



## Einführung in die Informatik II

Univ.-Prof. Dr. Andrey Rybalchenko, A. Herz, K. Apinis

Dieses Blatt behandelt Kapitel 6.5 - 6.7, 7, 14.0 - 14.3, 15.0 - 15.3 und 15.8 aus dem Buch zur Vorlesung.

**Aufgabe 6.16** Schreiben Sie eine Prozedur `append : 'a mylist → 'a mylist → 'a mylist` die zwei gemäß des Typkonstruktors `mylist (datatype 'a mylist = Nil | Cons of 'a * 'a mylist)` dargestellte Listen konkateniert.

## Aufgabe 15.6

Vervollständigen Sie die Deklaration `val (count, inc, reset) = so`, dass sie einen eingekapselten Zähler mit dem Anfangswert 0 und drei Prozeduren wie folgt liefert:

- `count : unit → int` liefert den Wert des Zählers.
- `inc : unit → unit` erhöht den Wert des Zählers um 1.
- `reset : unit → unit` setzt den Zähler auf 0 zurück.

## Aufgabe Signatur und Binärbaum

- a) Deklarieren Sie eine Struktur `structure StringBinTree` für binäre Suchbäume, welche folgende Signatur unterstützt:

```
signature sigBinTree = sig
  type tree (*datatype representing the tree*)
  val insert : tree → int → string → tree (*insert item into tree,
    invariant: left.key < key <
right.key*)
  val find : tree → int → string (*return item associated with key*)
  val create : tree (*return empty tree*)
  val depth : tree → int → int (*adds the depth of the tree to the given
    value*)
  exception NotFound (*raised by find if not found*)
  exception KeyConflict (*raised by insert in case of duplicate key*)
end
```

- b) Schreiben Sie die Prozedur `gen_tree : (int * string) list → StringBinTree.tree → StringBinTree.tree` welche die übergebenen Key - String Paare in den Baum einfügt.
- c) Konstruieren Sie einen Baum mittels `gen_tree`.
- d) Schreiben Sie eine Prozedur `find_safe : StringBinTree.tree → int → string → string` welche den übergebenen String zurückgibt, falls der Key nicht gefunden werden kann.
- e) Überlegen Sie sich das Resultat von `StringBinTree.depth (gen_tree [(1,"str1"),(5,"str5"),(7,"str7"),(2,"str2")] StringBinTree.create) 0`