

Übungsblatt 13

RECHENSTRUKTUREN

Abgabe: bis Freitag 04.02.2005, 13:00 Uhr in die Einwurfkästen im Untergeschoß des neuen Informatikgebäudes am Fasanengarten

**Erreichbare Punkte: 10 P / 50 T
(Praktische Aufgaben / Theoretische Aufgaben)**

Hinweis:

Alle Programmieraufgaben sind unter Verwendung der in der Vorlesung vorgestellten In-/Out-Klassen zu erstellen und (ähnlich wie die Beispielpprogramme im Skript) in ausführlicher Form zu kommentieren.

Versehen Sie die Programme mit Eingabeüberprüfungen, die sicherstellen, dass der Anwender korrekte Daten eingegeben hat.

1 RECHENSTRUKTUREN

1.1 Striktheit und Rechenstruktur BOOL (8T)

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

- Was heißt, dass die darin festgelegten Funktionen strikt sind? (2T)
- Legen Sie eine nicht strikte Funktionsdefinition für die Disjunktion mit der Bezeichnung `||` fest, die auf dem Prinzip der faulen Auswertung basiert. Geben Sie hierzu die Funktionalität der Funktion an und definieren Sie deren Wirkungsweise durch Gleichungen sowie eine Wertetabelle. (6T)

1.2 Strikte Funktionen (10T)

Geben Sie für eine nicht strikte logische Und-Verknüpfung (`a && b`) und eine strikte logische Oder-Verknüpfung (`a | b`)

- die Wertetabellen und (3T)
- die Gleichungen (3T)

an.

- Definieren Sie eine Rechenstruktur `BOOL2`, die diese beiden Verknüpfungen als Funktionen enthält. (4T)

1.3 Strikte Funktionen in Java (10P)

Realisieren Sie ein Programm, das zwei Parameter mit den Werten 1, 0 oder u (u für undefiniert) einliest und diese mittels einer Und-Verknüpfung und mittels einer Oder-Verknüpfung verknüpft.

Die Parameter (1, 0, u (für undefiniert)) sollen dabei nacheinander von der Tastatur eingelesen und das entsprechende Ergebnis auf dem Bildschirm ausgegeben werden. Es soll dabei über einen weiteren Parameter vom Anwender ausgewählt werden können, ob die Auswertung strikt oder nicht strikt erfolgt.

Implementieren Sie jeweils eine Methode, die die Und- und die Oder-Verknüpfung realisiert.

1.4 Rechenstruktur BOOLIF (12T)

Wir gehen von der Rechenstruktur BOOL aus, sowie einer Sorte m und einer zu m gehörigen Trägermenge M^+ . Wir erweitern im Folgenden die aus der Vorlesung bekannte Rechenstruktur BOOL zu BOOLIF um die Funktion $\text{fct } \text{if} . \text{then} . \text{else} . \text{fi} = \mathcal{B}^+ \times M^+ \times M^+ \rightarrow M^+$

- Stellen Sie die Signatur von BOOLIF graphisch in Form eines Signaturdiagramms dar. (4T)
- Geben Sie eine strikte Funktionsspezifikation von $\text{if} . \text{then} . \text{else} . \text{fi}$ an. (3T)
- Seien $x, y, z \in \mathcal{B}^+$. Geben Sie einen zu $\text{if } x \text{ then } y \text{ else } z \text{ fi}$ äquivalenten Term an, ohne $\text{if} . \text{then} . \text{else} . \text{fi}$ zu verwenden. Beweisen Sie Ihre Antwort. (5T)

1.5 Berechnungsformulare (10T)

Geben Sie Berechnungsformulare zu den folgenden Termen an:

- $\text{pred}(\text{add}(\text{succ}(\text{succ}(\text{zero})), \text{zero}))$ (3T)
- $(3+7)*4 - (9/2)^5$ (3T)
- $x + y / x - 13$ (3T)
- Wie nennt man ein Berechnungsformular wie Teilaufgabe c) ? (1T)

2 TERMERSETZUNG

2.1 Auswertungsstrategien für Termersetzungssysteme (10T)

Im Folgenden betrachten wir zwei Auswertungsstrategien für Termersetzungssysteme:

E (*eager*): Es wird in Termen immer möglichst weit innen reduziert. D.h.: Auf eine Funktion wird erst dann eine Regel angewandt, wenn sich auf keines Ihrer Argumente eine Regel anwenden lässt.

L (*lazy*): Es wird in Termen immer möglichst weit außen reduziert. D.h.: Auf Funktionsargumente werden nur dann Regeln angewandt, wenn weiter außen keine Regel anwendbar ist.

Wir erweitern die Rechenstruktur BOOLIF (siehe Aufgabe *Rechenstruktur BOOLIF*) zur Rechenstruktur NATSUM, indem wir neben den Sorten und Funktionssymbolen aus NAT die beiden Funktionen

$$\text{fct } \text{db} = (\text{nat}) \text{ nat} \quad \text{und} \quad \text{fct } \text{sum} = (\text{nat}) \text{ nat}$$

hinzunehmen. Für die Funktionen pred , add , $\frac{?}{?}$, db und sum ist folgendes Termersetzungssystem gegeben:

(P) $\text{pred}(\text{succ}(x)) \rightarrow x$

(G0) $\text{zero} \stackrel{?}{=} \text{zero} \rightarrow \text{true}$

(G1) $\text{zero} \stackrel{?}{=} \text{succ}(y) \rightarrow \text{false}$

(G2) $\text{succ}(x) \stackrel{?}{=} \text{zero} \rightarrow \text{false}$

(G3) $\text{succ}(x) \stackrel{?}{=} \text{succ}(y) \rightarrow x \stackrel{?}{=} y$

(A0) $\text{add}(\text{zero}, y) \rightarrow y$

(A1) $\text{add}(\text{succ}(x), y) \rightarrow \text{succ}(\text{add}(x, y))$

(D) $\text{db}(x) \rightarrow \text{add}(x, x)$

(S) $\text{sum}(x) \rightarrow \text{if } x \stackrel{?}{=} \text{zero} \text{ then zero else } \text{add}(x, \text{sum}(\text{pred}(x))) \text{ fi}$

(I0) $\text{if true then } x \text{ else } y \rightarrow x$

(I1) $\text{if false then } x \text{ else } y \rightarrow y$

Werten Sie folgende Terme jeweils nach den Strategien E und L aus:

a) $\text{add}(\text{succ}(\text{succ}(\text{zero})), \text{succ}(\text{zero})) \stackrel{?}{=} \text{zero}$ (5T)

b) $\text{db}(\text{add}(\text{succ}(\text{zero}), \text{zero}))$ (5T)

Geben Sie bei jedem Auswertungsschritt die verwendete Regel und ihre Operanden an.