

Compilerbau - Wintersemester 2021/22

Theoretisches Übungsblatt 7

Besprechung der Aufgaben am 10.12.21 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 7.1

Gegeben sei die folgende kontextfreie Grammatik G , mit dem Startsymbol R , den Nicht-terminal R und den Terminalen $\{ |, *, a, b, (,) \}$:

$$R \rightarrow R'|'R$$

$$R \rightarrow RR$$

$$R \rightarrow R*$$

$$R \rightarrow (R)$$

$$R \rightarrow a$$

$$R \rightarrow b$$

$$R \rightarrow c$$

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

- (a) Zeigen Sie, dass diese Grammatik mehrdeutig ist.
- (b) Konstruieren Sie eine äquivalente eindeutige Grammatik nach folgenden Operatorprioritäten (von hoch nach niedrig): $*$, Konkatination, Alternative $|$.
Wie ist die Operatorassoziativität Ihrer Lösung?
- (c) Geben Sie für beide Grammatiken den Syntaxbaum für den String $a|b*c$ an

Aufgabe 7.2

Gegeben sei folgende Attributgrammatik:

Produktion	Semantische Regel
$D \rightarrow TL$	$L.inh = T.type$
$T \rightarrow int$	$T.type = intger$
$T \rightarrow float$	$T.type = float$
$L \rightarrow L_1, id$	$L_1.inh = L.inh; L.entry = addType(id.entry, L.inh)$
$L \rightarrow id$	$L.entry = addType(id.entry, L.inh)$

Das Startsymbol ist D . Nichtterminale sind $\{D, T, L\}$. Hierbei fügt die Funktion `addType` einem Identifier-Knoten ein Typattribut hinzu.

- (a) Zeichnen Sie den Parsebaum für die Tokenfolge *float id₁, id₂, id₃*.
- (b) Zeichnen Sie den Abhängigkeitsgraph und führen Sie eine topologische Sortierung durch.
- (c) Berechnen Sie mit den semantischen Regeln alle Attribute für jeden Knoten

Aufgabe 7.3

Gegen sei eine kleine Teilmenge von Pascal:

1. Die erste 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 `writeln` 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 Bezeichnern, 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) Geben Sie eine Grammatik an.
- (b) Erweitern Sie diese um semantische Aktionen für einen Interpreter (Pseudocode).
Tipp: Speichern Sie den Wert einer Variablen nach einer Zuweisung ab.

Aufgabe 7.4

Gegeben sei die kontextfreie Grammatik G mit dem Startsymbol R_0 , den Nichtterminalen $\{R_0R_1R_2\}$ und den Terminalen $\{[, *, a, b, c, (,)\}$:

$$R_0 \rightarrow R_0 \text{ ' ' } R_1 | R_1$$

$$R_1 \rightarrow R_1 R_2 | R_2$$

$$R_2 \rightarrow R_2 *$$

$$R_2 \rightarrow (R_0)$$

$$R_2 \rightarrow a$$

$$R_2 \rightarrow b$$

$$R_2 \rightarrow c$$

Sowie eine zweite Grammatik G' mit dem Startsymbol R , den Nichtterminal R und den Terminalen $\{[, *, a, b, c, (,)\}$:

$$R \rightarrow R \text{ ' ' } R$$

$$R \rightarrow RR$$

$$R \rightarrow R *$$

$$R \rightarrow (R)$$

$$R \rightarrow a$$

$$R \rightarrow b$$

$$R \rightarrow c$$

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

- (a) Erweitern Sie Grammatik G um semantische Aktionen (Pseudocode) um einen AST zu erzeugen, welcher der abstrakten Grammatik G' entspricht.
- (b) Parsen Sie die Eingabe $(a|b)*abb$ und geben Sie den konkreten wie den abstrakten Syntaxbaum an. Geben Sie die Bäume direkt an.