

Einführung in die theoretische Informatik
Sommersemester 2019 – Hausaufgabenblatt Lösungsskizze 6

Handschriftliche Abgabe

Formale Kriterien zu handschriftlichen Abgaben entnehmen Sie bitte der Website <https://www21.in.tum.de/teaching/theo/SS19>.

AUFGABE 6.1. (*AutomataTutor*)

2 Punkte

Lösen Sie im AutomataTutor die Aufgaben H6.1 (a–d). Beachten Sie, dass im Gegensatz zu den Übungsaufgaben für jede Teilaufgabe maximal fünf Versuche erlaubt sind.

Die Lösungen müssen Sie nicht in ihre handschriftliche Abgabe mit aufnehmen, es reicht aus sie im AutomataTutor einzureichen. Diese Aufgabe ist einzeln zu bearbeiten, die restlichen Aufgaben in Zweiergruppen.

Beachten Sie außerdem, dass die Punkte für die AutomataTutor Aufgaben nicht auf Ihrer handschriftlichen Abgabe eingetragen werden.

AUFGABE 6.2. (*CYK-Algorithmus*)

2 Punkte

Wir betrachten die Grammatik $G = (\{S, T\}, \{a, b, c\}, P, S)$ mit den folgenden Produktionen:

$$\begin{aligned} S &\rightarrow T \mid SS \mid \varepsilon \\ T &\rightarrow aSbSb \mid c \end{aligned}$$

Konstruieren Sie eine zu G äquivalente Grammatik G' in Chomsky-Normalform, und bestimmen Sie mit dem CYK-Algorithmus, ob $cabcb \in L(G')$. Geben Sie dabei die vollständige Berechnungstabelle laut Vorlesung an.

Lösungsskizze

Nichtterminale für a und b einführen:

$$\begin{aligned} S &\rightarrow T \mid SS \mid \varepsilon & T &\rightarrow ASBBSB \mid c & A &\rightarrow a & B &\rightarrow b \end{aligned}$$

Lange Produktionen aufspalten:

$$\begin{aligned} S &\rightarrow T \mid SS \mid \varepsilon & T &\rightarrow AC_1 \mid c & A &\rightarrow a & B &\rightarrow b \\ C_1 &\rightarrow SC_2 & C_2 &\rightarrow BC_3 & C_3 &\rightarrow SB \end{aligned}$$

ε -Produktionen eliminieren (streng nach Konstruktion müsste man hier noch die Produktion $S \rightarrow S$ einfügen):

$$\begin{aligned} S &\rightarrow T \mid SS & T &\rightarrow AC_1 \mid c & A &\rightarrow a & B &\rightarrow b \\ C_1 &\rightarrow SC_2 \mid C_2 & C_2 &\rightarrow BC_3 & C_3 &\rightarrow SB \mid B \end{aligned}$$

Kettenproduktionen eliminieren:

$$\begin{aligned} S &\rightarrow AC_1 \mid SS \mid c & T &\rightarrow AC_1 \mid c & A &\rightarrow a & B &\rightarrow b \\ C_1 &\rightarrow SC_2 \mid BC_3 & C_2 &\rightarrow BC_3 & C_3 &\rightarrow SB \mid b \end{aligned}$$

Das Ergebnis ist in CNF und $L(G') = L(G) \setminus \{\varepsilon\}$. Da wir nur $cabcb \in L(G)$ entscheiden wollen, ist dieser Unterschied irrelevant. Um eine äquivalente Grammatik zu erhalten, müssten wir ein neues Startsymbol S' mit den Produktionen $S' \rightarrow AC_1 \mid SS \mid c \mid \varepsilon$ hinzufügen. Nach dem CYK-Algorithmus ergibt sich die folgende Berechnungstabelle:

¹⁵ S					
¹⁴ ∅	²⁵ S, T				
¹³ ∅	²⁴ ∅	³⁵ C ₁ , C ₂			
¹² ∅	²³ ∅	³⁴ ∅	⁴⁵ C ₃		
¹¹ S, T	²² A	³³ C ₃ , B	⁴⁴ S, T	⁵⁵ C ₃ , B	
	c	a	b	c	b

Damit ist $cabcb \in L(G')$.

AUFGABE 6.3. (*Kontextfrei oder nicht?*)

2 Punkte

Entscheiden Sie ob die folgenden Sprachen kontextfrei sind. Wenn ja, geben Sie eine Grammatik an und zeigen Sie, dass Ihre Grammatik die Sprache akzeptiert. Wenn nein, beweisen Sie dies durch einen Widerspruchsbeweis unter Verwendung des Pumping Lemmas für kontextfreie Sprachen.

(a) $L_1 = \{a^n b^m c^l \mid n, m, l \in \mathbb{N} \wedge n > m \wedge n > l\}$

(b) $L_2 = \{ww \mid w \in \{a, b, c\}^* \wedge w = w^R\}$

Lösungsskizze

(a) L_1 ist nicht kontextfrei. Widerspruchsbeweis unter Verwendung des Pumping Lemmas für kontextfreie Sprachen:

- (*) Angenommen, L_1 wäre kontextfrei. Dann können wir das Pumping-Lemma für kontextfreie Sprachen anwenden.
- (*) Sei $n \in \mathbb{N}$ eine Pumping-Lemma-Zahl für L .
- (*) Es gilt $z = a^{n+1} b^n c^n \in L$ und $|z| > n$.
- Es gibt also für z eine Zerlegung $z = uvwxy$ mit den Eigenschaften
 - (1) $vx \neq \varepsilon$
 - (2) $|vwx| \leq n$
 - (3) $\forall i \in \mathbb{N}_0. uv^i wx^i y \in L_1$
- (*) Wir unterscheiden die folgenden Fälle: (Achtung: Lösungsskizze)
 - $|vx|_a > 0$: Aus (2) und der Wahl von z folgt, dass $|vwx|_c = 0$. Folglich liegt das Wort $uv^0 wx^0 y$ nicht mehr in L_1 , da es höchstens so viele a 's wie c 's enthält. Dies ist ein Widerspruch zu (3).
 - $|vx|_a = 0$: In diesem Fall muss wegen (1) $|vx|_b > 0$ oder $|vx|_c > 0$ gelten. Folglich liegt das Wort $uv^3 wx^3 y$ nicht mehr in L_1 , da es entweder mehr b 's oder mehr c 's als a 's enthält. Dies ist ein Widerspruch zu (3).
- (*) Da jeder Fall zu einem Widerspruch führt und die obigen Fälle alle möglichen Zerlegungen abdecken, kann die ursprüngliche Annahme nicht gelten. Also ist L nicht kontextfrei.

(b) L_2 ist nicht kontextfrei. Widerspruchsbeweis unter Verwendung des Pumping Lemmas für kontextfreie Sprachen:

- (*) Angenommen, L_2 wäre kontextfrei. Dann können wir das Pumping-Lemma für kontextfreie Sprachen anwenden.
- (*) Sei $n \in \mathbb{N}$ eine Pumping-Lemma-Zahl für L .
- (*) Es gilt $z = a^n b^n a^{2n} b^n a^n \in L$ und $|z| > n$.
- Es gibt also für z eine Zerlegung $z = uvwxy$ mit den Eigenschaften
 - (1) $vx \neq \varepsilon$
 - (2) $|vwx| \leq n$
 - (3) $\forall i \in \mathbb{N}_0. uv^i wx^i y \in L_1$
- (*) Betrachten wir das Wort $uv^0 wx^0 y$. Es hat eine der folgenden Formen, wobei entweder $i > 0$ oder $j > 0$ gilt:
 - $a^{n-i} b^{n-j} a^{2n} b^n a^n$
 - $a^n b^{n-i} a^{2n-j} b^n a^n$
 - $a^n b^n a^{2n-i} b^{n-j} a^n$
 - $a^n b^n a^{2n} b^{n-i} a^{n-j}$
- (*) In jedem Fall gilt $uv^0 wx^0 y \notin L_2$ im Widerspruch zu (3). Also kann die ursprüngliche Annahme nicht gelten, d.h. L_2 nicht kontextfrei.

AUFGABE 6.4. (Myhill-Nerode)

2 Punkte

Ursprünglich war in dieser Aufgabe versehentlich für beide Sprachen das Alphabet $\Sigma = \{a, b, c\}$ angegeben. In der Korrektur werden sowohl Abgaben mit diesem Alphabet als auch mit den unten angegebenen Alphabeten als korrekt gewertet.

Entscheiden Sie, ob folgende Sprachen regulär sind. Bestimmen Sie hierzu die Äquivalenzklassen der dazugehörigen Myhill-Nerode-Relation. Falls die Sprache regulär ist, zählen Sie alle Äquivalenzklassen auf und zeichnen Sie den kanonischen Minimalautomaten. Falls die Sprache nicht regulär ist, reicht es eine unendliche Menge von Äquivalenzklassen anzugeben und zu zeigen dass die Elemente dieser Menge paarweise verschieden sind.

(a) $L_1 = L((ba|ab)^*)$ mit dem Alphabet $\Sigma = \{a, b\}$

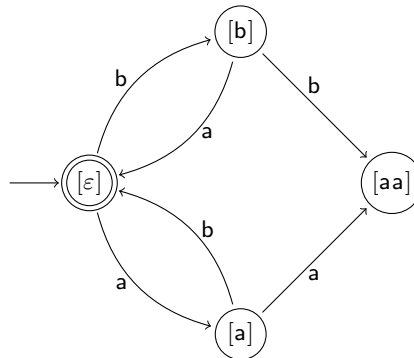
(b) $L_2 = \{a^n \mid \exists k \in \mathbb{N}_0. n = 2^k\}$ mit dem Alphabet $\Sigma = \{a\}$

Lösungsskizze

(a) • Bestimmen der Äquivalenzklassen:

$$[\varepsilon] = L_1 \quad [a] = \{wa \mid w \in L_1\} \quad [b] = \{wb \mid w \in L_1\} \quad [aa] = \{vw \mid v \in L_1, w \in L((aa|bb)(a|b)^*)\}$$

• Kanonischer DFA:



(b) Bestimmen einer unendlichen Menge von Äquivalenzklassen:

$$\{[a^{2^i}] \mid i \in \mathbb{N}_0\}$$

Beweis: Sei $i, j \in \mathbb{N}_0$ verschieden und ohne Einschränkung der Allgemeinheit $i < j$. Dann gilt $a^{2^i} a^{2^j} = a^{2^{i+1}} \in L_2$. Das Wort $w = a^{2^i} a^{2^j}$ hat die Länge $2^i + 2^j = 2^i + (2^{j-i} 2^i) = 2^i(2^{j-i} + 1)$. Damit hat $|w|$ den ungeraden Teiler $2^{j-i} + 1$ und ist daher keine Zweierpotenz. Es folgt $w \notin L_2$. Somit sind die Klassen unterscheidbar und L_2 ist keine reguläre Sprache.

AUFGABE 6.5. (Sprache und CFG)

Wir schreiben $u \leq v$ falls u ein Präfix von v ist, also falls es ein x gibt mit $v = ux$. Wir schreiben $u < v$, falls zusätzlich $x \neq \varepsilon$ (echtes Präfix).

Gegeben seien die Sprache

$$L = \{w \mid |w|_b = |w|_a + 1 \wedge (\forall u < w. |u|_a \geq |u|_b)\}$$

und die aus der Vorlesung bekannte Grammatik $G = (\{S\}, \{a, b\}, P, S)$ mit den Produktionen P:

$$S \rightarrow b \mid aSS$$

Ziel der Aufgabe ist es zu zeigen, dass die Grammatik zur Sprache passt, also $L(G) = L$. Beweisen Sie hierzu die folgenden Aussagen:

(a) $L(G) \subseteq L$

(b) $L \subseteq L(G)$

Hinweis: Da Teilaufgabe b) besonders schwer ist, vergeben wir einen Bonuspunkt, der nicht zu der regulär erreichbaren Maximalpunktzahl von 60 Punkten gezählt wird. Für das Erreichen der Maximalpunktzahl ist es also nicht notwendig, diese Aufgabe zu bearbeiten.

Lösungsskizze

Wir definieren zunächst $h(w) := |w|_a - |w|_b + 1$. Dann gilt $L = \{w \mid h(w) = 0 \wedge (\forall u < w. h(u) \geq 1)\}$. Man beachte $h(uv) = h(u) + h(v) - 1$.

(a) Wir fixieren $w \in L(G)$ und müssen zeigen $w \in L$. Es gilt $S \rightarrow^n w$ für ein $n > 0$. Wir zeigen $w \in L$ per vollständiger Induktion über n . Im Fall $n = 1$ hat die Ableitung die Form $S \rightarrow b$ mit $w = b$ und damit gilt die Aussage.

Für den Induktionsschritt nehmen wir an, dass die Aussage für alle w mit $w \leq n$ gilt (Induktionshypothese), und dass $n > 1$. Wir müssen die Aussage für $n + 1$ zeigen. Die Ableitung hat also folgende Form:

$$S \rightarrow aSS \xrightarrow{*} auv \quad \wedge \quad w = auv$$

Die IH gilt für u und v . Wir erhalten: $h(w) = h(auv) = 2 + h(u) - 1 + h(v) - 1 = h(u) + h(v) \stackrel{IH}{=} 0$.

Wir müssen noch zeigen, dass $h(x) \geq 1$ gilt, für alle x mit $x < w$. Fallunterscheidung:

- Die Fälle $x = \varepsilon$ und $x = a$ sind trivial.
- Fall $x = ay$ mit $y < u$. Dann gilt: $h(x) = h(y) + 1 \stackrel{IH}{\geq} 2$

- Fall $x = auy$ mit $y < v$. Dann gilt: $h(x) = h(u) + h(y) \stackrel{\text{IH}}{=} h(y) \stackrel{\text{IH}}{\geq} 1$

(b) Wir fixieren $w \in L$ und müssen zeigen $w \in L(G)$. Beweis per vollständiger Induktion über $|w|$. Die Fälle für $|w| = 0$ und $|w| = 1$ sind trivial.

Für den Induktionsschritt stellen wir zunächst fest, dass w nicht mit b beginnen kann, da $h(b) = 1$ und $b < w$. Wir betrachten also ein Wort der Form aw mit $|w| \geq 1$ und können als Induktionshypothese annehmen, dass $w' \in L \rightarrow w' \in L(G)$ für all w' mit $|w'| \leq |w|$.

Wir halten fest, dass die folgenden Eigenschaften gelten (wegen $aw \in L$): $h(w) = -1$ und $\forall u < w. h(u) \geq 0$. Damit muss ein u mit $u < w$ und $h(u) = 0$ existieren, Wir betrachten das *kürzeste* u dieser Art.

Wir wissen $u \neq \varepsilon$ (wegen $h(\varepsilon) = 1$) und $w = uv$ für ein v mit $v \neq \varepsilon$. Wir erhalten $-1 = h(uv) = h(v) - 1$, also $h(v) = 0$. Zudem gilt für alle $x < v$, dass $ux < w$, und somit $0 \leq h(ux) = h(x) - 1$ bzw. $h(x) \geq 1$. Wir haben also $v \in L(G)$ gezeigt.

Für jedes x mit $x < u$ gilt $h(x) \geq 0$ wegen $x < w$, und $h(x) \neq 0$ wegen der Minimalität von u . Insgesamt gilt also $h(x) \geq 1$, falls $x < u$. Also gilt auch $u \in L(G)$.

Unter Anwendung der IH für u und v können wir $w \in L(G)$ zeigen:

$$S \rightarrow aSS \rightarrow^* auS \rightarrow^* avv$$