

Übungen zu „Grundlagen des Compilerbau“, Winter 2011/12

Nr. 9, Abgabe der Aufgaben: 17. Januar 2012 vor der Vorlesung

Aufgaben

9.1 Attributgrammatik für Zuweisungen

7 Punkte

Die folgende Grammatik beschreibt eine Zuweisung an einen Bezeichner id von booleischem oder arithmetischem Typ. Dabei bezeichne $bval$ einen Wahrheitswert und $numval$ eine Zahl.

$$\begin{array}{ll}
 G_z : Z \rightarrow id := E & (1) \\
 E \rightarrow bval & (2) \\
 E \rightarrow numval & (3) \\
 E \rightarrow E + E & (4) \\
 E \rightarrow E \parallel E & (5) \\
 E \rightarrow E \&\& E & (6) \\
 E \rightarrow E = E & (7)
 \end{array}$$

- (a) Nur Zahlen dürfen mit $+$, nur Wahrheitswerte mit $\&\&$ und \parallel verknüpft werden. Der Vergleich $=$ ist nur für Argumente *gleichen* Typs definiert. Versehen Sie das Nonterminal E mit Attributen t mit $A^t = \{\text{Bool}, \text{Num}\}$ für den Typ des Ausdrucks und err mit $A^{err} = \{\text{true}, \text{false}\}$ als Typfehlerindikator.

- (b) Mit den nebenstehenden Produktionen erzeugt eine erweiterte Grammatik G'_z eine ganze Liste von Zuweisungen. Auch Bezeichner können nun auf der rechten Seite verwendet werden.

$$G'_z : Z \rightarrow id := E ; Z \quad (8)$$

$$E \rightarrow id \quad (9)$$

Erweitern Sie die Attributierung, so dass die gesamte Liste von Zuweisungen überprüft wird. Bezeichner dürfen nur auf der rechten Seite enthalten sein, falls ihnen bereits vorher ein Wert zugewiesen wurde. Die Verwendung muss typkonform erfolgen, wobei durch eine erneute Zuweisung der Typ nicht verändert werden darf.

Bsp1: $a := \text{True}; a' := 3; a := a'=3;$ mit $err.k_0 = \text{false}$

Bsp2: $a := \text{True}; a := a=3;$ mit $err.k_0 = \text{true}$

Bsp3: $a := 3; a := a=3;$ mit $err.k_0 = \text{true}$

(Dabei bezeichne k_0 den Wurzelknoten des Syntaxbaums)

Speichern Sie die Typinformationen in einer Hashtabelle mit den nebenstehenden Operationen. Verwenden Sie eine (leere) initiale Tabelle als inherites Attribut des Startsymbols Z (Parameter der Attributierung).

```

data Table = .. -- Hashtable
insert :: Table -> (key,value) -> Table
delete :: Table -> key -> Table
lookup :: Table -> key -> Maybe value

```

9.2 Zirkularitätstest

Geben Sie für die folgende Grammatik $syn(x)$ und $inh(x)$ für alle Symbole x an und untersuchen Sie mit Hilfe des in der Vorlesung vorgestellten Verfahrens, ob die Grammatik zirkulär ist:

$$\begin{array}{ll}
 G : S & \rightarrow AaB \quad s_{.0} = s_{2.3}, i_{1.1} = s_{2.3} \cdot s_{.2}, i_{2.1} = s_{1.3} \cdot s_{2.1}, i_{1.3} = 1, i_{2.3} = s_{1.1} \\
 A & \rightarrow Ba \quad s_{1.0} = 2 \cdot s_{2.1}, s_{2.0} = s_{1.1}, i_{1.1} = i_{1.0}, i_{2.1} = i_{2.0} + s_{.2} \\
 A & \rightarrow a \quad s_{1.0} = 1, s_{2.0} = s_{.1} \\
 B & \rightarrow b \quad s_{1.0} = i_{1.0}, s_{2.0} = s_{.1} \\
 B & \rightarrow c \quad s_{1.0} = s_{.1}, s_{2.0} = i_{2.0}
 \end{array}$$