

Übung zur Vorlesung Theoretische Informatik I

Abgabetermin: **Mittwoch, den 23. Juni 2004** bis spätestens 12:00 Uhr.

Aufgabe 17 *Myhill-Nerode* (10 Punkte)

Sei $L_1 = \{xx^Ry \mid x, y \in \{a, b\}^* \wedge |x| > 0\}$, wobei x^R das gespiegelte Wort von x ist. Zeigen Sie mit Hilfe des Satzes von Myhill und Nerode, dass L_1 nicht regulär ist.

Aufgabe 17 **(Lösungsvorschlag)** *Myhill-Nerode*

Man betrachte das Wort $w_k = ab^{2 \cdot k + 1}a$ für $k \geq 0$. Es soll zunächst gezeigt werden, dass kein $x \in \{a, b\}^+$ existiert, so dass xx^R Präfix von w_k ist. Aus $|x| > 0$ folgt, dass xx^R mit einem a beginnen und daher auch enden muss. Es müßte also $w_k = xx^R$ gelten. Daher folgt sofort der Widerspruch

$$2 \cdot k + 1 = \#_b w_k = \#_b x + \#_b x^R = 2 \cdot \#_b x$$

Es muss also $x = w_k$ gelten für jedes $k \geq 0$, d.h. $w_k \notin L_1$. Weiterhin gilt $w_k w_k \in L_1$. Man betrachte daher für $k \neq l$ das Wort $w_k w_l = ab^{2 \cdot k + 1} a ab^{2 \cdot l + 1} a$. Dann folgt wiederum für jede Zerlegung $xx^R y$, dass xx^R mit a beginnen und daher auch enden muss, so dass nur $xx^R \in \{w_k, w_k a, w_k w_l\}$ in Frage kommt.

Der Fall $xx^R = w_k$ kann ausgeschlossen werden, da in xx^R jedes Auftreten eines Zeichens gerade sein muss, jedoch in w_k nur eine ungerade Anzahl von b -Symbolen enthalten ist. Entsprechend folgt auf Grund der Anzahl von a -Symbolen, dass $xx^R = w_k a$ nicht gelten kann. Im Fall $xx^R = w_k w_l$ folgt schließlich, dass $|w_k| = |w_l|$ gelten muss, was aber nur für $l = k$ der Fall ist.

Daher folgt $\forall k \forall l: w_k w_l \in L_2 \Leftrightarrow l = k$. Damit befinden sich alle w_k in verschiedenen Äquivalenzklassen, und es folgt, dass L_1 nicht regulär ist.

Aufgabe 18 *kontextfreie Grammatik* \rightarrow *CNF* \rightarrow *GNF*, *CYK* (3+3+3 = 9 Punkte)

Gegeben sei die kontextfreie Grammatik $G = (\{S, A, B\}, \{a, b\}, P, S)$ mit $P = \{S \rightarrow aAA, A \rightarrow B \mid SBa, B \rightarrow b \mid \varepsilon\}$.

- Überführen Sie G in Chomsky Normalform.
- Überführen Sie G in Greibach Normalform.
- Überprüfen Sie mit dem CYK-Algorithmus, ob die Wörter *ababa* und *baba* in $L(G)$ enthalten sind.

a) Überführen von G in Chomsky Normalform:

- Entfernen von ε :
 $S \rightarrow aAA$
 $A \rightarrow B \mid SBa \mid \varepsilon \mid Sa$
 $B \rightarrow b$
- Entfernen von ε :
 $S \rightarrow aAA \mid aA \mid a$
 $A \rightarrow B \mid SBa \mid Sa$
 $B \rightarrow b$
- Entfernen von Kettenableitungen $A \rightarrow B$:
 $S \rightarrow aAA \mid aA \mid a$
 $A \rightarrow b \mid SBa \mid Sa$
 $B \rightarrow b$
- Hinzufügen der Nicht-Terminale für die Ableitung der Terminale:
 $S \rightarrow T_aAA \mid T_aA \mid a$
 $A \rightarrow b \mid SBT_a \mid ST_a$
 $B \rightarrow b$
 $T_a \rightarrow a$
- Aufspalten zu langer Regeln:
 $S \rightarrow [T_aA]A \mid T_aA \mid a$
 $A \rightarrow b \mid [SB]T_a \mid ST_a$
 $B \rightarrow b$
 $T_a \rightarrow a$
 $[T_aA] \rightarrow T_aA$
 $[SB] \rightarrow SB$

b) Überführen von G in Greibach Normalform:

- Nummerieren der Variablen:
 Seien $S = N_1, A = N_2, [T_aA] = N_3, [SB] = N_4, B = N_5, T_a = N_6$.
- Modifizieren der Grammatik:
 $N_1 \rightarrow N_3N_2 \mid N_6N_2 \mid a$
 $N_2 \rightarrow b \mid N_4N_6 \mid N_1N_6$
 $N_3 \rightarrow N_6N_2$
 $N_4 \rightarrow N_1N_5$
 $N_5 \rightarrow b$
 $N_6 \rightarrow a$
- Sicherstellen, daß $N_i \rightarrow N_j\alpha$ stets $i < j$ impliziert:
 $N_1 \rightarrow N_3N_2 \mid N_6N_2 \mid a$
 $N_2 \rightarrow b \mid N_4N_6 \mid N_3N_2N_6 \mid N_6N_2N_6 \mid aN_6$
 $N_3 \rightarrow N_6N_2$
 $N_4 \rightarrow N_6N_2N_2N_5 \mid N_6N_2N_5 \mid aN_5$
 $N_5 \rightarrow b$
 $N_6 \rightarrow a$

- Nun ersetzen wir von hinten nach vorne in jeder Regel $N_i \rightarrow N_j \alpha$ das Nichtterminal N_j durch sämtliche rechte Seiten der N_j -Regeln:

$$N_6 \rightarrow a$$

$$N_5 \rightarrow b$$

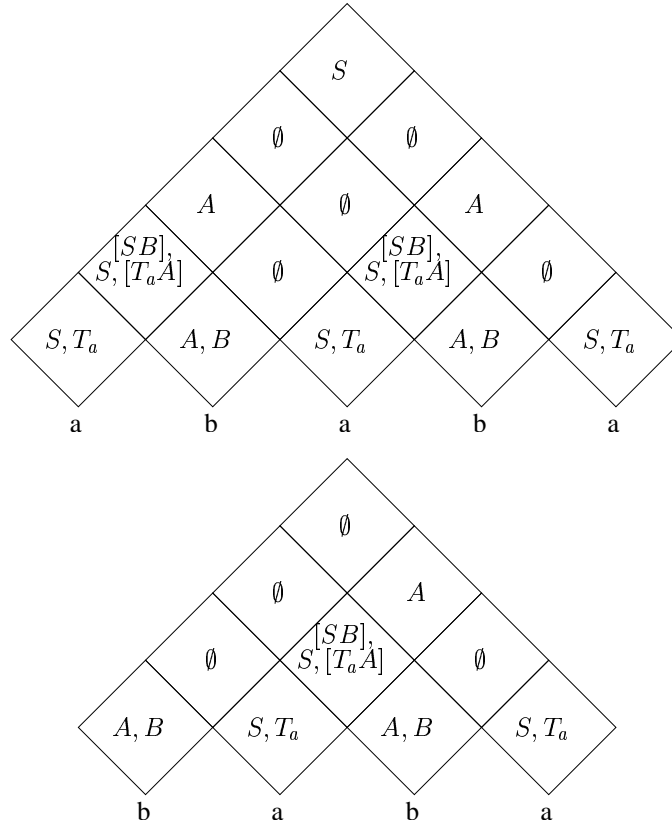
$$N_4 \rightarrow aN_2N_2N_5 \mid aN_2N_5 \mid aN_5$$

$$N_3 \rightarrow aN_2$$

$$N_2 \rightarrow b \mid aN_2N_2N_5N_6 \mid aN_2N_5N_6 \mid aN_5N_6 \mid aN_2N_2N_6 \mid aN_2N_6 \mid aN_6$$

$$N_1 \rightarrow aN_2N_2 \mid aN_2 \mid a$$

c) CYK-Algorithmus:



Aufgabe 19 Ogden's Lemma/Pumping Lemma (2+4+4+4 = 14 Punkte)

a) Beweisen Sie das folgende Lemma:

- Sei L eine reguläre Sprache. Dann existiert eine Konstante n_L , so dass für beliebiges $z \in L$ mit $|z| \geq n_L$ gilt: Markiert man in z beliebig mindestens n_L verschiedene Zeichen, so existiert eine Zerlegung $z = uvw$ mit:
 - * in v ist mindestens ein Zeichen markiert.
 - * in v sind höchstens n_L Zeichen markiert.
 - * für beliebiges $i \in \mathbb{N}$ gilt $uv^i w \in L$.
- Sei L eine kontextfreie Sprache. Dann existiert eine Konstante n_L , so dass für beliebiges $z \in L$ mit $\#z \geq n_L$ gilt: Markiert man in z beliebig mindestens n_L verschiedene Zeichen, so existiert eine Zerlegung $z = uvwxy$ mit:

- * in vx ist mindestens ein Zeichen markiert.
- * vw besitzt höchstens n_L markierte Zeichen.
- * für beliebiges $i \in \mathbb{N}$ gilt $uv^iwx^iy \in L$.

b) Zeigen Sie unter Verwendung von Teilaufgabe a) oder dem Pumping Lemma für kontextfreie Sprachen, dass die folgenden Sprachen nicht kontextfrei sind:

- (i) $L_3 = \{xx^R x \mid x \in \{a, b\}^*\}$
- (ii) $L_4 = \{a^n b^n c^i \mid i \neq n\}$
Hinweis: Das Wort $a^n b^n c^{n+n!}$ könnte hilfreich sein.

Aufgabe 19 **(Lösungsvorschlag)** *Ogden's Lemma/Pumping Lemma*

a) Ogden's Lemma

- (i) Wie bei dem Pumping Lemma für reguläre Sprachen sei n_L die Anzahl der Zustände eines DFA, welcher L akzeptiert. Sei weiterhin $z \in L$ mit $|z| \geq n_L$. Das längste Wort von L , für welches der DFA nicht in einen Zyklus laufen muß, hat die Länge $n_L - 1$. Damit liegt mindestens ein markiertes Zeichen von z in einem Zyklus und somit in v . Da der längste Zyklus höchstens n_L Zustände haben kann, besitzt v höchstens n_L markierte Zeichen.
- (ii) Wie bei dem Pumping Lemma für kontextfreie Sprachen sei $k = \#V$ die Anzahl der Variablen einer Grammatik, welche L produziert. Dabei sei die Grammatik ohne Einschränkung in CNF. Dann setze man $n_L := 2^k + 1$. Beginnend bei S konstruiere man einen Pfad durch den Ableitungsbaum zu z . Bei jedem inneren Knoten des Ableitungsbaums (dieser entspricht einer Regelanwendung der Art $A \rightarrow BC$) entscheide man sich für den Teilbaum, welcher mehr markierte Zeichen/Blätter trägt. Damit wird die Anzahl der erreichbaren markierten Zeichen bei jedem Knoten des so konstruierten Pfades höchstens halbiert. Da $n_L = 2^k + 1$ Zeichen markiert sind, muss dieser Pfad mindestens $k + 1$ Knoten/Verzweigungen besitzen, und bei mindestens $k + 1$ dieser Knoten/Verzweigungen wird die Anzahl der noch markierten Zeichen echt kleiner. Dabei muß aber mindestens ein Nicht-Terminal zweimal besucht worden sein. Man kann daher beginnend bei dem (markierten) Endknoten dieses Pfades rückwärts in dem Ableitungsbaum aufsteigen, bis das erste Mal ein mehrmals auf diesem Pfad liegendes Nicht-Terminal besucht wird. Der Rest des Beweises folgt exakt dem Beweis für das Pumping Lemma.

Für genauere Informationen siehe zum Beispiel Ingo Wegener, *Theoretische Informatik - eine algorithmenorientierte Einführung* oder John Hopcroft und Jeffrey Ullman, *Einführung in die Automatentheorie, formale Sprachen und Komplexitätstheorie*.

b) Pumping Lemma

- (i) $L_3 = \{xx^R x\}$
(Pumping Lemma): Betrachte $z = ba^n bba^n bba^n b$, wobei n die Konstante aus dem Pumping Lemma ist (unter der Annahme, dass L_3 kontextfrei ist). Sei $z = uvwx$ die Zerlegung aus dem Pumping Lemma. Wegen $1 \leq |vwx| \leq n$ existieren nur folgende Möglichkeiten von vw :

- * $vwx = ba^k$ mit $0 \leq k < n$
 - Gilt $vx = ba^m$, so kann durch Pumpen ein Wort erzeugt werden, in welchem die Anzahl der enthaltenen b -Symbole nicht durch drei teilbar ist, und somit auch keine Faktorisierung $uu^R u$ existieren kann
 - Gilt $vx = a^m$, dann wird ausschließlich in genau einem a -Block gepumpt. Da u genau zwei b -Symbole enthalten muß, folgt der Widerspruch.
- * $vwx = a^k$ mit $k \leq n$ (Analog zur 1. Möglichkeit)
- * $vwx = a^k b$ mit $k < n$ (Analog zur 1. Möglichkeit)
- * $vwx = a^k b b a^l$ mit $k + l \leq n - 2$
 - Enthält vx mindestens ein b , so kann durch Pumpen ein Wort mit einer nicht durch drei teilbaren Anzahl an b -Symbolen erzeugt werden, und somit kann auch keine Faktorisierung $uu^R u$ existieren.
 - Enthält vx ausschließlich a -Symbole, so wird entweder nur in einem oder in zwei a -Blöcken gepumpt, aber niemals in allen. Da die Anzahl der b -Symbole konstant 6 bleibt und u genau zwei b -Symbole enthalten muß, folgt der Widerspruch.

Allgemein gilt für $z \in L_3$, dass die Anzahl der Auftreten eines Zeichens in z ein Vielfaches von 3 sein muss.

In den Fällen, dass vx mindestens ein, höchstens zwei b -Symbole enthält, kann immer ein Wort konstruiert werden, welches eine nicht durch drei teilbare Anzahl von b -Symbolen hat, und daher auch nicht in L_3 liegen kann.

Besteht andererseits vx nur aus a -Symbolen, so wird höchstens in zwei durch b -Symbole getrennten Blöcken gepumpt. In diesem Fall bleibt die Anzahl der b -Symbole konstant gleich 6. Daher muss in jeder Faktorisierung $z = uu^R u$ der Faktor u genau zwei b -Symbole enthalten, und vor allem mit einem b beginnen und enden. Daher kann auch in diesem Fall immer ein Wort erzeugt werden, welches nicht in L_3 liegt.

(ii) $L_4 = \{a^n b^n c^i \mid i \neq n\}$

Sei L_4 kontextfrei und n die daher existierende Konstante aus einem der beiden Lemmata. Sei $z = a^n b^n c^{n+n!} \in L_4$. Dann existiert nach Annahme eine Zerlegung $z = uvwxy$, so dass für alle $i \in \mathbb{N}$ $uv^i w x^i y \in L_4$ gilt. Markiere die ersten n a -Symbole. Fälle, dass sich v oder x aus zwei oder drei verschiedenen Zeichen zusammensetzen, können auf Grund der geforderten Struktur des Wortes ausgeschlossen werden. Daher muss für jede mögliche Zerlegung auf Grund von Ogden's Lemma $v = a^k$ mit $k > 0$ (nur die a -Symbole sind markiert und vx enthält mindestens eine Markierung) und $x \in \{a^l, b^l, c^l \mid l \in \mathbb{N}\}$ gelten.

Es gelte $x = a^l$. Dann kann sofort ein Wort mit $\#_a \neq \#_b$ erzeugt werden. Entsprechendes gilt im Fall $x = c^l$. Sei also $v = a^k$ und $x = b^l$. Im Fall $k \neq l$ kann wieder ein Wort erzeugt werden, welches die Eigenschaft $\#_a = \#_b$ verletzt. Im Fall $l = k$ müsste jedoch, da $l \leq n$ ein Faktor von $n!$ ist, das Wort $a^{n+n!} b^{n+n!} c^{n+n!}$ in L_4 liegen. Widerspruch. Daher kann keine geeignete Zerlegung existieren, woraus folgt, dass L_4 nicht kontextfrei sein kann.

Aufgabe 20 kontextfreie Grammatiken (3+3 = 6 Punkte)

Geben Sie für die folgenden Sprachen kontextfreie Grammatiken an:

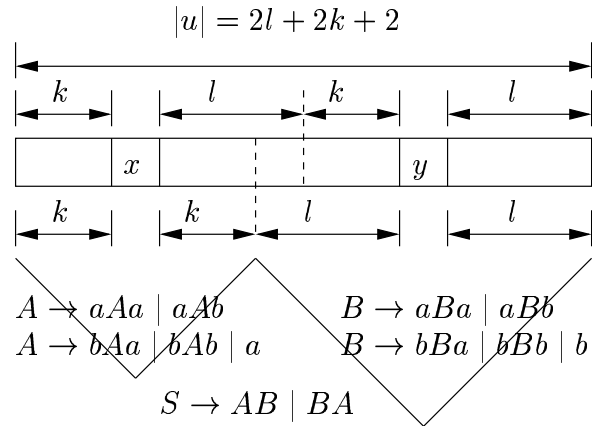
a) $L_5 = \{w \in \{a, b\}^* \mid \#_a w = 2 \cdot \#_b w\}$, wobei $\#_a w$ ($\#_b w$) die Anzahl von a -Zeichen (b -Zeichen) in w bezeichnet.

b) $L_6 = \{a, b\}^* \setminus \{ww \mid w \in \{a, b\}^*\}$

Aufgabe 20 (Lösungsvorschlag) *kontextfreie Grammatiken*

a) $S \rightarrow aSaSbS \mid aSbSaS \mid bSaSaS \mid \varepsilon$

b) Ein Wort u ist nicht von der Form ww , falls $|u|$ ungerade ist oder $|u|$ gerade ist, aber u eine "Fehlerstelle" enthält.



Die obige Graphik zeigt, dass ein Wort u der Länge $2k + 2l + 2$ nicht von der Form ww ist, wenn $x \neq y$ für $x, y \in \{a, b\}$, und wie solche Wörter erzeugt werden können.

Um auch die Wörter ungerader Länge zu erhalten, genügen zwei weitere Regeln:

$S \rightarrow AB \mid BA \mid A \mid B$
 $A \rightarrow xAy \mid a$ $x, y \in \{a, b\}$
 $B \rightarrow xBy \mid b$ $x, y \in \{a, b\}$