Universität Karlsruhe (TH) – Fakultät für Informatik
Institut für Logik, Komplexität und Deduktionssysteme
Lehrstuhl Prof. Dr. P. Deussen

Informatik III – Prof. Dr. P. Deussen im Wintersemester 2001/2002

C. Bierwald, Th. Käufl, F. Schuster   http://i10www.ira.uka.de/lehre/info3-winter2001   info3@ira.uka.de

Übungsblatt 2 – Lösungsvorschläge

26.10.2001

Aufgabe 1 Kontextfreie Grammatiken (Schuster)

Geben Sie kontextfreie Grammatiken an, welche die folgenden Sprachen erzeugen

a) $L_a = \{a^n c^k a^n : n \in \mathbb{N}, n \geq 1, k \in \mathbb{N}\}$ und $T = \{a, b, c\}$

b) $L_b = \{a^n b^{2n} : n \in \mathbb{N}, n \geq 1\}$ und $T = \{a, b\}$

c) $L_c = B(B \cup Z)^*$ mit $B = \{\mathrm{a, b, \ldots, z}\}$, $Z = \{0, 1, \ldots, 9\}$ und $T = B \cup Z$

a) Π_a: $S ::= aCa \mid aSa$, $C ::= Cc \mid \varepsilon$

b) Π_b: $S ::= abb \mid aSbb$

c) Π_c: $S ::= BR$, $B ::= a \mid b \mid \ldots \mid y \mid z$, $Z ::= 0 \mid 1 \mid \ldots \mid 9$, $R ::= \varepsilon \mid BR \mid ZR$

Aufgabe 2 Die Sprache „$a^n b^n$“ (Schuster)

Gegeben sei die Grammatik G mit dem Startzeichen S und den Produktionen $\Pi = \{S ::= aSb$?Operator-Alternativ? $ab\}$. Zeigen Sie, daß $L(G) = \{a^n b^n : n \geq 1\}$ gilt.

1. Zeige $L(G) \subseteq L$, d.h. für alle $x \in L(G)$ gilt $x \in L$

   a) Induktion über die Länge der Ableitung
   Induktionsanfang: $m = 1$
   D.h. $|S \Rightarrow x| = 1$ und somit $S \rightarrow x$ mit $x \in L(G)$. Da nur $S ::= ab$ direkt Terminale erzeugt, gilt $x = ab$ und dies liegt nach Definition in L.
   Induktionsschritt: m nach $m + 1$

$$
\begin{tikzcd}
m: & \begin{array}{c} |S \Rightarrow x| = m \\ x \in L(G) \end{array} \arrow[r, "\text{dann}"] & \begin{array}{c} \text{es gibt } n \text{ mit} \\ x = a^n b^n \in L \end{array} \arrow[d] & \text{Annahme} \\
m+1: & \begin{array}{c} |S \Rightarrow x'| = m+1 \\ x' \in L(G) \end{array} \arrow[u] \arrow[r, "\text{dann}"] & x' \in L & \text{Behauptung}
\end{tikzcd}
$$

1. Da $|S \Rightarrow x'| = m + 1 > 0$ gilt, zerglegt man $S \Rightarrow x'$ in $S \rightarrow aSb \Rightarrow x'$.
2. Die Induktionsannahme führt zu $S \rightarrow aSb \Rightarrow axb = x'$ – dies ist möglich, da nach Annahme jede Ableitung, d.h. auch jene, die zu x' führt verwendet werden darf.
3. Weil $x \in L$ gilt, gibt es ein n mit $x = a^n b^n$.
4. Letztendliche erhalten wir $S \rightarrow aSb \Rightarrow axb = x' = aa^n b^n b$ und $a^{n+1} b^{n+1}$ ist nach Definition in L enthalten.

b) Beweis über die Satzformen mit Induktion
Zeige, daß es für alle $n \geq 0$ Ableitungen $S \Rightarrow a^n S b^n$ gibt.
Induktion über die Länge der Ableitung
Induktionsanfang: $m = 0$
Es muß gelten $|S \Rightarrow a^n S b^n| = 0$, d.h. es muß $S = a^n S b^n$ gezeigt werden – was mit $n = 0$ sofort gilt.
Induktionsschritt: m nach $m + 1$

m:	$\lvert S \Rightarrow z' \rvert = m$	mit	$z' = a^n S b^n$	Annahme
$m+1$:	$\lvert S \Rightarrow z \rvert = m + 1$	mit	$z = a^{n+1} S b^{n+1}$	Behauptung

1. Da $|S \Rightarrow z| = m + 1 > 0$ gilt, kann die Ableitung in $S \Rightarrow z' \rightarrow z$ zerlegt werden.
2. Setzt man die Induktionsannahme ein, so ergibt dies $S \Rightarrow a^n S b^n \rightarrow z$.
3. Wird im letzten Ableitungsschritt $S ::= aSb$ eingesetzt, so erhält man letztendlich die gewünschte Satzform $S \Rightarrow a^n S b^n \rightarrow a^n aSbn^n = a^{n+1} S b^{n+1}$.

2. Zeige $L(G) \supseteq L$, d.h. für alle $x \in L$ gilt $x \in L(G)$
Induktion über die Wörter von L mittels n
Induktionsanfang: $n = 1$
Das erste Wort ist demnach $a^n b^n = ab$. Dies liegt in $L(G)$, da $S \rightarrow ab$ mit $S ::= ab$ gilt.
Induktionsschritt: n nach $n + 1$, $n \geq 1$

$$\begin{tikzcd}
n: & a^n b^n \in L \arrow[r, "\text{dann}"] & \begin{array}{c}\text{es gibt } n \text{ mit}\\ S \Rightarrow a^n b^n\end{array} \arrow[d] & \text{Annahme} \\
n+1: & a^{n+1} b^{n+1} \in L \arrow[r, "\text{dann}"] \arrow[u] & \begin{array}{c}\text{es gibt}\\ S \Rightarrow a^{n+1} b^{n+1}\end{array} & \text{Behauptung}
\end{tikzcd}$$

1. Es ist klar, wenn $a^{n+1} b^{n+1} \in L$ und $n \geq 1$, dann gilt auch $a^n b^n \in L$.
2. Es greift somit unmittelbar die Annahme und es gibt eine Ableitung $S \Rightarrow a^n b^n$.

3. Diese kann in $S \Rightarrow a^{n-1}Sb^{n-1} \rightarrow a^n b^n$ zerlegt werden.
4. Mit $S ::= aSb$ kann eine weitere direkte Ableitung „eingeschoben“ werden. Die erhaltene Ableitung $S \Rightarrow a^{n-1}Sb^{n-1} \rightarrow a^{n-1}aSbb^{n-1} \rightarrow a^{n-1}aabbb^{n-1} = a^{n+1}b^{n+1}$ zeigt die Behauptung.

Wenn in G alle Satzformen $a^n S b^n$ mit $n \geq 0$ erzeugt werden können und diese mit der Produktion $S ::= ab$ terminalisiert werden, so erhält man die Menge $\{a^{n+1}b^{n+1} : n \geq 0\}$. Diese entspricht L.

Aufgabe 3 Pump-Lemma (Bierwald)

Beweis durch Widerspruch.

Annahme. L kfG $\}$ Ex. Grammatik G mit $L(G)=L$

Wir wenden das Pump-Lemma an und betrachten das Wort $z = a^{k^2}$. Dabei sei $k > n$ und n die Zahl aus dem Pump-Lemma.

Offenbar $z \in L(G)$ und $|z| \geq n$.

Gemäß Pump-Lemma gibt es eine Zerlegung

$z = uvwxy$

und es gilt $\begin{cases} 1. & |vwx| \leq n \\ 2. & vx \neq \varepsilon \\ 3. & uv^iwx^iy \in L(G) \text{ für } i \geq 0 \end{cases}$

Wir betrachten das Wort

uv^2wx^2y

und versuchen, seine Länge abzuschätzen:

$$\begin{aligned} k^2 &< |uv^2wx^2y| && \text{(wegen 2.)} \\ &= |uvwxy| + |vx| \\ &= k^2 + |vx| \\ &\leq k^2 + |vwx| \\ &\leq k^2 + n && \text{(wegen 1.)} \\ &< k^2 + k \\ &= k(k+1) \\ &< (k+1)^2 \end{aligned}$$

Die Länge von uv^2wx^2y liegt echt zwischen zwei direkt aufeinander folgenden Quadratzahlen, kann also selbst keine Quadratzahl sein.

Dann aber $uv^2wx^2y \notin L$ ↯ zu 3.

Aufgabe 4 Eine „berühmte" Mehrdeutigkeit (Käufl)

Gegeben sei die Grammatik mit den Produktionen

S ::= B | z

B ::= if e then S | if e then S else S

Startzeichen ist S, T = {if, then, else, e, z} und N = {B, S}. Zeigen Sie mit Hilfe des Terminalworts if e then if e then z else z,

daß die Grammatik mehrdeutig ist.

Wir geben zwei Linksableitungen des Wortes an:

$S \to B \to$ if e then $S \to$ if e then if e then S else $S \Rightarrow$ if e then if e then z else z

$S \to B \to$ if e then S else $S \to$ if e then if e then S else $S \Rightarrow$ if e then if e then z else z

Aufgabe 5 Programmiersprachen sind vom Typ 1 (Käufl)

a) Geben Sie eine Chomsky-1-Grammatik an, deren Sprachschatz $\{ww\colon w \in T^*\}$ mit $T = \{a, b\}$ ist.
Anleitung. Führen Sie mit Hilfe der Produktionen ein Nichtterminalzeichen ein, das die Trennfuge zwischen den beiden Vorkommnissen von w kennzeichnet. Eine zweite Gruppe von Produktionen muß die Aufgabe haben, ein Vorkommnis des Wortes w zu erzeugen. Dabei müssen Sie sich merken, daß zu einem a (b) auch ein a (b) im zweiten Vorkommnis von w gehört. Dazu führen Sie geeignete Nichtterminalzeichen ein, für die Sie Produktionen vorsehen, die diese Zeichen an die Trennfuge zu verschieben erlauben und an der Trennfuge in Terminalzeichen verwandelt werden.

b) Seien T und U Mengen von Terminalzeichen, $U \subseteq T$ und $\varphi\colon U \to T^*$ eine Abbildung. Zeigen Sie: Ist $L \subseteq T^*$ eine Sprache vom Typ 2, dann auch $\varphi(L)$.

c) Zeigen Sie, daß die Menge der Programme `begin integer w; w := 1 end`, wobei `w` ein Wort über dem Alphabet {`a`, `b`} ist, nicht vom Typ 2 sein kann.

a) $Z ::= SMa \mid SNb \mid aa \mid bb \mid \varepsilon$

$S ::= SaA \mid SbB \mid aA \mid bB$

$Aa ::= aA$; $Ba ::= aB$; $Ab ::= bA$; $Bb ::= bB$

$AM ::= Ma$; $BM ::= Mb$; $AN ::= Na$; $BN ::= Nb$

$M ::= a$; $N ::= b$

Konstruktionsidee. M oder N markiert die „Trennfuge" zwischen den beiden Wörtern w. Zusätzlich geben M und N an, mit welchem Zeichen (das erste Vorkommnis von) w endet. Mit den Produktionen der zweiten Zeile wird das erste Vorkommnis des Wortes w (links von der Trennfuge) aufgebaut. Dabei erzeugt man zu jedem a (b) ein A (B). A und B können nach rechts verschoben werden, wobei kein A (B) ein B (A) „überholen" darf. (Produktionen der dritten Zeile.) Wird ein A oder B über die Trennfuge geschoben, dann wird rechts von der Trennfuge in das Terminalzeichen a bzw. b „verwandelt". (Vierte Zeile.) Schließlich erzeugt man aus M bzw. N ein Terminalzeichen.

b) Sei $G = (V, \Pi, S)$ die Grammatik, die L erzeugt. Wir können, Beweisskizze anschließend, annehmen, daß die Produktionen in der Form $B ::= r$ mit $r \in V^+$ oder $A_i ::= a_i$ mit $a_i \in T \cup \{\varepsilon\}$ sind. Die gesuchte kontextfreie Grammatik G' hat die Produktionen $B ::= r$ oder $A_i ::= \varphi(a_i)$.[1]

1. Grammatiken dieser Art heißen *separiert*.

Für den Beweis ist es zweckmäßig, φ auf Nichtterminalzeichen zu erweitern: $\varphi(Z) = Z$.

$L(G') \subseteq \varphi(L(G))$: Gelte $w_j' \to w_{j+1}'$ mittels $A_i ::= \varphi(a_i)$ und $\varphi(w_j) = w_j'$. Dann $w_j \to w_{j+1}$ mittels $A_i ::= a_i$ und $\varphi(w_{j+1}) = w_{j+1}'$. Beruht dieser Ableitungsschritt auf einer Produktion der Form $B ::= r$, dann gibt es diese Produktion auch in G und $\varphi(w_{j+1}) = w_{j+1}'$ gilt ebenso. Durch Induktion folgt, daß es zu jeder Ableitung $S \Rightarrow w'$ in G' eine Ableitung $S \Rightarrow w$ in G mit $\varphi(w) = w'$ gibt.

$\varphi(L(G)) \subseteq L(G')$: Gelte $w_j \to w_{j+1}$ in G mittels $A_i ::= a_i$ und sei $\varphi(w_j) = w_j'$. Dann auch $w_j' \to w_{j+1}'$ in G' mit Hilfe von $A_i ::= \varphi(a_i)$ und es ist $\varphi(w_{j+1}) = w_{j+1}'$. Dies gilt auch bei Anwendung einer Produktion der Form $B ::= r$, da in diesem Fall $\varphi(r) = r$.

Durch Induktion folgt $S \Rightarrow w$ in G, dann $S \Rightarrow \varphi(w)$ in G'.

Ist $u \in \varphi(L(G))$, dann gibt es ein $w \in L(G)$ mit $\varphi(w) = u$ und damit eine Ableitung $S \Rightarrow w$ in G und eine Ableitung $S \Rightarrow \varphi(w)$ in G.

Zum Nachweis, daß es zu jeder kontextfreien Grammatik G eine separierte G' gibt, nehmen wir $T = \{a_1, \ldots, a_n\}$ an. G' hat $N' = N \cup \{A_1, \ldots, A_n\}$, wobei die A_i neu sind, als Nichtterminalzeichen. Die Produktionenmenge $\Pi' = \Pi_1 \cup \{A_1 ::= a_1, \ldots, A_n ::= a_n\}$, wobei $B ::= r' \in \Pi_1$ genau dann, wenn $B ::= r \in \Pi$ und r' aus r durch Ersetzung von a_i durch A_i entsteht.

Im folgenden sei w' das Wort, das aus w durch Ersetzung der a_i durch A_i entsteht.

$L(G) \subseteq L(G')$: Ist $w_j \to w_{j+1}$ in G mittels $B ::= r$, dann auch $w_j' \to w_{j+1}'$ in G' mit Hilfe von $B ::= r'$. Ist $S \Rightarrow w$ in G, dann $S \Rightarrow w'$ in G' und die einzigen Nichtterminalzeichen in w' sind die A_i. Es gilt dann $w' \Rightarrow w$ mit Hilfe der Produktionen $A_i ::= a_i$.

$L(G') \subseteq L(G)$: Ist $w_j' \to w_{j+1}'$ in G' mit Hilfe einer Produktion $A_i ::= a_i$, dann $w_j \Rightarrow w_{j+1}$ (unechter Ableitungsschritt). Ist $w_j' \to w_{j+1}'$ in G' mit Hilfe einer Produktion $B ::= r'$, dann $w_j \to w_{j+1}$ mit Hilfe von $B ::= r$. Ist $w' \in L(G')$, dann gibt es eine Ableitung $S \Rightarrow w'$ und damit auch eine Ableitung $S \Rightarrow w$ in G und da $w' = w$ folgt $w \in L(G)$.

c) Den Nachweis, daß die Sprache P (Menge) dieser (einfachen) Programme nicht vom Typ 2 ist führen wir durch Vereinfachung dieser Sprache. φ sei eine Abbildung mit $\varphi(\text{begin}) = \varphi(;) = \varphi(:=) = \varphi(1) = \varphi(\text{end}) = \varepsilon$ und $\varphi(x) = x$ sonst. Dann ist $\varphi(P) = ww$. Ist P vom Typ 2, dann auch die Sprache mit den Wörtern ww, siehe Teil a. Wenn wir gezeigt haben, daß die Menge der Wörter ww nicht kontextfrei ist, kann dies auch nicht für P gelten.

Zunächst zeigen wir mit Hilfe des Pump-Satzes, daß die Menge der Wörter ww mit $w = ba^jba^jb$, $j \geq 0$ nicht vom Typ 2 ist.

Wäre diese Sprache vom Typ 2, dann gäbe es ein $n \geq 0$, sodaß für alle Worte ww mit $|ww| \geq n$ eine Zerlegung $uvwxy = ba^jba^jbba^jba^jb$ so gibt, daß $z_i = uv^iwx^iy$ ebenfalls zu dieser Sprache gehört.

1. Weder in v noch in x kommt das Zeichen b vor. Wäre dies der Fall, hätte z_2 sieben Vorkommnisse von b.

v und x enthalten also nur das Zeichen a.

2. v und x können nicht beide im linken Teilwort ba^jba^jb vorkommen. Wäre dies der Fall, enthielte in z_0 das linke der Teilwörter w weniger Vorkommnisse von a als das rechte.

3. Ebenso können v und x können nicht beide im linken Teilwort ba^jba^jb vorkommen.

4. Somit muß $v = a^r$ im linken Teilwort w, $x= a^s$ im rechten Teilwort w vorkommen. Kommen v und x jeweils im ersten Teilwort a^j von w vor, dann gehört nach dem Pump-Lemma auch $ba^jba^jbba^jba^jb$ zur Sprache, woraus folgt, daß auch $ba^{j+r}ba^jbba^{j+s}ba^jb$ zur Sprache gehört, ein Widerspruch, da $r > 0$ oder $s > 0$. In der gleichen Weise behandelt man die übrigen Fälle.

Ist ww eine Sprache vom Typ 2, dann gibt es eine kontextfreie Grammatik, die diese Sprache erzeugt und damit auch alle Worte der Form $ba^jba^jbba^jba^jb$. Dies führt, wie oben gezeigt auf einen Widerspruch. Damit ist ww nicht vom Typ 2.