

## 5. Übung „Prinzipien von Programmiersprachen“ Bearbeitung bis zum 30. November 2004

---

### Programm für die Übung am 25. November

- Besprechung letzte Übung
- Diskussion der Lösung für die Türme von Hanoi
- Parameterübergabe bei Prozeduren

### Aufgabe 1

In der Vorlesung wurde die Relation  $\vdash^R$  zur Herleitung des Wertes eines Ausdrucks definiert. Diese Relation wurde um Funktionsaufrufe erweitert. Es wurde jedoch vorausgesetzt, daß der Speicher  $M$  durch die Funktionsaufrufe nicht verändert wird, d.h. Funktionen dürfen keine Seiteneffekte haben.

Geben Sie nun eine Ableitungsrelation  $\langle E, M \rangle \vdash^{RM} e : v, M'$  an, bei der  $e$  ein Ausdruck,  $v$  ein Wert und  $M'$  der möglicherweise veränderte Speicher ist. Die Umgebung  $E$  soll während der Auswertung nicht verändert werden. Für einen Ausdruck  $e$  sind folgende Fälle zu betrachten:

```
e =      x                % Variable
        | c                % Konstante
        | e1 ⊗ e2        % ⊗ ∈ {+, -, *, /}
        | if (B) then e1 else e2
        | f(e1, ..., en)
```

Wie sind die Regeln für die Zuweisung und den Funktionsaufruf zu modifizieren?

### Aufgabe 2

In der Vorlesung wurden Exceptions in Java nur informell behandelt. In dieser Aufgabe sollen Sie die Semantik der Exceptions formalisieren.

Als Vereinfachung beschränken wir uns auf folgende Exceptions

```
ArrayIndexOutOfBoundsException
DivisionByZeroException
NullPointerException
```

Zur weiteren Vereinfachung ignorieren wir in dieser Aufgabe auch die Fehler-Nachricht, welche eigentlich in einer Exception abgelegt werden kann.

- Überlegen Sie zunächst, wie Sie den Zustand erweitern können, so daß Sie erfolgreiche Berechnungen von Berechnungen, in denen eine Exception aufgetreten ist, unterscheiden können.

- b) Erweitern Sie die Division, so daß im Fehlerfall eine `DivisionByZeroException` auftritt. Geben Sie auch Fehlerregeln für den Array-Zugriff an, die bei Zugriff auf einen ungültigen Index eine `ArrayIndexOutOfBoundsException` erzeugen.
- c) Diskutieren Sie, in welchen Fällen eine `NullPointerException` auftritt. Da Java nicht direkt die Programmierung mit Zeigern ermöglicht, können Sie von einem Pascal-ähnlichen Zeigerkonzept (wie in der Vorlesung vorgestellt) ausgehen.
- d) Definieren Sie die Semantik der `throw`-Anweisung und skizzieren Sie die Effekte der Exception durch Angabe der modifizierten Inferenzregeln für die Sequenz.
- e) Geben Sie die Semantik des `try-catch-finally` Konstrukts an. Beachten Sie hierbei auch, was passiert, wenn Teile (z.B. der `catch`- oder der `finally`-Block) wegfallen bzw. mehrere `catch`-Anweisungen hintereinander stehen.

### Aufgabe 3

Programmieren Sie die Türme von Hanoi in Java (Diskussion der Aufgabe in der Übung). Stellen Sie die Türme als drei Zahlenarrays dar. Die Zahlen entsprechen der Größe der Kugeln: 1 (kleinste Kugel) bis  $k$  (größte Kugel). Der Eintrag Null in einem Turmarray entspricht einem freien Platz, auf den eine Kugel gelegt werden kann. Achten Sie darauf, daß die Füllung der Turmarrays immer wie folgt aussehen muß:

$$\{x_0, \dots, x_n, 0, \dots, 0\} \quad , \text{ wobei } \forall 1 \leq i \leq n \leq k : x_{i-1} > x_i$$

Ziel ist es einen Turm der Höhe  $k$  vom ersten in das zweite Array zu versetzen, wobei in jedem Zwischenschritt obige Eigenschaft nicht verletzt werden darf. Die Ausgabe Ihres Programms soll die Zugfolge wie folgt ausgeben:

```
1 -> 2
1 -> 3
```

falls die oberste Kugel von Turm eins auf Turm zwei und danach die nächste Kugel von Turm eins auf Turm drei verschoben wird.

Geben Sie Lösungen für Türme der Höhen  $k = 3$  und  $k = 4$  an.