1 Forelesning 9

1.1 Typeklasser

Typeklassene ligner på "interface" i Java. Typeklasser er mer fleksible:

- Et Java interface må defineres før klassen.
- En typeklasse kan defineres når som helst.

Idea. Noen funknsjoner fungerer for alle typer (polymorfe), men noen fungerer bare på typer med en viss struktur. For eksempel likhet, rodning, eller + og −.

En typeklasse bestemmer typen til en liste med funksjoner. En hver type som kører til typeklassen skriver in egen implementasjon.

Eksempel. Klassen `Eq` er den enkleste innebygde klassen:

```
class Eq aa where
    (==) :: a -> a -> Bool
```

For eksempel Bool:

```
data Bool = True | False

instance Eq Bool where
    instance Eq Bool where
    True == True = True
    False == False = True
    _ == _ = False
```

For eksempel for lister:

```
instance (Eq a) => Eq [a] where
[] == [] = True
(a:as) == (b:bs) = (a == b) && (as == bs)
_ == _ = False
```

◇

Eksempel. Eksempel på rotering av lister

```
class Rotateable a where
    rotateLeft :: a -> a
    rotateRight :: a -> a

instance Rotateable [b] where
    rotateLeft [] = []
    rotateLeft (a:as) = as ++ [a]
    rotateLeft [] = []
    rotateLeft as = last as : init as
```

◇

1.2 Binære søkretrær

1.2.1 Antagelse

Vi antar at vi jobber med en type som implementerer Ord typen. (F. eks heltall, eller strenger)

1.2.2 Basis ide

Binære søketrær er basert på samme ide som QuickSort: Hvis vi har et element x kan vi dele opp en hver mengde i tre deler:

- De som er mindre enn x (til venstre)
- Elementet x selv (i midten)
- De som er større enn x (til høyre)