

---

**Compilerbau**

<http://proglang.informatik.uni-freiburg.de/teaching/compilerbau/2004/>

---

**Übungsblatt 7**

Abgabe: 15.12.2004

**Aufgabe 1 (Typüberprüfung für Listen):**

Die folgende Grammatik erzeugt Programme,  $P$ , die aus einer Folge von Deklarationen,  $D$ , und einem Ausdruck,  $E$ , bestehen, wobei Ausdrücke Listen über Literallisten repräsentieren und Deklarationen Variablen bestimmte Listentypen zuweisen.

$$\begin{aligned}
 P &\rightarrow D ; E \\
 D &\rightarrow D ; D \mid \text{id} : T \\
 T &\rightarrow \text{list}(T) \mid \text{char} \mid \text{integer} \\
 E &\rightarrow (L) \mid \text{literal} \mid \text{num} \mid \text{id} \\
 L &\rightarrow E , L \mid E
 \end{aligned}$$

- (i) Erweitere die Grammatik zu einer Attributgrammatik, so dass die Attributierung gerade die Typen aller Ausdrücke ( $E$ ) und Listen ( $L$ ) bestimmt.
- (ii) Die Produktion

$$E \rightarrow \text{nil}$$

erweitert die Grammatik so, dass nun auch leere Listen durch Ausdrücke darstellen werden können. Passe Deine Attributgrammatik aus (i) an diese Erweiterung an, wobei leere Listen für Listen eines beliebigen Typs stehen können.

**Aufgabe 2 (Semantische Analyse einfacher sequentieller Programme):**

Die folgende Grammatik erzeugt Anweisungen,  $S$ , einer einfachen sequentiellen Programmiersprache.

$$\begin{aligned}
 S &\rightarrow \text{var id} \\
 S &\rightarrow \text{if } E \text{ then } S \text{ else } S \\
 S &\rightarrow \text{while } E \text{ do } S \\
 S &\rightarrow \text{id} := E \\
 S &\rightarrow \text{return } E \\
 S &\rightarrow S ; S \\
 E &\rightarrow \text{const} \\
 E &\rightarrow \text{id} \\
 E &\rightarrow E + E
 \end{aligned}$$

Formuliere eine semantische Analyse mit Hilfe einer Attributgrammatik, die überprüft, dass

- (i) Variablen erst dann referenziert werden (durch  $\text{id}$  bzw.  $\text{id} := E$ ), nachdem sie deklariert wurden (mit Hilfe einer Variablendeklaration  $\text{var id}$ ), und

(ii) die zuletzt ausgeführte Anweisung immer eine `return`-Anweisung ist.

**Aufgabe 3 (Auflösung von Überladung):**

Der Algorithmus zur Auflösung von Überladung aus der Vorlesung läuft in zwei Phasen ab. Ist es möglich die Überladung in nur einem Durchgang aufzulösen? Falls ja, gib den passenden Algorithmus dazu an.