

# Lena & neučakana evalvacija

$f(e)$

neučakana:

najprej izračunamo  $e \rightarrow v$   
in nato izvedemo klic  $f(v)$

Lena:

tahoj izvedemo  $f(e)$

Primer:

$(\lambda x. (\lambda y. 7 + y) 3) (2 + 2)$

• neučakana

$(\lambda x. (\lambda y. 7 + y) 3) (2 + 2)$

$(\lambda x. (\lambda y. 7 + y) 3) 4$

$(\lambda y. 7 + y) 3$

$7 + 3$

10

• Lena

$(\lambda x. (\lambda y. 7 + y) 3) (2 + 2)$

$(\lambda y. 7 + y) 3$

$7 + 3$

10

Primer:  $(\lambda x. x+x) (2+2)$

lena:

$(\lambda x. x+x) (2+2)$

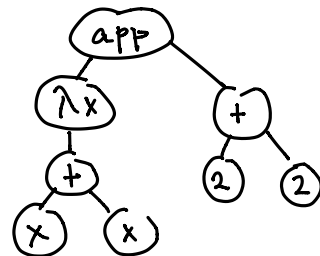
$(2+2) + (2+2)$

4 + 4

8

izboljšava  
call-by-need

$(\lambda x. x+x) (2+2)$



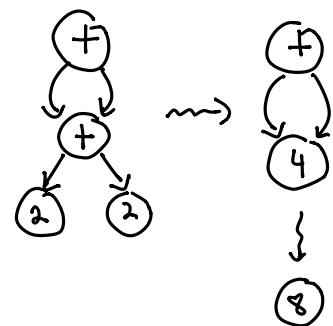
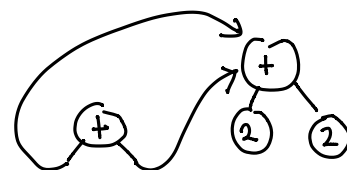
neučakano:

$(\lambda x. x+x) (2+2)$

$(\lambda x. x+x) \quad 4$

4 + 4

8

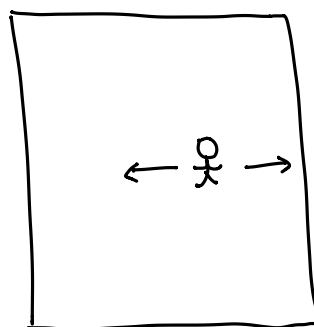
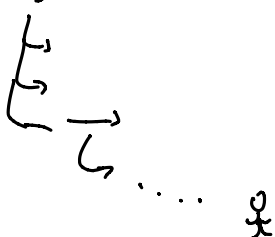


Len programski jezik:

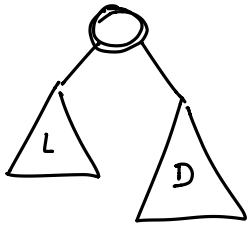
uporabi leno evalvacijo, oz. call-by-need

Haskell, Elm

HTML  
↳ HEAD  
↳ BODY



Spletna stran  
HTML



Odmor do 9:20

# Razredi tipov (Type classes)

Razred tipov (specifikacija, vmesnik)

Primerki (implementacije)

Razred tipov — tipi, ki imajo neko dodatno skupno funkcionalnost, kot jo opisuje razred

Primer: tipi, katerih vrednosti imajo velikost

Bool	1
Int	64
[Int]	64 x dolžina seznamu

v C je to sizeof

Class ImeRazreda a where

foo :: ----

bar :: ---

opišemo, kaj vse mora imeti tipa, da ga štejemo za primerele razreda ImeRazreda

instance ImeRazred Bool where

foo = ----

bar = ----

Bool opreunimo s funkcionalnostjo razreda ImeRazreda

# Monoid

Množica

$M$

Element

$e \in M$

Operacija

$*$

velja:  $(x * y) * z = x * (y * z)$

$e * x = x$

$x * e = x$

Primeri:

$(\mathbb{N}, 0, +)$

$(\mathbb{N}, 1, \times)$

$(\mathbb{N}, 0, \max)$

$(\mathbb{R}^{n \times n}, [1 \dots 1], \text{množenje matrik})$

↑  
matrice velikosti  $n \times n$

plus [1,2,3] = 1+2+3

$(A \rightarrow A, (\lambda x. x), 0)$

↑  
kompozicija

$(\text{List } \mathbb{N}, [], ++)$

↳ stikanje seznamov

Functor : ideja

$f a$  "struktura, ki vsebuje a-je"

$$h: a \rightarrow b$$

$$fmap h: f a \rightarrow f b$$

$$map: (a \rightarrow b) \rightarrow [a] \rightarrow [b]$$

$$map f [x_1, \dots, x_n] = [f x_1, \dots, f x_n]$$

Primer:

Maybe a =  
Nothing  
| Just a

$$h: a \rightarrow b$$

$$fmap h: Maybe a \rightarrow Maybe b$$

$$\text{Nothing} \mapsto \text{Nothing}$$

$$\text{Just } x \mapsto \text{Just } (h x)$$

## Razred Applicative

Pogosto imamo podatkovni tip, ki "hrani" podatke z dodatno lastnostjo, da lahko shranimo hatanho en "čisti" podatek.

Primer: Seznam

$$[x_1, \dots, x_n]$$

hranimo n-podatkov

$$[x]$$

"čisti" podatek

Primer: Maybe

Nothing

Just x      "čisti"

Primer: program, ki dela I/O

"čist" je program, ki ne uporabi I/O

Applicative:

pure : a → fa      naredi "čisti" podatke

$h \langle * \rangle x =$  nečist rezultat  
↓                 ↓  
nečista          nečist  
funkcija        podatek

ODMOR DO 10:30