

Übungen zu Einführung in die Informatik II

Aufgabe 1 **Verifikation funktionaler Programme**

Gegeben sei die folgende Mini-OCaml-Funktion zur Berechnung der Summe einer Liste von Integern:

```
let rec sum = fun list -> match list with
  [] -> 0
  | x::xs -> x + (sum xs)
```

- Zeigen Sie für die in der Vorlesung angegebene Small-Step operationelle Semantik für Mini-OCaml, dass $\text{sum } (42 :: []) \rightarrow^* 42$ gilt. (Dabei sei \rightarrow^* die transitive Hülle der Relation \rightarrow .)
- Zeigen Sie entsprechend für die in der Vorlesung angegebene Big-Step operationelle Semantik für Mini-OCaml, dass $\text{sum } (42 :: []) \Rightarrow 42$ gilt.

Aufgabe 2 **Verifikation funktionaler Programme**

Gegeben sei folgende OCaml-Funktion zur Umkehrung einer Liste:

```
let rec rev = fun list ->
  match list with
  | [] -> []
  | e1 :: rest -> app (rev rest) [e1]
```

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
```

Ziel dieser Aufgabe ist es zu zeigen, dass die Funktion `rev` involutorisch (oder selbst-invers) ist, d.h. es gilt:

$$\text{rev } (\text{rev } \text{list}) = \text{list} \tag{1}$$

- Um die Gültigkeit der Aussage (1) zu zeigen, muss zunächst Folgendes gezeigt werden:

$$\text{rev } (\text{app } \text{xs } \text{ys}) = \text{app } (\text{rev } \text{ys}) (\text{rev } \text{xs}) \tag{2}$$

- Zeigen Sie nun die Gültigkeit des Prädikates (1)!