

3. Übung „Prinzipien von Programmiersprachen“ Bearbeitung bis zum 16. November 2004

Programm für die Übung am 11. November

- Besprechung letzte Übung
- operationelle Semantik imperativer Sprachen

Aufgabe 1

In der Vorlesung wurden Inferenzsysteme zur Herleitung der R- und L-Werte von Ausdrücken vorgestellt. Ein Ausdruck sei aufgebaut aus Namen, Zahlen und den Operatoren + und *.

- Geben Sie ein Inferenzsystem zur Herleitung des Typs eines Ausdruckes an. Es stehen die drei Typen `int`, `float` und `double` zur Verfügung. Die Anwendung von + und * sei dabei nur dann zulässig, falls beide Argumente vom gleichen Typ sind.
- Erweitern Sie das System aus Teil a) so, daß Typkonversionen implizit erlaubt sind. Zum Beispiel kann der Typ `int` in den Typ `float` umgewandelt werden.
- Geben Sie eine Inferenzregel an, die die Semantik einer Zuweisung mit Typumwandlung spezifiziert.

Aufgabe 2

In der Vorlesung wurden Inferenzregeln für die Deklaration, Erzeugung und den Zugriff auf Arrays vorgestellt. Geben Sie analog dazu Inferenzregeln für die Deklaration eines Verbundes und den Zugriff auf einzelne Werte (L- und R-Werte) an. Falls `class V { $\tau_1 x_1$; ... ; $\tau_n x_n$;}` der Verbund ist, so soll bei der Deklaration „`V x`“ ein Eintrag `x : record($\tau_1 x_1, \dots, \tau_n x_n, l$)` in die Umgebung eingefügt werden. Es wird hierbei davon ausgegangen, daß jede Recordkomponente genau eine Speicherzelle benötigt und daß die Komponenten eines Records in aufeinanderfolgenden Zellen abgelegt werden.

Aufgabe 3

Implementieren Sie eine Java-Klasse `newton`, die eine Methode zur Berechnung einer Näherung für die Wurzel einer Zahl `x` zur Verfügung stellt. Das Newton-Verfahren berechnet ausgehend von einem Startwert `x0 = 1.0` eine Folge von Werten, die gegen

die Wurzel von x konvergiert. Das $n+1$ -te Folgenglied x_{n+1} ergibt sich aus dem n -ten Folgenglied x_n und dem Argument x nach folgender Formel:

$$x_{n+1} = \frac{1}{2} \left(x_n + \frac{x}{x_n} \right)$$

Die Iteration soll abgebrochen werden, wenn sich das Quadrat der Näherung maximal um 10^{-6} von dem Argument unterscheidet. Verwenden Sie die Programmierumgebung *BlueJ*, um Ihre Klasse zu testen. *BlueJ* erlaubt einzelne Methoden direkt aufzurufen.