

1. Übung zur Vorlesung „Prinzipien von Programmiersprachen“ Wintersemester 2008/2009

Abgabe: 11. November 2008 in der Vorlesung

Aufgabe 1

(Präsenzaufgabe)

Gegeben sei die folgende Grammatik in Backus-Naur-Form, die für das Nichtterminalsymbol $\langle \text{Exp} \rangle$ eine Menge von Ausdrücken definiert.

```
 $\langle \text{Exp} \rangle ::= \langle \text{Number} \rangle \mid \langle \text{Id} \rangle \mid \langle \text{Exp} \rangle \langle \text{Op} \rangle \langle \text{Exp} \rangle$   
           $\mid \langle \text{Exp} \rangle \langle \text{Relop} \rangle \langle \text{Exp} \rangle \mid (\langle \text{Exp} \rangle)$   
 $\langle \text{Op} \rangle ::= + \mid - \mid * \mid /$   
 $\langle \text{Relop} \rangle ::= = \mid < \mid <= \mid > \mid >=$   
 $\langle \text{Id} \rangle ::= \langle \text{Letter} \rangle \mid \langle \text{Id} \rangle \langle \text{Letter} \rangle \mid \langle \text{Id} \rangle \langle \text{Digit} \rangle$   
 $\langle \text{Number} \rangle ::= \langle \text{Digit} \rangle \mid \langle \text{Number} \rangle \langle \text{Digit} \rangle$   
 $\langle \text{Letter} \rangle ::= a \mid b \mid c \mid \dots \mid z \mid A \mid B \mid C \mid \dots \mid Z$   
 $\langle \text{Digit} \rangle ::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$ 
```

Ein Ableitungsbaum für ein Wort x ist ein geordneter Baum. Die inneren Knoten sind mit Nichtterminalsymbolen markiert. Ist dabei n ein beliebiger innerer Knoten, markiert mit dem Nichtterminal A , und sind seine Kinder von links nach rechts markiert mit N_1, \dots, N_k , so ist $A ::= \dots \mid N_1 \dots N_k \mid \dots$ eine Regel der Grammatik. Das Blattwort des Ableitungsbaums ergibt sich durch Konkatenation der Markierungen der Blätter von links nach rechts. Geben Sie einen Ableitungsbaum für folgende Ausdrücke an:

- a) $(a * 34) <= a * a - 67$
- b) $(e - b * 36) > 45$

Die Multiplikation und die Division sollen eine höhere Priorität als die Addition und die Subtraktion besitzen.

Aufgabe 2

(Präsenzaufgabe)

Geben Sie eine terminierende Berechnung des Zustands

```
 $\langle \text{if } x < 0 \text{ then } x := 0 - x \text{ else skip, } \{x \mapsto -42\} \rangle$ 
```

in der in der Vorlesung eingeführten *strukturierten operationalen Semantik* an. Zeigen Sie anhand des angegebenen Inferenzsystems, dass die Berechnung korrekt ist.

Aufgabe 3

Der folgende Programmcode zeigt die Definition einer Funktion `f` in Java.

```
public static double f( int y, double x ) {  
    y = 4;  
    x = 2 * y;  
    return x / 3;  
}
```

Die Schlüsselwörter `public` und `static` können in dieser Reihenfolge zu Beginn der Funktionsdefinition auftreten. Das Resultat des Beispielprogramms ist ein Wert vom Typ `double`. Die Funktion `f` besitzt zwei Argumente und im Rumpf können finden mehrere Zuweisungen statt. Das Funktionsresultat wird mit `return` zurückgegeben.

- a) Geben Sie eine EBNF an, die Funktionsdefinitionen der obigen Struktur beschreibt, wobei natürlich allgemeinere Formen (z.B. `public` oder `static` weglassen, weniger oder mehr als zwei Argumente) zulässig sein sollen. Verwenden Sie mindestens die Nichtterminalsymbole `Parameterliste`, `Parameter`, `Typ`, `Zahl`, `Rumpf` und `Ausdruck`. Die Ausdrücke sind aus Bezeichnern, Zahlen und den Funktionssymbolen `+`, `*`, `-` und `/` aufgebaut. Es stehen nur die Typen `int` und `double` zur Verfügung.
- b) Geben Sie ein Syntaxdiagramm an, das die Funktionsdefinitionen aus Aufgabe 3 charakterisiert. Der Übersichtlichkeit halber können Sie dieses Diagramm in mehrere Teildiagramme aufspalten.