

# Real World Haskell

## *Blatt 3*

Julian Fleischer, Alexander Steen

Mittwoch, den 24.07.2013

### 1: Kinds, Types:

Gegeben die folgenden algebraischen Datentypen (ADTs):

```
1 data Bool = True | False
2 data Maybe a = Nothing | Just a
3 data List a = Cons a (List a) | Nil
4 data Tree a = Branch (Tree a) a (Tree a) | Empty
5 data Rose a = Node a [Rose a]
6 data Either a b = Left a | Right b
7 newtype Parser s u t a
8 newtype St s a = S (s -> (a, s))
```

Schauen Sie nach im folgenden unbekanntenen Funktionen und Typen bitte in der Hoogle-Suchmaschine!

Was ist der kind der folgenden Typen?

Ein Beispiel: `Rose` hat kind `* -> *`, `Rose Int` hat kind `*`.

1. `Word64`
2. `Maybe`
3. `Maybe String`
4. `Tree`
5. `Either Int32 [Char]`
6. `Either (Maybe Int)`
7. `Either`
8. `Parser`
9. `Parser String [(String, String)]`
10. `St (Map k v) Int32`

Welchen Typ haben die folgenden Ausdrücke?

Ein Beispiel: `Nil` hat den Typ `List a`, `Cons` hat den Typ `a -> List a -> List a`.

1. Just 'c'
2. Nothing
3. Left "Hooray"
4. Node 7 []
5. Node 7
6. Node
7. Cons True
8. \x -> Just x
9. ()
10. S (\s -> (s, s))
11. S
12. (\a b -> (+a))
13. (\[a] -> a) [1]
14. return 7
15. \x -> if x == 0 then Nothing else return x
16. maybe
17. either

Tipp: Sollten Sie keine Ahnung haben können Sie mit Hilfe von `:t` und `:k` die Kinds und Typen der obigen Ausdrücke checken.

## 2: Netzwerk

Bauen Sie einen kleinen Echo-Server, den sie mit Netcat (`nc`) ansprechen können. Der Ping-Server soll auf eingehende Verbindungen warten, eine Zeile lesen, und diese wieder zurück an den Nutzer schicken. Danach kann die Verbindung beendet werden, er kann aber auch gerne weitere Zeilen abwarten und zurückschicken (das bleibt ihnen überlassen).

Schreiben Sie ihren Ping-Server von Anfang an als Cabal-Projekt (siehe Slides der ersten Vorlesung), so dass sie ihn mittels `cabal configure` und `cabal build` bauen und starten können.

Netcat ist unter Unix/Linux/OS X normalerweise immer verfügbar, aber auch für Windows<sup>1</sup>. Orientieren Sie sich an dem Webserver, der in der ersten Vorlesung vorgestellt wurde und machen Sie sich mit dem `network` package<sup>2</sup> vertraut (Sie benötigen nur das Modul `Network`).

<sup>1</sup><http://joncraton.org/blog/46/netcat-for-windows>

<sup>2</sup><http://hackage.haskell.org/package/network>

### 3: Grafikprogrammierung mit Gloss

**(Installation)** Installieren Sie gloss mit Hilfe von `cabal update` gefolgt von `cabal install gloss` und `cabal install gloss-examples`.

**(Einfache Formen)** Zeichnen Sie ein Kreis, Rechteck (or whatever) mit Hilfe der `display` Funktion.

**(Animation)** Lassen Sie ihre Form aus der vorigen Aufgabe durch die Gegend springen oder gleiten (die Bewegung ist ihnen überlassen, Sie können Dinge im Kreis fliegen lassen, auf und ab hüpfen, etc.). Bauen Sie mindestens 3 verschiedene Animationen.