



Universität Karlsruhe

Informatik 1 WS 00/01

Institut für Telematik, Forschungsgruppe C&M
Prof. Dr. S. Abeck, R. Scholderer

Abgabe bis: 19.12.2000 14:00 Uhr

Übungsblatt 8

Aufgabe

1. Striktheit und Rechenstruktur BOOL

Wir gehen von der in der Vorlesung eingeführten Rechenstruktur BOOL aus.

Teilaufgabe 1) Was bedeutet Striktheit für eine in einer Rechenstruktur definierten Funktion?

Teilaufgabe 2) Legen Sie eine nicht strikte Funktionsdefinition für die Disjunktion mit der Bezeichnung \vee fest. Geben Sie hierzu die Signatur der Funktion an, und legen Sie ihre Wirkungsweise durch Gleichungen (Axiome) sowie eine Wertetabelle fest.

Teilaufgabe 3) Entwickeln Sie zwei Java-Programme, wobei das eine die strikte Definition der Disjunktion verwendet, das andere die nicht strikte.

Bei welchen Eingaben wird der Unterschied zwischen beiden Formen deutlich?

Woran erkennt man bei Java, dass eine Funktion Bottom als Ergebnis hat?

Hinweis: Bei einer Division durch eine Null im Nenner bricht ein Java-Programm ab und liefert die Ausnahme: `java.lang.ArithmeticException: / by zero` zurück.

Aufgabe

2. Rechenstruktur BOOLIF

- Wir gehen von der aus der Vorlesung bekannten Rechenstruktur $BOOL = (\Sigma_B, F_B)$ mit der Signatur $\Sigma_B = (S_B, F_B)$ aus, sowie einer Trägermenge M aus.

Es sei nun $BOOLIF = (S \cup \{M\}, F \cup \{ \text{if } . \text{ then } . \text{ else } . \text{ fi}^{BOOLIF} \})$ eine Rechenstruktur mit der zugehörigen Signatur $\Sigma = (S_B \cup \{m\}, F_B \cup \{ \text{fct if } . \text{ then } . \text{ else } . \text{ fi} = (\text{bool}, m, m) \})$ eine weitere Rechenstruktur, wobei gelte $m^{BOOLIF} = M$.

Die Funktion "fct if . then . else . fi" führt eine Fallunterscheidung in der aus Java bekannten Art und Weise durch.

- Geben Sie die Signatur von BOOLIF als Signaturdiagramm an.

b) Geben Sie eine strikte Funktionsspezifikation von `if . then . else . fi` in Form von Gleichungen (Axiomen) an.

c) Es sei nun $m^{\text{BOOLIF}} = B^+$ und `x`, `y`, `z` seien Identifikatoren der Sorte `m`. Geben Sie einen zu `if x then y else z` äquivalenten Term aus `BOOLIF` an, ohne `if . then . else . fi` zu verwenden.
Beweisen Sie Ihre Behauptung, indem Sie die Axiome aus b) nachweisen.

Aufgabe

3. Berechnungsformulare

Geben Sie Rechenformulare bzw. Berechnungsschemata zu den folgenden Termen an

a) `succ(add(pred(succ(zero)),succ(zero)))`

b) $(3*7)+4 - (6-2)^2$

c) `if x ≠ 0 then y / x else print(`Fehler´)`

[*Hinweis:* Bei der Beschreibung dieses Terms als Berechnungsschema treten Probleme auf. Wie müssen Sie die Struktur des Berechnungsschemas erweitern, um diese zu beheben?]

d) Setzen Sie bei dem Term aus c) `x=3` und `y=12` und erstellen Sie ein Rechenformular.

Aufgabe

4. Java: Ganzzahlige Quadratwurzel

Erstellen Sie ein Java-Programm, das von einer natürlichen Zahl die Quadratwurzel bestimmt.

Bei Zahlen, für die keine ganzzahlige Lösung existiert, genügt eine ganzzahlige Annäherung: Es seien $n, i \in \mathbb{N}$; n sei die natürliche Zahl, i die (Annäherung der) Quadratwurzel. Dann gilt: $i^2 \leq n < (i+1)^2$

Beispiel: Quadratwurzel von 4 ist 2; Quadratwurzel von 5 ist angenähert 2.
