

Applikative Funktoren

Funktionale Programmierung

Prof. Dr. Oliver Braun

Letzte Änderung: 09.10.2018 06:54

- Erinnerung:

```
class Functor (f :: * → *) where  
  fmap :: (a → b) → f a → f b
```

- Problem: kann nicht mit Funktionen mit mehreren Argumenten verwendet werden
- Beispiel:

```
Prelude> :t fmap (+) (Just 1)  
fmap (+) (Just 1) :: Num a ⇒ Maybe (a → a)
```

- also nicht **applikativ**

Ein applikativer Funktor

- applikativer Funktor

```
class Functor f  $\Rightarrow$  Applicative f where  
  pure :: a  $\rightarrow$  f a  
  (<*>) :: f (a  $\rightarrow$  b)  $\rightarrow$  f a  $\rightarrow$  f b
```

- Beispiel:

```
Prelude> pure (+) <*> Just 1 <*> Just 3  
Just 4
```

- um das noch schöner schreiben zu können, ist definiert

```
(<$>) = fmap
```

- damit:

```
(+) <$> Just 1 <*> Just 3
```

```
instance Applicative Maybe where
```

```
  pure = Just
```

```
  Nothing <*> _ = Nothing
```

```
  _ <*> Nothing = Nothing
```

```
  Just f <*> Just a = Just (f a)
```

Beispiel: Liste als applikativer Funktor

- als Funktor: Anwendung **einer** Funktion auf alle Elemente einer Liste

```
Prelude> fmap (2^) [1, 2, 3]
```

```
[2,4,8]
```

```
Prelude> fmap (^2) [1, 2, 3]
```

```
[1,4,9]
```

- als applikativer Funktor: Anwendung **aller** Funktionen auf **alle** Elemente einer Liste

```
Prelude> [(+1), (*2)] <*> [1..3]
```

```
[2,3,4,2,4,6]
```

```
Prelude> [(+), (*)] <*> [1..3] <*> [5,6]
```

```
[6,7,7,8,8,9,5,6,10,12,15,18]
```

```
Prelude> concat [[x+y, x*y] | x<-[1..3], y<-[5,6]]
```

```
[6,5,7,6,7,10,8,12,8,15,9,18]
```

Gesetze für applikative Funktoren

- Identität

$$\text{pure id} \langle * \rangle v \equiv v$$

- Komposition

$$\begin{aligned} \text{pure } (.) \langle * \rangle u \langle * \rangle v \langle * \rangle w &\equiv \\ u \langle * \rangle (v \langle * \rangle w) & \end{aligned}$$

- Homomorphismus

$$\text{pure } f \langle * \rangle \text{pure } x \equiv \text{pure } (f x)$$

- (spezielle Art von) Kommutativität

$$u \langle * \rangle \text{pure } y \equiv \text{pure } (\$ y) \langle * \rangle u$$