

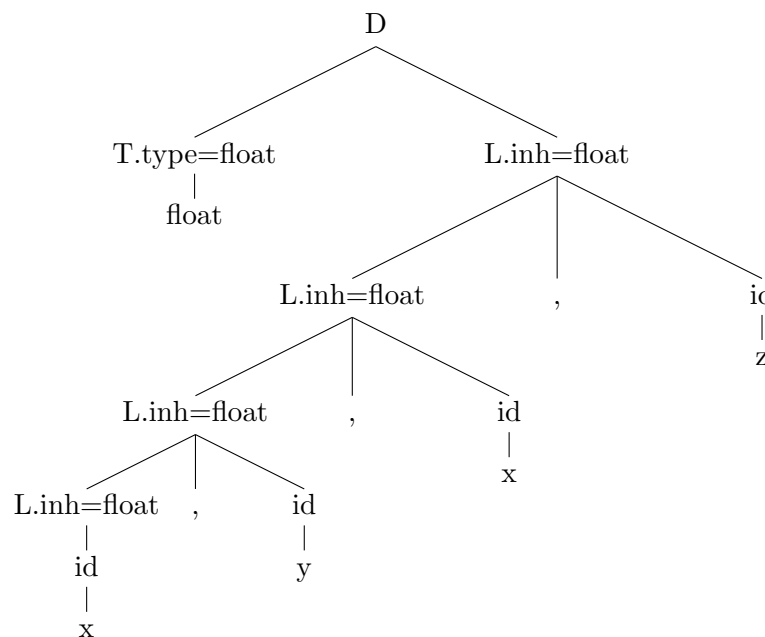
Übersetzerbau: Übung 11

von

Naja v. Schmude (4127652), Lisa Dohrmann (4130066)

Aufgabe 1

- a) Eingabe ist `float x, y, x, z`. Der dazugehörige annotierte Parsebaum sieht wie folgt aus:



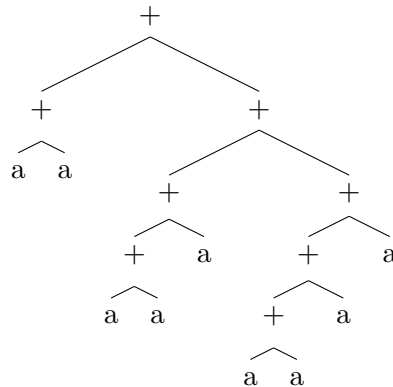
- b) Die SDD soll so erweitert werden, dass eine Warnung ausgegeben wird, falls ein Bezeichner mehr als einmal in einer Deklaration vorkommt. Das sollte am besten in der Funktion `addType` passieren. In ihr wird der Typ eines Bezeichners in die Tabelle eingetragen. Wird dabei festgestellt, dass bereits ein Typ vorhanden ist, sollte eine Meldung ausgegeben werden.

Aufgabe 2

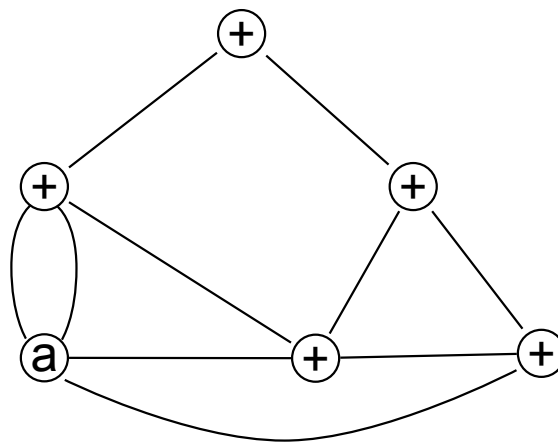
SDD für Ausdrücke über a mit der linksrekursiven Addition.

Produktion	semantische Regel
$E \rightarrow E_1 + T$	$E.n = mkNode('+', E_1.n + T.n)$
$E \rightarrow T$	$E.n = T.n$
$T \rightarrow (E)$	$T.n = E.n$
$T \rightarrow id$	$T.n = mkLeaf(id, a)$

Beispiel: Eingabe $a+a+(a+a+a+(a+a+a+a))$. Der Syntaxbaum dazu sähe so aus



Ein GAG, der Knoten wiederverwendet, sieht hingegen wie folgt aus:



Aufgabe 3

a) SDD für Berechnung der Werte von Binärzahlen:

Produktion	semantische Regel
$B \rightarrow B_1 0$	$B.val = B_1.val \cdot 2$
$B \rightarrow B_1 1$	$B.val = B_1.val \cdot 2 + 1$
$B \rightarrow 1$	$B.val = 1$

b) Übersetzungsschema. Wir fügen eine neues Startsymbol S hinzu, um das Ergebnis zum Schluss ausgeben zu können.

$$\begin{aligned}
 S &\rightarrow B \{print(B.val)\} \\
 B &\rightarrow B_1 0 \{B.val = B_1.val \cdot 2\} \\
 B &\rightarrow B_1 1 \{B.val = B_1.val \cdot 2 + 1\} \\
 B &\rightarrow 1 \{B.val = 1\}
 \end{aligned}$$

c) Entfernen der Linksrekursion. Die eingefügten Codefragmente werden dabei wie Terminalsymbole behandelt. Sie müssen aber noch angepasst werden, damit der errechnete Wert am Ende zur Wurzel durchgereicht werden kann. Dazu führen wir

das synthetisierten Attribute *syn* und das ererbte Attribut *inh* ein.

$$\begin{aligned}
 S &\rightarrow B \{print(B.syn)\} \\
 B &\rightarrow 1 \{B'.inh = 1\} B' \{B.syn = B'.syn\} \\
 B' &\rightarrow 0 \{B'_1.inh = B'.inh \cdot 2\} B'_1 \{B'.syn = B'_1.syn\} \\
 B' &\rightarrow 1 \{B'_1.inh = B'.inh \cdot 2 + 1\} B'_1 \{B'.syn = B'_1.syn\} \\
 B' &\rightarrow \epsilon \{B'.syn = B'.inh\}
 \end{aligned}$$

Beispiel: Eingabe 101, Ausgabe 5

