

8. Übung zu „Grundlagen des Compilerbaus“, WS 2007/08

Abgabe der Aufgaben: Mi, 12.Dezember 2007 (vor der Vorlesung)

8.1 Attributgleichungssysteme

5 Punkte

Gegeben sei die kontextfreie attributierte Grammatik G mit der Attributmengem $Att = \{s, s_1, s_2, i_1, i_2\}$ und der folgenden Attributzuordnung:

$$syn(S) := \{s\}, \quad syn(A) := \{s_1, s_2\}, \quad inh(A) := \{i_1, i_2\}$$

Der Wertebereich in den Attributgleichungen sei stets \mathbb{N} und $succ(n) = n + 1$ die Nachfolgerfunktion.

$$\begin{array}{llll}
 G : S & \rightarrow AA & (\pi_1) & \pi_1 : s.0 = succ(s_2.2) & \pi_2 : s_1.0 = succ(i_2.0) \\
 A & \rightarrow a & (\pi_2) & i_1.1 = 1 & s_2.0 = succ(i_1.0) \\
 A & \rightarrow b & (\pi_3) & i_2.1 = succ(s_1.2) & \pi_3 : s_1.0 = succ(i_1.0) \\
 A & \rightarrow c & (\pi_4) & i_1.2 = succ(s_2.1) & s_2.0 = succ(i_2.0) \\
 & & & i_2.2 = succ(s_1.1) & \pi_4 : s_1.0 = succ(i_1.0) \\
 & & & & s_2.0 = succ(i_1.0)
 \end{array}$$

- (a) Bestimmen Sie zu den Ableitungsbäumen der Wörter ab , ac und ba die jeweiligen Attributgleichungssysteme und berechnen Sie den Wert des Attributs s daraus.
- (b) Erstellen Sie die Teil-Abhängigkeitsgraphen der einzelnen Produktionen. Konstruieren Sie zu einem konkreten Ableitungsbaum t einen Abhängigkeitsgraphen DG_t , der Zyklen enthält. Welche Attributwerte von t sind berechenbar, welche nicht?

8.2 L-Attributgrammatiken

Gegeben sei die folgende L-Attributgrammatik, wobei der Wertebereich aller Attribute die ganzen Zahlen seien:

$$\begin{array}{l}
 G : S \rightarrow AB \quad s.0 = s.2, i.2 = s.1, i.1 = 0 \\
 A \rightarrow aA \quad i.2 = i.0 + 1, s.0 = s.2 \\
 \quad \quad \quad | \varepsilon \quad s.0 = i.0 \\
 B \rightarrow bB \quad i.2 = i.0 - 1, s.0 = s.2 \\
 \quad \quad \quad | \varepsilon \quad s.0 = i.0
 \end{array}$$

- (a) Der in der Vorlesung definierte TDA für Attributgrammatiken berechnet das Attribut s des Startsymbols und ermittelt dabei alle benötigten Attribute im Ableitungsbaum. Berechnen Sie s für die Analyse des Wortes $aabbb$ und geben Sie alle dabei durchlaufenen Konfigurationen an.

Die Berechnung von Attributen kann auch mittels der von Martin Jourdan entwickelten rekursiven Berechnung [Jou84] durchgeführt werden:

Jedem synthetischen Attribut eines Nonterminals wird eine (rekursive) Funktion zugeordnet, deren Argumente die inheriten Attribute und die Nachfolgerknoten sind. Die Abhängigkeit von den synthetischen Attributen der Nachfolger im Baum ergibt die Rekursion.

```

----- Haskell Code -----
-- Syntaxbaum
data S = S A B
      deriving Show
data A = N A
      | E
      deriving Show
-- für B gleiche Darstellung
type B = A

-- Berechnung
s_inS :: S -> Int
s_inA :: A -> Int -> Int
s_inB :: B -> Int -> Int

```

- (b) Programmieren Sie für die gegebene Grammatik rekursive Funktionen, welche das synthetische Attribut s des jeweiligen Nonterminals berechnen.

[Jou84] Martin Jourdan. Strongly non-circular attribute grammars and their recursive evaluation. *ACM SIGPLAN Notices*, 19(6):81–93, June 1984.