



Praktische Informatik 3: Funktionale Programmierung

Vorlesung 3 (01.11.2022): Übungen

Christoph Lüth



Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH



Universität
Bremen

Wintersemester 2022/23

Striktheit Revisited

- **Konjunktion** definiert als

```
a && b = case a of False → False  
                    True → b
```

- Alternative Definition als Wahrheitstabelle:

```
and :: Bool → Bool → Bool  
and False True = False  
and False False = False  
and True True = True  
and True False = False
```

Übung 3.1: Kurze Frage: Gibt es einen Unterschied zwischen den beiden?

Striktheit Revisited

- **Konjunktion** definiert als

```
a && b = case a of False → False  
           True → b
```

- Alternative Definition als Wahrheitstabelle:

```
and :: Bool → Bool → Bool  
and False True = False  
and False False = False  
and True True = True  
and True False = False
```

Übung 3.1: Kurze Frage: Gibt es einen Unterschied zwischen den beiden?

Lösung:

- Erste Definition ist **nicht-strikt** im zweiten Argument.
- Merke: wir können Striktheit von Funktionen (ungewollt) **erzwingen**

Jetzt seit ihr dran

Übung 3.2: Refaktorierungen

Was passiert bei folgenden Änderungen an `preis` :

- 1 Vorletzte Zeile zu `Cent (round (l* case bio of Chemie → 69; Bio→ 119`
- 2 Vorletzte Zeile zu `Cent (round (l* case bio of Bio→ 119; _ → 69`
- 3 Vertauschung der zwei vorletzten und letzten Zeile.

Jetzt seit ihr dran

Übung 3.2: Refaktorierungen

Was passiert bei folgenden Änderungen an `preis` :

- ① Vorletzte Zeile zu `Cent (round (l* case bio of Chemie → 69; Bio→ 119`
- ② Vorletzte Zeile zu `Cent (round (l* case bio of Bio→ 119; _ → 69`
- ③ Vertauschung der zwei vorletzten und letzten Zeile.

Lösung:

- ① Nichts, unterschiedliche Fälle können getauscht werden.
- ② Nichts, da `_` nur `Chemie` sein kann
- ③ Der letzte Fall wird nie aufgerufen — der Milchpreis wäre `Ungueltig`