TECHNISCHE UNIVERSITÄT MÜNCHEN
FAKULTÄT FÜR INFORMATIK

Lehrstuhl für Sprachen und Beschreibungsstrukturen WS 2008/09
Einführung in die Informatik 2 **Übungsblatt 11**
Prof. Dr. Helmut Seidl, T. M. Gawlitza, S. Pott, M. Schwarz 23.12.2008

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:

a) twice g 7 $\Rightarrow$ 35

b) unzip [(1,2)] $\Rightarrow$ ([1],[2])

Lösungsvorschlag 11.1

a) Zur Vereinfachung nehmen wir folgende Setzungen vor:

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

$\pi_g :=$

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

$$\dfrac{\dfrac{\dfrac{twice = fun\ f1 \rightarrow e_{fun}}{twice \Rightarrow fun\ f1 \rightarrow e_{fun}}\ (GD) \qquad e_{fun}[g/f1] \Rightarrow fun\ a \rightarrow g\ (g\ a)}{twice\ g \Rightarrow fun\ a \rightarrow g\ (g\ a)}\ (App) \qquad \dfrac{\pi_g \qquad \dfrac{\pi_g \qquad \dfrac{2*7 \Rightarrow 14 \quad 14+7 \Rightarrow 21}{2*7+7 \Rightarrow 21}\ (Op)}{g\ 7 \Rightarrow 21}\ (App) \qquad \dfrac{2*7 \Rightarrow 14 \quad 14+21 \Rightarrow 35}{2*7+21 \Rightarrow 35}\ (App)}{g\ (g\ 7) \Rightarrow 35}\ (App)}{twice\ g\ 7 \Rightarrow 35}\ (App)$$

b) Zur Vereinfachung nehmen wir folgende Setzungen vor:

$e_{match} := \texttt{match unzip xs with } (\texttt{xs1}, \texttt{xs2}) \rightarrow ((\texttt{x1} :: \texttt{xs1}), (\texttt{x2} :: \texttt{xs2}))$

$e_{fun} := \texttt{match l with } [] \rightarrow ([], []) \;|(\texttt{x1}, \texttt{x2}) :: \texttt{xs} \rightarrow (\texttt{e}_{\texttt{match}})$

$$\dfrac{\dfrac{\texttt{unzip} = \texttt{fun l} \rightarrow \texttt{e}_{\texttt{fun}}}{\texttt{unzip} \Rightarrow \texttt{fun l} \rightarrow \texttt{e}_{\texttt{fun}}}\,(GD) \qquad \dfrac{[(1,2)] \Rightarrow [(1,2)] \equiv (1,2) :: [] \qquad \dfrac{\dfrac{\dfrac{\texttt{unzip} = \texttt{fun l} \rightarrow \texttt{e}_{\texttt{fun}}}{\texttt{unzip} \Rightarrow \texttt{fun l} \rightarrow \texttt{e}_{\texttt{fun}}}\,(GD) \quad \dfrac{[] \Rightarrow [] \quad ([], []) \Rightarrow ([], [])}{\texttt{e}_{\texttt{fun}}[[]/\texttt{l}] \Rightarrow ([], [])}\,(PM)}{\texttt{unzip}\ [] \Rightarrow ([], []) \equiv (\texttt{xs1}, \texttt{xs2})[[]/\texttt{xs1}, []/\texttt{xs2}]}\,(App) \qquad ((1 :: []), (2 :: [])) \Rightarrow ([1], [2])}{\texttt{e}_{\texttt{match}}[1/\texttt{x1}, 2/\texttt{x2}, []/\texttt{xs}] \Rightarrow ([1], [2])}\,(PM)}{\texttt{e}_{\texttt{fun}}[[(1,2)]/\texttt{l}] \Rightarrow ([1], [2])}\,(PM)}{\texttt{unzip}\ [(1,2)] \Rightarrow ([1], [2])}\,(App)$$

Aufgabe 11.2 (H) Abgeleitete Regeln

Es sei die folgende abgeleitete Regel betrachtet.

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

a) Zeigen Sie anhand eines Gegenbeispiels, dass auf die Voraussetzung „$\mathtt{e_1}$ terminiert“ nicht verzichtet werden kann.

b) Zeigen Sie die Gültigkeit obiger Regel.

Lösungsvorschlag 11.2

a) **let rec** f = **fun** x –> f (x+1)
let e0 = **fun** y –> 0
Offensichtlich terminiert die Auswertung des Ausdrucks `f 0` nicht.

Die Auswertung des Ausdrucks $\mathtt{e0\ (f\ 0)}$ terminiert nicht (*). Andernfalls hätte ein Beweis für $\mathtt{e0\ (f\ 0)}$ folgende Form:

$$\frac{\mathtt{e0} \Rightarrow \mathtt{fun\ y} \rightarrow \mathtt{0} \quad \mathtt{f\ 0} \Rightarrow \mathtt{v_1} \quad \mathtt{0}[\mathtt{v_1}/\mathtt{y}] \Rightarrow \mathtt{v}}{\mathtt{e0\ (f\ 0)} \Rightarrow \mathtt{v}}\ (App)$$

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

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

Aus (*) und (**) folgt, dass $\mathtt{e0\ (f\ 0)} = \mathtt{0}[(\mathtt{f\ 0})/\mathtt{y}]$ nicht gilt.

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

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

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

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

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

exisitieren. Da $\mathtt{e_1} = \mathtt{v_1}$ gilt, folgt mit dem Substitutionslemma, dass die Auswertung des Ausdrucks $\mathtt{e}[\mathtt{e_1}/\mathtt{x}] = \mathtt{v}$, d.h., da $\mathtt{v}$ ein Wert ist, dass die Auswertung des Ausdrucks $\mathtt{e}[\mathtt{e_1}/\mathtt{x}]$ 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,

$$\texttt{fact\_iter}\ \texttt{n} = \texttt{fact}\ \texttt{n}$$

für alle nicht-negativen ganzen Zahlen $\texttt{n} \in \mathbf{N}_0$ 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,

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

für alle `f` und g gilt.

Lösungsvorschlag 11.3

a) Per Induktion über `n` wird gezeigt, dass

$$\texttt{fact\_aux}\ \texttt{x}\ \texttt{n} = \texttt{x} * \texttt{fact}\ \texttt{n} \tag{1}$$

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

Induktionsverankerung: Es gilt $\texttt{n} = 0$. Es folgt

$$\texttt{fact\_aux}\ \texttt{x}\ \texttt{n} = \texttt{fact\_aux}\ \texttt{x}\ 0 = \texttt{x} = \texttt{x} * 1 = \texttt{x} * \texttt{fact}\ 0.$$

Induktionsschritt: Es gilt $\texttt{n} > 0$. Es folgt:

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

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

$$\texttt{fact\_iter}\ \texttt{n} = \texttt{fact\_aux}\ 1\ \texttt{n} = 1 * \texttt{fact}\ \texttt{n} = \texttt{fact}\ \texttt{n}$$

b) Es ist zu zeigen, dass

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

für alle Werte `l` gilt. Dies geschieht per Induktion:

Induktionsverankerung: Es gilt $\texttt{l} = []$. Es folgt:

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

Induktionsschritt: Es gilt $\texttt{l} = \texttt{v} :: \texttt{vs}$ und damit folgt:

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

Es folgt:

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

Big-Step Operationelle Semantik

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

Tupel:
$$\frac{e_1 \;\Rightarrow\; v_1 \qquad \ldots \qquad e_k \;\Rightarrow\; v_k}{(e_1,\ldots,e_k) \;\Rightarrow\; (v_1,\ldots,v_k)}\;(T)$$

Listen:
$$\frac{e_1 \;\Rightarrow\; v_1 \qquad e_2 \;\Rightarrow\; v_2}{e_1 \,\texttt{::}\, e_2 \;\Rightarrow\; v_1 \,\texttt{::}\, v_2}\;(L)$$

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

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

Funktionsaufrufe:
$$\frac{e_1 \;\Rightarrow\; \texttt{fun}\; x \,\texttt{->}\, e_0 \qquad e_2 \;\Rightarrow\; v_2 \qquad e_0[v_2/x] \;\Rightarrow\; v_0}{e_1\; e_2 \;\Rightarrow\; v_0}\;(App)$$

Pattern Matching:
$$\frac{e_0 \;\Rightarrow\; v' \equiv p_i[v_1/x_1,\ldots,v_k/x_k] \qquad e_i[v_1/x_1,\ldots,v_k/x_k] \;\Rightarrow\; v}{\texttt{match}\; e_0 \;\texttt{with}\; p_1 \,\texttt{->}\, e_1 \;\texttt{|}\; \ldots \;\texttt{|}\; p_m \,\texttt{->}\, e_m \;\Rightarrow\; v}\;(PM)$$

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

Eingebaute Operatoren:
$$\frac{e_1 \;\Rightarrow\; v_1 \qquad e_2 \;\Rightarrow\; v_2 \qquad v_1 \;\mathsf{op}\; v_2 \Rightarrow v}{e_1 \;\mathsf{op}\; e_2 \;\Rightarrow\; v}\;(Op)$$

— Unäre Operatoren werden analog behandelt.

Substitutionslemma

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