

Übungsblatt 12

Aufgabe 1. Zeigen Sie, dass die folgenden Haskell-Ausdrücke wohlgetypt sind, indem Sie jeweils Typherleitungen angeben.

(a) `\x -> ((+) x) 1`

(b) `let x = \y -> x y in x True`

(c) Wir definieren zunächst

```
data IntPair = MkPair Int Int
```

und betrachten dann

```
case (MkPair 0) 1 of { MkPair x y -> x }
```

Aufgabe 2. Wir betrachten wieder Ausdrücke der Form `if e1 then e2 else e3`.

(a) Geben Sie eine geeignete Typregel dafür an.

(b) Wie im letzten Übungsblatt übersetzen wir den Ausdruck wie folgt in Basissyntax:

```
case e1 of { True -> e2 ; False -> e3 }
```

Geben Sie so weit wie möglich eine Typherleitung für diesen Ausdruck an. Erreichen Sie dabei dieselben Annahmen wie bei Ihrer Regel für If-Then-Else?

Aufgabe 3. Führen Sie jeweils Typinferenz für die folgenden Ausdrücke durch. Notieren Sie das Typgleichungssystem, das Sie am Ende erhalten. Sie müssen dieses nicht lösen.

(a) `\x -> ((+) x) 1`

(b) `case Nil of { Nil -> 0 ; Cons x y -> x }`

Hier nehmen wir an, dass

```
Nil :: List Int
```

```
Cons :: Int -> List Int -> List Int
```