



Universität Karlsruhe

Informatik 1 WS 00/01

Institut für Telematik, Forschungsgruppe C&M

Prof. Dr. S. Abeck, R. Scholderer

Abgabe bis: 06.02.2001 14:00 Uhr

Übungsblatt 13

Aufgabe

1. Semantik rekursiver Funktionsdeklarationen

Gegeben ist die rekursive Funktionsdeklaration der Funktion exp (siehe auch: Aufgabe "Einbettung der Funktionsdeklaration zur Funktion exp ")

fct $\text{exp} = (\text{nat } x, \text{nat } y) \text{ nat}$:

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

Geben Sie für $n \in \mathbb{N}$ eine geschlossene Form der Funktion $\text{exp}_n(x, y)$, an, und beweisen Sie deren Richtigkeit mit vollständiger Induktion über n . Wie lautet die geschlossene Form für die Funktion $\text{exp}(x, y) := \text{exp}_\infty(x, y)$?

Verwenden Sie das zu exp gehörende Funktional, welches gegeben ist durch:

$\tau: (\mathbb{N}^\perp \times \mathbb{N}^\perp \rightarrow \mathbb{N}^\perp) \rightarrow (\mathbb{N}^\perp \times \mathbb{N}^\perp \rightarrow \mathbb{N}^\perp)$

$$\tau[f](x, y) = \begin{cases} \perp, & \text{falls } x = \perp \vee y = \perp \\ 1, & \text{falls } x \vee \perp \vee y = 0 \\ x * f(x, y - 1), & \text{falls } x \vee \perp \vee y > 0 \end{cases}$$

Hinweis:

$$\text{Betrachten sie die Funktion } g(x, y) = \begin{cases} 1, & \text{falls } y = 0 \\ x^y & \text{sonst} \end{cases}$$

Verwenden Sie bei Ihrem Beweis die Funktionenfolge $(f_i)_{i \in \mathbb{N}}$ zur Ermittlung der induktiven Deutung einer Funktion f . Es ist stets $f_0 = \perp$, $f_{i+1} = \tau[f_i]$. f_0 sollte als Induktionsanfang gewählt werden.

Aufgabe

2. Beweis mittels Parameterinduktion

Gegeben seien die Funktionen exp und exp_embed aus der Aufgabe "Einbettung der Funktionsdeklaration zur Funktion exp " mit den folgenden rekursiven Funktionsdeklarationen:

```
fct exp = (nat x, nat y) nat:           fct exp_embed = (nat x, nat y, nat z) nat:
    if y  $\stackrel{?}{\neq}$  0                          if y  $\stackrel{?}{\neq}$  0
    then 1                               then z
    else x * exp(x, y - 1) fi          else exp_embed(x, y - 1, z * x) fi
```

exp_embed entsteht aus der exp , indem man das Zwischenergebnis von exp im else-Fall ($x * \text{exp}(x, y - 1)$) in den Parameter z von exp_embed einbettet.

Zeigen sie mittels Parameterinduktion:

Die Funktion exp_embed ist eine Verallgemeinerung der Funktion exp für die

$$\text{exp}(m, n) = \text{exp_embed}(m, n, 1) \text{ für alle } m, n \in \mathbb{N}.$$

gilt.

Hinweis: Beweisen Sie zunächst die allgemeinere Aussage

$$p * \text{exp}(m, n) = \text{exp_embed}(m, n, p) \text{ für alle } m, n, p \in \mathbb{N}.$$

Aufgabe

3. Terminierungsbeweise für rekursive Funktionsdeklarationen

Gegeben sei die folgende rekursive Funktionsdeklaration der Funktion f .

```
fct f = (nat x. nat y) nat:
    if x > y then 1
    elif x  $\stackrel{?}{\neq}$  y then x
    else f(x, (x + y)  $\div$  2)  $\neq$  f(1 + (x + y)  $\div$  2, y)
    fi
```

Zeigen Sie mit Hilfe einer Abstiegsfunktion die Terminierung von f für Eingaben $(x, y) \in \mathbb{N}^2$ (also $x, y \neq \perp$)

Aufgabe

4. Gültigkeit und Lebensdauer (I)

Gegeben sei der folgende Programmabschnitt eines applikativen Programms:

```
┌ nat x = 2;
```

nat y = 3;

┌ **fst** teilen = (**int** x, **nat** y : y > 0) **int**:
if x < y **then** 0 **else** 1 + teilen(x - y, y) **fi** ┘

teilen(x, y) ┘

Teilaufgabe 1:

Ordnen Sie den auftretenden Identifikatoren durch Pfeile Bindungen zu.

Teilaufgabe 2:

Geben Sie für die im Programmabschnitt auftretenden Identifikatoren ihre Lebensdauer und ihre Gültigkeit in einer Tabelle an. Benennen Sie die Aufrufparameter der Funktion teilen zur besseren Unterscheidung mit x* und y*.

Aufgabe

5. Java: Schiffe versenken

In dieser Aufgabe soll das Spiel "Schiffe Versenken" in Java programmiert werden.

"Schiffe versenken" wie folgt: Jeder Spieler hat ein Blatt Papier mit zwei gleich großen Spielfeldern vor sich. In einem trägt er vor dem Beginn des Spieles - unsichtbar für den Gegner - die Positionen seiner eigenen Schiffe ein. Auf dem anderen werden während des Spieles alle seine erfolgreichen Treffer auf die Schiffe der feindlichen Flotte vermerkt.

Abwechselnd gibt ein Spieler dem anderen ein Koordinatenpaar an, und dieser antwortet mit "Treffer" oder "Wasser", je nachdem, ob er an der angegebenen Stelle ein Schiff plazierte hatte oder nicht.

Schreiben Sie ein Programm, das "Schiffe versenken" mit Ihnen spielt. Das Spiel soll nur in eine Richtung gehen: Der Computer verwaltet eine Flotte und gibt auf Ihre Fragen nach Schiffen Antworten ("Treffer", "Wasser", "schon versenkt" (falls an dieser Stelle schon ein erfolgreicher Treffer gelandet wurde)). Alle Treffer werden vom Computer vermerkt und das Spiel wird beendet, wenn Sie alle Schiffe getroffen haben. Eine grafische Darstellung des Spielfelds braucht nicht implementiert zu werden.

Beispiel für einen Schuß:

Buchstabe? c

Ziffer? 6

Treffer !!!

Das 8 x 8 Felder große Spielfeld soll wie folgt belegt werden:

	a	b	c	d	e	f	g	h
1					X			
2		X	X	X	X			
3			X			X		
4	X					X		
5						X		
6		X	X	X				

7		X	X					
8								

Überlegen Sie sich, welche Datenstruktur am besten geeignet ist, um das Spielfeld zu modellieren. Überlegen Sie sich weiterhin eine geeignete Kodierung für den Zustand eines Feldes ("Schiff", "Wasser" oder "schon versenkt").

Teilaufgabe 1:

Überlegen Sie sich zuerst, was Sie programmieren müssen, bevor Sie loslegen. Legen Sie eine Datenstruktur für das Spielfeld und eine Kodierung für die einzelnen Zustände fest und schreiben Sie die Arbeitsschritte, die Ihr Programm ausführen muß, in Pseudocode auf.

Teilaufgabe 2:

Implementieren Sie die Programmschritte, die Sie in Teilaufgabe 1 angegeben haben, und dokumentieren Sie die einzelnen Schritte ausreichend.

Hinweis: Verwenden Sie den Aufruf `Console.in.readWord().charAt(0)`, um Zeichen von der Tastatur einzulesen. Man kann mit Zeichen rechnen: So gilt ``f`-`a`==5`.
