



EDAF95/EDAN40: Functional Programming

Assignment 1F: Verify Sudoku

Assignment 1N: Chatterbots

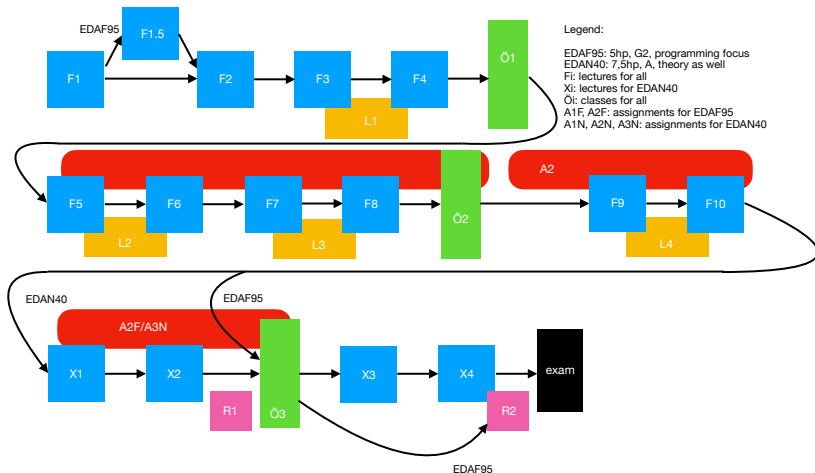
Jacek Malec

Dept. of Computer Science, Lund University, Sweden

April 3rd, 2023



The course(s)





Programming partner search

1 EDAN40:

- Go to canvas
- If you are alone, register into an assignment group as first member, then
- Wait for contact, announce it on discord, or
- Contact someone else registered alone in some other group.
- You should be able to move yourself between groups.

2 EDAF95

- work in lab groups



foldl and infinite arguments

`foldl` $:: (a \rightarrow b \rightarrow a) \rightarrow a \rightarrow [b] \rightarrow a$

`foldl f z []` = `z`

`foldl f z (x:xs)` = `foldl f (f z x) xs`

`foldr` $:: (a \rightarrow b \rightarrow b) \rightarrow b \rightarrow [a] \rightarrow b$

`foldr f z []` = `z`

`foldr f z (x:xs)` = `f x (foldr f z xs)`



fold and infinite arguments

Consider folding a list of n values $[x_1, x_2, x_3, x_4 \dots x_n]$ with some function f and seed z .

`foldl` is:

- 1 Left associative:
 $f (\dots (f (f (f (f z x_1) x_2) x_3) x_4) \dots) x_n$
- 2 Tail recursive: It iterates through the list, producing the value afterwards
- 3 Lazy: Nothing is evaluated until the result is needed
- 4 Backwards: `foldl (flip (:)) []` reverses a list.



foldr and infinite arguments

Consider folding a list of n values $[x_1, x_2, x_3, x_4 \dots x_n]$ with some function f and seed z .

`foldr` is:

- 1 Right associative:
`f x1 (f x2 (f x3 (f x4 ... (f xn z)...)))`
- 2 Recursive into an argument: Each iteration applies f to the next value and the result of folding the rest of the list.
- 3 Lazy: Nothing is evaluated until the result is needed
- 4 Forwards: `foldr (:) []` returns a list unchanged.



IO and Random

A fragment of `Chatterbot.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.



Sudoku (F1)

2	6		3					
5						7		
					1		4	
6			5			2		
		4			8			1
	5		9					
		7						3
					4		1	6



The task (F1)

All solutions are easily found on the net. Please don't use them.

Learn!

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



Chatterbots (N1)

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 (not any longer)
- Elbot (www.elbot.com) Artificial Solutions, SAS, IKEA
- Cleverbot (www.cleverbot.com), fun
- 85% of the customer support nowadays is done without human intervention!




But be careful!

chat.openai.com/chat

educational Software SynBio Semantic Web stuff EU Admin/Ledning Banki Wydatki Malców Meeting agen...Google Docs News Travel jacek https://mina...

chat DALL... GPT L... Tesla'... Civilia... AI can... Weap... A Cha... Part... The D...

JA explain why is multiplication not commutative

 Multiplication is not commutative because the order of the numbers being multiplied can change the result. For example, 2×3 is not the same as 3×2 , the former equals 6 and the latter is 9. Commutativity applies only to operations where the order of the numbers doesn't change the outcome, like addition. ■



The task (N1)

All solutions are easily found on the net. Please don't use them.

Learn!

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