Universität Karlsruhe

Informatik 1 WS 00/01

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

Abgabe bis: 30.01.2001 14:00 Uhr

Übungsblatt 12

Aufgabe

1. Funktional und induktive Deutung der Fakultätsfunktion

Gegeben ist folgende rekursive Funktionsdeklaration, die die Fakultät (x!) für natürliche Zahlen berechnet:

> **fct** fac = (**nat** *x*) **nat**:
>
> > **if** $\stackrel{?}{=}$ 0
> > **then** 1
> > **else** *x* * **fac**(x-1) **fi**

Teilaufgabe 1:
Geben Sie das zu fac gehörende Funktional τ an. Legen Sie auch den Definitions- und Wertebereich von τ fest.

Teilaufgabe 2:
Die Funktionenfolge für die induktive Deutung von fac ist wie üblich definiert:
Es seien fac_n: $N^I \rightarrow N^I$ ($n \in$ N) Funktionen mit
$fac_0(x) = I$
$fac_{n+1}(x) = t[fac_n]\,(x)$
für alle $x \in$ N
Berechnen Sie die drei ersten Glieder der Folge (fac_i): fac_0, fac_1 und fac_2.
Es genügt nicht, die Folgeglieder ohne Angabe des Lösungswegs aufzuschreiben.

Hinweis: Sowohl die in fac verwendeten Funktionen als auch die Funktion fac selbst sind strikt.

Aufgabe

2. Funktional und induktive Deutung der Funktion teile1

Gegeben ist folgende rekursive Funktionsdeklaration:

fct teile1 = (**int** *n*, **nat** *k* : *k* > 0) **int**:

if $n \stackrel{?}{=} 0$ **then** 0 **else** 1 + teilel(*n* - *k*, *k*) **fi**

Teilaufgabe 1: Zeichnen Sie die Aufrufbäume für die Aufrufe teile1(6,2) und teile1(8,3). Was berechnet die Funktion teile1. Für welche Eingaben terminiert sie? (Mit Begründung, aber ohne Beweis!)

Hinweis: Beim Aufrufbaum einer Funktion stellen die Knoten die (rekursiven) Funktionsaufrufe (z.B. teile1(4,2)) dar. Zwischen zwei Knoten (d.h. Funktionsaufrufen) existiert eine Kante, wenn der Aufruf der einen Funktion im Rumpf der anderen Funktion erfolgt.)

Teilaufgabe 2: Geben Sie das zu teile1 gehörende Funktional τ an. Legen Sie auch den Definitions- und Wertebereich von τ fest.

Teilaufgabe 3: Geben Sie eine geschlossene Darstellung für das i-te Folgeglied teile1_i der Funktionenfolge $(\text{teile1}_i)_{i \in N}$ zur Ermittlung induktiven Deutung der Funktion teile1 an. Beweisen sie Ihre Annahme durch vollständige Induktion über i.
Wie lautet die induktive Deutung teile1_∞ zur Funktion teile1?
Hinweis: Die Definition der einzelnen Folgeglieder erfolgt genauso wie bei der Folge $(\text{fac}_i)_{i \in N}$ (siehe Aufgabe Funktional und induktive Deutung der Fakultätsfunktion). Jegliche Funktionen sind selbstverständlich strikt.

Aufgabe

3. Einbettung der Funktionsdeklaration zur Funktion exp

Gegeben ist die Funktion exp(x, y)

fct exp = (**nat** *x*, **nat** *y*) **nat**:

if $y \stackrel{?}{=} 0$
then 1
else *x* * exp(*x*,*y* - 1) **fi**

Teilaufgabe 1:
Warum ist die folgende Funktionsdeklaration nicht repetitiv rekursiv?

fct exp_embed_try = (**nat** *x*, **nat** *y*, **nat** *z*)

if $y \stackrel{?}{=} 0$
then 1
else *x* * exp_embed_try(*x*, *y*-1, *x*)

Teilaufgabe2:
Geben Sie eine repetitiv rekursive Funktionsdeklaration für eine Funktion exp_embed an, die durch Einbettung der Funktion exp entsteht.

exp_embed habe die Funktionalität

fct exp_embed = (**nat,nat,nat**) **nat**

und es gelte:
$exp(m, n) = exp\_embed(m, n, 1)$ für alle $m, n \in$ N
(Diesen Zusammenhang müssen Sie für Ihre Funktionsdeklatation nicht beweisen).
Geben Sie (ebenfalls ohne Beweis) eine geschlossene Darstellung für Ihre
Deklaration von exp_embed an.

Hinweis: Verwenden Sie die Funktion $g(x,y) = \begin{cases} x^y, \text{ falls } y \neq 0 \\ 1, \text{ falls } y = 0 \end{cases}$

Aufgabe

4. Gofer: Klammergebirge

Schreiben Sie ein Gofer-Programm welches ein als Eingabe übergebenes Klammergebirge auf Korrektheit überprüft.
Zur Erinnerung: Ein Klammergebirge heißt korrekt, wenn in jeder Klammertiefe zu einer öffnenden auch eine schließende Klammer existiert: "(()(()))" ist korrekt "(()(())" ist nicht korrekt, es fehlt eine schließende Klammer.

Lösen Sie die Aufgabe wie folgt durch Einbettung:
Gegeben ist eine Funktion korrekt, die in Gofer wie folgt umgesetzt wird:

korrekt :: String → Bool
korrekt s = korrekt_([],s)

korrekt überprüft (durch Aufruf von _korrekt) für eine Zeichenreihe s ∈ {(,)}* als Eingabe, ob s ein korrektes Klammergebirge ist oder nicht. Beispielhafte Eingabe: korrekt "((()))" , Ausgabe True oder korrekt "(()(())", Ausgabe False.
korrekt_ ist eine in korrekt eingebettete Hilfsfunktion, die durch geeigneten Vergleich der öffnenden und schließenden Klammern feststellt, ob s ein korrektes Klammergebirge ist. Die Ausgabe (True oder False) von _korrekt ist gleichzeitig die Ausgabe von korrekt.

Aufgabe

5. Bindungen

Gegeben sei der folgende applikative Programmabschnitt:

⌈ **nat** n = 2;
⌈ **fct** erhoehe = (**nat** x) **nat**:

```
        x + n ⌋

    ⌈ nat n = 7;
      if n<l0 then erhoehe(n)

              else 1 fi

    ⌋
  * n ⌋
```

a) Ordnen Sie den Identifikatoren durch Pfeile ihre Bindungen zu.
b) Welches Ergebnis liefert obiger Abschnitt?

Hinweis: Setzen Sie bei beiden Teilaufgaben statische Bindung voraus.
