Rekurzija

g g

```
let f = rek (fun self -> (fun n -> if n = 0 then 1 else n * self (n - 1)))
```

$$((\alpha \to \beta) \to (\alpha \to \beta)) \to (\alpha \to \beta)$$

domena — kodomena

$t := \alpha \to \beta$

$(t \to t) \to t$

h

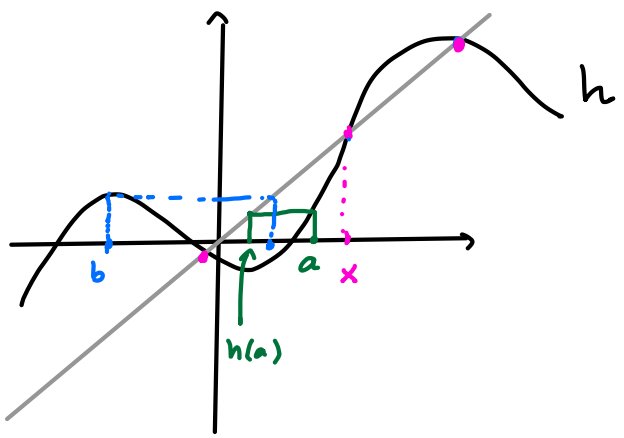
rek h = h (rek h)

Negibna točka:

$h : \mathbb{R} \to \mathbb{R}$

Negibna točka za h je tak $x \in \mathbb{R}$, da

$h(x) = x$



x_0

$x_1 = h(x_0)$

$x_2 = h(x_1) = h(h(x_0))$

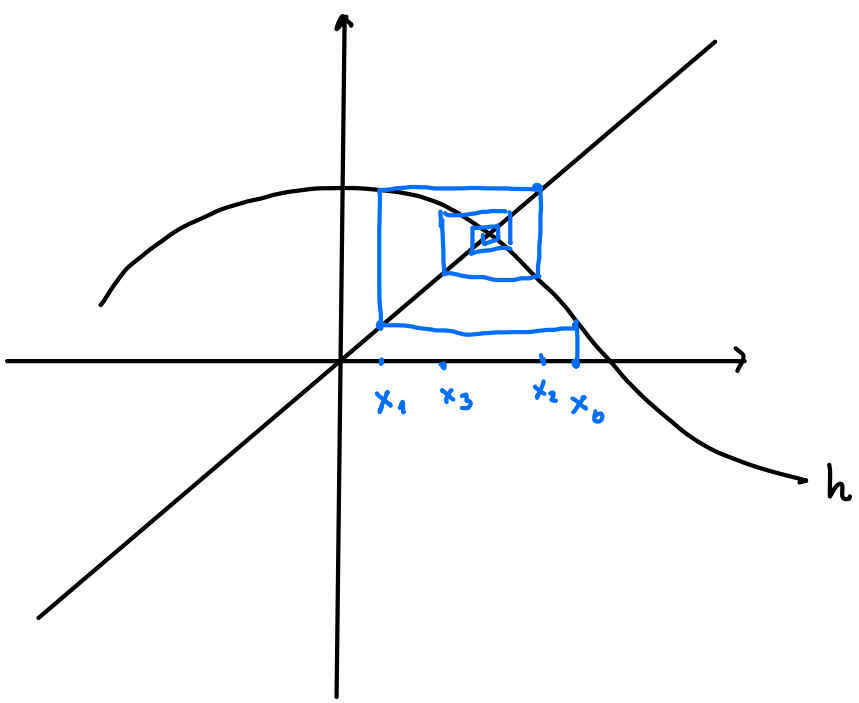
$x_3 = h(x_2) = h(h(h(x_0)))$

$\vdots$

$x_{n+1} = h(x_n)$

Negibna: $\lim_{n \to \infty} x_n$

$h(x) = x$



$$l(x) = d(x)$$

$$l(x) + x = d(x) + x$$

$$\underbrace{l(x) + x - d(x)}_{h(x)} = x$$

$$h(x) = x$$

REKURZIJA = NEGIBNA TOČKA

$$f = \underbrace{\lambda n.\ \text{if } n = 0 \text{ then } 1 \text{ else } n \cdot f(n-1)}_{h(f)}$$

$$f = h(f)$$

$$h = \lambda g.\ \lambda n.\ \text{if } n = 0 \text{ then } 1 \text{ else } n \cdot f(n-1)$$

Hkratna rekurzija:

```
(f, g) = ((λ x . if x = 0 then 1 + f (x - 1) else 2 + g (x - 1)),
          (λ y . if y = 0 then 1 else 3 * f (y - 1)))
```

```
t = λ (f', g') . ((λ x . if x = 0 then 1 + f' (x - 1) else 2 + g' (x - 1)),
                  (λ y . if y = 0 then 1 else 3 * f' (y - 1)))
```

$$(f, g) = t\,(f, g)$$

Rekurzivne podatkovne strukture

Seznam:

Rekurzivna definicija podatkovne strukture

- prazen

ALI

- sestavljen iz glave in repa, ki je seznam

Ocaml:

[] prazen seznam

x :: l

↑ glava, ↖ rep

[x₁; x₂; x₃; ...] =

x₁ :: (x₂ :: (x₃ :: ... [])...)

Rekurzivna def. posameznega seznama:

$$l = 1 :: l$$

$$= [1; 1; 1; 1; \ldots]$$

LISP:

prazen seznam: nil

sestavljen (cons x l)

TEORIJA MNOŽIC

$$Seznam = \{Nil\} \cup \{Cons(x, l) \mid x \in \mathbb{Z}, l \in Seznam\}$$

↑ množica vseh seznamov

Cons(1, Cons(2, Cons(3, Nil)))

$$Seznam = \mathbb{1} + \mathbb{Z} \times Seznam$$

$\iota_1()$ $\iota_2(x, l)$

$\iota_2(1, \iota_2(2, \iota_2(3, \iota_1())))$

PODATKOVNI TIP

```
type seznam =
  | Nil
  | Cons of int * seznam
```

Seznam = prazen ali sestavljen

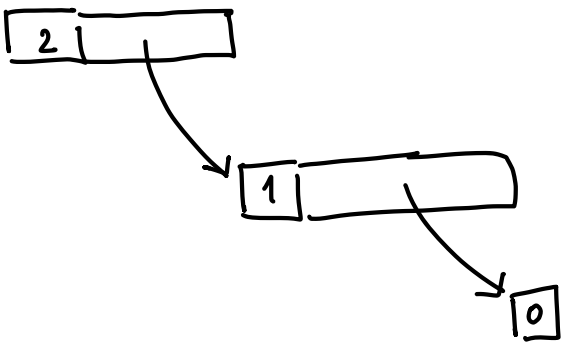
$$\mathbb{N} = \{0\} \cup \{ succ(n) \mid n \in \mathbb{N} \}$$

$\infty = succ(\infty)$ PREPOVEDANO ZARADI PRINCIPA INDUKCIJE

$1010_2 = 8+2=10$

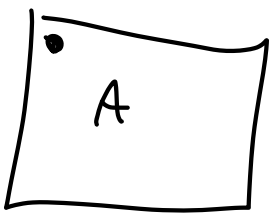
0
00
01
000
001
010
011
0000
0001

Shl1 (Shl0 zero)



if x < y

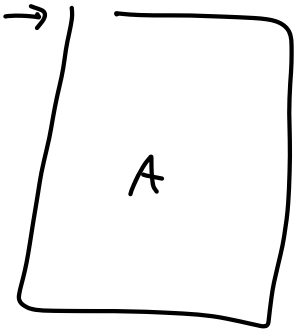
then

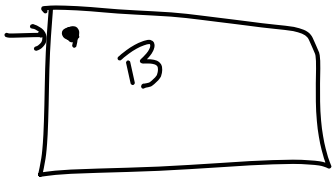


else

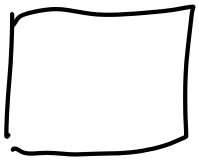
match Compare x y with

| Less →





| NotLess →



boolean blindness

spol : bool ;

type spol =
| Muski
| Zenski
| Other of string

9