

Lösungsvorschlag zur Übung 1 zur Vorlesung  
Theoretische Informatik für Studierende der Medieninformatik

**TIMI1-1 Operationen auf formalen Sprachen**

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  $L_1 \cap L_2 = \{w \in \Sigma^* \mid w \in L_1 \wedge w \in L_2\}$ . 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  $\Sigma$  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  $|\Sigma|^0 + |\Sigma|^1 + \dots + |\Sigma|^k < \infty$  Wörter.

- c) 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  $w_1 \cdot \dots \cdot w_n$  mit  $w_i \in L$

für  $i = 1, \dots, n$ . 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

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

### TIMI1-2 Grammatiken angeben

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  $\Sigma$  erzeugt. Verwenden Sie keine  $\varepsilon$ -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:

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

Mit der Produktion  $S \rightarrow SS$  generieren wir die Satzformen  $SS, SSS, \text{etc.}$  bis zu beliebiger Länge. Die Produktionen  $S \rightarrow a$  und  $S \rightarrow b$  erlauben uns dann, beliebige Wörter aus  $\{a, b\}^+$  aus diesen Satzformen abzuleiten.

Die Grammatik  $G_1$  ist vom Typ 2, denn für alle Produktionen  $l \rightarrow r$  gilt  $l \in \{S\}$ .  $G_1$  ist nicht vom Typ 3, denn die Produktion  $S \rightarrow SS$  ist nicht von der Form  $S \rightarrow c$  oder  $S \rightarrow cA$  mit  $c \in \Sigma$  und  $A \in \{S\}$ . Allerdings gäbe es

auch eine Typ-3-Grammatik, die  $L_1$  erkennt:

$$G'_1 = (\{S\}, \Sigma, \{S \rightarrow a, S \rightarrow b, S \rightarrow aS, S \rightarrow bS\}, S)$$

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

**LÖSUNGSVORSCHLAG:**

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

Da die Sprache  $L_2$  endlich ist (vergleiche Aufgabe TIM11-1b)), 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  $S \rightarrow aa$  nicht von der Form  $S \rightarrow c$  oder  $S \rightarrow cA$  mit  $c \in \Sigma$  und  $A \in \{S\}$  ist.

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

**LÖSUNGSVORSCHLAG:**

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

Die Produktion  $X \rightarrow bXa$  generiert Satzformen  $bXa, bbXaa, \dots, b^k X a^k$  für  $k \in \mathbb{N}_{>0}$ . Die Produktion  $X \rightarrow ba$  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  $S \rightarrow aSb$  Satzformen  $a^l S b^l$  für  $l \in \mathbb{N}_{>0}$  und die Produktion  $S \rightarrow aXb$  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.