EDAN40: Functional Programming
Assignment 1: Chatterbots

Jacek Malec
Dept. of Computer Science, Lund University, Sweden
April 8th, 2019

Programming partner search

- Raise your hands
- Talk during the break
- Use the sign-up facility
- On Wednesday I will try to pair the lone individuals via the sign-up system.

folds and infinite arguments

foldl :: (a -> b -> a) -> a -> [b] -> a
foldl f z [] = z
foldl f z (x:xs) = foldl f (f z x) xs

foldr :: (a -> b -> b) -> b -> [a] -> b
foldr f z [] = z
foldr f z (x:xs) = f x (foldr f z xs)
Consider folding a list of $n$ values $[x_1, x_2, x_3, x_4 \ldots x_n]$ with some function $f$ and seed $z$.

foldl is:
- Left associative: $f(\ldots(f(f(z,m_1,m_2),m_2,m_3),m_2,m_3,\ldots,m_n))$
- Tail recursive: It iterates through the list, producing the value afterwards
- Lazy: Nothing is evaluated until the result is needed
- Backwards: foldl (flip (:)) [] reverses a list.

foldr is:
- Right associative: $f x_1 (f x_2 (f x_3 (f x_4 \ldots (f x_n z)\ldots)))$
- Recursive into an argument: Each iteration applies $f$ to the next value and the result of folding the rest of the list.
- Lazy: Nothing is evaluated until the result is needed
- Forwards: foldr (:) [] returns a list unchanged.

A fragment of Chatterbot.hs:

```hs
stateOfMind :: BotBrain -> IO (Phrase -> Phrase)
{-# TO BE WRITTEN #-}
stateOfMind _ = return id

return is of type:

return :: a -> m a

while in System.Random you have

randomIO :: IO n

where n is an (almost) arbitrary numeric type.

A chatterbot is a program that attempts to simulate typed conversation, with the aim of at least temporarily fooling a human into thinking they are talking to another person.

- Eliza (Joseph Weizenbaum, 1966), Analiza:-), M-x doctor
- Turing test
- Loebner prize
- Elbot (www.elbot.com) Artificial Solutions, SAS, IKEA
- Cleverbot (www.cleverbot.com)
The task (N1)

- Read the existing code and make sure you understand it;
- Write your code in places marked by \{- TO BE WRITTEN -\};
- When you get Eliza running, polish your code to get point-free style as much as possible. You will be helped by the TAs;
- Use \texttt{HLint} for suggestions how to improve it;
- Work in pairs. Ask others if you need. Ask TAs. Ask me. \textbf{Don't copy!} Do learn.
- Enjoy.

Jacek Malic, http://rss.cs.lth.se

The task (F1)

- Read the task description and make sure you understand it. What do you need to verify?
- Build up your code bottom-up to get the desired functionality.
- Polish your code to get point-free style as much as possible. You will be helped by the LA and TAs;
- Use \texttt{HLint} for suggestions how to improve it;
- Work in pairs. Ask others if you need. Ask TAs. Ask me. \textbf{Don't copy!} Do learn.
- Enjoy.

Jacek Malic, http://rss.cs.lth.se