

# Compilerbau

## Übungsblatt 4

Yangzi Zhang, Michael Gottschalk, Felix J. Oppermann

6. Juni 2006

## Aufgabe 11 (1+1 Punkte)

- a) Bei s-attribuierten Grammatiken handelt es sich um Attribuierte Grammatiken die nur synthetisierte Attribute enthalten. Zusätzlich zur Grammatik müssen Berechnungsvorschriften für die Attribute angegeben werden. Die Attribute einer s-attribuierten Grammatik lassen sich durch eine Bottom-up-Traversierung bestimmen. Eine s-attribuierte Grammatik kann direkt beim Parsen mit einem Bottom-up-Parser ausgewertet werden.
- b) Übersetzungsschemata sind eine spezielle Form Attribuiertes Grammatiken bei der die Berechnungsvorschriften direkt in die Grammatikregeln eingestreut werden. Die Vorschriften können dabei als Pseudo-Terminalsymbole betrachtet werden. Da eine Auswertung immer beim Erreichen der Berechnungsvorschrift erfolgt dürfen diese nur von zuvor erreichten Berechnungen abhängen. Diese Bedingung ist für synthetisierte Attribute immer erfüllt. Ererbte Attribute müssen in einer links von der aktuellen Berechnungsvorschrift stehenden Berechnung ermittelt worden sein. Übersetzungsschemata erlauben anders als s-attribuierte Grammatiken unter bestimmten Bedingungen auch ererbte Attribute. Sie sind folglich allgemeiner als s-attribuierte Grammatiken.

## Aufgabe 12 (2+1+3+3 Punkte)

$$\begin{array}{lll} \text{a) } S \rightarrow AaB & S.x := 2A.y + B.y & \text{synthetisiert} \\ & A.z := B.x + 1 & \text{ererbte} \\ & B.z := A.x + 1 & \text{ererbte} \end{array}$$

$$\begin{array}{lll} B \rightarrow AB_1 & B.x := 2A.x & \text{synthetisiert} \\ & B.y := 2A.y + B_1.y & \text{synthetisiert} \\ & A.z := B_1.x - 1 & \text{ererbte} \\ & B_1.z := 2B, z & \text{ererbte} \end{array}$$

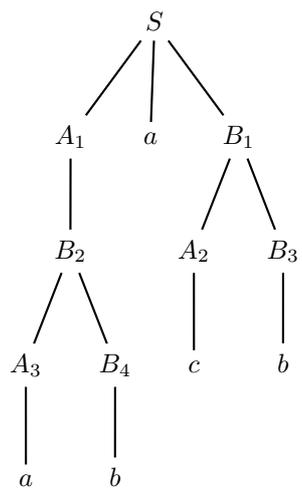
$$\begin{array}{lll} A \rightarrow B & A.x := B.x & \text{synthetisiert} \\ & A.y := B.y & \text{synthetisiert} \\ & B.z := A.z & \text{ererbte} \end{array}$$

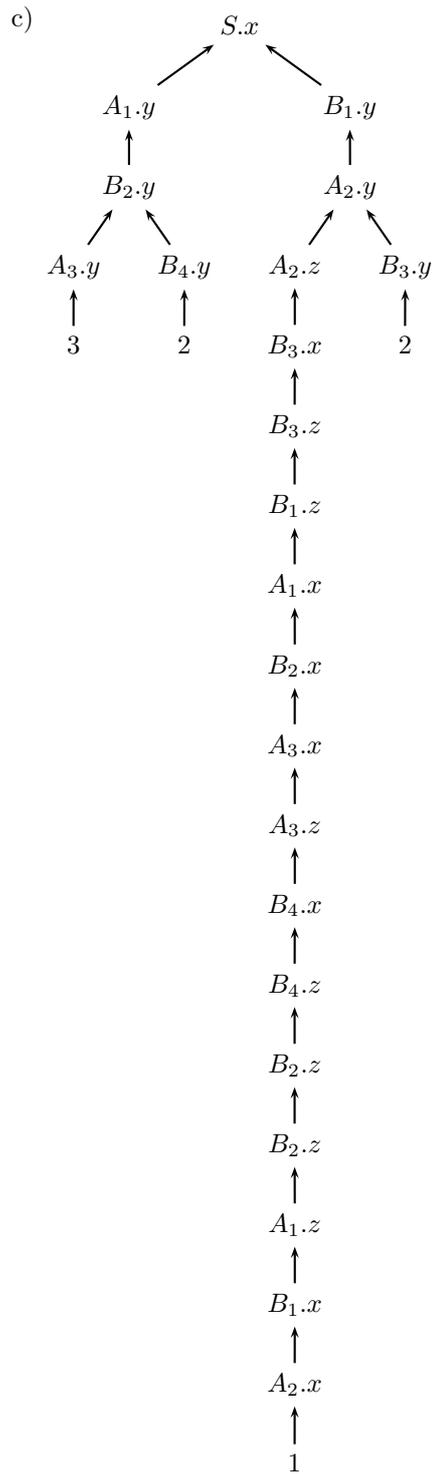
$$\begin{array}{lll} A \rightarrow a & A.x := A.z - 1 & \text{synthetisiert} \\ & A.y := 3 & \text{synthetisiert} \end{array}$$

$$\begin{array}{lll} A \rightarrow c & A.x := 1 & \text{synthetisiert} \\ & A.y := A.z & \text{synthetisiert} \end{array}$$

$$\begin{array}{lll} B \rightarrow b & B.x := B.z - 1 & \text{synthetisiert} \\ & B.y := 2 & \text{synthetisiert} \end{array}$$

b)





d)  $A_2.x = 1$   
 $B_1.x = 2A_1.x = 2 * 1 = 2$   
 $A_1.z = B_1.x + 1 = 2 + 1 = 3$

$B_2.z = A_1.z = 3$   
 $B_4.z = 2B_2.z = 2 * 3 = 6$   
 $B_4.x = B_4.z - 1 = 6 - 1 = 5$   
 $A_3.z = B_4.x - 1 = 5 - 1 = 4$   
 $A_3.x = A_3.z - 1 = 4 - 1 = 3$

$$\begin{aligned}
 B_2.x &= 2A_3.x = 2 * 3 = 6 \\
 A_1.x &= B_2.x = 6 \\
 B_1.z &= A_1.x + 1 = 6 + 1 = 7 \\
 B_3.z &= 2B_1.z = 2 * 7 = 14 \\
 B_3.x &= B_3.z - 1 = 14 - 1 = 13
 \end{aligned}$$

$$\begin{aligned}
 A_3.y &= 3 \\
 B_4.y &= 2 \\
 A_2.z &= B_3.x - 1 = 13 - 1 = 12
 \end{aligned}$$

$$\begin{aligned}
 B_2.y &= 2A_3.y + B_4.y = 2 * 3 + 2 = 8 \\
 A_2.y &= A_2.z = 12 \\
 B_3.y &= 2
 \end{aligned}$$

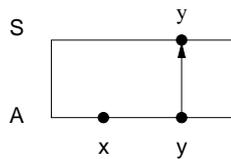
$$\begin{aligned}
 A_1.y &= B_1.y = 8 \\
 B_1.y &= 2A_2.y + B_3.y = 2 * 12 + 2 = 26
 \end{aligned}$$

$$S.x = 2A_1.y + B_3.y = 2 * 8 + 26 = 42$$

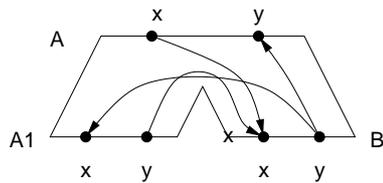
### Aufgabe 13 (2+3+3 Punkte)

a)

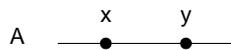
$S \rightarrow A$ :



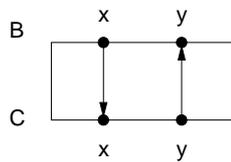
$A \rightarrow A_1B$ :



$A \rightarrow a$ :



$B \rightarrow C$ :



$C \rightarrow c$ :



b)

#### 1. Schritt

$$\text{is}[S] := \text{is}[A] := \text{is}[B] := \text{is}[C] := \emptyset$$

**2. Schritt**

1. Iteration:

$$p: A \rightarrow a : \text{is}[A] := \{ \begin{array}{c} x \quad y \\ \bullet \text{---} \bullet \end{array} \}$$

$$p: C \rightarrow c : \text{is}[C] := \{ \begin{array}{c} x \quad y \\ \bullet \text{---} \bullet \\ \quad \curvearrowright \end{array} \}$$

$$p: B \rightarrow C : \text{is}[B] := \text{is}(\text{D}(B \rightarrow C)[\begin{array}{c} x \quad y \\ \bullet \text{---} \bullet \\ \quad \curvearrowright \end{array}]) = \{ \begin{array}{c} x \quad y \\ \bullet \text{---} \bullet \\ \quad \curvearrowright \end{array} \}$$

$$p: A \rightarrow A_1 B : \text{is}[A] := \text{is}[A] \cup \text{is}(\text{D}(A \rightarrow A_1 B)[\begin{array}{c} x \quad y \\ \bullet \text{---} \bullet \\ \quad \curvearrowright \end{array}, \begin{array}{c} x \quad y \\ \bullet \text{---} \bullet \\ \quad \curvearrowright \end{array}]) = \{ \begin{array}{c} x \quad y \\ \bullet \text{---} \bullet \\ \quad \curvearrowright \end{array}, \begin{array}{c} x \quad y \\ \bullet \text{---} \bullet \\ \quad \curvearrowright \end{array} \}$$

$$p: S \rightarrow A : \text{is}[S] := \text{is}(\text{D}(S \rightarrow A)[\begin{array}{c} x \quad y \\ \bullet \text{---} \bullet \end{array}]) \cup \text{is}(\text{D}(S \rightarrow A)[\begin{array}{c} x \quad y \\ \bullet \text{---} \bullet \\ \quad \curvearrowright \end{array}]) = \{ \begin{array}{c} x \quad y \\ \bullet \text{---} \bullet \\ \quad \curvearrowright \end{array} \}$$

2. Iteration:

$$p: A \rightarrow A_1 B : \text{is}[A] := \text{is}[A] \cup \text{is}(\text{D}(A \rightarrow A_1 B)[\begin{array}{c} x \quad y \\ \bullet \text{---} \bullet \\ \quad \curvearrowright \end{array}, \begin{array}{c} x \quad y \\ \bullet \text{---} \bullet \\ \quad \curvearrowright \end{array}]) = \{ \begin{array}{c} x \quad y \\ \bullet \text{---} \bullet \\ \quad \curvearrowright \end{array}, \begin{array}{c} x \quad y \\ \bullet \text{---} \bullet \\ \quad \curvearrowright \end{array} \}$$

*Keine weitere Änderung, Algorithmus abgeschlossen*

$$\Rightarrow \text{is\_set}(X) := \text{is}(X) \text{ für } X \in \{A, B, C, S\}$$

**c)**

Nein, es gibt kein Wort in der von der Grammatik definierten Sprache, dessen Ableitungsbaum eine zirkuläre Abhängigkeit der Attribute zur Folge hat, denn die Grammatik ist nicht zirkulär, d.h. es gibt keinen Ableitungsbaum, der einen Zyklus enthält. Dass die Grammatik nicht zirkulär ist, erkennt man daran, dass es für keine Produktion p in der Grammatik einen in b) erzeugten Abhängigkeitsgraphen D(p) gibt, der einen Zyklus enthält.