Prof. Dr. Jasmin Blanchette
Jannis Limperg

Ludwig-Maximilians-Universität München
Institut für Informatik
Abgabe bis 26. April 2023, 10:00 Uhr

# Lösungsvorschlag zur Übung 1 zur Vorlesung
# Formale Sprachen und Komplexität

**FSK1-1** ***Operationen auf Formalen Sprachen*** (2 Punkte)

Beweisen oder widerlegen Sie jede der folgenden Aussagen:

a) Seien $L_1$ und $L_2$ formale Sprachen über dem Alphabet $\Sigma = \{a, b\}$, sodass alle Wörter in $L_1$ eine gerade Anzahl an $a$'s haben und alle Wörter in $L_2$ eine gerade Anzahl an $b$'s haben. Dann haben alle Wörter in $L_1 \cup L_2$ eine gerade Anzahl an $a$'s und eine gerade Anzahl an $b$'s.

**LÖSUNGSVORSCHLAG:**

Stimmt nicht. Gegenbeispiel: $L_1 = \{aab\}$, $L_2 = \{\}$ aber $aab \in L_1 \cup L_2$.

b) Sei die formale Sprache $L$ definiert als $L = \{w \in \{a, b\}^* \mid \#_a(w) \leq \#_b(w)\}$.
Dann gilt $L \cap \{a\}^* = \{\}$.

**LÖSUNGSVORSCHLAG:**

Stimmt nicht. Sowohl $L$ als auch $\{a\}^*$ enthalten das Wort $\varepsilon$. Daher ist $L \cap \{a\}^* \neq \{\}$.

c) Sei $L$ eine Sprache über dem Alphabet $\Sigma = \{a, b\}$, die alle Wörter aus $\Sigma^*$ enthält, deren Wortlänge kleiner als 1000 ist (d.h. $w \in L$ gilt genau dann wenn $w \in \Sigma^*$ und $|w| < 1000$). Dann ist $L$ eine endliche Sprache.

**LÖSUNGSVORSCHLAG:**

Stimmt. Es gibt für ein Alphabet mit zwei Elementen genau $2^i$ Wörter der Länge $i$, also enthält $L$ $2^0 + 2^1 + \cdots + 2^{1000} < \infty$ Wörter.

d) Sei $L$ eine formale Sprache über dem Alphabet $\Sigma = \{a, b\}$, die nur Wörter mit dreimal sovielen $b$'s wie $a$'s enthält, d.h. für alle $w \in L$ gilt $\#_b(w) = 3 \cdot \#_a(w)$.

Dann enthält $L^*$ ebenfalls nur Wörter, die dreimal soviele $b$'s wie $a$'s enthalten.

**LÖSUNGSVORSCHLAG:**

Stimmt. Jedes Wort $w$ in $L^*$ lässt sich schreiben als $w_1 \cdots w_n$, wobei $w_i \in L$ für $i = 1, \cdots, n$. Es gilt also

$$\begin{aligned}
\#_b(w) &= \#_b(w_1 \cdots w_n) \\
&= \#_b(w_1) + \cdots + \#_b(w_n) \\
&= 3 \cdot \#_a(w_1) + \cdots + 3 \cdot \#_a(w_n) \\
&= 3 \cdot (\#_a(w_1) + \cdots + \#_a(w_n)) \\
&= 3 \cdot \#_a(w)
\end{aligned}$$

Alternativ, und etwas formeller, können wir die Aussage auch durch Induktion über $n$ beweisen. (In dem Beweis oben versteckt sich die Induktion in der Notation $\cdots$.)

**Induktionsbasis $n = 0$:** Die Aussage gilt, da $0 = \#_b(\varepsilon) = 3 \cdot \#_a(\varepsilon) = 0$.

**Induktionsschritt $n - 1 \to n$:** Wende die Induktionshypothese auf $w_1 \cdots w_{n-1}$ an. Das liefert $\#_b(w_1 \cdots w_{n-1}) = 3 \cdot \#_a(w_1 \cdots w_{n-1})$. Es gilt

$$\begin{aligned}
\#_b(w_1 \cdots w_n) &= \#_b(w_1 \cdots w_{n-1}) + \#_b(w_n) \\
&= 3 \cdot \#_a(w_1 \cdots w_{n-1}) + \#_b(w_n) \\
&= 3 \cdot \#_a(w_1 \cdots w_{n-1}) + 3 \cdot \#_a(w_n) \\
&= 3 \cdot (\#_a(w_1 \cdots w_{n-1}) + \#_a(w_n)) \\
&= 3 \cdot \#_a(w_1 \cdots w_n)
\end{aligned}$$

**FSK1-2** ***Grammatiken angeben*** (2 Punkte)

Sei $\Sigma = \{a, b\}$. Geben Sie jeweils eine Grammatik $G_i$ an, sodass $L(G_i)$ die Sprache $L_i$ über $\Sigma$ erzeugt. Verwenden Sie keine $\varepsilon$-Produktionen.[1] Erläutern Sie, warum $L(G_i) = L_i$ gilt, indem Sie die „Aufgabe“ der einzelnen Variablen und Produktionen erläutern. Geben Sie außerdem jeweils den Typ Ihrer Grammatik an (mit Begründung).

a) $L_1 = \{aba, abba, abbababababa\}$

**LÖSUNGSVORSCHLAG:**

$$G_1 = (\{S\}, \Sigma, \{S \to aba,\ S \to abba,\ S \to abbababababa\}, S)$$

[1]Eine $\varepsilon$-Produktion ist eine Produktion, deren rechte Seite aus dem leeren Wort besteht.

Die Grammatik erzeugt $L(G_1)$, da das Startsymbol genau diese drei Wörter mit je einer Produktion erzeugt. Die Grammatik ist von Typ 2 (und damit auch Typ 0, Typ 1), da es keine verkürzenden Regeln gibt, und alle linken Seiten der Produktionen Variablen sind. Die Grammatik ist nicht Typ 3, da alle Produktionen das Format für reguläre Grammatiken verletzen.

b) $L_2 = \{a\}^+$

**LÖSUNGSVORSCHLAG:**

$$G_2 = (\{S\}, \Sigma, \{S \to a,\ S \to aS\}, S)$$

Die Variable $S$ erzeugt ein $a$ und möglicherweise eine Variable $S$ dazu (siehe erste bzw. zweite Produktion). Um ein $a$ zu generieren, nutzt man die erste Produktion. Um $n$ viele $a$'s zu generieren, nutzt man die zweite Produktion $(n-1)$-mal und die erste Produktion einmal. Die Grammatik ist vom Typ 3 (und 0 und 1 und 2).

c) $L_3 = \{a^i b^j a^j b^i \mid i, j > 0\}$.

**LÖSUNGSVORSCHLAG:**

$$G_3 = (\{S, A\}, \Sigma, \{S \to aSb,\ S \to aAb,\ A \to bAa,\ A \to ba\}, S)$$

Die Produktionen $A \to bAa$ und $A \to ba$ erzeugen $b^j a^j$. Durch $A \to ba$ wird sichergestellt, dass mindestens ein $ba$-Paar erzeugt wird ($j > 0$). Außen werden immer links bzw. rechts davon durch $S \to aSb$ und $S \to aAb$ je ein $a$ bzw. ein $b$ erzeugt. Insgesamt erzeugen diese zwei Produktionen einen Rahmen von der Form $a^i \dots b^i$ um $b^j a^j$ herum. Durch $S \to aAb$ wird sichergestellt, dass mindestens ein $a \dots b$-Paar erzeugt wird ($i > 0$). Die Grammatik ist vom Typ 2 (und 0 und 1), da es keine verkürzenden Regeln gibt und links nur einzelne Variablen stehen. Die Grammatik ist aber nicht vom Typ 3, da das Format für reguläre Grammatiken z.B. durch $S \to aAa$ verletzt ist.

**FSK1-3** ***Spracheigenschaft per Induktion beweisen*** (0 Punkte)

Die Grammatik $G$ sei definiert durch $G = (A, \{a, b\}, \{A \to AA,\ A \to aAbb,\ A \to bab\}, A)$. Beweisen Sie, dass gilt:

$$w \in L(G) \implies \#_b(w) = 2 \cdot \#_a(w)$$

Hinweis: Verwenden Sie vollständige Induktion über die Länge der Linksableitung von $w$.

**LÖSUNGSVORSCHLAG:**

Vollständige Induktion über die Länge der Linksableitung von $w$:

**Induktionsbasis:** Die einzige Ableitung der Länge 1 ist $A \Rightarrow bab$. Da $\#_b(bab) = 2 = 2\#_a(bab)$, gilt die Behauptung.

**Induktionsschritt $n-1 \to n$:** Induktionshypothese: Für jede Linksableitung $A \Rightarrow^k w$ eines Wortes $w \in L(G)$ mit $k < n$ Schritten gilt $\#_b(w) = 2\#_a(w)$.

Sei nun $A \Rightarrow^n w$ mit $n > 1$. Wir unterscheiden zwei Fälle:

- $A \Rightarrow aAbb \Rightarrow^{n-1} w$. Mit $w = aubb$ gilt $A \Rightarrow^{n-1} u$. Aus der Induktionshypothese folgt $\#_b(u) = 2\#_a(u)$ und damit

$$\#_b(w) = \#_b(aubb) = 2 + \#_b(u) = 2 + 2\#_a(u) = 2\#_a(aubb) = 2\#_a(w)$$

- $A \Rightarrow AA \Rightarrow^{n-1} w$. Dann ist diese Ableitung von der Form $AA \Rightarrow^{k_1} uA \Rightarrow^{k_2} uv$ mit $u, v \in \{a,b\}^*$ und $uv = w$ und $k_1 + k_2 = n-1$. Daraus folgt $A \Rightarrow^{k_1} u$ und $A \Rightarrow^{k_2} v$. Anwenden der Induktionshypothese zeigt $\#_b(u) = 2\#_a(u)$ und $\#_b(v) = 2\#_a(v)$ und damit folgt

$$\begin{aligned}\#_b(w) = \#_b(uv) = \#_b(u) + \#_b(v) = 2\#_a(u) + 2\#_a(v) &= 2(\#_a(u) + \#_a(v)) \\ &= 2\#_a(uv) = 2\#_a(w)\end{aligned}$$

**FSK1-4** *Klammersprache* (0 Punkte)

Die Grammatik $G$ sei definiert durch $G = (\{S, A, B\}, \{(,), [,]\}, P, S)$, wobei

$$P = \{S \to (S),\ \ S \to [S],\ \ S \to A,\ \ S \to B,\ \ A \to (),\ \ A \to [],\ \ B \to S,\ \ B \to BB\}.$$

a) Von welchem Typ ist die Grammatik $G$?

**LÖSUNGSVORSCHLAG:** Typ 2, da es keine verkürzenden Regeln gibt (daher auch Typ 0 und 1) und alle linken Seiten der Produktionen nur aus einer Variablen bestehen. Aber nicht Typ 3, da es Regeln gibt (sogar alle), die nicht in der für Typ 3-Grammatiken zugelassenen Form sind (also deren rechte Seite weder von der Form $aV$ noch von der Form $a$ ist).

b) Stellen Sie für folgende Zeichenketten fest, ob sie Wörter in $L(G)$ sind. Begründen Sie Ihre Antwort. Bei Wörtern, die in $L(G)$ sind, geben Sie eine Ableitung an.

- $()[[()]]$

**LÖSUNGSVORSCHLAG:**
In $L(G)$. Ableitung:
$S \Rightarrow B \Rightarrow BB \Rightarrow SB \Rightarrow SS \Rightarrow AS \Rightarrow ()S \Rightarrow ()[S] \Rightarrow ()[[S]] \Rightarrow ()[[A]] \Rightarrow ()[[()]]$

- $[[BA]]$

**LÖSUNGSVORSCHLAG:** Nicht in $L(G)$. Dies ist nicht einmal ein Wort über dem zugrundeliegenden Alphabet, sondern enthält noch Variablen.

- $(][[]])$

**LÖSUNGSVORSCHLAG:** Nicht in $L(G)$. Es gibt keine Regel, die ein $]$ erzeugt, ohne dass links davon auch ein $[$ erzeugt würde.

- $[()()()]$

**LÖSUNGSVORSCHLAG:**
In $L(G)$. Ableitung:

$$\begin{aligned} S &\Rightarrow [S] \Rightarrow [B] \Rightarrow [BB] \Rightarrow [BBB] \Rightarrow [SBB] \Rightarrow [SSB] \Rightarrow [SSS] \\ &\Rightarrow [ASS] \Rightarrow [AAS] \Rightarrow [AAA] \Rightarrow [()AA] \Rightarrow [()()A] \Rightarrow [()()()] \end{aligned}$$

- $(([[]]]))$

**LÖSUNGSVORSCHLAG:** Nicht in $L(G)$. Jede Regel, die $[$ erzeugt, erzeugt auch $]$ und umgekehrt. Somit muss es in einem Wort in $L(G)$ immer gleich viele $[$ wie $]$ geben. In diesem Wort ist das aber nicht der Fall.

c) Geben Sie vier verschiedene Wörter aus $L(G)$ an, die nicht in Teilaufgabe b) vorkommen.

**LÖSUNGSVORSCHLAG:** $[], (), []() , ([]), [()], \ldots$

d) Geben Sie 4 verschiedene Wörter aus $\{[,],(,)\}^*$ an, die nicht in $L(G)$ liegen und die nicht in Teilaufgabe b) vorkommen.

**LÖSUNGSVORSCHLAG:** $[(], ([), []](), ([[]), [([)], \ldots$

e) Beschreiben Sie die Sprache $L(G)$ in natürlicher Sprache möglichst kurz und genau.

**LÖSUNGSVORSCHLAG:** $G$ akzeptiert die Sprache der zueinander passenden Klammern, wobei es eckige [] und runde () Klammern gibt.