

Übung 4 zu KMPS

Besprechung siehe Homepage

Aufgabe 21:

- a) Welchen Platzaufwand hat die `sum`-Funktion unten auf Folie „HigherOrderProgrammierung“ 3?
- b) Implementieren Sie `sum` tail-rekursiv, um den Platzaufwand konstant zu halten, indem Sie das folgende Programmgerüst ergänzen:

```
def sum(f: Int => Int) (a: Int, b: Int): Int = {  
  def iter(a: Int, result: Int): Int = {  
    if (??) ??  
    else iter(??, ??)  
  }  
  iter(??, ??)  
}
```

- c) Geben Sie den Aufruf von `sum` an, um die Summe der Zahlen zwischen 3 und 5 zu erhalten?

Aufgabe 22:

- a) Definieren Sie eine Scala-Funktion `add`, die die Summe der Elemente einer Integer-Liste liefert.
- b) Definieren Sie ausschließlich unter Verwendung der Funktionen `add` und `map` eine Higher-Order Funktion `addMap`, die eine Funktion auf alle Listenelemente anwendet und dann die Summe aller erhaltenen Listenelemente liefert.

Aufgabe 23:

- a) Definieren Sie eine Higher-Order-Funktion `foldl`, die für alle nichtleeren Integer-Listen mittels einer zweistelligen Funktion `f` alle Listenelemente von links nach rechts verknüpft. Dabei wird bei der leeren Liste der anzugebenden Startwert zurückgegeben.
Bsp.: Falls es sich bei `f` um die Addition handelt und der Startwert 0 ist, erhält man die Summe aller Listenelemente.
- b) Wie muss man `foldl` aufrufen, um das Produkt der Listenelemente zu erhalten?

Aufgabe 24:

Implementieren Sie eine Funktion `range`, die eine Liste der Integerzahlen zwischen `a` und `b` erzeugt.

Aufgabe 25:

Definieren Sie eine Scala-Funktion zur Addition zweier Integer-Zahlen durch currying und erläutern Sie, wie man damit welche anderen Funktionen definieren kann und wie man diese aufruft.

Aufgabe 26:

- a) Erweitern Sie die Funktion `map` auf Binärbäume mit Integerwerten als Einträge, so dass die Funktion `f` auf alle Knoten des Binärbaums angewendet wird.
- b) Wie muss man das erweiterte `map` aufrufen, um die Einträge des Binärbaums zu verdoppeln?

Übung 4 zu KMPS

Besprechung siehe Homepage

Aufgabe 27:

- a) Erweitern Sie die Funktion `filter` auf Binärbäume mit Integerwerten als Einträge, so dass eine Liste die Knoten des Binärbaums in Präorderreihenfolge enthält, die die boolesche Funktion erfüllen.
- b) Wie muss man das erweiterte `filter` aufrufen, um alle geraden Einträge des Binärbaumes in der Ergebnisliste zu speichern?

Aufgabe 28:

Implementieren Sie eine Scala-Funktion `infix`, die überprüft, ob eine Liste `xs` in einer Liste `ys` enthalten ist. Verwenden Sie dabei die Scala-interne Listenstruktur. Sie dürfen dabei die Funktion `präfix` aus Aufgabe 12 verwenden (`präfix(xs, ys)` bedeutet, dass `xs` Präfix von `ys` ist).