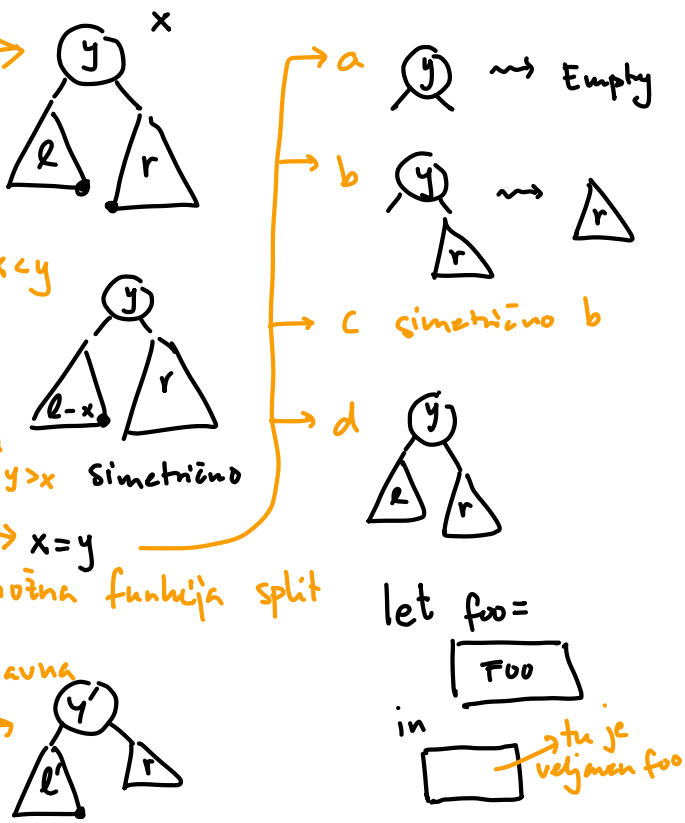
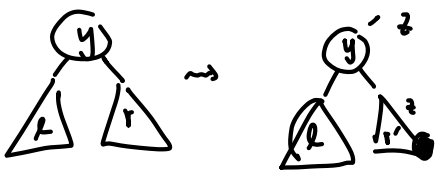
```
(* Odstrani element iz iskalnega drevesa *)
let rec remove x = function
  | Empty -> Empty

  | Node (y, l, r) ->
     begin
       match cmp x y with
       | Less -> Node (y, remove x l, r)
       | Greater -> Node (y, l, remove x r)
       | Equal ->
        begin match (l, r) with
      a | (Empty, Empty) -> Empty
      b | (Empty, _) -> r
      c | (_, Empty) -> l
      d | (_, _) ->
            let rec split = function
          i | Empty -> failwith "impossible"
          ii| Node (y, l, Empty) -> (l, y)
          iii| Node (y, l, r) ->
                let (r', z) = split r in
                (Node (y, l, r'), z)
            in
            let (l', y') = split l in
            Node (y', l', r)
        end
     end
```



split:

i) x ii)  iii) 

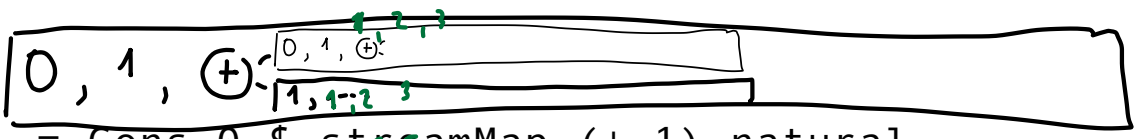
Koinduktivni tipi

končni in neskončni podatki

Matematika	OCaml	Haskell
$A \times B$	a * b	(a,b) ← tip
(u, v)	(u,v)	(u,v) ← element
{*}	unit	() ← tip
*	()	() ← element
$e \in T$	e : T	e :: T
	x :: l	x : l
	type	data
	type	newtype

type
'a list

type
[a] ← tip




```
natural = Cons 0 $ streamMap (+ 1) natural
```

thunk ("zavlačevanje")

A
x

a

unit → a

1 → A

```
fibonacci = Cons 0 $ Cons 1 $ streamZip (+) fibonacci (rest fibonacci)
```

$\sum_{n=0}^{\infty} a_n \cdot x^n$

$\sum_{n=0}^{\infty} \frac{x^n}{n!} = e^x$

$\sum_{k=0}^{\infty} a_k x^k$

$S = a_0, a_1, a_2, a_3, \ldots a_n, \ldots$

```
fun x -> x + 3
```

x

$\sum_{k=0}^{n} a_k x^k$ = eval n x s

Izpeljava tipov

42 int, float

$\lambda x.x$ bool → bool
int → int

$\beta \to \beta$ β poljuben
↖ parameter

Primeri

```
fun x -> x + 3
```

$x : \alpha$

α int
int

α = int
int = int

$\alpha \to$ int
int → int

Rešimo:

α = int ✓ $\alpha \mapsto$ int

~~int = int~~

Primer:

```
if 3 < 5 then (fun x -> x) else (fun y -> y + 3)
```

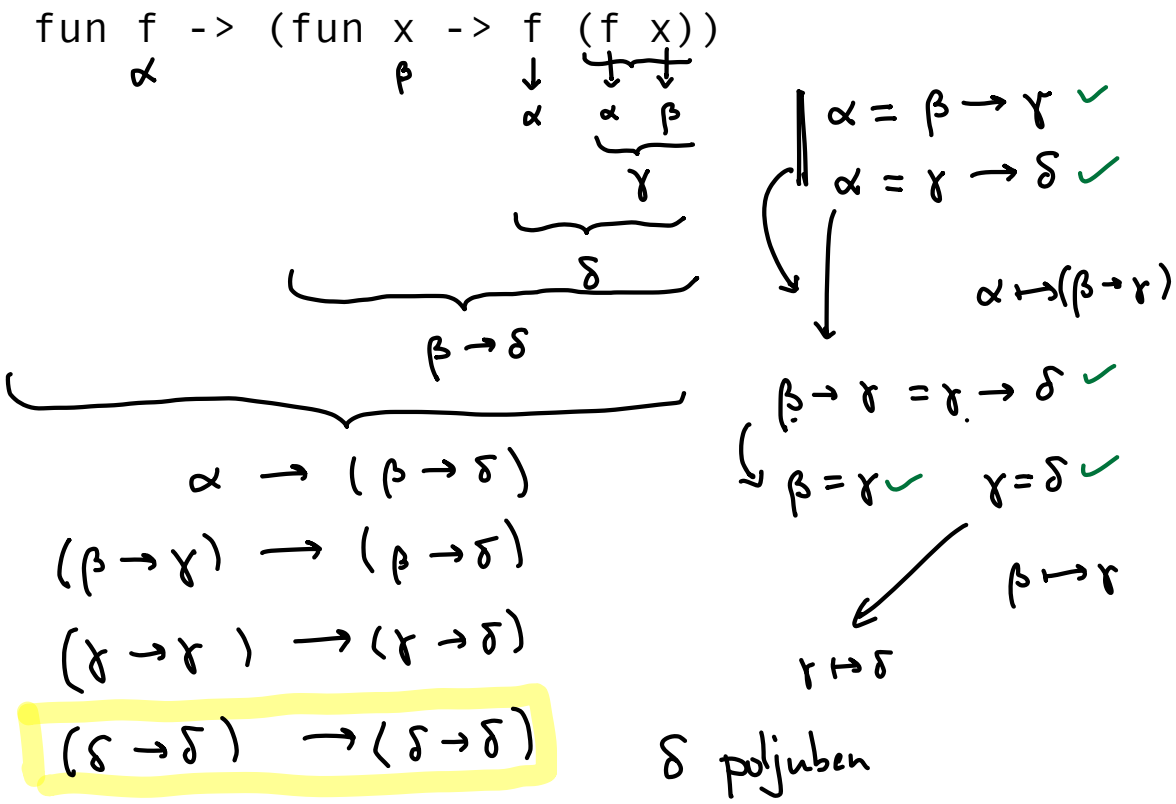
bool ✓

α α
$\alpha \to \alpha$

β=int int ✓
int → int

$\alpha \to \alpha$
int → int

pri $\alpha \to \alpha$ = int → int
⇝ α = int $\alpha \mapsto$ int
α = int


fun f -> (fun x -> f (f x))
α
β
α
α β
γ
δ
β → δ
α → (β → δ)
(β → γ) → (β → δ)
(γ → γ) → (γ → δ)
(δ → δ) → (δ → δ)
α = β → γ ✓
α = γ → δ ✓
α ↦ (β → γ)
β → γ = γ → δ ✓
β = γ ✓
γ = δ ✓
β ↦ γ
γ ↦ δ
δ poljuben