



Übungen zu „Konzepte von Programmiersprachen“, WS 2010/11

Prof. Dr. R. Loogen · Fachbereich Mathematik und Informatik · Hans-Meerwein-Straße, D-35032 Marburg

Nr. 9, Abgabe: Dienstag, 21. Dezember 2010 vor der Vorlesung

22. Typinferenz

4 Punkte

Führen Sie für die folgende Funktion eine Typinferenz mit dem in der Vorlesung vorgestellten Verfahren durch:

```
go f q0 []      = [q0]
go f q0 (x:xs) = let qs = go f q0 xs
                  in f x (head qs) : qs
```

23. Typklassen

6 Punkte

Gegeben seien die folgenden Deklarationen:

```
-- verschiedene Baumdefinitionen
data BinTree a  = Leaf a  | Node (BinTree a) (BinTree a)
data LabTree l a = LLeaf a | LNode l (LabTree l a) (LabTree l a)
data STree a    = Empty  | Split a (STree a) (STree a)
data RoseTree a = RNode a [RoseTree a]
data AExp a     = Val a | Plus (AExp a) (AExp a) | Mult (AExp a) (AExp a)
-- Typklasse Tree
class Tree t where
  subtrees :: t -> [t] -- liefert direkte Teilbaeume
```

(a) Geben Sie für jede Baumart eine Instanzendeklaration für die Klasse `Tree` an. / 1

(b) Definieren Sie die folgenden Funktionen:

`depth :: Tree t => t -> Int` bestimmt die Tiefe eines Baumes. / 1

`size :: Tree t => t -> Int` bestimmt die Knotenanzahl eines Baumes. / 1

`dfs :: Tree t => t -> [t]` / 1

liefert die Liste *aller* Teilbäume in Präfixordnung.

`bfs :: Tree t => t -> [t]` liefert die Liste *aller* Teilbäume ebeneweise. / 2

24. Fehlermeldungen im Unifikationsprogramm

2 Punkte

Erweitern Sie die auf der Vorlesungsseite angegebene Implementierung des Unifikationsalgorithmus von Robinson (Datei `robinson.hs`) im Fall scheiternder Unifikationen um die Ausgabe einer Beschreibung der Fehlerursache.