Lehrstuhl für Softwaretechnik und Programmiersprachen
Professor Dr. Michael Leuschel

Compilerbau - Wintersemester 2021/22

Theoretisches Übungsblatt 8

Besprechung der Aufgaben am 17.12.21 um 14:30 Uhr in 25.12.02.55 und gleichzeitig online per BBB
Fragen an Lukas.Lang@hhu.de
Die Bearbeitung ist freiwillig.

Aufgabe 8.1

Gegeben sei die Grammatik G mit dem Startsymbol E, dem Nichtterminal E und den Terminalen $\{*, +, nat, id, (,)\}$:

$$E \rightarrow E + E | E * E | id | nat$$

Gegeben sei die Grammatik G' mit dem Startsymbol E, den Nichtterminalen $\{E, T, E', T', F\}$ und den Terminalen $\{*, +, nat, id, (,)\}$:

$$E \rightarrow TE'$$

$$E' \rightarrow +E | \epsilon$$

$$T \rightarrow FT'$$

$$T' \rightarrow *T | \epsilon$$

$$F \rightarrow nat | id | (E)$$

a. Erweitern Sie die Grammatik G' um semantische Aktionen um einen AST zu erzeugen (welcher G entspricht).

b. Geben Sie den Parsebaum für die Eingabe 3*5+4 für G sowie G' an.

c. Zeichnen Sie den Attributabhängigkeitsgraph in den Parsebaum. Gibt es Zyklen? Ist dies eine S-Attributgrammatik?

Tipp: Verwenden Sie auch vererbte Attribute.

Aufgabe 8.2

Gegeben sei die Grammatik G mit dem Startsymbol E, den Nichtterminal E und den Terminalen $\{+, -, *, /, <, >, ==, true, false, num\}$:

$$E \rightarrow E + E | E - E | E * E | E / E | E < E | E > E | E == E | true | false | num$$

a. Geben Sie eine konkrete Grammatik G' mit korrekter Operatorpräzedenz an, mit welcher sich alle Wörter in G erzeugen lassen.

b. Erweitern Sie die Grammatik G' um semantische Aktionen um einen AST zu erzeugen (welcher G entspricht)

c. Erweitern Sie die Grammatik G um semantische Aktionen für einen AST Interpreter.

d. Erweitern Sie die Grammatik G um semantische Aktionen für einen Typechecker.

Aufgabe 8.3

Gegeben sei eine kleine Programmiersprache:

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

Ein Beispielprogramm:

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

Ein falsches Programm erzeugt eine Fehlermeldung.

a. Geben Sie die Symboltabellen für das Programm (also global), die Funktion f und Funktion neg an.

b. Wie funktioniert der lookup in Zeile 6 ($z := neg(x + 5 < 4)$;) ? D.h. welche Tabellen werden mit welchen Schlüsseln in welcher Reihenfolge abgefragt ?