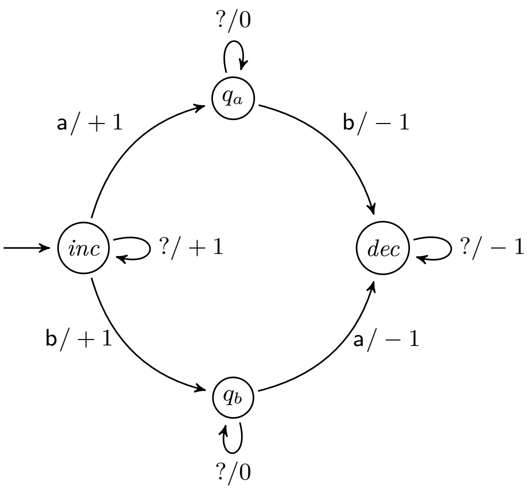


Hier bezeichnet Z, dass der Zähler gleich null ist, und P, dass der Zähler ungleich null ist. Es ist auch möglich diesen Automaten zu determinisieren.

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

Knobelaufgaben:

(a) Idee: Wir raten eine Stelle k, an der der k-te Buchstabe nicht mit dem k-letzten Buchstaben übereinstimmt. Zusätzlich müssen wir noch Wörter erkennen die ausschließlich aus as bzw. bs bestehen und ungerade Länge haben, da es bei diesen Wörtern keinen Mismatch gibt. Das lässt sich mit einem weiteren nichtdeterministischen Zweig lösen der hier nicht eingezeichnet ist.



(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. □

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 $\overline{L}$ 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 $\overline{L}$ akzeptiert, ist diese Klasse strikt mächtiger.