Lehrstuhl für Softwaretechnik und Programmiersprachen
Professor Dr. Michael Leuschel

Compilerbau - Wintersemester 2021/22

Praktisches Übungsblatt 8

Besprechung der Aufgaben am 17.12.21 um 16:30 Uhr in 25.12.02.55 und gleichzeitig online per BBB
(evtl. auch früher)
Fragen an Lukas.Lang@hhu.de
Die Bearbeitung ist freiwillig.
Bringen Sie Ihre Lösung mit zum Übungstermin

Aufgabe 8.1

Gegen sei eine kleine Teilmenge von Pascal:

1. Die ersten Zeile aller Programme ist immer gleich: *program Foo; var x,y,z: boolean;*
2. Die Programme bestehen aus einer Abfolge von Anweisungen innerhalb eines Begin-End Blockes. Jede Anweisung endet mit einem Semikolon.
3. Bei Anweisungen handelt es sich entweder um Zuweisungen oder die writln Funktion.
4. Links einer Zuweisung steht ein Bezeichner. Dieser besteht aus einer Abfolge von Klein- und Grossbuchstaben.
5. Rechts eines Zuweisungsoperators (:=) sowie rechts des 'writeln'-Funktion steht ein Ausdruck.
6. Ein Ausdruck kann aus Bezeichern, den Konstanten 'true' und 'false' sowie den Verknüpfungen 'and', 'or' und 'not' sowie runden Klammern bestehen.

Ein Beispielprogramm:

```
program Foo; var x,y,z: boolean;
begin
  x := true and false;
  writeln(x);
  y := true and (false or false);
  z := true or false;
  writeln(x or(y or z));
end.
```

Erzeugt die Ausgabe:

```
FALSE
TRUE
```

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

b. Schreiben Sie einen AST-Interpreter. Wie kann man hier das Visitorpattern anwenden?

Aufgabe 8.2
Gegeben sei die folgende kontextfreie Grammatik mit dem Startsymbol R_0, den Nicht-Terminal $\{R_0, R_1, R_2\}$ und den Terminalen $\{|, *, a, b, c, (,)\}$:

$R_0 \rightarrow R_0\ '|'\ R_1 | R_1$

$R_1 \rightarrow R_1 R_2 | R_2$

$R_2 \rightarrow R_2 *$

$R_2 \rightarrow (R_0)$

$R_2 \rightarrow a$

$R_2 \rightarrow b$

$R_2 \rightarrow c$

Betrachten Sie, dass der erste senkrechte Strich das „oder"-Symbol ist und kein Seperator zwischen Alternativen.

(a) Schreiben Sie mit Hilfe von SableCC ein Java Programm welches hierzu einen AST generiert.

(b) Schreiben Sie einen AST-visitor welcher für reguläre Ausdrücke in dieser Grammatik nichtdeterministische endliche Automaten berechnet.

Hinweis:

1. Erben Sie hierbei von der Klasse DepthFirstAdapter und überschreiben Sie die relevanten Methoden.
2. Die Verarbeitung eines Teilbaums sollte min. die Attribute start und end vom Typ Node setzen. Diese geben den Start- und Endknoten des Automaten an.
3. Eine mögliche Konstruktionsanleitung finden Sie auf den Folienen zu Kapitel 3 ab Seite 50.

Zu diesem Zettel wird es keine Musterlösung geben!