

Übungen zu „Grundlagen des Compilerbau“, Winter 2009/10

Nr. 8, Abgabe der Aufgaben: 13. Januar 2010 vor der Vorlesung

Frohe Weihnachten und alles Gute in 2010

Aufgaben

8.1 Attributgrammatik für Zuweisungen

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

Bsp2: $a := \mathbf{True}; a := a=3;$ 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

```

Bitte wenden!

8.2 Typen von Attributgrammatiken

5 Punkte

Im folgenden sind zwei mal die Produktionen der Grammatik $G = (N, \Sigma, P, A)$ mit insgesamt fünf verschiedenen Attributierungen $E_1 - E_5$ über der Attributmenge $Att = \{w, w'\}$ aufgeführt.

P	E_1	E_2	E_3
$A \rightarrow SD$	$w.0 = w.1, w'.1 = w.2$	$w.0 = w'.1, w.1 = w.2$	$w.0 = w.1 * w.2$
$D \rightarrow Dz$	$w.0 = w.1 * 10 + getVal(z)$	$w.0 = w.1 * 10 + getVal(z)$	$w.0 = w.1 * 10 + getVal(z)$
z	$w.0 = getVal(z)$	$w.0 = getVal(z)$	$w.0 = getVal(z)$
$S \rightarrow +$	$w.0 = w'.0$	$w.0 = w'.0$	$w.0 = 1$
$-$	$w.0 = (-1) * w'.0$	$w.0 = (-1) * w'.0$	$w.0 = -1$

P	E_4	E_5
$A \rightarrow SD$	$w.0 = w.1 * w.2, w'.1 = w.2, w'.2 = w.1$	$w.0 = w.1 * w.2, w'.2 = 1$
$D \rightarrow Dz$	$w.0 = w'.0 * getVal(z), w'.1 = w'.0$	$w.0 = w.1 + w'.0 * getVal(z), w'.1 = w'.0 * 10$
z	$w.0 = w'.0 * getVal(z)$	$w.0 = w'.0 * getVal(z)$
$S \rightarrow +$	$w.0 = w'.0$	$w.0 = 1$
$-$	$w.0 = (-1) * w'.0$	$w.0 = -1$

Geben Sie für jedes $AG_i := (G, E_i), i \in \{1, \dots, 5\}$, an, ob es eine gültige Attributgrammatik beschreibt, ob es sich um eine L-Attributgrammatik, eine zyklfreie-Attributgrammatik oder eine S-Attributgrammatik handelt. Füllen Sie dazu die folgende Tabelle mit **Ja** und **Nein** Einträgen aus. Jeder richtige Eintrag zählt einen **Viertelpunkt**, jeder falsche führt zu ebensoviel **Abzug**.

	AG_1	AG_2	AG_3	AG_4	AG_5
gültige Attributgrammatik					
L-Attributgrammatik					
gültige, zyklfreie AG					
S-Attributgrammatik					

8.3 Weihnachtspunkt

1 Punkte