



Abgabe: 13.01.2009 (vor der Vorlesung)

Aufgabe 11.1 (H) In großen Schritten zum Ziel

Gegeben seien folgende MiniOCaml-Definitionen:

```
let f = (fun x -> (fun y -> 2 * x + y))  
  
let g = f 7  
  
let twice = fun f1 -> fun a -> f1 (f1 a)  
  
let rec unzip =  
  fun l ->  
    match l with  
    [] -> ([],[])  
  | (x1,x2)::xs ->  
    (  
      match unzip xs with  
      (xs1,xs2) -> ((x1::xs1),(x2::xs2))  
    )
```

Konstruieren Sie die Beweise für folgende Aussagen:

- $\text{twice } g \ 7 \Rightarrow 35$
- $\text{unzip } [(1,2)] \Rightarrow ([1],[2])$

Lösungsvorschlag 11.1

a) Zur Vereinfachung nehmen wir folgende Setzungen vor:

$$e_{fun} := fun\ a \rightarrow f1\ (f1\ a)$$

$$\pi_g :=$$

$$g = f\ 7 \frac{\frac{f = fun\ x \rightarrow (fun\ y \rightarrow 2 * x + y)}{f \Rightarrow fun\ x \rightarrow (fun\ y \rightarrow 2 * x + y)} (GD)}{f\ 7 \Rightarrow fun\ y \rightarrow 2 * 7 + y} (App)$$

$$\frac{g = f\ 7 \quad f\ 7 \Rightarrow fun\ y \rightarrow 2 * 7 + y}{g \Rightarrow fun\ y \rightarrow 2 * 7 + y} (GD)$$

$$\begin{array}{c}
\frac{\text{twice} = \text{fun } f1 \rightarrow e_{\text{fun}}}{\text{twice} \Rightarrow \text{fun } f1 \rightarrow e_{\text{fun}}} \quad (GD) \quad \frac{\pi_g \quad e_{\text{fun}}[(\text{fun } y \rightarrow 2 * 7 + y)/f1] \Rightarrow \text{fun } a \rightarrow (\text{fun } y \rightarrow 2 * 7 + y) ((\text{fun } y \rightarrow 2 * 7 + y) a)}{\text{twice } g \Rightarrow \text{fun } a \rightarrow (\text{fun } y \rightarrow 2 * 7 + y) ((\text{fun } y \rightarrow 2 * 7 + y) a)} \quad (App) \\
\frac{\frac{\frac{2 * 7 \Rightarrow 14 \quad 14 + 7 \Rightarrow 21}{2 * 7 + 7 \Rightarrow 21} \quad (Op)}{\frac{\text{fun } y \rightarrow 2 * 7 + y}{(\text{fun } y \rightarrow 2 * 7 + y) 7} \Rightarrow 21} \quad (App)}{\frac{\text{fun } y \rightarrow 2 * 7 + y}{(\text{fun } y \rightarrow 2 * 7 + y) ((\text{fun } y \rightarrow 2 * 7 + y) 7)} \Rightarrow 21} \quad (App)}{\frac{2 * 7 \Rightarrow 14 \quad 14 + 21 \Rightarrow 35}{2 * 7 + 21 \Rightarrow 35} \quad (App)} \quad (App) \\
\frac{\text{twice } g \ 7 \Rightarrow 35}{\text{twice } g \ 7 \Rightarrow 35}
\end{array}$$

Aufgabe 11.2 (H) Abgeleitete Regeln

Es sei die folgende abgeleitete Regel betrachtet.

$$\frac{e_0 = \text{fun } x \rightarrow e \quad e_1 \text{ terminiert}}{e_0 \ e_1 = e[e_1/x]}$$

- a) Zeigen Sie anhand eines Gegenbeispiels, dass auf die Voraussetzung „ e_1 terminiert“ nicht verzichtet werden kann.
- b) Zeigen Sie die Gültigkeit obiger Regel.

Lösungsvorschlag 11.2

- a) **let rec** $f = \text{fun } x \rightarrow f(x+1)$
let $e_0 = \text{fun } y \rightarrow 0$

Offensichtlich terminiert die Auswertung des Ausdrucks $f\ 0$ nicht.

Die Auswertung des Ausdrucks $e_0\ (f\ 0)$ terminiert nicht (*). Andernfalls hätte ein Beweis für $e_0\ (f\ 0)$ folgende Form:

$$\frac{e_0 \Rightarrow \text{fun } y \rightarrow 0 \quad f\ 0 \Rightarrow v_1 \quad 0[v_1/y] \Rightarrow v}{e_0\ (f\ 0) \Rightarrow v} \text{ (App)}$$

Dies würde bedeuten, dass die Auswertung des Ausdrucks $f\ 0$ terminiert.

Die Auswertung des Ausdrucks $0[(f\ 0)/y]$ terminiert (**).

Aus (*) und (**) folgt, dass $e_0\ (f\ 0) = 0[(f\ 0)/y]$ nicht gilt.

- b) Es sei angenommen, dass die Prämissen gelten. Aus $e_0 = \text{fun } x \rightarrow e$ folgt $e_0 \Rightarrow \text{fun } x \rightarrow e$. Aus e_1 terminiert folgt, dass ein Wert v_1 existiert, so dass $e_1 \Rightarrow v_1$ und damit $e_1 = v_1$ gelten.

1. Fall: Die Auswertung des Ausdrucks $e[e_1/x]$ terminiert. Daraus folgt, dass ein Wert v existiert, so dass $e[e_1/x] \Rightarrow v$ und damit auch $e[e_1/x] = v$ gelten. Mit dem Substitutionslemma und $e_1 = v_1$ folgt, dass $e[v_1/x] \Rightarrow v$ gilt. Der folgende Beweis zeigt, dass sich $e_0 \ e_1$ unter diesen Voraussetzungen auch zu v auswertet:

$$\frac{e_0 \Rightarrow \text{fun } x \rightarrow e \quad e_1 \Rightarrow v_1 \quad e[v_1/x] \Rightarrow v}{e_0 \ e_1 \Rightarrow v} \text{ (App)}$$

2. Fall: Die Auswertung des Ausdrucks $e[e_1/x]$ terminiert nicht. Um zu zeigen, dass die Auswertung des Ausdrucks $e_0 \ e_1$ ebenfalls nicht terminiert, sei zur Herleitung eines Widerspruchs angenommen, dass ein Wert v mit $e_0 \ e_1 \Rightarrow v$ existiert. Damit muss ein Beweis der Form

$$\frac{e_0 \Rightarrow \text{fun } x \rightarrow e \quad e_1 \Rightarrow v_1 \quad e[v_1/x] \Rightarrow v}{e_0 \ e_1 \Rightarrow v} \text{ (App)}$$

existieren. Da $e_1 = v_1$ gilt, folgt mit dem Substitutionslemma, dass die Auswertung des Ausdrucks $e[e_1/x] = v$, d.h., da v ein Wert ist, dass die Auswertung des Ausdrucks $e[e_1/x]$ terminiert — Widerspruch. \square

Aufgabe 11.3 (P) Neues von fact, map und comp

a) Es seien folgende, bekannte Definitionen gegeben:

```

let rec fact = fun n ->
  match n with
    0 -> 1
  | n -> n * fact (n-1)

let rec fact_aux = fun x n ->
  match n with
    0 -> x
  | n -> fact_aux (n*x) (n-1)

let fact_iter = fact_aux 1

```

Zeigen Sie, dass unter der Voraussetzung, dass alle Aufrufe terminieren,

$$\text{fact_iter } n = \text{fact } n$$

für alle nicht-negativen ganzen Zahlen $n \in \mathbb{N}_0$ gilt.

b) Es seien folgende, bekannte Definitionen gegeben:

```

let comp = fun f g x -> f (g x)

let rec map = fun op l ->
  match l with
    [] -> []
  | x :: xs -> op x :: map op xs

```

Zeigen Sie, dass unter der Voraussetzung, dass alle Aufrufe terminieren,

$$\text{map (comp f g)} = \text{comp (map f) (map g)}$$

für alle f und g gilt.

Lösungsvorschlag 11.3

a) Per Induktion über n wird gezeigt, dass

$$\text{fact_aux } x \ n = x * \text{fact } n \tag{1}$$

für alle $n \in \mathbb{N}_0$ und alle $x \in \mathbb{Z}$ gilt.

Induktionsverankerung: Es gilt $n = 0$. Es folgt

$$\text{fact_aux } x \ n = \text{fact_aux } x \ 0 = x = x * 1 = x * \text{fact } 0.$$

Induktionsschritt: Es gilt $n > 0$. Es folgt:

$$\begin{aligned}
 \text{fact_aux } x \ n &= \text{fact_aux } (n * x) \ (n - 1) && \text{(Def. fact_aux)} \\
 &= n * x * \text{fact } (n - 1) && \text{(Induktionsannahme)} \\
 &= x * (n * \text{fact } (n - 1)) \\
 &= x * \text{fact } n && \text{(Def. fact)}
 \end{aligned}$$

Damit ist (1) gezeigt. Mithilfe von (1) folgt schließlich:

$$\text{fact_iter } n = \text{fact_aux } 1 \ n = 1 * \text{fact } n = \text{fact } n$$

b) Es ist zu zeigen, dass

$$\text{map } (\text{comp } f \ g) \ l = \text{comp } (\text{map } f) \ (\text{map } g) \ l$$

für alle Werte l gilt. Dies geschieht per Induktion:

Induktionsverankerung: Es gilt $l = []$. Es folgt:

$$\begin{aligned} \text{map } (\text{comp } f \ g) \ l &= \text{map } (\text{comp } f \ g) \ [] \\ &= [] && \text{(Def. map)} \\ &= \text{map } f \ [] && \text{(Def. map)} \\ &= \text{map } f \ (\text{map } g \ []) && \text{(Def. map)} \\ &= \text{comp } (\text{map } f) \ (\text{map } g) \ [] && \text{(Def. comp)} \\ &= \text{comp } (\text{map } f) \ (\text{map } g) \ l \end{aligned}$$

Induktionsschritt: Es gilt $l = v :: vs$ und damit folgt:

$$\text{map } (\text{comp } f \ g) \ vs = \text{comp } (\text{map } f) \ (\text{map } g) \ vs$$

Es folgt:

$$\begin{aligned} \text{map } (\text{comp } f \ g) \ l &= \text{map } (\text{comp } f \ g) \ (v :: vs) \\ &= \text{comp } f \ g \ v :: \text{map } (\text{comp } f \ g) \ vs && \text{(Def. map)} \\ &= \text{comp } f \ g \ v :: \text{comp } (\text{map } f) \ (\text{map } g) \ vs && \text{(Induktionsannahme)} \\ &= f \ (g \ v) :: \text{map } f \ (\text{map } g \ vs) && \text{(Def. comp)} \\ &= \text{map } f \ (g \ v :: \text{map } g \ vs) && \text{(Def. map)} \\ &= \text{map } f \ (\text{map } g \ (v :: vs)) && \text{(Def. map)} \\ &= \text{map } f \ (\text{map } g \ l) \\ &= \text{comp } (\text{map } f) \ (\text{map } g) \ l && \text{(Def. comp)} \end{aligned}$$

Big-Step Operationelle Semantik

Axiome: $v \Rightarrow v$ für jeden Wert v

Tupel:
$$\frac{e_1 \Rightarrow v_1 \quad \dots \quad e_k \Rightarrow v_k}{(e_1, \dots, e_k) \Rightarrow (v_1, \dots, v_k)} (T)$$

Listen:
$$\frac{e_1 \Rightarrow v_1 \quad e_2 \Rightarrow v_2}{e_1 :: e_2 \Rightarrow v_1 :: v_2} (L)$$

Globale Definitionen:
$$\frac{f = e \quad e \Rightarrow v}{f \Rightarrow v} (GD)$$

Lokale Definitionen:
$$\frac{e_1 \Rightarrow v_1 \quad e_0[v_1/x] \Rightarrow v_0}{\text{let } x = e_1 \text{ in } e_0 \Rightarrow v_0} (LD)$$

Funktionsaufrufe:
$$\frac{e_1 \Rightarrow \text{fun } x \rightarrow e_0 \quad e_2 \Rightarrow v_2 \quad e_0[v_2/x] \Rightarrow v_0}{e_1 e_2 \Rightarrow v_0} (App)$$

Pattern Matching:
$$\frac{e_0 \Rightarrow v' \equiv p_i[v_1/x_1, \dots, v_k/x_k] \quad e_i[v_1/x_1, \dots, v_k/x_k] \Rightarrow v}{\text{match } e_0 \text{ with } p_1 \rightarrow e_1 \mid \dots \mid p_m \rightarrow e_m \Rightarrow v} (PM)$$

— sofern v' auf keines der Muster p_1, \dots, p_{i-1} passt

Eingebaute Operatoren:
$$\frac{e_1 \Rightarrow v_1 \quad e_2 \Rightarrow v_2 \quad v_1 \text{ op } v_2 \Rightarrow v}{e_1 \text{ op } e_2 \Rightarrow v} (Op)$$

— Unäre Operatoren werden analog behandelt.

Substitutionslemma

$$\frac{e_1 = e_2}{e[e_1/x] = e[e_2/x]}$$