Prof. Dr. Jasmin Blanchette
Jannis Limperg
Luca Maio

Ludwig-Maximilians-Universität München
Institut für Informatik
Abgabe bis 24. April 2024, 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 von a's haben und alle Wörter in L_2 eine gerade Anzahl von b's haben. Dann haben alle Wörter in $L_1 \cap L_2$ eine gerade Anzahl von a's und eine gerade Anzahl von b's.

> **LÖSUNGSVORSCHLAG:**
> Stimmt. Es ist $L_1 \cap L_2 = \{w \in \Sigma^* \mid w \in L_1 \land w \in L_2\}$. Aus $w \in L_1$ folgt, dass w eine gerade Anzahl von a's hat, und aus $w \in L_2$ folgt, dass w eine gerade Anzahl von b's hat.

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

> **LÖSUNGSVORSCHLAG:**
> Stimmt. Es ist $\{b\}^* \subseteq L$, da für jedes Wort $w = b \cdots b \in \{b\}^*$ gilt: $\#_a(w) = 0 \leq \#_b(w)$. Somit $L \cup \{b\}^* = L$.

c) Sei Σ ein Alphabet und $k \in \mathbb{N}$. Sei L die Sprache $\{w \in \Sigma^* \mid |w| \leq k\}$. Dann ist L eine endliche Sprache.

> **LÖSUNGSVORSCHLAG:**
> Stimmt. Alphabete sind per Definition endlich, d.h. $|\Sigma|$ ist eine natürliche Zahl. Es gibt somit für jedes $i \in \mathbb{N}$ $|\Sigma|^i$ verschiedene Wörter der Länge i. Insgesamt enthält L also $|\Sigma|^0 + |\Sigma|^1 + \cdots + |\Sigma|^k < \infty$ Wörter.

d) Über dem Alphabet $\Sigma = \{a, b, c\}$ definieren wir die Sprache $L = \{w \in \Sigma^* \mid \#_a(w) + \#_b(w) = \#_c(w)\}$, also die Sprache der Wörter, die so viele a's und b's wie c's enthalten. Es gilt: $L^* \subseteq L$.

LÖSUNGSVORSCHLAG:
Stimmt. Wir müssen zeigen, dass für jedes Wort $w \in L^*$ gilt: $\#_a(w) + \#_b(w) = \#_c(w)$. Das Wort w lässt sich schreiben als $w_1 \cdots w_n$ mit $w_i \in L$ für $i = 1, \ldots, n$. Es gilt also

$$\begin{aligned}\#_a(w) + \#_b(w) &= \left(\sum_{i=1}^{n} \#_a(w_i)\right) + \left(\sum_{i=1}^{n} \#_b(w_i)\right) = \sum_{i=1}^{n} \#_a(w_i) + \#_b(w_i)\\ &= \sum_{i=1}^{n} \#_c(w_i) = \#_c(w)\end{aligned}$$

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 $w_1 \cdots w_n$.)

- Für $n = 0$: Die Aussage gilt, da $\#_c(\varepsilon) = 0 = 0 + 0 = \#_a(\varepsilon) + \#_b(\varepsilon)$.
- Für $n > 0$: Wende die Induktionshypothese auf $w_1 \cdots w_{n-1}$ an. Das liefert $\#_c(w_1 \cdots w_{n-1}) = \#_a(w_1 \cdots w_{n-1}) + \#_b(w_1 \cdots w_{n-1})$. Nun gilt

$$\begin{aligned}\#_c(w) &= \#_c(w_1 \cdots w_n) = \#_c(w_1 \cdots w_{n-1}) + \#_c(w_n)\\ &\overset{\text{IH}}{=} \#_a(w_1 \cdots w_{n-1}) + \#_b(w_1 \cdots w_{n-1}) + \#_a(w_n) + \#_b(w_n)\\ &= \#_a(w_1 \cdots w_n) + \#_b(w_1 \cdots w_n) = \#_a(w) + \#_b(w)\end{aligned}$$

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

Sei $\Sigma = \{a, b\}$. Geben Sie für jede der folgenden Teilaufgaben eine Grammatik G_i als 4-Tupel an, sodass $L(G_i)$ die Sprache L_i über Σ erzeugt. Verwenden Sie keine ε-Produktionen. 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) $L_1 = \{a, b\}^+$

LÖSUNGSVORSCHLAG:

$$G_1 = (\{S\}, \Sigma, \{S \to SS, S \to a, S \to b\}, S)$$

Mit der Produktion $S \to SS$ generieren wir die Satzformen SS, SSS, etc. bis zu beliebiger Länge. Die Produktionen $S \to a$ und $S \to b$ erlauben uns dann, beliebige Wörter aus $\{a, b\}^+$ aus diesen Satzformen abzuleiten.
Die Grammatik G_1 ist vom Typ 2, denn für alle Produktionen $l \to r$ gilt $l \in \{S\}$. G_1 ist nicht vom Typ 3, denn die Produktion $S \to SS$ ist nicht von

der Form $S \to c$ oder $S \to cA$ mit $c \in \Sigma$ und $A \in \{S\}$. Allerdings gäbe es auch eine Typ-3-Grammatik, die L_1 erkennt:

$$G_1' = (\{S\}, \Sigma, \{S \to a, S \to b, S \to aS, S \to bS\}, S)$$

b) $L_2 = \{w \in \Sigma^+ \mid |w| \leq 2\}$

LÖSUNGSVORSCHLAG:

$$G_2 = (\{S\}, \Sigma, \{S \to w \mid w \in L_2\}, S)$$

Da die Sprache L_2 endlich ist (vergleiche Aufgabe FSK1-1c)), können wir einfach eine Produktion von S zu jedem Wort in L_2 verwenden.
Die Grammatik G_2 ist vom Typ 2, weil die linke Seite jeder Produktion ein Nichtterminal ist. G_2 ist nicht vom Typ 3, weil beispielsweise die Produktion $S \to aa$ nicht von der Form $S \to c$ oder $S \to cA$ mit $c \in \Sigma$ und $A \in \{S\}$ ist.

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

LÖSUNGSVORSCHLAG:

$$G_3 = (\{S, X\}, \Sigma, \{S \to aSb, S \to aXb, X \to bXa, X \to ba\}, S)$$

Die Produktion $X \to bXa$ generiert Satzformen bXa, $bbXaa$, $\ldots$, $b^k X a^k$ für $k \in \mathbb{N}_{>0}$. Die Produktion $X \to ba$ vervollständigt diese Satzformen zu Wörtern $b^{k+1}a^{k+1}$, also $b^j a^j$ für $j \in \mathbb{N}_{>0}$. Analog generiert die Produktion $S \to aSb$ Satzformen $a^l S b^l$ für $l \in \mathbb{N}_{>0}$ und die Produktion $S \to aXb$ vervollständigt diese zu $a^i X b^i$ für $i \in \mathbb{N}_{>0}$.
Die Grammatik G_3 ist vom Typ 2. Die Begründung ist analog zu den vorigen Teilaufgaben.

FSK1-3 ***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 (daher auch Typ 0 und 1), da es keine verkürzenden Regeln gibt 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 aC 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 Linksableitung, eine Rechtsableitung und einen Syntaxbaum an.

i) $()[[()]]$

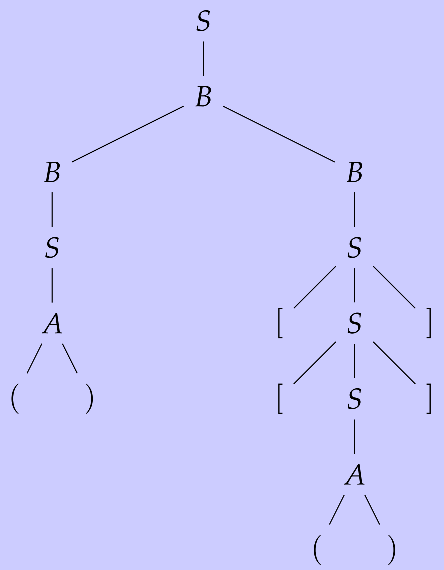
LÖSUNGSVORSCHLAG:
In $L(G)$.
Linksableitung:

$$S \Rightarrow B \Rightarrow BB \Rightarrow SB \Rightarrow AB \Rightarrow ()B \Rightarrow ()S \Rightarrow$$
$$()[S] \Rightarrow ()[[S]] \Rightarrow ()[[A]] \Rightarrow ()[[()]]$$

Rechtsableitung:

$$S \Rightarrow B \Rightarrow BB \Rightarrow BS \Rightarrow B[S] \Rightarrow B[[S]] \Rightarrow B[[A]] \Rightarrow$$
$$B[[()]] \Rightarrow S[[()]] \Rightarrow A[[()]] \Rightarrow ()[[()]]$$

Syntaxbaum:



ii) $[[BA]]$

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

iii) $(][[]])$

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

iv) $(([[]]]))$

> **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.

FSK1-4 ***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: Beweisen Sie zunächst die allgemeinere Aussage, dass für jede Satzform u gilt: Wenn u von G in n Schritten erzeugt wird (d.h. $A \Rightarrow^n u$), dann ist $\#_b(u) = 2 \cdot \#_a(u)$. Verwenden Sie vollständige Induktion über n.

LÖSUNGSVORSCHLAG:
Wir beweisen die allgemeinere Aussage durch vollständige Induktion über n:

- Für $n = 0$: Wegen $A \Rightarrow^0 u$ muss $u = A$ sein, also ist $\#_b(u) = \#_a(u) = 0$.
- Für $n > 0$:

  Induktionshypothese: Für jede Satzform v und jedes $k < n$ gilt: $A \Rightarrow^k v$ impliziert $\#_b(v) = 2 \cdot \#_a(v)$.

  Angenommen $A \Rightarrow^n u$. Dann gilt $A \Rightarrow^{n-1} v$ für eine Satzform v mit $v \Rightarrow u$. Folgende Fälle sind möglich, je nachdem, welche Regel zuletzt angewendet wurde:

  - Fall $A \to AA$: Dann ist $v = v_1 A v_2$ und $u = v_1 AA v_2$. Dann ist $\#_b(u) = \#_b(v)$ und $\#_a(u) = \#_a(v)$, also mit der Induktionshypothese $\#_b(u) = 2 \cdot \#_a(u)$.
  - Fall $A \to aAbb$ oder $A \to bab$: Dann ist $v = v_1 A v_2$ und wir haben entweder $u = v_1 aAbb v_2$ oder $u = v_1 bab v_2$. Dann ist

$$\#_b(u) = \#_b(v) + 2 \overset{\text{IH}}{=} (2 \cdot \#_a(v)) + 2 = 2 \cdot (\#_a(v) + 1) = 2 \cdot \#_a(u)$$

Aus der allgemeineren Aussage folgt die gewünschte Aussage, denn jedes Wort w ist auch eine Satzform und wenn $w \in L(G)$ ist gibt es eine Ableitung $A \Rightarrow^* w$. Somit ist $\#_b(w) = 2 \cdot \#_a(w)$.