Übungen zur Vorlesung Grundbegriffe der Theoretischen Informatik

Thomas Schwentick

technische universität dortmund

Gaetano Geck, Lutz Oettershagen,
Christopher Spinrath, Marco Wilhelm

Lehrstuhl 1
Logik in der Informatik

SoSe 2018 | Übungsblatt 6 | 22.05.2018

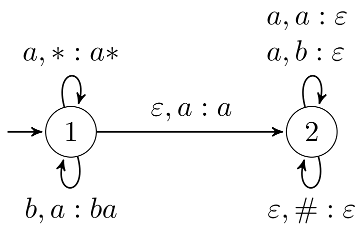
Aufgabe 6.1 [Verbindung zwischen PDAs und kontextfreien Grammatiken] **7 Punkte**

a) Es sei die kontextfreie Grammatik G durch die folgenden Regeln gegeben.

$$\begin{aligned} S &\to A \\ A &\to aAB \mid bA \mid cA \mid \varepsilon \\ B &\to a \end{aligned}$$

Bestimmen Sie einen zu G äquivalenten PDA (Kellerautomaten), der bei leerem Keller akzeptiert. Begründen Sie die Äquivalenz, indem Sie die von G erzeugte Sprache angeben und argumentieren, warum der von Ihnen konstruierte PDA diese Sprache entscheidet. **(2 Punkte)**

b) Es sei der PDA $\mathcal{A}$, der bei leerem Keller akzeptiert, wie folgt gegeben.



(5 Punkte)

(i) Entscheiden Sie für die folgenden Wörter w_i, $i = 1, 2, 3$, ob sie von $\mathcal{A}$ akzeptiert werden. Falls Sie der Meinung sind, dass $\mathcal{A}$ ein Wort w_i akzeptiert, so geben Sie eine akzeptierende Berechnung von $\mathcal{A}$ bei Eingabe von w_i an. Andernfalls begründen Sie kurz, warum das Wort nicht akzeptiert wird.

$$\begin{aligned} w_1 &= ab \\ w_2 &= abaa \\ w_3 &= abaaaa \end{aligned}$$

[3 Punkte]

(ii) Bestimmen Sie eine zu $\mathcal{A}$ äquivalente kontextfreie Grammatik. Folgen Sie dabei der Konstruktion aus der Beweisidee zu Satz 9.3 aus der Vorlesung. **[2 Punkte]**

Hinweis

Die Transitionen $(1, a, \gamma, 1, a\gamma)$ für $\gamma \in \{\#, a, b\}$ führen zu den Regeln

$$\begin{aligned}
X_{1,\#,1} &\to aX_{1,a,1}X_{1,\#,1} \mid aX_{1,a,2}X_{2,\#,1} \\
X_{1,\#,2} &\to aX_{1,a,1}X_{1,\#,2} \mid aX_{1,a,2}X_{2,\#,2} \\
X_{1,a,1} &\to aX_{1,a,1}X_{1,a,1} \mid aX_{1,a,2}X_{2,a,1} \\
X_{1,a,2} &\to aX_{1,a,1}X_{1,a,2} \mid aX_{1,a,2}X_{2,a,2} \\
X_{1,b,1} &\to aX_{1,a,1}X_{1,b,1} \mid aX_{1,a,2}X_{2,b,1} \\
X_{1,b,2} &\to aX_{1,a,1}X_{1,b,2} \mid aX_{1,a,2}X_{2,b,2}
\end{aligned}$$

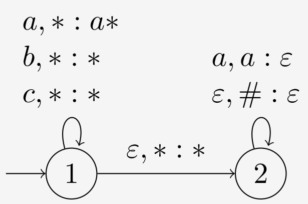
Diese Regeln sind Teil der gesuchten kontextfreien Grammatik und müssen nicht erneut angegeben werden.

Lösung:

a) Die Regeln $A \to aAB \mid bA \mid cA \mid \varepsilon$ in G erlauben es, die Zeichen a, b und c beliebig oft und in beliebiger Reihenfolge zu erzeugen. Jedesmal wenn ein a erzeugt wird, wird zudem die Satzform um die Variable B am Ende ergänzt, welche wiederum jedesmal durch ein a ersetzt werden muss, um ein Wort abzuleiten. Damit beginnt ein aus G abgeleitetes Wort mit einem Teilwort $u \in \{a, b, c\}^*$, gefolgt von ebenso vielen a's, wie a's in u enthalten sind. Die von G erzeugte Sprache lautet demnach

$$L(G) = \{uv \mid u \in \{a, b, c\}^*,\ v = a^{\#_a(u)}\}.$$

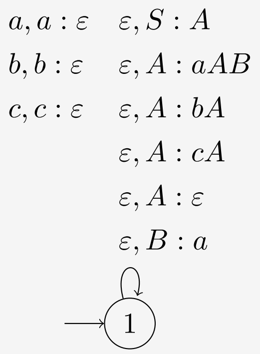
Ein zu G äquivalenter PDA, der bei leerem Keller akzeptiert, ist z.B. gegeben durch



Durch die Übergänge vom Startzustand 1 zum Zustand 1 selbst lassen sich beliebige Folgen $u \in \{a, b, c\}^*$ von a's, b's und c's einlesen. Für jedes gelesene a legt der PDA ein a im Keller ab. Für jedes im Zustand 2 gelesene a wird dann wiederum ein a aus dem Keller entfernt. Da der PDA bei leerem Keller akzeptiert, müssen im Zustand 2 demnach genauso viele a's gelesen werden, wie in Zustand 1. Der PDA akzeptiert folglich nur solche Wörter, die mit u beginnen und auf die ebenso viele a's folgen, wie a's im Keller abgelegt wurden, d.h. wie viele a's in u enthalten sind.

Alternative Konstruktion eines geeigneten PDAs:

Alternativ kann man einen geeigneten PDA auch nach der Vorschrift im Beweis zu Satz 9.2 aus der Vorlesung konstruieren. Der resultierende PDA hat die Form



wobei das unterste Kellersymbol S ist. Die Korrektheit ergibt sich dann aus dem Satz 9.2.

Hinweis

Bei der Sprache $L(G)$ handelt es sich um die Sprache L_2 aus Aufgabe 4.1b. Beide Aufgaben gemeinsam liefern die Erkenntnis, dass $L(G)$ zwar keine reguläre, aber eine kontextfreie Sprache ist.

b) **(i)** **Der PDA akzeptiert das Wort $\boldsymbol{w_1 = ab}$ nicht:** Geht der PDA nach Lesen des Zeichens a von Zustand 1 in Zustand 2 über, hat er keine Möglichkeit mehr das Zeichen b zu lesen. Alternativ befindet er sich nach vollständigem Lesen des Wortes w_1 in Zustand 1, kann nun den Keller aber nicht mehr leeren, da dies nur in Zustand 2 möglich ist, den der PDA aber nicht mehr erreichen kann.

Der PDA akzeptiert das Wort $\boldsymbol{w_2 = abaa}$ nicht: Unter Berücksichtigung der Argumentation zu w_1 befindet sich der PDA nach Lesen von ab in Zustand 1. Nun muss er mindestens ein weiteres a lesen, ehe er zu Zustand 2 wechseln kann. Dies liegt daran, dass das oberste Zeichen im Keller ein b ist, die ε-Transition zu Zustand 2 aber nur anwendbar ist, falls das oberste Zeichen ein a ist. Es folgt, dass der Kellerinhalt mindestens Länge 3 hat, wenn der PDA Zustand 2 erreicht. Dort kann er aber höchstens ein Zeichen aus dem Keller entfernen.

Der PDA akzeptiert das Wort $\boldsymbol{w_3 = abaaaa}$: Eine akzeptierende Berechnung von $\mathcal{A}$ bei Eingabe von w_3 ist

$$\begin{aligned}
& (1, abaaaa, \#) \\
\vdash\ & (1, baaaa, a\#) \\
\vdash\ & (1, aaaa, ba\#) \\
\vdash\ & (1, aaa, aba\#) \\
\vdash\ & (2, aaa, aba\#) \\
\vdash\ & (2, aa, ba\#) \\
\vdash\ & (2, a, a\#) \\
\vdash\ & (2, \varepsilon, \#) \\
\vdash\ & (2, \varepsilon, \varepsilon)
\end{aligned}$$

(ii) Für das Startsymbol S werden nach der Beweisidee zu Satz 9.3 die Regeln

$$S \to X_{1,\#,1} \mid X_{1,\#,2}$$

erzeugt. Die übrigen Übergänge des PDAs werden zudem in die folgenden Regeln überführt:

$(1, b, a, 1, ba)$	$X_{1,a,1} \to bX_{1,b,1}X_{1,a,1} \mid bX_{1,b,2}X_{2,a,1}$
	$X_{1,a,2} \to bX_{1,b,1}X_{1,a,2} \mid bX_{1,b,2}X_{2,a,2}$
$(1, \varepsilon, a, 2, a)$	$X_{1,a,1} \to \varepsilon X_{2,a,1}$
	$X_{1,a,2} \to \varepsilon X_{2,a,2}$
$(2, a, a, 2, \varepsilon)$	$X_{2,a,2} \to a$
$(2, a, b, 2, \varepsilon)$	$X_{2,b,2} \to a$
$(2, \varepsilon, \#, 2, \varepsilon)$	$X_{2,\#,2} \to \varepsilon$

Aufgabe 6.2 [PDA modellieren] 3 Punkte

Tina lebt in einer wahrlich eindimensionalen Welt. Sie kann lediglich nach Norden oder nach Süden gehen, dafür aber jeweils beliebig weit. Zu jedem Zeitpunkt entscheidet sie sich auf's Neue, ob sie

- einen Schritt Richtung Norden macht oder
- einen (gleich langen) Schritt Richtung Süden macht oder
- einen Sprung Richtung Norden (entspricht der Länge von zwei Schritten Richtung Norden) macht oder
- einen (gleich langen) Sprung Richtung Süden macht.

Geben Sie einen PDA an, der genau die Spaziergänge (= Folgen von Schritten und Sprüngen) von Tina akzeptiert, die sie zu ihrem Ausgangspunkt zurückführen. Verwenden Sie dabei das Eingabealphabet $\Sigma = \{N_1, N_2, S_1, S_2\}$ mit den Bedeutungen

- N_1 : ein Schritt Richtung Norden,
- N_2 : ein Sprung Richtung Norden,
- S_1 : ein Schritt Richtung Süden,
- S_2 : ein Sprung Richtung Süden.

Das Kelleralphabet sollen Sie selbst passend wählen. Argumentieren Sie, warum ihr PDA sinnvoll gewählt ist.

Hinweis

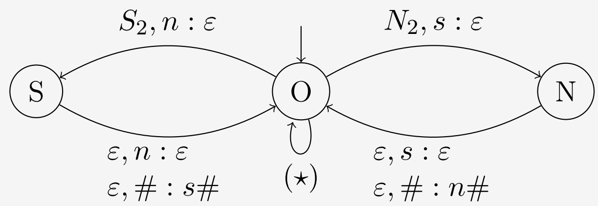
Auch eine leere Folge von Schritten und Sprüngen soll als Spaziergang gelten. Beachten Sie zudem, dass PDAs auch ε-Transitionen beinhalten dürfen.

Lösung:

Die Grundidee ist es, Tinas Abstand von ihrem Ausgangspunkt in Schrittlängen zu messen und im Keller durch eine entsprechende Anzahl an Kellersymbolen abzuspeichern. Dabei ist es wichtig, zwischen zwei Kellersymbolen zu unterscheiden, um kenntlich zu machen, ob sich Tina nördlich oder südlich von ihrem Ausgangspunkt befindet. Nur so kann beurteilt werden,

ob sie sch mit einem weiteren Schritt vom Ausgangspunkt entfernt oder sich ihm nähert. Wir entscheiden uns für das Kelleralphabet $\{\#, s, n\}$ (# ist das unterste Kellersymbol). Ferner muss darauf geachtet werden, dass Zeichenketten im Keller nie beide Symbole (s und n) enthalten, da sie sich gegeneinander „aufheben “ würden und man gegebenenfalls fälschlicherweise nicht erkennt, dass sich Tina im Ausgangspunkt befindet.
Um Schrittweiten von Sprüngen korrekt im Keller repräsentieren zu können, ist es wichtig zu wissen, welches (falls vorhanden) das zweitoberste Symbol im Keller ist. Diese Abfrage wird realisiert, indem Sprünge in zweit Teilschritten abgearbeitet werden.

Ein PDA, der genau die Spaziergänge akzeptiert, die Tina wieder zurück zu ihrem Ausgangspunkt führen, ist z.B. der folgende PDA, der mit leerem Keller akzeptiert:



wobei an der Stelle $(\star)$ steht:

$\varepsilon, \# : \varepsilon$	
$N_1, \# : n\#$	$S_1, \# : s\#$
$N_1, s : \varepsilon$	$S_1, n : \varepsilon$
$N_1, n : nn$	$S_1, s : ss$
$N_2, \# : nn\#$	$S_2, \# : ss\#$
$N_2, n : nnn$	$S_2, s : sss$

Im Keller wird, wie erläutert, die Entfernung vom Ausgangspunkt in Schrittlängen gemessen (z.B. bedeutet der Kellerinhalt $nnnn\#$, dass sich Tina vier Schrittlängen nördlich von ihrem Ausgangspunkt befindet, während $sss\#$ meint, dass Tina drei Schrittlängen südlich ist). Der PDA ist so konstruiert, dass sich zu keiner Zeit sowohl ein n als auch ein s im Keller befinden kann. Dadurch ist sicher gestellt, dass man durch Auslesen des obersten Zeichens im Keller den groben, momentanen Aufenthaltsort von Tina erfährt (nördlich vom Ausgangspunkt, südlich vom Ausgangspunkt, im Ausgangspunkt). Durch das oberste Zeichen im Keller wird damit eindeutig bestimmt, wie bei einem Lesen von N_1 (bzw. S_1) vorzugehen ist: Befindet sich Tina am Ausgangspunkt oder nördlich davon, wird ein n im Keller abgelegt; befindet sich Tina südlich, wird ein s entfernt (für S_1 umgekehrt). Dies alles ist in Zustand O realisiert.
Werden N_2 (bzw. S_2) gelesen, gestaltet sich die Situation schwieriger, da ggf. die obersten beiden Zeichen im Keller in Betracht gezogen werden müssen. Wir diskutieren den Fall N_2 (S_2 geht wieder analog): Befindet sich Tina am Ausgangspunkt oder nördlich davon, werden einfach zwei nn im Keller abgelegt; befindet sich Tina südlich, hängt das weitere Vorgehen davon ab, ob Tina genau eine Schrittlänge oder noch weiter südlich ist. Zunächst wird das oberste s aus dem Keller entfernt und ein Zustandswechsel (O nach N) durchgeführt. N fungiert als Hilfszustand, in dem überprüft wird, welches nun das oberste Zeichen im Keller ist. Ist es #, so befand sich Tina vor dem Sprung nach Norden (N_2) nur einen Schritt weit südlich vom Ausgangspunkt und sie springt über ihren Ausgangspunkt hinweg. Entsprechend wird ein n im Keller abgelegt und in Zustand O gewechselt. Ist es s, so kann ein weiteres s aus dem Keller entfernt und in den Zustand O gewechselt werden.

Wenn weder ein s noch ein n im Keller abgelegt sind, befindet sich Tina im Ausgangspunkt und der Keller kann durch $(\varepsilon, \# : \varepsilon)$ für die Akzeptanz geleert werden.

Aufgabe 6.3 [Pumping-Lemma für kontextfreie Sprachen] **5 Punkte**

Zeigen Sie mit dem Pumping-Lemma für kontextfreie Sprachen, dass die folgenden Sprachen nicht kontextfrei sind.

a) $L_1 = \{a^\ell b^m c^p d^q \mid \ell, m, p, q \in \mathbb{N}_0, \ell < p \text{ und } q < m\}$ **(2,5 Punkte)**

b) $L_2 = \{ww^R w \mid w \in \{a, b\}^*\}$, wobei w^R, wie aus den Übungen bekannt, für das Spiegelwort von w steht. **(2,5 Punkte)**

Hinweis

Wählen Sie in Teilaufgabe b) für den Nachweis mit dem Pumping-Lemma das Wort $z = ba^n bba^n bba^n b \in L_2$ mit $n \in \mathbb{N}$ beliebig. Sie können ohne Beweis verwenden, dass L_2 nur solche Wörter der Form $ba^h bba^i bba^j b$ enthält, die $h = i = j$ erfüllen.

Lösung:

a) Wir verwenden das Korollar 10.2 des Pumping-Lemmas aus der Vorlesung. Es sei $n \in \mathbb{N}$ beliebig. Wähle $z = a^n b^{n+1} c^{n+1} d^n$, womit $|z| \geq n$ und $z \in L_1$ gelten.

Es sei nun $z = uvwxy$ eine beliebige Zerlegung, welche die Bedingungen

(1) $vx \neq \varepsilon$,
(2) $|vwx| \leq n$,

erfüllt. Nach Bedingung (1) enthält vx mindestens ein Zeichen. Wir betrachten vier Fälle, die sich gegenseitig nicht ausschließen:

- ***vx* enthält mindestens ein *a*.** Wegen $|vwx| \leq n$ enthält vx kein c. Wähle $k = 2$. Das Wort uv^2wx^2y ist nicht in L_1 enthalten, da in uv^2wx^2y mindestens so viele as wie cs enthalten sind.
- ***vx* enthält mindestens ein *b*.** Wegen $|vwx| \leq n$ enthält vx kein d. Wähle $k = 0$. Das Wort $uv^0wx^0y = uwy$ ist nicht in L_1 enthalten, da uwy höchstens so viele bs wie ds enthält.
- ***vx* enthält mindestens ein *c*.** Wegen $|vwx| \leq n$ enthält vx kein a. Wähle $k = 0$. Das Wort $uv^0wx^0y = uwy$ ist nicht in L_1 enthalten, da in uwy höchstens so viele cs wie as vorkommen.
- ***vx* enthält mindestens ein *d*.** Wegen $|vwx| \leq n$ enthält vx kein b. Wähle $k = 2$. Das Wort uv^2wx^2y ist nicht in L_1 enthalten, da in uv^2wx^2y mindestens so viele ds wie bs vorkommen.

Da es in allen Fällen ein k gibt, so dass uv^kwx^ky nicht in L_1 enthalten ist, ist L_1 nach dem Pumping-Lemma für kontextfreie Sprachen nicht kontextfrei.

b) Wir verwenden das Korollar 10.2 des Pumping-Lemmas aus der Vorlesung. Nach der angegebenen Wahl von $z = ba^nbba^nbba^nb$ gelten $|z| \geq n$ und $z \in L_2$. Wir bezeichnen den Teilstring von z, der mit dem ersten b beginnt und dem zweiten b endet, als *ersten Block*, den Teilstring, der mit dem dritten b beginnt und dem vierten b endet, als *zweiten Block* und den Teilstring, der der mit dem fünften b beginnt und dem letzten b endet, als *dritten Block*.

Es sei nun $z = uvwxy$ eine beliebige Zerlegung von z, welche die Bedingungen

(1) $vx \neq \varepsilon$

(2) $|vwx| \leq n$

erfüllt. Nach Bedingung (1) enthält vx mindestens ein Zeichen. Wir betrachten zwei sich ausschließende Fälle:

- **vx enthält mindestens ein b.** Wegen (2) kann vx maximal zwei bs enthalten. Wir wählen $k = 0$. Das Wort $z' = uv^0wx^0y = uwy$ enthält dann vier oder fünf bs. Für jedes Wort aus L_2 muss aber gelten, dass die Anzahl der bs in dem Wort durch drei teilbar ist (ist $s = tt^Rt \in L_2$, dann gilt $\#_b(s) = 3 \cdot \#_b(t)$). Also gilt $z' \notin L_2$.
- **vx enthält keine bs.** Wir unterscheiden drei Fälle:
  - <u>vx enthält mindestens ein a aus dem ersten Block.</u> Aufgrund von (2) kann vx keine as aus dem dritten Block enthalten. Sei $vx = a^l$ mit $1 \leq l \leq n$. Wir wählen $k = 0$. Sei $z' = uv^0wx^0y = uwy$. Da z'
    - ∗ mit einem b beginnt,
    - ∗ genau sechs bs beinhaltet und
    - ∗ das zweite und dritte b bzw. das vierte und fünfte b direkt aufeinanderfolgen,

    muss jedes Wort t mit $z' = tt^Rt$ mit genau einem b beginnen und genau einem b enden, wobei zwischen diesen bs höchstens as vorkommen können. Während zwischen dem fünften und sechsten b (also im letzten Block) von z' genau n as enthalten sind, befinden sich zwischen dem ersten und dem vierten b (also in den ersten beiden Blöcken) aber insgesamt $2n - l < 2n$ as. Also gilt $z' \notin L_2$.
  - <u>vx enthält mindestens ein a aus dem dritten Block.</u>
    (analog zum vorigen Fall)
  - <u>vx enthält nur as aus dem zweiten Block.</u> Es sei $vx = a^l$ mit $1 \leq l \leq n$. Wir wählen $k = 0$. Es sei $z' = uv^0wx^0y = uwy$. Auch hier gilt, dass jedes Wort t mit $z' = tt^Rt$ mit genau einem b beginnen und genau einem b enden muss, wobei zwischen diesen beiden bs höchstens as vorkommen können. Während im ersten und dritten Block von z' insgesamt $2n$ as enthalten sind, befinden sich im zweiten Block nur $n - l < n$ as. Also kann z' nicht in L_2 enthalten sein.

Da es in allen Fällen ein k gibt, so dass uv^kwx^ky nicht in L_2 enthalten ist, ist L_2 nach dem Pumping-Lemma für kontextfreie Sprachen nicht kontextfrei.