Programmier-Paradigmen

Tutorium – Gruppe 2 & 8

Henning Dieterichs

Besprechung der Aufgaben

1. Typen und Typklassen in Haskell

```
fun1 xs = (xs == [])
    :: (Eq t) => [t] -> Bool

fun2 f a = foldr f "a"
    :: Foldable t => (a -> String -> String) -> r -> t a -> String

fun3 f a xs c = foldl f a xs c
    :: Foldable t =>  ((r -> s) -> a -> (r -> s)) -> (r -> s) -> t a -> (r -> s)

fun4 f xs = map f xs xs
    :: (untypisierbar)
```

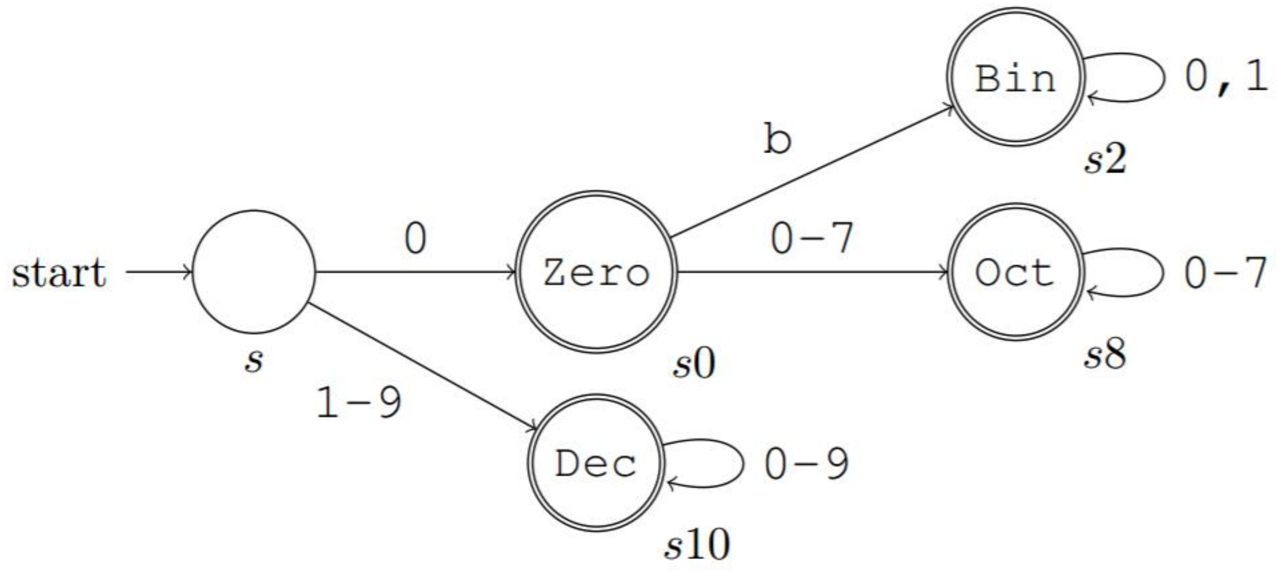
1. Typen und Typklassen in Haskell

```
fun5 a b c = (maximum [a..b], 3 * c)
  :: (Enum t, Ord t, Num n) =>  t -> t -> n -> (t, n)

fun6 x y = succ (toEnum (last [fromEnum x..fromEnum y]))
  :: (Enum a, Enum b, Enum c) => a -> b -> c

fun7 x = if show x /= [] then x else error ""
  :: (Show (String -> a)) => (String -> a) -> (String -> a)
```

Zustandsautomat



Monaden

- http://funktionale-programmierung.de/2013/04/18/haskell-monaden.html
- http://adit.io/posts/2013-04-17-functors,_applicatives,_and_monads_in_pictures.html