Prof. Javier Esparza
Philipp Czerner, Kevin Kappelmann, Valentin Krasotin

Technische Universität München
Lehrstuhl für Theoretische Informatik

Einführung in die Theoretische Informatik

Sommersemester 2024 – Hausaufgabenblatt 8

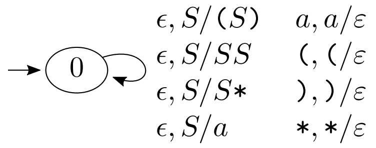
- Die Aufgaben werden in folgender Reihenfolge korrigiert: H8.2, H8.3.
- Die Knobelaufgabe bitte separat auf Moodle abgeben. Sie wird korrigiert.

AT-Aufgabe H8.1. (*almost recfg*) unkorrigiert (4 Punkte)

Bearbeiten Sie diese Aufgabe auf Automata Tutor. Sei $\Sigma := \{a, \texttt{(}, \texttt{)}, \texttt{*}\}$ ein Alphabet und G die kontextfreie Grammatik, die über folgende Produktionen gegeben ist: $S \to \texttt{(}S\texttt{)} \mid SS \mid S\texttt{*} \mid a$.

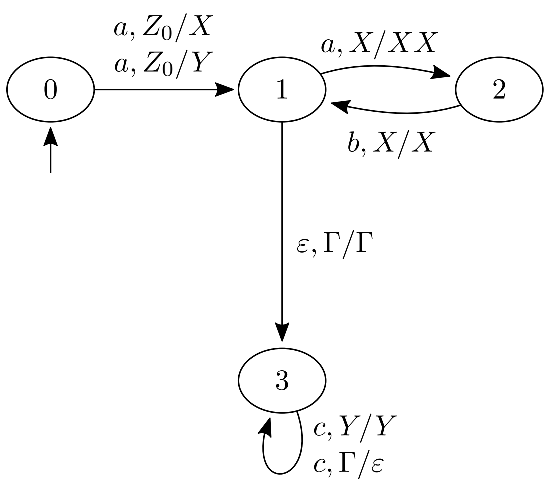
Konstruieren Sie einen PDA für $L(G)$, der über leeren Keller akzeptiert.

Lösungsskizze. Wir erstellen den PDA mit dem Verfahren aus der Vorlesung. Das initiale Kellersymbol ist S, und der PDA akzeptiert über leeren Keller.

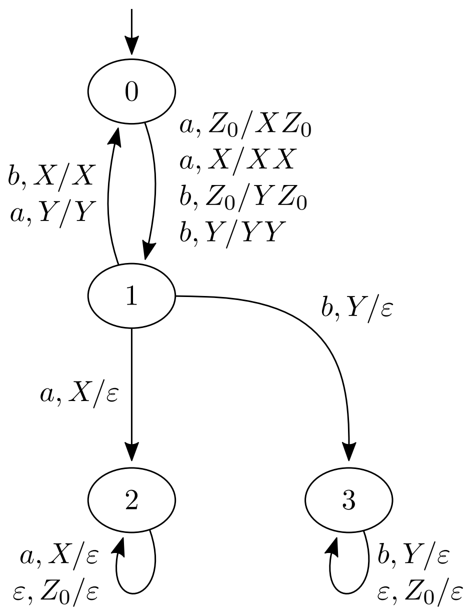


Aufgabe H8.2. (*Sprachfindung*) 2+2+2+2 Punkte

Geben Sie für die folgenden Kellerautomaten jeweils die akzeptierte Sprache in Mengennotation an. (Ihre Angabe soll sich insbesondere nicht auf den PDA beziehen.) Die PDAs akzeptieren über leeren Keller. Bitte beachten Sie die Hinweise zur Notation von PDAs auf Übungsblatt 8.

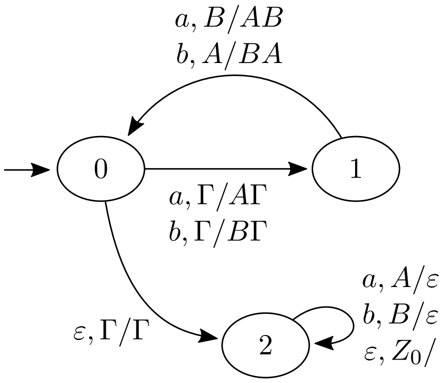


(a)

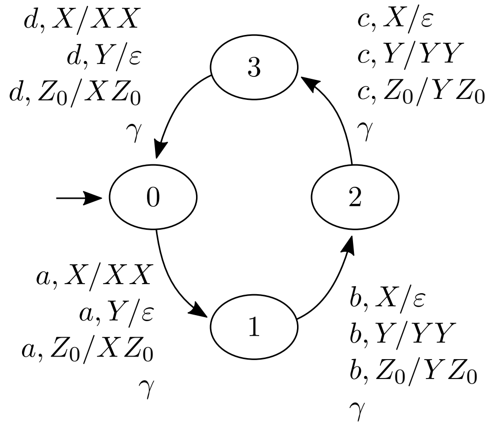


(b)

$\gamma := (\ \varepsilon, \Gamma/\Gamma \quad \varepsilon, Z_0/\varepsilon\ )$



(c)



(d)

Lösungsskizze.

(a) $\{a(ab)^n c^{n+1} : n \in \mathbb{N}\} \cup L(ac^+)$

(b) $\{(ab)^n a^{n+2} : n \in \mathbb{N}\} \cup \{(ba)^n b^{n+2} : n \in \mathbb{N}\}$, oder auch $\{a(ba)^n a^{n+1} : n \in \mathbb{N}\} \cup \{b(ab)^n b^{n+1} : n \in \mathbb{N}\}$

(c) $\{ww^R : w \in L((ab|ba)^*)\}$

(d) $\{w \in \{a,b,c,d\}^* : |w|_a + |w|_d = |w|_b + |w|_c\}$

Aufgabe H8.3. (*I made a typ0*) 3+7+2 Punkte

Erinnerung: Eine Grammatik vom Typ 0 kann sowohl Terminale als auch Nichtterminale auf beiden Seiten ihrer Produktionen besitzen, d.h. $P \subseteq (V \cup \Sigma)^* \times (V \cup \Sigma)^*$.

(a) Gegeben eine Grammatik $G = (V, \Sigma, P, S)$, sei $G' := (V \cup \{S'\}, \Sigma, P', S')$, wobei $S' \notin V$ und $P' := P \cup \{S' \to SS\}$. Geben Sie eine Grammatik G mit $L(G') \neq L(G)L(G)$ und maximal 3 Produktionen an. Begründen Sie die Korrektheit Ihres Beispiels.

(b) Geben Sie ein Verfahren an, das als Input eine Grammatik $G = (V, \Sigma, P, S)$ erhält und eine Grammatik H mit $L(H) = L(G)L(G)$ und höchstens $3(|\Sigma| + |P|)$ Produktionen konstruiert. Erläutern Sie die Idee Ihres Verfahrens auch informell.

(c) Wenden Sie Ihr Verfahren auf die folgende Grammatik mit Startsymbol S an:

$$S \to XaYa \qquad aY \to bc \qquad Xb \to cc$$

Lösungsskizze.

(a) Sei $G = (\{S\}, \{a\}, \{SS \to a\}, S)$. Es gilt $L(G) = \emptyset$ und $L(G') = \{a\}$.

(b) Wir möchten die Idee aus Aufgabe (a) weiterverfolgen, müssen dabei aber verhindern, dass die beiden Ableitungen vom ursprünglichen Startsymbol durch Konkatenation die linke Seite einer Produktion bilden können. Hierfür stellen wir sicher, dass (1) Terminale nicht mehr auf der linken Seite von Produktionen vorkommen und (2) wir disjunkte Nichtterminale für die beide Ableitungen (beginnend vom neuen Startsymbol) verwenden.

Zwei Schritte:

(1) Wir ersetzen jedes Vorkommen eines Terminals a in G durch ein neues Nichtterminal X_a und fügen die Produktion $X_a \to a$ hinzu. Damit erhalten wir eine Grammatik $G_1 = (V_1, \Sigma, P_1, S)$ mit Produktionen der Gestalt $X_1 \cdots X_n \to Y_1 \cdots Y_m$ und $X_a \to a$.

(2) Sei $V_1' \coloneqq \{X' \mid X \in V_1\}$ eine Kopie von V_1, disjunkt mit V_1, und sei P_1' die Menge von Produktionen, die man erhält, indem man jedes Vorkommen eines Nichtterminals X durch X' ersetzt. Sei $S_0 \notin V_1 \cup V_1'$. Sei H die Grammatik

$$H \coloneqq (V_1 \cup V_1' \cup \{S_0\}, \Sigma, P_1 \cup P_1' \cup \{S_0 \to SS'\}, S_0).$$

Es gilt $L(H) = L(G)L(G)$: Jedes Wort aus Nichtterminalen, das von S_0 abgeleitet werden kann, ist von der Gestalt $\alpha\beta$ mit $\alpha \in V_1^*, \beta \in (V_1')^*$. Jede Produktion in P kann nur auf α und jede Produktion in P' kann nur auf β angewendet werden. Da P bzw. P' von S bzw. S' aus $L(G)$ erzeugen, folgt somit $L(H) = L(G)L(G)$.

Alternative:

Wir möchten die Idee aus Aufgabe (a) weiterverfolgen, müssen dabei aber verhindern, dass die beiden Ableitungen vom ursprünglichen Startsymbol durch Konkatenation die linke Seite einer Produktion bilden können. Hierfür stellen wir sicher, dass (1) Terminale nicht mehr auf der linken Seite der ursprünglichen Produktionen vorkommen, (2) die erzeugten Symbolketten der beiden Ableitungen durch ein neues Symbol getrennt werden und (3) das Trennsymbol erst entfernt werden kann, sobald beide Ableitungen für S beendet sind (d.h. alle Nichtterminale entfernt sind).

Zwei Schritte:

(1) Wir ersetzen jedes Vorkommen eines Terminals a in G durch ein neues Nichtterminal X_a und fügen die Produktion $X_a \to a$ hinzu. Damit erhalten wir eine Grammatik $G_1 = (V_1, \Sigma, P_1, S)$ mit Produktionen der Gestalt $X_1 \cdots X_n \to Y_1 \cdots Y_m$ und $X_a \to a$.

(2) Sei $\# \notin V_1$. Für jedes Terminalzeichen x führen wir die Produktionen $\#x \to x\#$ und $x\# \to \#x$ ein, formal $P' \coloneqq \{\#x \to x\# \mid x \in \Sigma\} \cup \{x\# \to \#x \mid x \in \Sigma\}$. Sei $S_0 \notin (V_1 \cup \{\#\})$. Sei H die Grammatik

$$H \coloneqq (V_1 \cup \{\#, S_0\}, \Sigma, P_1 \cup P' \cup \{S_0 \to \#S\#S\#, \#\#\# \to \varepsilon\}, S_0).$$

(c) 1. Schritt: $S \to XAYA, \quad AY \to BC, \quad XB \to CC, \quad A \to a, \quad B \to b, \quad C \to c$

2. Schritt: H enthält die Produktionen

$$\begin{aligned}
&S_0 \to SS',\\
&S \to XAYA, \quad AY \to BC, \quad XB \to CC, \quad A \to a, \quad B \to b, \quad C \to c\\
&S' \to X'A'Y'A', \quad A'Y' \to B'C', \quad X'B' \to C'C',\\
&A' \to a, \quad B' \to b, \quad C' \to c
\end{aligned}$$

Alternative:

$$\begin{aligned}
&S_0 \to \#S\#S\#, \quad \#x \to x\#, \quad x\# \to \#x \ (\forall x \in \{a,b,c\}), \quad \#\#\# \to \varepsilon,\\
&S \to XAYA, \quad AY \to BC, \quad XB \to CC, \quad A \to a, \quad B \to b, \quad C \to c
\end{aligned}$$

Quizaufgabe H8.4. (*Hochstaplerei*) unkorrigiert (7 Punkte)

Dora ist wütend. Ihr Kindergartenrivalin, Eva Pirsalz, behauptet einen größeren Bausteinturm als Dora gebaut zu haben. Aber als Dora sich den Turm anschauen wollte, war er nicht mehr da – Eva behauptet, sie hat ihn schon wieder abgebaut. Dora will der Sache auf den Grund gehen, und besorgt sich eine Kopie von Evas Labortagebuch. Da muss für jeden Baustein jeweils vermerkt sein, wann der er aus der Bausteinbox **g**eholt wurde, wann er **v**erbaut wurde, und wann er wieder **z**urückgelegt wurde.

Die Einträge werden mit dem Alphabet $\Sigma := \{g, v, z\}$ notiert. In Doras Labortagebuch steht z.B. das Protokoll $ggvgvvzgvzzz$. Ein Baustein muss immer zuerst geholt, dann verbaut, und dann zurückgegeben werden. Das Protokoll $gvvgzz$ wäre also nicht gültig, da hier ein Stein verbaut wurde, bevor er geholt wurde. Ebenso wäre $ggvgvvzz$ nicht in Ordnung, da einer der Steine nicht zurückgegeben wurde, oder gz, da einer der Steine nicht verbaut wurde. Sei $L := \{w \in \Sigma^* : w \text{ ist gültiges Protokoll}\}$. Um Eva zu überführen, will Dora nun einen Kellerautomaten bauen, der jedes Protokoll in Evas Labortagebuch überprüft.

(a) Zeigen Sie, dass L nicht kontextfrei ist und es somit keinen Kellerautomaten für L gibt.

Dora ist enttäuscht, hat aber noch eine Idee: Für ihre Zwecke wäre es genauso gut, einen Kellerautomaten zu haben, der die *ungültigen* Protokolle akzeptiert.

(b) Konstruieren Sie einen Kellerautomaten für $\overline{L}$ und beschreiben Sie die Idee hinter Ihrer Konstruktion.

Hinweis: Bitte beachten Sie die Anmerkungen zur Notation von Kellerautomaten auf Übungsblatt 7 und 8.

Lösungsskizze. **(a)** Angenommen, L wäre kontextfrei. Dann können wir das Pumping-Lemma für kontextfreie Sprachen anwenden.

Sei $n \in \mathbb{N}$ PL-Zahl. Wir wählen das Wort $r := g^n v^n z^n$. (Wir bezeichnen dies meist als z, das wäre hier aber ungünstig, da im Alphabet Σ. Also wählen wir r statt z für den Namen des Wortes, und f statt v für den zweiten Teil der Zerlegung.)

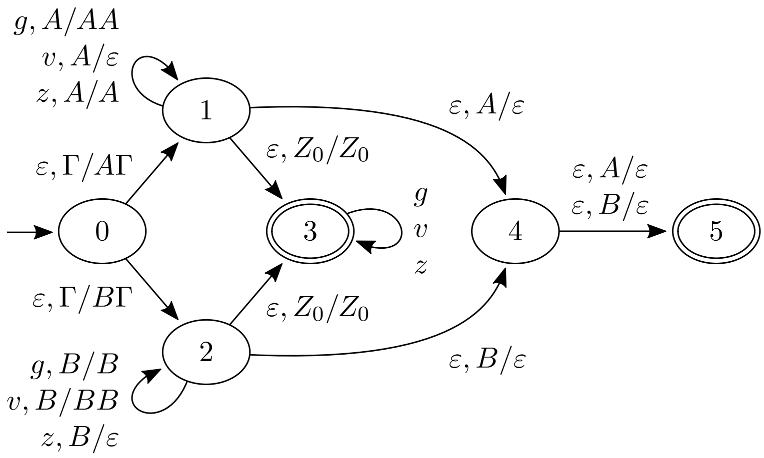
Sei $ufwxy$ eine Zerlegung unseres Wortes mit $|fwx| \leq n$ und $fx \neq \varepsilon$. Wir betrachten die folgenden Fälle:

- $|fx|_z = 0$: Wir pumpen also kein z. Wir pumpen ab, betrachten also das Wort $r' := uf^0wx^0y = uwy$. Da $fx \neq \varepsilon$, haben wir zumindest ein g oder ein v entfernt. Daher gilt $|r'|_g < |r'|_z$ oder $|r'|_v < |r'|_z$. Das heißt, ein Stein wurde zurückgegeben bevor er geholt oder verbaut wurde. Daher ist in beiden Fällen das Wort $r' \notin L$.
- $|fx|_z > 0$: Aufgrund der Form unseres Wortes und $|fwx| \leq n$ gilt also $|fwx|_g = 0$. Wir pumpen 2-mal, betrachten also $r' := uf^2wx^2y$. Es gilt $|r'|_g = n < n + |fx|_z = |r'|_z$. In r' wurde daher ein Stein zurückgegeben, bevor er geholt wurde, und $r' \notin L$.

(b) Um zu überprüfen, dass ein Wort in L enthalten ist, können wir folgendermaßen vorgehen: Wir gehen das Wort Zeichen für Zeichen durch, und merken uns stets die Anzahl der Steine die wir geholt, aber nicht verbaut haben, und die der Steine, die wir verbaut, aber nicht zurückgelegt haben. (Erstere Zahl nennen wir A, und letztere B.)

Keine dieser Zahlen darf negativ werden, und am Ende müssen beide 0 sein. Mit einem Kellerautomaten kann man dies nicht überprüfen, da er sich nur eine dieser Zahlen merken kann. Aber: Falls das Wort nicht in der Sprache ist, muss also einer dieser

Zähler negativ werden oder nicht auf 0 enden. Dies können wir überprüfen, indem wir den Nichtdeterminismus ausnutzen.



(Der PDA verwendet die auf Übungsblatt 8 eingeführte Kurznotation.) Zuerst entscheidet sich der Automat, ob er A oder B überprüft. Wir betrachten nun den Fall, dass der Automat A überprüft, das Vorgehen für B ist analog. Anfangs setzten wir den Zähler auf 1, indem er ein A auf den Stapel legt. Danach liest er das Wort ein. Jedes g legt ein weiteres A auf den Stapel, jedes v nimmt eines herunter. Es gibt zwei Möglichkeiten, wie das Wort inkorrekt sein könnte: (1) Der Zähler erreicht zwischendurch die Zahl 0 (es wurde also ein Stein verbaut, der nicht geholt wurde). (2) Der Zähler endet nicht bei 1 (es wurden nicht alle Steine verbaut).

Im Fall (1) geht der Automat zu Zustand 3 über, und kann dort den Rest des Wortes akzeptieren. (Umgekehrt kann man Zustand 3 auch nur in Fall (1) erreichen.) Bei (2) müssen am Ende des Wortes zwei A auf dem Stapel liegen (der Zähler ist also mindestens 2) – hier gehen wir nach 4 und nach 5, von wo aus keine weiteren Zeichen mehr eingelesen werden können. Auch hier gilt die Umkehrung: wir können nur nach 5 gehen, wenn wir am Ende des Wortes sind und mindestens zwei A auf dem Stapel liegen.

Knobelaufgabe H8.5. (*Die Sterne sind ganz schön hoch*)

Sei $n > 2$ und $\Sigma := \{a, b\}$. Zeigen Sie, dass es eine DFA M über Σ mit n^{100} Zuständen gibt, sodass die Sprache $L' := \{w : \{w\}^* \subseteq L(M)\}$ mindestens $n!$ Residualsprachen hat.

Hinweis: Sie dürfen bekannte Aussagen über Primzahlen ohne Beweis verwenden, etwa den Primzahlsatz, oder den Chinesischen Restsatz, und insbesondere, dass die Summe der ersten n Primzahlen kleiner als n^3 ist.