

# Übungsaufgaben Theoretische Informatik I

## Blatt 8

Marco Möller <Marco.Moeller@gmxpro.de>  
Matrikel Nummer: 23220320

20. Dezember 2004

### 1 Pumping Lemma

Hier das Pumping Lemma in Quantorenschreibweise:

$$L \in \mathcal{L}_2 \Rightarrow [\exists_{\mathbb{N}}^n : (\forall_L^z |z| \geq n : (\exists z = uvwxy \wedge |vx| \geq 1 \wedge |vwx| \leq n : (\forall i \geq 0 : (uv^iwx^iy \in L)))))]$$

Vorweg einmal die Kontraposition des Pumping Lemmas, die ich brauche, um zu zeigen, das es sich nicht um eine reguläre Sprache handelt:

$$[\forall_{\mathbb{N}}^n : (\exists_L^z |z| \geq n : (\forall z = uvwxy \wedge |vx| \geq 1 \vee |vwx| \leq n : (\exists i \geq 0 : (uv^iwx^iy \notin L)))))] \Rightarrow L \notin \mathcal{L}_2$$

In Textform:

Für alle  $n$ , existiert ein  $z \in L$  mit  $|z| \geq n$ , bei dem für alle Zerlegungen  $z = uvwxy$  mit  $|vx| \geq 1$  und  $|vwx| \leq n$  es ein  $i$  gibt, bei dem  $uv^iwx^iy$  nicht in  $L$  liegt. Dann ist  $L$  keine kontextfreie Sprache.

#### 1.1 Prüfen der Sprache $M_1 = \{a^l b^m c^n \mid l, n, m \in \mathbb{N} \wedge l < n < m\}$

Als erstes benenne ich die Symbole in der Definition von  $M_1$  um, damit keine Verwechslungen mit denen des Pumping Lemma (P.L.) vorkommen. Zudem werde ich die Schreibweise ein wenig ändern:

$$M_1 = \{a^k b^{k+m+l} c^{k+m} \mid k, m, l \in \mathbb{N}\}$$

Zum Beweis wähle ich das Wort  $z = a^n b^{n+2} c^{n+1}$ . Dieses erfüllt die Bedingung  $|z| \geq n$ . Wegen der Voraussetzung  $|uvw| \leq n$  kann ein konkretes  $uvw$  nicht a's, b's und c's gleichzeitig enthalten, sondern nur maximal 2 davon. Sobald ich nur zwei der Buchstabenkombinationen anfasse und sie in Ihrer Häufigkeit verändere, ergibt es entweder für  $i = 0$  (ein Buchstabe der eigentlich häufiger als ein anderer sein sollte, ist plötzlich genausooft oder seltener vorhanden) oder für  $i = 2$  (ein Buchstabe der eigentlich seltener als ein anderer sein sollte, ist plötzlich genausooft oder häufiger vorhanden) Probleme. Dies lässt sich in 5 Fälle unterscheiden:

1.  $uvw$  enthält nur a's  $\Rightarrow$  z.B. mit  $i = 2$  gibt es zuviele a's
2.  $uvw$  enthält a's und b's  $\Rightarrow$  z.B. mit  $i = 0$  gibt es zuwenige b's
3.  $uvw$  enthält nur b's  $\Rightarrow$  z.B. mit  $i = 0$  gibt es zuwenige b's
4.  $uvw$  enthält b's und c's  $\Rightarrow$  z.B. mit  $i = 0$  gibt es zuwenige c's
5.  $uvw$  enthält nur c's  $\Rightarrow$  z.B. mit  $i = 0$  gibt es zuwenige c's / mit  $i = 2$  gibt es zuviele c's

## 1.2 Prüfen der Sprache $M_2 = \{a^l b^m c^n \mid l, n, m \in \mathbb{N} \wedge l > n > m\}$

Als erstes benenne ich die Symbole in der Definition von  $M_2$  um, damit keine Verwechslungen mit denen des Pumping Lemma (P.L.) vorkommen. Zudem werde ich die Schreibweise ein wenig ändern:

$$M_2 = \{a^{k+l+m} b^k c^{k+m} \mid k, m, l \in \mathbb{N}\}$$

Zum Beweis wähle ich das Wort  $z = a^{n+2} b^n c^{n+1}$ . Dieses erfüllt die Bedingung  $|z| \geq n$ . Wegen der Voraussetzung  $|uvw| \leq n$  kann ein konkretes  $uvw$  nicht a's, b's und c's gleichzeitig enthalten, sondern nur maximal 2 davon. Sobald ich nur zwei der Buchstabenkombinationen anfasse und sie in Ihrer Häufigkeit verändere, ergibt es entweder für  $i = 0$  (ein Buchstabe der eigentlich häufiger als ein anderer sein sollte, ist plötzlich genausooft oder seltener vorhanden) oder für  $i = 2$  (ein Buchstabe der eigentlich seltener als ein anderer sein sollte, ist plötzlich genausooft oder häufiger vorhanden) Probleme. Dies lässt sich in 5 Fälle unterscheiden:

1.  $uvw$  enthält nur a's  $\Rightarrow$  z.B. mit  $i = 0$  gibt es zuwenige a's
2.  $uvw$  enthält a's und b's  $\Rightarrow$  z.B. mit  $i = 2$  gibt es zuviele b's
3.  $uvw$  enthält nur b's  $\Rightarrow$  z.B. mit  $i = 2$  gibt es zuviele b's
4.  $uvw$  enthält b's und c's  $\Rightarrow$  z.B. mit  $i = 2$  gibt es zuviele c's
5.  $uvw$  enthält nur c's  $\Rightarrow$  z.B. mit  $i = 0$  gibt es zuwenige c's / mit  $i = 2$  gibt es zuviele c's

## 2 Chomsky-Normalform

Gegeben ist die Grammatik

$$\begin{aligned} G &= (V, \Sigma, P, S) \\ V &= \{S, A, B, C, D\} \\ \Sigma &= \{a, b, c\} \\ P &= \{S \rightarrow AB, \\ &\quad A \rightarrow ab \mid aAbcDa \mid C, \\ &\quad B \rightarrow c \mid cB \mid D, \\ &\quad C \rightarrow A \mid D \mid \varepsilon, \\ &\quad D \rightarrow a \mid c\} \end{aligned}$$

### 2.1 Entfernen von Kettenregeln

Um eine Grammatik in Chomsky Normalform bringen zu können, muss sie als allererstes von Kettenregeln befreit werden.

Hierzu beginnt man damit, alle  $\varepsilon$ -Regeln zu entfernen. Die Menge  $V_1 := \{A \in V \mid A \Rightarrow_G^* \varepsilon\}$  enthält alle Variablen, die sich auf  $\varepsilon$  ableiten lassen können. Bei unserer Grammatik ist das  $V_1 = \{A, C\}$ . Diese  $\varepsilon$ -Regeln, und alle Vorkommen von Variablen die dort hinführen müssen gestrichen werden. Wenn eine Variable allerdings nicht nur zum  $\varepsilon$  führt, müssen neue Regeln eingefügt werden:  $\forall B \rightarrow$

$xAy \wedge A \in V_1 \wedge xy \neq \varepsilon$  : die neue Regeln  $B \rightarrow xy$  einfügen. Dies führt zu folgender Grammatik  $G'$

$$\begin{aligned} G' &= (V, \Sigma, P', S) \\ V &= \{S, A, B, C, D\} \\ \Sigma &= \{a, b, c\} \\ P' &= \{S \rightarrow AB|B, \\ &\quad A \rightarrow ab|aAbcDa|abcDa|C, \\ &\quad B \rightarrow c|cB|D, \\ &\quad C \rightarrow A|D, \\ &\quad D \rightarrow a|c\} \end{aligned}$$

Nun sind die Kettenregeln an der Reihe. Hierfür muss für jede Variable  $X \in V$  die Äquivalenzklasse bestimmt werden (Variablen die sich ineinander ableiten lassen):

$$\begin{aligned} [A] &= [C] = \{A, C\} \\ [S] &= \{S\} \\ [B] &= \{B\} \\ [D] &= \{D\} \end{aligned}$$

Alle Variablen die in einer gemeinsamen Äquivalenzklasse liegen, haben genau die gleiche Funktion, und können somit zu einer zusammengefasst werden.

$$\begin{aligned} G'' &= (V, \Sigma, P'', S) \\ V &= \{S, A, B, D\} \\ \Sigma &= \{a, b, c\} \\ P'' &= \{S \rightarrow AB|B, \\ &\quad A \rightarrow ab|aAbcDa|abcDa|D, \\ &\quad B \rightarrow c|cB|D, \\ &\quad D \rightarrow a|c\} \end{aligned}$$

Bei allen Regeln, die auf der rechten Seite nur eine Variable enthalten, kann diese direkt ersetzt werden. Dies führt zu:

$$\begin{aligned} G''' &= (V, \Sigma, P''', S) \\ V &= \{S, A, B, D\} \\ \Sigma &= \{a, b, c\} \\ P''' &= \{S \rightarrow AB|cB|a|c, \\ &\quad A \rightarrow ab|aAbcDa|abcDa|a|c, \\ &\quad B \rightarrow cB|a|c, \\ &\quad D \rightarrow a|c\} \end{aligned}$$

## 2.2 Chomsky-Normalisierung

Als erstes wird für jedes Terminal, eine neue Variable angelegt. In allen Produktionen wird nun auf der Rechten Seite ein Terminal durch die zugehörige Variable ersetzt, wenn es nicht alleine vorkommt. Zudem müssen noch Ableitungen von entsprechenden Variable / Terminalpaaren

eingefügt werden. Dies sieht im Ergebniss wie folgt aus:

$$\begin{aligned}
G^{(4)} &= (V^{(4)}, \Sigma, P^{(4)}, S) \\
V^{(4)} &= \{S, A, B, D, V_a, V_b, V_c\} \\
\Sigma &= \{a, b, c\} \\
P^{(4)} &= \{S \rightarrow AB|V_cB|a|c, \\
&\quad A \rightarrow V_aV_b|V_aAV_bV_cDV_a|V_aV_bV_cDV_a|a|c, \\
&\quad B \rightarrow V_cB|a|c, \\
&\quad D \rightarrow a|c, \\
&\quad V_a \rightarrow a, \\
&\quad V_b \rightarrow b, \\
&\quad V_c \rightarrow c\}
\end{aligned}$$

Der letzte und abschließende Schritt entfernt alle zu langen Regeln, indem Sie in mehrere Regeln aufgebrochen werden. So haben alle rechten Seiten maximal 2 Variablen.

$$\begin{aligned}
G^{(5)} &= (V^{(5)}, \Sigma, P^{(5)}, S) \\
V^{(5)} &= \{S, A, B, D, V_a, V_b, V_c, X_1, X_2, X_3, X_4, X_5, X_6, X_7\} \\
\Sigma &= \{a, b, c\} \\
P^{(5)} &= \{S \rightarrow AB|V_cB|a|c, \\
&\quad A \rightarrow V_aV_b|V_aX_1|V_aX_5|a|c, \\
&\quad B \rightarrow V_cB|a|c, \\
&\quad D \rightarrow a|c, \\
&\quad V_a \rightarrow a, \\
&\quad V_b \rightarrow b, \\
&\quad V_c \rightarrow c, \\
&\quad X_1 \rightarrow AX_2, \\
&\quad X_2 \rightarrow V_bX_3, \\
&\quad X_3 \rightarrow V_cX_4, \\
&\quad X_4 \rightarrow DV_a, \\
&\quad X_5 \rightarrow V_bX_6, \\
&\quad X_6 \rightarrow V_cX_7, \\
&\quad X_7 \rightarrow DV_a\}
\end{aligned}$$

Wenn man will, kann man hier noch gleiche Ableitungsstränge, die durch das aufbrechen der Regeln

entstanden sind, zusammenfassen:

$$\begin{aligned}
 G^{(6)} &= (V^{(6)}, \Sigma, P^{(6)}, S) \\
 V^{(5)} &= \{S, A, B, D, V_a, V_b, V_c, X_1, X_2, X_3, X_4\} \\
 \Sigma &= \{a, b, c\} \\
 P^{(5)} &= \{S \rightarrow AB|V_cB|a|c, \\
 &\quad A \rightarrow V_aV_b|V_aX_1|V_aX_2|a|c, \\
 &\quad B \rightarrow V_cB|a|c, \\
 &\quad D \rightarrow a|c, \\
 &\quad V_a \rightarrow a, \\
 &\quad V_b \rightarrow b, \\
 &\quad V_c \rightarrow c, \\
 &\quad X_1 \rightarrow AX_2, \\
 &\quad X_2 \rightarrow V_bX_3, \\
 &\quad X_3 \rightarrow V_cX_4, \\
 &\quad X_4 \rightarrow DV_a\}
 \end{aligned}$$

### 3 Algorithmen

#### 3.1 Entfernen von $\varepsilon$ -Regeln

1.  $V_1 = \emptyset$
2.  $V_{1neu} = V_1$
3. Für jedes  $A$  aus  $V_1$ :
  - (a) Für jede Produktion  $p$  aus  $P$ :
    - i. Wenn sich die rechte Seite der Produktion nur aus Elementen von  $V_1$  besteht, dann füge  $A$  zu  $V_{1neu}$  hinzu
4. Wenn  $V_{1neu} \neq V_1$  setze  $V_1 = V_{1neu}$  und mache bei 3. weiter
5. setze  $V_1 = V_{1neu}$
6. Für jede Produktion  $p$  aus  $P$ :
  - (a) Wenn die rechte Seite von  $p = \varepsilon$ , entferne  $p$  aus  $P$
7. Für jede Produktion  $p$  aus  $P$ :
  - (a) Ist ein Symbol der rechten Seite von  $p$  Element aus  $V_1$  und die rechte Seite hat noch weitere Symbole, füge eine neue Regel  $p'$  zu  $P$  hinzu, bei der das Symbol aus  $p$  entfernt wurde

### 3.2 Entfernen von äquivalenten Variablen

1. Für jede Variable  $A$  aus  $V$ :
  - (a)  $[A] = \{A\}$
  - (b) Für jede Variable  $B$  aus  $V$ :
    - i. prüfe ob es eine Regel der Form  $A \rightarrow B$  und  $B \rightarrow A$  in  $P$  gibt, wenn ja füge  $B$  zu  $[A]$  hinzu.
  - (c) Wenn  $|[A]| > 1$ :
    - i. wähle irgendein Element  $C$  aus  $[A]$  aus
    - ii. Für jedes  $D$  aus  $[A]$ :
      - A. Ersetze jedes  $D$  in  $G$  durch  $C$
2. Für alle  $p$  aus  $P$ 
  - (a) Wenn bei  $p$  die linke und rechte Seite übereinstimmen, entferne  $p$  aus  $P$

### 3.3 Entfernen von Kettenregeln

1. // Die Menge der Variablen lässt sich über  $V[i]$  adressieren (wie ein Array)
2. Für  $i = |V| - 1$  bis 1:
  - (a) Für  $j = |V|$  bis  $i + 1$ :
    - i. Existiert eine Regel  $p$  mit  $V[i] \rightarrow V[j]$  in  $P$ :
      - A. Für jedes  $g$  in  $P$ 
        - Ist die linke Seite von  $g = V[j]$ :
          - Füge eine Regel der Form  $V[i] \rightarrow$  rechte Seite von  $g$  zu  $P$  hinzu
          - entferne  $p$  aus  $P$

### 3.4 Chomsky-Normalform

1. Für jedes Terminal  $t$  aus  $\Sigma$ 
  - (a) Füge eine neue Variable  $V_t$  zu  $V$  hinzu
  - (b) Füge eine neue Regel  $V_t \rightarrow t$  zu  $P$  hinzu
2. Für jede Produktion  $p$  aus  $P$  mit der Form  $p = l \rightarrow r$ 
  - (a) ist  $|r| > 1$ 
    - i. ersetze jedes Terminal  $t$  aus  $r$  durch  $V_t$
3.  $k = 1$
4. Für jede Produktion  $p$  aus  $P$  mit der Form  $p = l \rightarrow r$ 
  - (a) ist  $|r| = n > 2$ 
    - // die Symbole von  $r$  lassen sich mit  $r[i]$  adressieren
    - i. Füge eine Regel der Form  $l \rightarrow r[1] X_{k++}$  zu  $P$  hinzu
    - ii. für  $i = 2$  bis  $n - 2$ 
      - A. Füge eine Regel der Form  $X_k \rightarrow r[i] X_{k++}$  zu  $P$  hinzu

- iii. Füge eine Regel der Form  $X_k \rightarrow r[n-1]r[n]$  zu  $P$  hinzu
- iv. entferne  $p$  aus  $P$

5. Für  $i = 1$  bis  $k$

- (a) füge eine Variable  $X_i$  zu  $V$  hinzu