

Einführung in die funktionale Programmierung

Wintersemester 2024/2025

Aufgabenblatt Nr. 4

Abgabe: bis Dienstag 10.12.2024 online

Aufgabe 1 (30 Punkte)

Ziel dieser Aufgabe ist Äquivalenz bzw. Nicht-Äquivalenz von Ausdrücken/Programmen in KFPTSProb zu erkennen und zu begründen, und evtl. Wahrscheinlichkeitswerte anzugeben, für die das gilt.

1. Schreiben Sie eine Funktion `multiCoin`, die bei Eingabe einer Liste $[(w_1, p_1), (w_2, p_2), \dots]$ eine Funktion ist, die die Verteilung $[(w_1, p_1), (w_2, p_2), \dots]$ hat. Hierbei nehmen wir an, dass $w_i \in \{0, 1, 2, \dots\}$ und dass $0 \leq p_i \leq 1$ und $\sum_i p_i \leq 1$.
2. Berechnen Sie die Verteilung von `result`.

```
result = (coin 0.4 1 2) + (coin 0.6 2 1)
```

3. Haben `result1` und `result2` die gleiche Verteilung?

```
wuerfle x = coin 0.5 1 2
result1 = let x = wuerfle () in
           if x == 1 then 2 else wuerfle ()
result2 = 2
```

4. Für welche Konstanten $0 \leq p_1, p_2 \leq 1$ sind `result1` und `result2` Verteilungs-äquivalent?

```
result1 = (coin 0.5 1 2) + (coin 0.5 2 3)
result2 = coin p1 3 (coin p2 4 5)
```

5. Sind `result1` und `result2` Verteilungs-äquivalent?

```
result1 = map (\x -> coin 0.5 0 1) [1..]
result2 = map (\x -> if x == 1 then 0 else 1) [1..]
```

6. Welche Ausdrücke aus `result1`, `result2` und `result3` sind Verteilungs-äquivalent und welche nicht?

```
liste1 = map (\x -> coin 0.5 0 1) [1..]
liste2 = map (\x -> coin 0.5 1 0) [1..]
result1 = map (\(x,y) -> x*y) (zip liste1 liste1)
result2 = map (\(x,y) -> x*y) (zip liste1 liste2)
result3 = map (\(x,y) -> x*y) (zip liste2 liste2)
```

Aufgabe 2 (20 Punkte)

Ziel dieser Aufgabe ist ein Gegentest mit Funktionen die ChatGPT oder ein anderer Chatbot evtl. vorschlagen. D.h. fragen Sie den Chatbot nach der Definition einer Haskell Funktion, die das Problem löst. Überprüfen Sie dann, ob diese das Richtige leistet.

Aufgabe 1: Schreibe eine Funktion, die eine (endliche) Liste von ganzen Zahlen daraufhin prüft, ob es ein Element m in der Liste gibt, das alle Zahlen der Liste als Teiler hat.

Aufgabe 2: Schreibe eine Funktion, die testet, ob zwei gegebene (endliche) Listen von Listen von positiven ganzen Zahlen gleich sind bzgl. folgender Relation: zwei Unterlisten sind gleich, wenn sie durch Ändern der Reihenfolge gleich gemacht werden können.