Fachbereich Mathematik
Prof. Dr. Martin Otto
Alexander Kartzow
Alexander Kreuzer
Benno van den Berg

TECHNISCHE
UNIVERSITÄT
DARMSTADT

SS 2010
19.05.2010

# 6. Übungsblatt zu FGdI 1

## Gruppenübung

**Aufgabe G1**

Betrachten Sie die kontextfreie Sprache $L$, die von der folgenden Grammatik $G$ erzeugt wird:

$$G: \quad X_0 \rightarrow ab \mid ba \mid X_0X_0 \mid aX_0b \mid bX_0a$$

(a) Beschreiben Sie $L$ umgangssprachlich.

(b) Bringen Sie $G$ in Chomsky-Normalform.

(c) Wenden Sie den CYK Algoritmus an, um zu bestimmen, ob $bbab \in L$ und $aabbab \in L$.

(d) Konstruieren Sie einen Kellerautomaten, welcher $L$ erkennt.

**Musterlösung:**

(a) $L$ besteht aus den Wörtern in $\{a,b\}^*$, die nicht leer sind und genauso viele $a$ wie $b$ enthalten.
Klar ist, dass alle Wörter $w$, die man ableiten kann, genauso viele $a$ wie $b$ enthalten und dass das leere Wort nicht ableitbar ist. Andererseits kann man das Argument in Aufgabe G1 auf dem 5. Übungsblatt einfach zu einem Beweis umschreiben, dass alle nicht-leeren Wörter mit genauso vielen $a$ wie $b$ in $G$ ableitbar sind.

(b) Es gibt keine Regeln der Form $X \rightarrow Y$ oder $X \rightarrow \epsilon$, deshalb können wir direkt damit beginnen, Buchstaben durch Variablen zu ersetzen und erhalten:

$$\begin{array}{lcl} X_0 & \rightarrow & Z_aZ_b \mid Z_bZ_a \mid X_0X_0 \mid Z_aX_0Z_b \mid Z_bX_0Z_a \\ Z_a & \rightarrow & a \\ Z_b & \rightarrow & b \end{array}$$

Nun werden die Regeln der Form $X \rightarrow Y_1Y_2 \dots Y_n$ mit $n \geq 3$ aufgelöst und wir erhalten als Endergebnis:

$$\begin{array}{lcl} X_0 & \rightarrow & Z_aZ_b \mid Z_bZ_a \mid X_0X_0 \mid Z_aX \mid Z_bY \\ X & \rightarrow & X_0Z_b \\ Y & \rightarrow & X_0Z_a \\ Z_a & \rightarrow & a \\ Z_b & \rightarrow & b \end{array}$$

(c) Erzeugbare Teilworte der Länge 1:

$$\begin{array}{lcl} a & : & Z_a \\ b & : & Z_b \end{array}$$

Erzeugbare Teilworte der Länge 2:

$$\begin{array}{lcl} aa & : & - \\ ab & : & X_0 \\ bb & : & - \\ ba & : & X_0 \end{array}$$

Erzeugbare Teilworte der Länge 3:

$$\begin{array}{lcl} aab & : & - \\ abb & : & X \\ bba & : & - \\ bab & : & X \end{array}$$

Erzeugbare Teilworte der Länge 4:

$$\begin{array}{lcl} aabb & : & X_0 \\ abba & : & X_0 \\ bbab & : & - \end{array}$$

Erzeugbare Teilworte der Länge 5:

$$\begin{array}{lcl} aabba & : & Y \\ abbab & : & X \end{array}$$

Deshalb ist *bbab* nicht erzeugbar und *aabbab* ist erzeugbar (z.B. mittels $X_0 \to Z_a X$).

(d) Sei $\mathcal{P} = (\Sigma, Q, q_0, \Delta, A, \Gamma, \#)$ der Kellerautomat mit Eingabealphabet $\Sigma = \{a, b\}$, Zustandsmenge $Q = \{q_0, q_1\}$, $q_0$ als Anfangzustand, $A = \{q_1\}$ als Menge der akzeptierenden Zustände, Kelleralphabet $\Gamma = \{\#, A, B\}$ und Übergangsrelation $\Delta$ gegeben durch

$$\begin{array}{ll} \{ & (q_0, \#, a, A\#, q_1) \\ & (q_0, \#, b, B\#, q_1) \\ & (q_1, A, a, AA, q_1) \\ & (q_1, A, b, \epsilon, q_1) \\ & (q_1, B, a, \epsilon, q_1) \\ & (q_1, B, b, BB, q_1) \\ & (q_1, \#, a, A\#, q_1) \\ & (q_1, \#, b, B\#, q_1) \\ & (q_1, \#, \epsilon, \epsilon, q_1) \qquad \}. \end{array}$$

$\mathcal{P}$ erkennt die Sprache $L$. Der Automat ist in Zustand $q_1$ genau dann, wenn ein nichtleeres Wort bereits gelesen wurde. Außerdem wird der Keller benutzt um die Differenz der $a$ und $b$ im bisher gelesenen Wort zu zählen: Wenn bisher das Wort $w$ gelesen wurde und $|w|_a > |w|_b$ dann befinden sich genau $|w|_a - |w|_b$ viele $A$ im Speicher, falls $|w|_b > |w|_a$ gilt die analoge Aussage für $B$.

Alternativ kann man auch die Konstruktion aus dem Beweis des Satzes 4.1.5 aus dem Skript anwenden, die eine beliebige kontextfreie Grammatik in einen Kellerautomaten übersetzt, der die von der Grammatik erzeugte Sprache akzeptiert.

**Aufgabe G2**

(a) Zeigen Sie, dass die Sprache

$$L = \{a^n b^m a^n b^m \,:\, m, n \geq 0\}$$

nicht kontextfrei ist.

(b) Sei $L$ eine kontextfreie Sprache und $M$ eine reguläre $\Sigma$-Sprache. Zeigen Sie, dass $L \cap M$ eine kontextfreie $\Sigma$-Sprache ist.

Hinweis: Sei $\mathcal{P}$ ein Kellerautomat für $L$, und $\mathcal{A}$ ein NFA für $M$. Konstruieren Sie daraus (wie in Lemma 2.2.11(a) im Skript) einen Kellerautomat $\mathcal{Q}$, der $L \cap M$ erkennt.

(c) Benutzen Sie Aufgabenteilen (a) und (b) um zu schließen, dass

$$N = \{ww \,:\, w \in \{a, b\}^*\}$$

keine kontextfreie Sprache ist.

(d) Zusatzaufgabe: Warum kann man nicht 2 Kellerautomaten parallel simulieren? Warum ist der Durchschnitt kontextfreier Sprachen im allgemeinen nicht kontextfrei?

**Musterlösung:**

(a) Wir zeigen, dass die Bedingung des Pumping Lemmas verletzt ist. Das heißt:

> Für jedes $i \in \mathbb{N}$ gibt es ein $x \in L$ mit $|x| \geq i$, so dass für alle Zeichenreihen $y, u, v, w, z$, mit $x = y \cdot u \cdot v \cdot w \cdot z$, $|uw| > 0$ und $|uvw| \leq i$, es ein $j \in \mathbb{N}$ gibt mit
>
> $$y \cdot u^j \cdot v \cdot w^j \cdot z \notin L.$$

Um dies zu zeigen, sei $i \in \mathbb{N}$ beliebig und

$$x = a^i b^i a^i b^i.$$

Offensichtlich ist $x \in L$ und $|x| \geq i$. Seien also $y, u, v, w, z$, mit $x = y \cdot u \cdot v \cdot w \cdot z$, $|uw| > 0$ und $|uvw| \leq i$. Wir wählen $j = 2$ und behaupten

$$x' = y \cdot u^2 \cdot v \cdot w^2 \cdot z \notin L.$$

**Beweis:** Nenne die Teilwörter der Form $a^i$ und $b^i$ in $x$ *Blöcke*. Es gibt jetzt drei Möglichkeiten:

- Entweder $u$ oder $w$ überschreitet eine Blockgrenze (beides zusammen ist wegen $|uvw| \leq n$ unmöglich!). Dann enthält entweder $u$ oder $w$ sowohl $a$ als $b$. Das bedeutet, dass $x'$ nicht die Form $a^*b^*a^*b^*$ hat und deshalb nicht in $L$ enthalten sein kann.
- $u$ und $w$ sind in demselben Block enthalten. Dann hat $x'$ einen Block der länger ist als die andere drei und ist deshalb nicht in $L$ enthalten.
- $u$ und $w$ sind in verschiedenen Blöcken. Diese Blöcke müssen benachbart sein, da $|uvw| \leq i$. Das heißt, dass entweder $u$ nur aus $a$ besteht und $w$ aus $b$, oder umgekehrt. Dann haben die zwei $a$-Blöcke in $x'$ verschiedene Länge (wenn $uw$ $a$ enthält), oder die zwei $b$-Blöcke (wenn $uw$ $b$ enthält; beides ist natürlich auch möglich!). Jedenfalls ist $x'$ nicht in $L$ enthalten.

(b) Sei $\mathcal{P} = (\Sigma, Q^1, q_0^1, \Delta^1, A^1, \Gamma, \#)$ ein Kellerautomat, der $L$ erkennt, und $\mathcal{A} = (\Sigma, Q^2, q_0^2, \Delta^2, A^2)$ ein NFA der $M$ erkennt. Wir konstuieren einen Kellerautomat $\mathcal{Q}$, der $\mathcal{P}$ und $\mathcal{A}$ parallel simuliert analog zur Bildung von Produktautomaten für den Schnitt regulärer Sprachen. Sei $\mathcal{Q} := (\Sigma, Q, q_0, \Delta, A, \Gamma, \#)$, wobei:

$$\begin{aligned} Q &= Q^1 \times Q^2 \\ q_0 &= (q_0^1, q_0^2) \\ A &= A^1 \times A^2, \end{aligned}$$

und

$$\begin{aligned} ((q,p), \gamma, x, \beta, (q', p')) \in \Delta \quad :\Leftrightarrow \quad & \big((q, \gamma, x, \beta, q') \in \Delta^1 \text{ und } (p, x, p') \in \Delta^2 \text{ und } x \neq \epsilon\big) \text{ oder} \\ & \big((q, \gamma, x, \beta, q') \in \Delta^1 \text{ und } p = p' \text{ und } x = \epsilon\big) \end{aligned}$$

(c) Sei $L$ die Sprache aus Aufgabe (a) und $M := L(a^*b^*a^*b^*)$. $M$ ist regulär und es gilt $L = M \cap N$. Wäre $N$ also kontextfrei, dann wäre nach Aufgabe (b) $L$ auch kontextfrei. Wir haben aber in Aufgabe (a) gezeigt, dass $L$ nicht kontextfrei ist.

# Hausübung

**Aufgabe H1** (6 Punkte)

Der Parser eines Compilers arbeitet üblicherweise in zwei Schritten. Zunächst wird als Vorverarbeitung die Eingabe in eine Folge sogenannter >Tokens< zerlegt. Die eigentliche Syntaxanalyse arbeitet dann mit Wörtern über dem Alphabet, welches aus diesen Tokens besteht. Beispielsweise könnte die Eingabe

```
fac n := if n = 0 then 1 else n * fac (n-1);
```

in die Token-Folge

$$id\ id := if\ id\ op\ num\ then\ num\ else\ id\ op\ id\ (\ id\ op\ id\ )\ ;$$

konvertiert werden. Wir betrachten das Tokenalphabet

$$\Sigma := \{id, op, num, if, then, else, :=, (, ), ;\}$$

und die Grammatik $G = (\Sigma, V, P, S)$ mit Produktionen

$$\begin{aligned}
&S \to D \mid D\ S\\
&D \to id\ A := E\ ;\\
&A \to \varepsilon \mid id\ A\\
&E \to id \mid num \mid (\ E\ ) \mid E\ op\ E \mid id\ P \mid if\ E\ then\ E\ else\ E\\
&P \to E \mid E\ P
\end{aligned}$$

(a) Transformieren Sie $G$ in Chomsky-Normalform und bestimmen Sie mit Hilfe des CYK-Algorithmus, ob das Wort

$$id\ id\ :=\ id\ (\ id\ op\ id\ op\ num\ )\ ;$$

zu $L(G)$ gehört.

(b) Konstruieren Sie einen Kellerautomaten, welcher $L(G)$ erkennt.

**Musterlösung:**

(a) 1. Schritt: $X \to X_0$ eliminieren.

$$\begin{aligned}
S &\to id\ A := E\ ; \mid D\ S\\
D &\to id\ A := E\ ;\\
A &\to \varepsilon \mid id\ A\\
E &\to id \mid num \mid (\ E\ ) \mid E\ op\ E \mid id\ P \mid if\ E\ then\ E\ else\ E\\
P &\to id \mid num \mid (\ E\ ) \mid E\ op\ E \mid id\ P \mid if\ E\ then\ E\ else\ E \mid E\ P
\end{aligned}$$

2. Schritt: $\epsilon$-Produktionen eliminieren.

$$\begin{aligned}
S &\to id\ A := E\ ; \mid id\ := E\ ; \mid D\ S\\
D &\to id\ A := E\ ; \mid id\ := E\ ;\\
A &\to id \mid id\ A\\
E &\to id \mid num \mid (\ E\ ) \mid E\ op\ E \mid id\ P \mid if\ E\ then\ E\ else\ E\\
P &\to id \mid num \mid (\ E\ ) \mid E\ op\ E \mid id\ P \mid if\ E\ then\ E\ else\ E \mid E\ P
\end{aligned}$$

3. Schritt: Variablen für Buchstaben.

$$
\begin{array}{lcl}
S & \to & IAGEK \mid IGEK \mid DS \\
D & \to & IAGEK \mid IGEK \\
A & \to & id \mid IA \\
E & \to & id \mid num \mid XEY \mid EOE \mid IP \mid FETELE \\
P & \to & id \mid num \mid XEY \mid EOE \mid IP \mid FETELE \mid EP \\
I & \to & id \\
O & \to & op \\
G & \to & := \\
K & \to & ; \\
X & \to & ( \\
Y & \to & ) \\
F & \to & if \\
T & \to & then \\
L & \to & else
\end{array}
$$

4. Schritt: $X \to X_0 \dots X_k$ mit $k \geq 2$ eliminieren.

$$
\begin{array}{ll}
S \to ID_0 \mid ID_1 \mid DS & I \to id \\
D \to ID_0 \mid ID_1 & O \to op \\
D_0 \to AD_1 & G \to := \\
D_1 \to GD_2 & K \to ; \\
D_2 \to EK & X \to ( \\
E \to id \mid num \mid XE_0 \mid EE_1 \mid IP \mid FE_2 & Y \to ) \\
E_0 \to EY & F \to if \\
E_1 \to OE & T \to then \\
E_2 \to EE_3 & L \to else \\
E_3 \to TE_4 & A \to id \mid IA \\
E_4 \to EE_5 & \\
E_5 \to LE & \\
P \to id \mid num \mid XE_0 \mid EE_1 \mid IP \mid FE_2 \mid EP &
\end{array}
$$

| | $id$ | $id$ | $:=$ | $id$ | $($ | $id$ | $op$ | $id$ | $op$ | $num$ | $)$ | $;$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | $A, I, E, P$ | $A, I, E, P$ | $G$ | $A, I, E, P$ | $X$ | $A, I, E, P$ | $O$ | $A, I, E, P$ | $O$ | $E, P$ | $Y$ | $K$ |
| 2 | $A, E, P$ | – | – | – | – | – | $E_1$ | – | $E_1$ | $E_0$ | – | |
| 3 | – | – | – | – | – | $E, P$ | – | $E, P$ | – | – | | |
| 4 | – | – | – | – | – | – | $E_1$ | $E_0$ | – | | | |
| 5 | – | – | – | – | – | $E, P$ | – | – | | | | |
| 6 | – | – | – | – | – | $E_0$ | – | | | | | |
| 7 | – | – | – | – | $E, P$ | – | | | | | | |
| 8 | – | – | – | $E, P$ | $D_2$ | | | | | | | |
| 9 | – | – | – | $D_2$ | | | | | | | | |
| 10 | – | – | $D_1$ | | | | | | | | | |
| 11 | – | $D, D_0, S$ | | | | | | | | | | |
| 12 | $D, S$ | | | | | | | | | | | |

Da das Startsymbol $S$ in der untersten Zeile steht, gehört das Wort zu $L(G)$.

(b) $\mathcal{P} = (\Sigma, Q, q_0, \Delta, A, \Gamma, \#)$ mit $Q = \{q_0\}$, $A = \{q_0\}$, $\Gamma = \{S, D, A, E, P, \#\} \cup \Sigma$ und

Transitionen

$$
\begin{array}{llll}
(q_0,\ \#,\ \varepsilon,\ S, & q_0) & (q_0,\ id, & id, \quad \varepsilon,\ q_0) \\
(q_0,\ S,\ \varepsilon,\ D, & q_0) & (q_0,\ op, & op, \quad \varepsilon,\ q_0) \\
(q_0,\ S,\ \varepsilon,\ DS, & q_0) & (q_0,\ num, & num,\ \varepsilon,\ q_0) \\
(q_0,\ D,\ \varepsilon,\ id\,A := E\,;, & q_0) & (q_0,\ if, & if, \quad \varepsilon,\ q_0) \\
(q_0,\ A,\ \varepsilon,\ \varepsilon, & q_0) & (q_0,\ then, & then,\ \varepsilon,\ q_0) \\
(q_0,\ A,\ \varepsilon,\ id\,A, & q_0) & (q_0,\ else, & else, \quad \varepsilon,\ q_0) \\
(q_0,\ E,\ \varepsilon,\ num, & q_0) & (q_0,\ :=, & :=, \quad \varepsilon,\ q_0) \\
(q_0,\ E,\ \varepsilon,\ (E), & q_0) & (q_0,\ (, & (, \quad \varepsilon,\ q_0) \\
(q_0,\ E,\ \varepsilon,\ E\ op\ E, & q_0) & (q_0,\ ), & ), \quad \varepsilon,\ q_0) \\
(q_0,\ E,\ \varepsilon,\ id\,P, & q_0) & (q_0,\ ;, & ;, \quad \varepsilon,\ q_0) \\
(q_0,\ E,\ \varepsilon,\ if\ E\ then\ E\ else\ E, & q_0) & & \\
(q_0,\ P,\ \varepsilon,\ E, & q_0) & & \\
(q_0,\ P,\ \varepsilon,\ EP, & q_0) & &
\end{array}
$$

**Aufgabe H2** (6 Punkte)

(a) Welche Sprache beschreibt die Grammatik $G = (\Sigma, V, P, S)$ mit $\Sigma := \{a, b, c, \$\}$, $V := \{S, A, B, X, Y\}$ und Produktionen

$$
\begin{array}{ll}
S \to AB & Xa \to aX \\
A \to aXA \mid \$ & X\$ \to \$Y \\
B \to bYB \mid \$ & Yb \to bY \\
 & Y\$ \to \$c
\end{array}
$$

Beweisen Sie ihre Behauptung durch Induktion über die Menge der in $G$ ableitbaren $(V \cup \Sigma)$-Wörter. Stellen Sie hierzu zuächst eine geeignete Induktionsbehauptung auf.

(b) Geben Sie eine kontextfreie Grammatik für die Sprache aus (a) an, d. h. eine Grammatik, bei welcher die linke Seite jeder Produktion aus einer einzigen Variablen besteht.

**Musterlösung:**

(a) Wir behaupten: $L(G) = \{a^k\$b^n\$c^{k+n} \,:\, k, n \in \mathbb{N}\}$.

($\supseteq$) Jedes Wort der Form $w := a^k\$b^n\$c^{k+n}$ kann in $G$ abgeleitet werden:

$$
\begin{aligned}
S &\to_G AB \to_G aXAB \to_G aXaXAB \to_G \cdots \to_G (aX)^k AB \to_G (aX)^k\$B \\
&\to_G (aX)^k\$bYB \to_G \cdots \to_G (aX)^k\$(bY)^nB \to_G (aX^k)\$(bY)^n\$ \\
&\to_G \cdots \to_G a^kX^k\$(bY)^n\$ \to_G \cdots \to_G a^k\$Y^k(bY)^n\$ \\
&\to_G \cdots \to_G a^k\$b^nY^{k+n}\$ \to_G \cdots \to_G a^k\$b^n\$c^{k+n}
\end{aligned}
$$

($\subseteq$) Wir beweisen die folgenden Aussagen durch Induktion über den Ableitungsprozeß.

(1) Gilt $S \to_G^* w$, dann ist $w = S$ oder $w$ ist von der Form $w = uxvyz$ für geeigenete Wörter

$$u \in \{a, X\}^*, \quad v \in \{b, Y\}^*, \quad z \in \{c\}^*, \quad x \in \{A, \$\} \quad \text{und} \quad y \in \{B, \$\}\,.$$

(2) Gilt $S \to_G^* w$, so ist $|w|_a + |w|_b = |w|_X + |w|_Y + |w|_c$.

Induktionsanfang: Für $w = S$ sind die Aussagen offensichtlich erfüllt.

Induktionsschritt: Angenommen die Ableitung ist $S \to_G^* w' \to_G w$. Falls $w' = S$, dann ist $w = AB$ und (1) und (2) erfüllt. Wir nehmen deshalb an, dass $w' \neq S$. Nach Induktionshypothese erfüllt $w'$ die Bedingungen (2) und es gibt eine Zerlegung $w' =$

$u'x'v'y'z'$, wie unter (1). Für jede mögliche Produktion, die von $w'$ nach $w$ führen könnte, müssen wir untersuchen, ob sie die Eigenschaften (1) und (2) erhält. Wir betrachten exemplarisch einige Fälle.

($A \to aXA$) In diesem Fall ist $x' = A$ und $w = u'aXAv'y'z'$. Desweiteren gilt

$$|w|_a + |w|_b = |w'|_a + |w'|_b + 1 = |w'|_X + |w'|_Y + |w'|_c + 1 = |w|_X + |w|_Y + |w|_c\,.$$

($Yb \to bY$) In diesem Fall ist $w = u'x'vy'z'$, wobei das Wort $v$ aus $v'$ entsteht, indem zwei Buchstaben vertauscht wurden. An der Anzahl der verschiedenen Buchstaben ändert sich nichts.

($Y\$ \to \$c$) Dann ist $w = u'x'v\$cz'$ mit $v' = vY$ und $y' = \$$. Es verringert sich die Anzahl der $Y$ um 1. Dafür steigt die Zahl der $c$.

Die verbleibenden Fälle zeigt man analog.

(b)

$$S \to aSc \mid \$X$$
$$X \to bXc \mid \$$$