Lehrstuhl für Softwaretechnik und Programmiersprachen
Professor Dr. Michael Leuschel

Compilerbau - Wintersemester 2021/22

Praktisches Übungsblatt 6

Besprechung der Aufgaben am 03.12.21 um 16:30 Uhr (evtl. früher) in 25.12.02.55 und gleichzeitig online per BBB
Fragen an Lukas.Lang@hhu.de
Die Bearbeitung ist freiwillig.
Bringen Sie Ihre Lösung mit zum Übungstermin

Aufgabe 6.1

Schreiben Sie einen Typechecker für eine kleine Sprache von booleschen und arithmetischen Ausdrücken:

1. Die Programme bestehen aus einer Abfolge von Zuweisungen.
2. Links einer Zuweisung steht ein Bezeichner. Dieser besteht aus einer Abfolge von Klein- und Grossbuchstaben. Rechts davon steht ein Ausdruck. Der Zuweisungsoperator ist ':='.
3. Ein Ausdruck kann aus Bezeichern, den Konstanten 'true' und 'false', Zahlen, den Verknüpfungen '+', '-', '*', '/' ,'and', 'or' und 'not' sowie runden Klammern bestehen.
4. Es gibt die zwei Typen integer und boolean.

Ein Beispielprogramm:

```
x := 17+1*3                    1
y := x+1                       2
z := false and true            3
a := y*3+x                     4
```

Erzeugt folgende Ausgabe:

```
a = integer
z = boolean
y = integer
x = integer
```

Ein falsches Programm erzeugt eine Fehlermeldung

a. Erzeugen Sie mit Hilfe von SableCC einen Parser und einen AST.

b. Schreiben Sie einen Visitor der den AST durchläuft und die Typen aller Ausdrücke und Variablen inferiert und ausgibt ob das Programm Fehler enthält.

Aufgabe 6.2

Die Sprache aus Aufgabe 6.1 ermöglicht nun mehrere Funktionen und Funktionsaufrufe.

1. Die Sichtbarkeit einer Parametervariable ist auf ihre Funktion beschränkt.
2. Variablen die in einer Funktion nicht deklariert wurden, müssen global deklariert sein.
3. Funktionsaufrufe liefern einen Wert vom Typ integer oder boolean zurück.
4. Das Schlüesselwort return gibt den Wert einer Funktion zurüeck.
5. Auf Integerwerten sind zusätzlich die Operatoren $=$, $<$ und $>$ definiert.

Ein Beispielprogramm:

```
1   integer x;
2   boolean z;
3   x := f(4);
4   z := true;
5
6   function boolean neg(boolean b) {
7       return not b;
8   }
9
10  function integer f(integer x) {
11      z := neg(x+5<4);
12      return x;
13  }
```

Ein falsches Programm erzeugt eine Fehlermeldung.

a. Erzeugen Sie mit Hilfe von SableCC einen Parser und einen AST.

b. Schreiben Sie einen Visitor der den AST durchläuft und die Typen aller Ausdrücke und Variablen inferiert und ausgibt ob das Programm Fehler enthält. Verwendeen Sie eine Symboltabelle.

Hinweis zu b): Es ist auch erlaubt mehr als einen Visitor zu verwenden.

Zu diesem Zettel wird es keine Musterlösung geben!