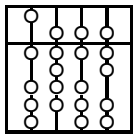
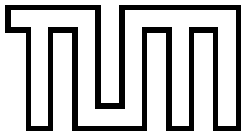


TECHNISCHE UNIVERSITÄT MÜNCHEN
FAKULTÄT FÜR INFORMATIK



**Lehrstuhl für Sprachen und Beschreibungsstrukturen**
*Prof. Dr. Helmut Seidl*
*A. Berlea, F. Forster, T. Gawlitza*

SoS 2007
Übungsblatt 7
**Ausgegeben am:** 12. Juli 2007
**Abgabe bis:** -

# Übungen zu Einführung in die Informatik II

## Aufgabe 1 Terminierung

Gegeben sei folgende MiniOcaml-Definition der Funktion `map`:

```
let map = fun f -> fun l ->
    match l with
        [] -> []
      | x::xs -> (f x)::(map f xs)
```

Sei weiterhin `g : t1 -> t2` eine Funktion (ein Wert), die für jedes Argument terminiert. D.h. für jeden Wert $v$ existiert ein Wert $v'$, so dass $\mathtt{g}\ \mathtt{v} \Rightarrow \mathtt{v}'$. Zeigen Sie mit Hilfe der Big-Step operationellen Semantik, dass der Aufruf `map g l` für jedes `l`, das sich zu einer Liste auswertet, terminiert.

## Aufgabe 2 Verifikation funktionaler Programme

Gegeben sei folgende MiniOcaml-Funktion zur Umkehrung einer Liste

```
let rec rev = fun list ->
    match list with
          [] -> []
        | el :: rest -> app (rev rest) [el]
```

sowie die in der Vorlesung wie folgt definierte Funktion `app`

```
let rec app = fun x -> fun y ->
  match x with
      [] -> y
    | x::xs -> x :: app xs y
```

Zeigen Sie unter der Annahme, dass alle vorkommenden Funktionsaufrufe terminieren, dass

a) $\mathtt{rev}\ (\mathtt{app}\ \mathtt{xs}\ \mathtt{ys}) = \mathtt{app}\ (\mathtt{rev}\ \mathtt{ys})\ (\mathtt{rev}\ \mathtt{xs})$

b) $\mathtt{rev}\ (\mathtt{rev}\ \mathtt{list}) = \mathtt{list}$

# Big-Step Operationelle Semantik

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

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

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

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

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

Funktionsaufrufe:
$$\frac{e_1 \Rightarrow \texttt{fun}\ x\texttt{->}e_0 \quad e_2 \Rightarrow v_2 \quad e_0[v_2/x] \Rightarrow v_0}{e_1\ e_2 \Rightarrow v_0}$$

Pattern Matching:
$$\frac{e_0 \Rightarrow v' \equiv p_i[v_1/x_1, \ldots, v_k/x_k] \quad e_i[v_1/x_1, \ldots, v_k/x_k] \Rightarrow v}{\texttt{match}\ e_0\ \texttt{with}\ p_1\ \texttt{->}e_1\ \texttt{|}\ \ldots\ \texttt{|}\ p_m\ \texttt{->}e_m \Rightarrow v}$$
— sofern $v'$ auf keines der Muster $p_1, \ldots, p_{i-1}$ passt ;-)

Eingebaute Operatoren:
$$\frac{e_1 \Rightarrow v_1 \quad e_2 \Rightarrow v_2 \quad v_1\,\texttt{op}\,v_2 = v}{e_1\,\texttt{op}\,e_2 \Rightarrow v}$$
— Unäre Operatoren werden analog behandelt.