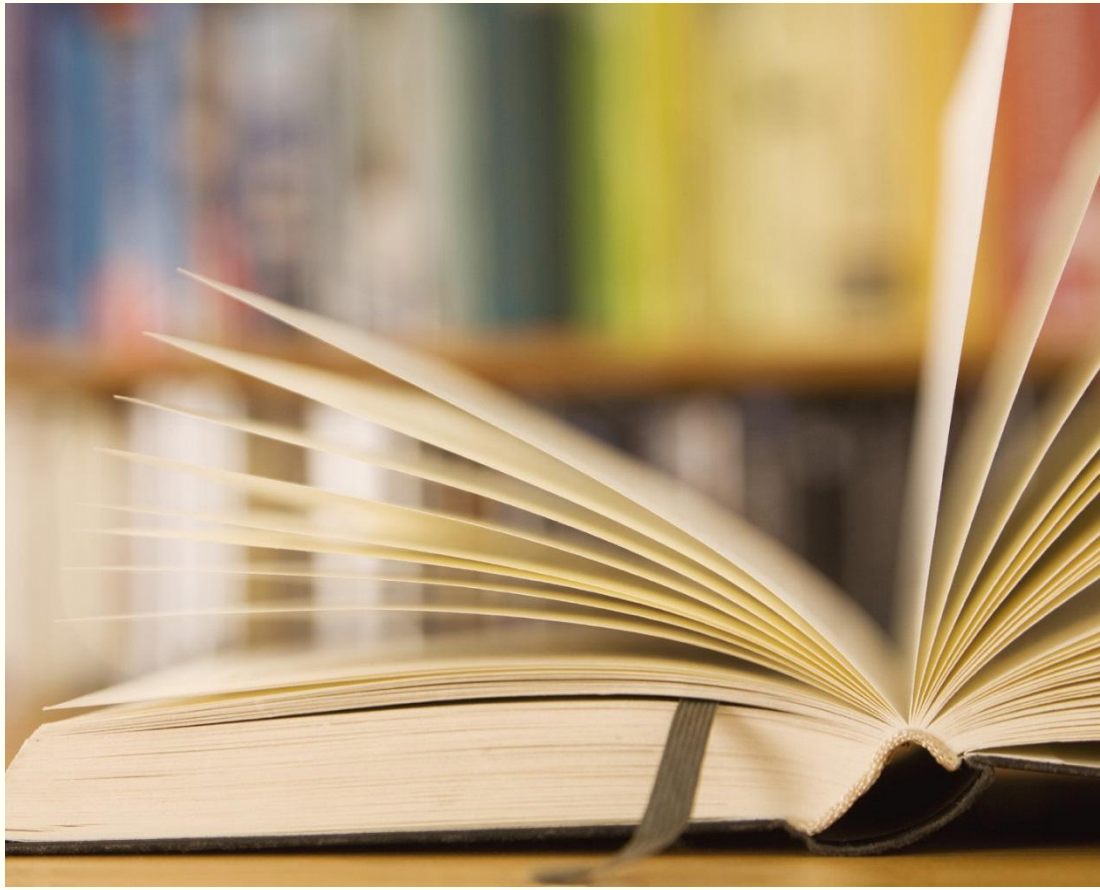
# FUNKCIONALNO PROGRAMIRANJE

(MATERIJALI PREUZETI OD DOC. DR MILANE GRBIĆ)

Lambda račun



# LAMBDA RAČUN

# Lambda račun

- Funkcije u matematici imaju imena.

$$\text{Neka je } f : \mathbb{R} \to \mathbb{R} \text{ definisana sa:}$$
$$f(x) = 0, \text{ ako je } x = 0$$
$$f(x) = x^2 \sin(1/x^2), \text{ ako je } x \neq 0$$

- Funkcije u programskim jezicima imaju imena.

```
int suc(int n)
 { return n + 1;
 }
```

# Lambda račun

- Lambda račun možemo da shvatimo kao račun anonimnih (neimenovanih) funkcija.

$$x \mapsto t[x] \qquad \lambda x.\, t[x]$$

  - Metod za predstavljanje funkcija;
  - Skup pravila za sintaksnu transformaciju;

- Istorijat lambda računa:
  - Alonzo Čerč (Alonzo Church), početkom 1930-ih;
  - 1932/1933. – prva verzija (deo logičkog sistema; pokazalo se da je protivrečna);
  - 1936. – lambda račun kao model izračunavanja; 1941. – knjiga „The Calculi of Lambda-Conversion”;

# Zadaci

1. U lambda notaciji zapisati funkciju identiteta.
2. U lambda notaciji napisati funkciju koja vraća dvostruku vrednost svog argumenta.

# Rešenje zadataka 1 i 2

- Funkcija identiteta:

$\lambda x.\ x$

- Dvostruka vrednost argumenta:

$\lambda x.\ * 2\ x \qquad (\text{ili } \lambda x.\ +\ x\ x)$

- Primena = zamena parametra argumentom:

$(\lambda x.\ x)\ 5\ =\ 5$

$(\lambda x.\ * 2\ x)\ 7\ =\ * 2\ 7\ =\ 14$

- Formalno pravilo (β-redukcija) sledi u delu „Izračunavanje vrednosti”

# Prednosti upotrebe lambda računa

- f(x) zapisujemo kao f x;
- f x y znači (f x) y, tj. (f(x))(y) – primena je levo asocijativna;
- λx.λy. t[x,y] skraćeno pišemo λx y. t[x,y];
- λx. x y znači λx. (x y) – telo apstrakcije se proteže što dalje udesno;

$$(\lambda x\ y.\ x + y)\ 1\ 2 = (\lambda y.\ 1 + y)\ 2 = 1 + 2$$

- Vezane promenljive: m u prvom, y u drugom i n u trećem izrazu; slobodne: n u prvom, x i a u drugom izrazu;

$$\sum_{m=1}^{n} m = \frac{n(n+1)}{2}$$

$$\int_0^x 2y + a\ dy = x^2 + ax$$

```
int suc(int n)
 { return n + 1;
 }
```

# Primer: funkcija sa dva argumenta

$(\lambda x\ y.\ x + y)\ 1\ 2$

$=\ ((\lambda x.\lambda y.\ x + y)\ 1)\ 2$ [skraćenica λx y; primena je levo asocijativna]

$\rightarrow_\beta\ (\lambda y.\ 1 + y)\ 2$ [x := 1]

$\rightarrow_\beta\ 1 + 2$ [y := 2]

$\rightarrow_\delta\ 3$ [ugrađeno sabiranje]

- $\rightarrow_\beta$: primena funkcije na argument (detaljno kasnije)
- $\rightarrow_\delta$: računanje ugrađene operacije (npr. sabiranja)
- U tri primera sa prethodnog slajda:
- Σ m od m = 1 do n: m je vezana, n slobodna
- $\int_0^x (2y + a)\ dy$: y je vezana, x i a su slobodne
- int suc(int n) { return n + 1; }: n je vezana (parametar)

# Prednosti upotrebe lambda računa

- Promena imena vezane promenljive (alfa konverzija);

$$\int_0^x 2y + a \; dy = x^2 + ax$$

$$\int_0^x 2z + a \; dz = x^2 + ax$$

$$\lambda x.\ E[x]$$

$$\lambda y.\ E[y]$$

$$\int_0^x 2a + a \; da \neq x^2 + ax$$

- Ista imena slobodnih i vezanih promenljivih;

$$\int_0^x 2x + a \; dx = x^2 + ax$$

# Primer: preimenovanje vezane promenljive

$\lambda x.\ +\ x\ a\ \equiv_\alpha\ \lambda y.\ +\ y\ a$ [x preimenovano u y; y se ne javlja u telu]

$\lambda x.\ +\ x\ a\ \not\equiv_\alpha\ \lambda a.\ +\ a\ a$ [a je slobodna u telu – bila bi „uhvaćena”]

- Isto kao kod integrala: $\int_0^x (2y + a)\, dy = \int_0^x (2z + a)\, dz$,
- ali $\int_0^x (2a + a)\, da = 3x^2/2 \neq x^2 + ax$
- Pravilo: novo ime ne sme da se javlja slobodno u telu apstrakcije

# Lambda račun kao formalni sistem

- Lambda izrazi (termovi):
  - Promenljive;
  - Konstante;
  - Aplikacije (primene) – primena funkcije s na argument t; s i t mogu biti proizvoljni lambda izrazi. Zapis: s t.
  - Apstrakcija – lambda izraza s po promenljivoj x, u oznaci λx. s.
- Induktivna definicija lambda izraza.
  - Funkcije nad lambda izrazima se definišu rekurzivno.
- Sintaksa lambda izraza u BNF (Backus-Naurovoj formi):

$$\text{Exp} ::= \text{Var} \mid \text{Const} \mid \text{Exp Exp} \mid \lambda\text{Var. Exp}$$

# Primer: kako se čita lambda izraz

- Konvencije zapisa:

$f\ x\ y \equiv (f\ x)\ y$ [primena je levo asocijativna]

$\lambda x.\ x\ y \equiv \lambda x.\ (x\ y)$ [telo ide što dalje udesno]

$\lambda x\ y.\ M \equiv \lambda x.\lambda y.\ M$ [skraćenica za više argumenata]

- Primeri:

$\lambda x\ y.\ x\ y\ z \equiv \lambda x.(\lambda y.((x\ y)\ z))$

$(\lambda x.\ x)\ y\ z \equiv ((\lambda x.\ x)\ y)\ z$

- $\lambda x.\ x\ y$ (apstrakcija) i $(\lambda x.\ x)\ y$ (primena) su različiti izrazi

# Zadaci

3. Da li su sledeći izrazi lambda izrazi ili ne? Obrazložiti odgovor.

   *a)* $42$
   *b)* $(+\ 6)$
   *c)* $\lambda y.* \ 2 \ y$
   *d)* $\lambda v.v \ x \ y$
   *e)* $(\lambda f.\lambda a.\lambda b.f \ a \ b)(\lambda x.(\lambda y.x))$

# Rešenje zadatka 3

- Svi su lambda izrazi (primenjeni lambda račun, sa konstantama):
- a) 42 – konstanta
- b) (+ 6) – primena konstante + na konstantu 6
- c) λy. * 2 y – apstrakcija; telo je primena (* 2) y
- d) λv. v x y – apstrakcija; telo je (v x) y; x i y su slobodne
- e) primena apstrakcije λf.λa.λb. f a b na apstrakciju λx.(λy. x)
- Vrednost izraza e) (β-redukcija se uvodi kasnije):

(λf.λa.λb. f a b) (λx.λy. x)

$\rightarrow_\beta$ λa.λb. (λx.λy. x) a b [f := λx.λy. x]

$\rightarrow_\beta$ λa.λb. (λy. a) b [x := a]

$\rightarrow_\beta$ λa.λb. a [y := b]

# Slobodne i vezane promenljive

- Pojavljivanje promenljive v u izrazu E je vezano ako se nalazi unutar nekog podizraza od E oblika λv. E1.
- Pojavljivanje koje nije vezano je slobodno (ista promenljiva može imati i slobodna i vezana pojavljivanja).
- Lambda račun se naziva čist ako ne uključuje konstante.
- Ako lambda račun uključuje konstante, onda se naziva primenjen.

# Primer: slobodna i vezana pojavljivanja

(λx. x y) x

- x u telu λx. x y je vezano (za λx)
- y je slobodno – nijedno λy ga ne vezuje
- poslednje x (argument) je slobodno – ista promenljiva, drugo pojavljivanje

λx.λy. x y z

- pojavljivanja x i y su vezana, z je slobodno

λx. y i λy. y

- u prvom je y slobodno, u drugom vezano

# Slobodne i vezane promenljive

- Skup slobodnih promenljivih u izrazu s označavamo sa FV(s) i definišemo rekurzivno na sledeći način:

$$
\begin{aligned}
FV(x) &= \{x\} \\
FV(c) &= \emptyset \\
FV(s\ t) &= FV(s) \cup FV(t) \\
FV(\lambda x.\ s) &= FV(s) - \{x\}
\end{aligned}
$$

# Primer: računanje FV

$FV((\lambda x.\ x\ y)\ x)$

$= FV(\lambda x.\ x\ y) \cup FV(x)$ [pravilo za primenu]

$= (FV(x\ y) - \{x\}) \cup \{x\}$ [pravilo za apstrakciju; $FV(x) = \{x\}$]

$= ((FV(x) \cup FV(y)) - \{x\}) \cup \{x\}$ [pravilo za primenu]

$= (\{x, y\} - \{x\}) \cup \{x\}$ [FV promenljive]

$= \{y\} \cup \{x\} = \{x, y\}$

- $x \in FV$: poslednje pojavljivanje x je slobodno, iako je x u telu vezano

# Slobodne i vezane promenljive

- Skup vezanih promenljivih u izrazu s označavamo sa BV(s) i definišemo rekurzivno na sledeći način:

$$
\begin{aligned}
BV(x) &= \emptyset \\
BV(c) &= \emptyset \\
BV(s\ t) &= BV(s) \cup BV(t) \\
BV(\lambda x.\ s) &= BV(s) \cup \{x\}
\end{aligned}
$$

# Zadaci

4. Za lambda izraz

$$s = (\lambda x\ y.\ x)(\lambda x.\, z\ x)$$

odrediti $FV(s)$ i $BV(s)$.

# Rešenje zadatka 4

$s = (\lambda x.\lambda y.\ x)\ (\lambda x.\ z\ x)$

$FV(s) = FV(\lambda x.\lambda y.\ x) \cup FV(\lambda x.\ z\ x)$ [pravilo za primenu]

$= (FV(\lambda y.\ x) - \{x\}) \cup (FV(z\ x) - \{x\})$ [pravilo za apstrakciju]

$= ((\{x\} - \{y\}) - \{x\}) \cup (\{z, x\} - \{x\})$

$= \emptyset \cup \{z\} = \{z\}$

$BV(s) = BV(\lambda x.\lambda y.\ x) \cup BV(\lambda x.\ z\ x)$ [pravilo za primenu]

$= (BV(\lambda y.\ x) \cup \{x\}) \cup (BV(z\ x) \cup \{x\})$ [pravilo za apstrakciju]

$= ((\emptyset \cup \{y\}) \cup \{x\}) \cup ((\emptyset \cup \emptyset) \cup \{x\})$

$= \{x, y\}$

# Zamena

- Lambda apstrakcija i primena;

$$(\lambda x\ y.\ x + y)\ 1\ 2 = (\lambda y.\ 1 + y)\ 2 = 1 + 2$$

- Sa s[t/x] označavamo zamenu svih slobodnih pojavljivanja promenljive x u izrazu s izrazom t.

$x[t/x] = t$
$y[t/x] = y,$ ako je $x \neq y$
$c[t/x] = c$
$(s1\ s2)[t/x] = s1[t/x]\ s2[t/x]$
$(\lambda x.\ s)[t/x] = \lambda x.\ s$
$(\lambda y.\ s)[t/x] = \lambda y.\ (s[t/x]),$ ako je $x \neq y$ i ($x \notin FV(s)$ ili $y \notin FV(t)$)
$(\lambda y.\ s)[t/x] = \lambda z.\ (s[z/y][t/x]),$ inače; z je novo ime: $z \notin FV(s) \cup FV(t)$

# Primeri zamene

$(x\ y)[N/x] = N\ y$ [x je slobodno – menja se]

$(\lambda x.\ x\ y)[N/x] = \lambda x.\ x\ y$ [x je vezano – ne dira se]

$(\lambda y.\ x\ y)[z/x] = \lambda y.\ z\ y$ [$x \neq y$ i $y \notin FV(z)$ – bez preimenovanja]

$(\lambda y.\ x\ y)[y/x] = \lambda w.\ (x\ w)[y/x]$ [$y \in FV(y)$: prvo $\lambda y \equiv_\alpha \lambda w$]

$= \lambda w.\ y\ w$

- Bez preimenovanja: $\lambda y.\ y\ y$ – slobodno y bi postalo vezano (hvatanje)
- Zašto je bitan uslov sa FV:

$(\lambda y.\ x + y)[y/x]$ („dodaj x argumentu”)

pogrešno: $\lambda y.\ y + y$ [y uhvaćeno – sada „dupliraj argument”]

ispravno: $\lambda z.\ y + z$ [prvo $\lambda y \equiv_\alpha \lambda z$, pa zamena $x := y$]

# Izračunavanje vrednosti lambda izraza

- δ-pravila
  - Primena ugrađenih funkcija na konstante, npr. + 3 4 $\rightarrow_{\delta}$ 7.
- β-redukcija (najvažnija: izračunava vrednost funkcije λv. E1 za dati argument E2)

$$(\lambda \mathrm{v}.E_1)\ E_2 \longrightarrow_\beta E_1[E_2/\mathrm{v}]$$

  - β-redeks – izraz oblika: (λv.E1) E2
- η-redukcija: λx. E x $\rightarrow_{\eta}$ E, ako x ∉ FV(E) (za programiranje manje važna).
- α-konverzija
  - Preimenovanje vezanih promenljivih – potrebno da bi se izbeglo hvatanje promenljivih (primeri posle zadataka 5 i 6).

# Primeri: β-redukcija i δ-pravila

$(\lambda x.\ +\ x\ 1)\ 4$

$\rightarrow_\beta\ +\ 4\ 1$ [redeks $(\lambda x.\ +\ x\ 1)\ 4$; $x := 4$]

$\rightarrow_\delta\ 5$ [ugrađeno sabiranje]

- Izraz može da sadrži više β-redeksa:

$(\lambda x.\ x)\ ((\lambda y.\ y)\ z)$

- Redeksi: ceo izraz i $(\lambda y.\ y)\ z$

$\rightarrow_\beta\ (\lambda y.\ y)\ z$ [spoljašnji redeks; $x := (\lambda y.\ y)\ z$]

$\rightarrow_\beta\ z$ [$y := z$]

- $\lambda x.\ x\ y$ nije redeks (nema argumenta); $x\ y$ nije redeks ($x$ nije apstrakcija)

# Zadaci

5. Izračunati vrednost sledećih lambda izraza:

$(\lambda v.v)\ c$
$(\lambda v.x\ (v\ c)\ v)\ (a\ y)$
$(\lambda v.c)\ (x\ v)$
$(\lambda v.v\ c)\ (\lambda x.x\ a)$
$(\lambda x.plus\ x\ 1)\ ((\lambda y.times\ y\ y)\ 3)$

# Rešenje zadatka 5 (a–d)

- a)

$(\lambda v.\ v)\ c \rightarrow_\beta c \quad [v := c]$

- b)

$(\lambda v.\ x\ (v\ c)\ v)\ (a\ y) \rightarrow_\beta x\ ((a\ y)\ c)\ (a\ y) \quad [v := a\ y]$

- c)

$(\lambda v.\ c)\ (x\ v) \rightarrow_\beta c \quad [v := x\ v;\ v \text{ se ne javlja u telu } c]$

- d)

$(\lambda v.\ v\ c)\ (\lambda x.\ x\ a)$

$\rightarrow_\beta (\lambda x.\ x\ a)\ c \quad [v := \lambda x.\ x\ a]$

$\rightarrow_\beta c\ a \quad [x := c]$

# Rešenje zadatka 5 (e)

(λx. plus x 1) ((λy. times y y) 3)

$\rightarrow_\beta$ plus ((λy. times y y) 3) 1 [spoljašnji redeks; x := (λy. times y y) 3]

$\rightarrow_\beta$ plus (times 3 3) 1 [y := 3]

$\rightarrow_\delta$ plus 9 1 [times 3 3 = 9]

$\rightarrow_\delta$ 10 [plus 9 1 = 10]

- Drugi redosled, isti rezultat:

$\rightarrow_\beta$ (λx. plus x 1) (times 3 3) [prvo argument; y := 3]

$\rightarrow_\delta$ (λx. plus x 1) 9

$\rightarrow_\beta$ plus 9 1 [x := 9]

$\rightarrow_\delta$ 10

# Zadaci

6. Izračunati vrednost lambda izraza:

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

# Rešenje zadatka 6

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

$=\ ((\lambda f.\lambda x.\ f\ 4\ x)\ (\lambda y.\lambda x.\ +\ x\ y))\ 3$ [primena je levo asocijativna]

$\rightarrow_\beta\ (\lambda x.\ (\lambda y.\lambda x.\ +\ x\ y)\ 4\ x)\ 3$ [f := λy.λx. + x y – zatvoren izraz]

$\rightarrow_\beta\ (\lambda y.\lambda x.\ +\ x\ y)\ 4\ 3$ [x := 3; unutrašnje λx ima svoje x]

$\rightarrow_\beta\ (\lambda x.\ +\ x\ 4)\ 3$ [y := 4]

$\rightarrow_\beta\ +\ 3\ 4$ [x := 3]

$\rightarrow_\delta\ 7$

# Primer: α-konverzija pre β-redukcije

$(\lambda x.\lambda y.\ x)\ y$

pogrešno: $\lambda y.\ y$ [x := y bez preimenovanja – slobodno y je uhvaćeno]

- Ispravno: prvo preimenujemo vezano y (α), pa tek onda β:

$(\lambda x.\lambda y.\ x)\ y \equiv_\alpha (\lambda x.\lambda z.\ x)\ y$ [$\lambda y.\ x \equiv_\alpha \lambda z.\ x$]

$\rightarrow_\beta \lambda z.\ y$ [x := y]

- Rezultat: konstantna funkcija koja uvek vraća (slobodno) y

# Primer sa preimenovanjem: λx.((λy.λx. + x y) x)

λx. ((λy.λx. + x y) x)

- Redeks: (λy.λx. + x y) x
- y := x u λx. + x y: unutrašnje λx bi uhvatilo slobodno x!

$\equiv_\alpha$ λx. ((λy.λz. + z y) x) [unutrašnje λx preimenovano u λz]

$\rightarrow_\beta$ λx.λz. + z x [y := x]

- Bez α: λx.λx. + x x – pogrešno (drugi sabirak postaje unutrašnje x)
- α-ekvivalentni izrazi: razlikuju se samo po imenima vezanih promenljivih
- α-ekvivalentne izraze smatramo jednakim

# Primeri: η-redukcija

$\lambda x.\, f\, x \rightarrow_\eta f \quad [x \notin FV(f)]$

$\lambda x.\, +\, 1\, x \rightarrow_\eta +\, 1 \quad [x \notin FV(+\, 1)]$

- Iste funkcije – na primer na argumentu 5:

$(\lambda x.\, +\, 1\, x)\, 5 \rightarrow_\beta +\, 1\, 5 \quad [x := 5]$

$\rightarrow_\delta 6$

$(+\, 1)\, 5 = +\, 1\, 5 \rightarrow_\delta 6$

- $\lambda x.\ x\ x$ nije η-redeks: x je slobodno u x
- $\lambda x.\ +\ x\ 1$ nije η-redeks: telo nije oblika E x

# Jednakost lambda izraza

- Dva lambda izraza su jednaka (s = t) ako se jedan može dobiti iz drugog konačnim nizom α-, β- i η-koraka, u bilo kom smeru i na bilo kom podizrazu.

Ako $s \rightarrow_\alpha t$, $s \rightarrow_\beta t$ ili $s \rightarrow_\eta t$, onda $s = t$

$t = t$

Ako $s = t$, onda $t = s$

Ako $s = t$ i $t = u$, onda $s = u$

Ako $s = t$, onda $s\ u = t\ u$, $u\ s = u\ t$ i $\lambda x.\ s = \lambda x.\ t$

$\equiv_\alpha$

$(\lambda x.\ x)y \equiv_\alpha (\lambda y.\ y)y.$

# Primeri: jednakost i redukcija

$(\lambda x.\ x)\ y \equiv_\alpha (\lambda y.\ y)\ y$ $[\lambda x.\ x \equiv_\alpha \lambda y.\ y;\ y \notin FV(x)]$

$(\lambda x.\ x)\ y \to_\beta y$ $[x := y]$

$(\lambda y.\ y)\ y \to_\beta y$ $[y := y]$

- Redukcija $s \to t$: isto kao jednakost, ali samo u smeru izračunavanja

$(\lambda x.\ x)\ a \to_\beta a$ [zato $(\lambda x.\ x)\ a = a$, a i $a = (\lambda x.\ x)\ a$]

- ali NE važi $a \to (\lambda x.\ x)\ a$
- $(\lambda x.\ x)\ a = (\lambda y.\ a)\ b$: oba se redukuju u a, pa su jednaki

# Normalne forme

- **Definicija:**

Za izraz E kažemo da je u *normalnoj formi* ako ne može dalje da se redukuje, tj. ako više ne sadrži nijedan redeks.

- **Definicija:**

Izraz E ima normalnu formu ako postoji konačan niz redukcija

$$E \rightarrow E1 \rightarrow E2 \rightarrow \ldots \rightarrow En$$

takav da je En u normalnoj formi; pišemo $E \rightarrow_* En$.

# Primeri: normalna forma

- $\lambda x.\ x$ – već je u normalnoj formi (nema redeksa)

$(\lambda x.\ x)\ y \rightarrow_\beta y$ [x := y; y je u normalnoj formi]

$(\lambda x.\ x\ x)\ (\lambda x.\ x\ x) \rightarrow_\beta (\lambda x.\ x\ x)\ (\lambda x.\ x\ x)$ [x := λx. x x]

- ... i tako zauvek: ovaj izraz (Ω) nema normalnu formu

$(\lambda x.\ y)\ \Omega \rightarrow_\beta y$ [x := Ω; x se ne javlja u telu]

- ima normalnu formu y, ali do nje ne vodi svaki redosled (sledeći slajdovi)

# Položaj redeksa

- Spoljašnji redeks: nije sadržan ni u jednom drugom redeksu
- Unutrašnji redeks: ne sadrži nijedan drugi redeks
- Najlevlji: redeks čije je λ najviše levo u zapisu
- Primer:

(λx. (λy. y) x) ((λz. z) a)

- R1 = ceo izraz – spoljašnji
- R2 = (λy. y) x – unutrašnji, najlevlji među unutrašnjim
- R3 = (λz. z) a – unutrašnji
- Normalni redosled bira R1, aplikativni bira R2

# Strategije redukcije

- Teorijska razmatranja i funkcionalno programiranje?
- Izbor narednog redeksa na dva najčešća načina:
  - Najlevlji-najunutrašnjiji redeks (aplikativni redosled; srodno: prenos po vrednosti) – izračunavanje ne mora da se završi.

$$
\begin{aligned}
& (\lambda x.\, y)\ ((\lambda x.\, x\ x\ x)\ (\lambda x.\, x\ x\ x)) \\
\longrightarrow\ & (\lambda x.\, y)\ ((\lambda x.\, x\ x\ x)\ (\lambda x.\, x\ x\ x)\ (\lambda x.\, x\ x\ x)) \\
\longrightarrow\ & (\lambda x.\, y)\ ((\lambda x.\, x\ x\ x)\ (\lambda x.\, x\ x\ x)\ (\lambda x.\, x\ x\ x)\ (\lambda x.\, x\ x\ x)) \\
\longrightarrow\ & \ldots
\end{aligned}
$$

  - Najlevlji-najspoljašnji redeks (normalni redosled; srodno: prenos po imenu, lenjo izračunavanje u Haskell-u) – sigurno pronalazi normalnu formu ako ona postoji.

$$(\lambda x.\, y)\ ((\lambda x.\, x\ x\ x)\ (\lambda x.\, x\ x\ x)) \longrightarrow y$$

# Primer sa prethodnog slajda, korak po korak

- Neka je $W \equiv \lambda x.\ x\ x\ x$
- Aplikativni redosled:

$(\lambda x.\ y)\ (W\ W)$

$\rightarrow_\beta\ (\lambda x.\ y)\ (W\ W\ W)$ [unutrašnji redeks W W; x := W]

$\rightarrow_\beta\ (\lambda x.\ y)\ (W\ W\ W\ W)$ [ponovo najlevlji unutrašnji W W]

$\rightarrow_\beta$ ... [izraz raste – nikad se ne završava]

- Normalni redosled:

$(\lambda x.\ y)\ (W\ W)$

$\rightarrow_\beta\ y$ [spoljašnji redeks; x := W W; x se ne javlja u telu]

# Primer: dva redosleda, ista normalna forma

$(\lambda x.\ (\lambda y.\ y)\ x)\ ((\lambda z.\ z)\ a)$

- Normalni:

$\rightarrow_\beta\ (\lambda y.\ y)\ ((\lambda z.\ z)\ a)\ \ [R1;\ x := (\lambda z.\ z)\ a]$

$\rightarrow_\beta\ (\lambda z.\ z)\ a\ \ [y := (\lambda z.\ z)\ a]$

$\rightarrow_\beta\ a\ \ [z := a]$

- Aplikativni:

$\rightarrow_\beta\ (\lambda x.\ x)\ ((\lambda z.\ z)\ a)\ \ [R2;\ y := x]$

$\rightarrow_\beta\ (\lambda x.\ x)\ a\ \ [R3;\ z := a]$

$\rightarrow_\beta\ a\ \ [x := a]$

- Ako se do normalne forme stigne, ona je ista bez obzira na redosled

# Primer: isti rezultat, različit broj koraka

$(\lambda x.\ +\ x\ x)\ (+\ 1\ 2)$

- Normalni:

$\rightarrow_\beta\ +\ (+\ 1\ 2)\ (+\ 1\ 2)$ [x := + 1 2]

$\rightarrow_\delta\ +\ 3\ (+\ 1\ 2)$

$\rightarrow_\delta\ +\ 3\ 3$

$\rightarrow_\delta\ 6$ [4 koraka; + 1 2 računato dva puta]

- Aplikativni:

$\rightarrow_\delta\ (\lambda x.\ +\ x\ x)\ 3$ [prvo argument]

$\rightarrow_\beta\ +\ 3\ 3$ [x := 3]

$\rightarrow_\delta\ 6$ [3 koraka]

- Lenjo izračunavanje (Haskell) = normalni redosled + deljenje:
- svaki argument se računa najviše jednom

# Zadaci

7. Izračunati vrednost lambda izraza

$(\lambda x.+\ x\ x)((\lambda y.*\ y\ y)(+5\ 5))$

a) normalnim redosledom redukcije;

b) aplikativnim redosledom redukcije.

8. Izračunati vrednost lambda izraza

$(\lambda g.\ g + \big(g\ *(g+5)\big))(\lambda f.\lambda x.f\ x\ x)$

a) normalnim redosledom redukcije;

b) aplikativnim redosledom redukcije.

# Rešenje zadatka 7a – normalni redosled

$(\lambda x.\ +\ x\ x)\ ((\lambda y.\ *\ y\ y)\ (+\ 5\ 5))$

$\rightarrow_\beta\ +\ ((\lambda y.\ *\ y\ y)\ (+\ 5\ 5))\ ((\lambda y.\ *\ y\ y)\ (+\ 5\ 5))$ [x := (λy. * y y) (+ 5 5)]

$\rightarrow_\beta\ +\ (*\ (+\ 5\ 5)\ (+\ 5\ 5))\ ((\lambda y.\ *\ y\ y)\ (+\ 5\ 5))$ [prvi argument; y := + 5 5]

$\rightarrow_\delta\ +\ (*\ 10\ (+\ 5\ 5))\ ((\lambda y.\ *\ y\ y)\ (+\ 5\ 5))$

$\rightarrow_\delta\ +\ (*\ 10\ 10)\ ((\lambda y.\ *\ y\ y)\ (+\ 5\ 5))$

$\rightarrow_\delta\ +\ 100\ ((\lambda y.\ *\ y\ y)\ (+\ 5\ 5))$

$\rightarrow_\beta\ +\ 100\ (*\ (+\ 5\ 5)\ (+\ 5\ 5))$ [y := + 5 5]

$\rightarrow_\delta\ +\ 100\ (*\ 10\ (+\ 5\ 5))$

$\rightarrow_\delta\ +\ 100\ (*\ 10\ 10)$

$\rightarrow_\delta\ +\ 100\ 100$

$\rightarrow_\delta\ 200$ [10 koraka; + 5 5 računato četiri puta]

# Rešenje zadatka 7b – aplikativni redosled

$(\lambda x.\ +\ x\ x)\ ((\lambda y.\ *\ y\ y)\ (+\ 5\ 5))$

$\rightarrow_{\delta}\ (\lambda x.\ +\ x\ x)\ ((\lambda y.\ *\ y\ y)\ 10)$ [najunutrašnjiji redeks + 5 5]

$\rightarrow_{\beta}\ (\lambda x.\ +\ x\ x)\ (*\ 10\ 10)$ [y := 10]

$\rightarrow_{\delta}\ (\lambda x.\ +\ x\ x)\ 100$

$\rightarrow_{\beta}\ +\ 100\ 100$ [x := 100]

$\rightarrow_{\delta}\ 200$ [5 koraka]

- Isti rezultat kao normalnim redosledom, ali u 5 umesto 10 koraka

# Zadatak 8 – priprema

$(\lambda g.\ g + (g * (g + 5)))\ (\lambda f.\lambda x.\ f\ x\ x)$

- Ovde su + i $*$ argumenti funkcije g: g + 5 znači (g +) 5
- Neka je D $\equiv$ $\lambda f.\lambda x.$ f x x („primeni f na dupliran argument”)

D h a

$\rightarrow_\beta$ $(\lambda x.\ h\ x\ x)\ a$ $[f := h]$

$\rightarrow_\beta$ h a a $[x := a]$

- Dakle: D + 5 $\rightarrow_\beta^*$ + 5 5 i D $*$ a $\rightarrow_\beta^*$ $*$ a a

# Rešenje zadatka 8a – normalni redosled (1/2)

$(\lambda g.\ g + (g * (g + 5)))\ D$

$\rightarrow_\beta\ D + (D * (D + 5))$ [g := D]

$\rightarrow_\beta\ (\lambda x.\ + x\ x)\ (D * (D + 5))$ [redeks D +; f := +]

$\rightarrow_\beta\ +\ (D * (D + 5))\ (D * (D + 5))$ [x := D * (D + 5)]

$\rightarrow_\beta\ +\ ((\lambda x.\ * x\ x)\ (D + 5))\ (D * (D + 5))$ [redeks D *; f := *]

$\rightarrow_\beta\ +\ (*\ (D + 5)\ (D + 5))\ (D * (D + 5))$ [x := D + 5]

$\rightarrow_\beta\ +\ (*\ ((\lambda x.\ + x\ x)\ 5)\ (D + 5))\ (D * (D + 5))$ [redeks D +; f := +]

$\rightarrow_\beta\ +\ (*\ (+\ 5\ 5)\ (D + 5))\ (D * (D + 5))$ [x := 5]

$\rightarrow_\delta\ +\ (*\ 10\ (D + 5))\ (D * (D + 5))$ [+ 5 5 = 10]

$\rightarrow_\beta\ +\ (*\ 10\ ((\lambda x.\ + x\ x)\ 5))\ (D * (D + 5))$ [redeks D +; f := +]

$\rightarrow_\beta\ +\ (*\ 10\ (+\ 5\ 5))\ (D * (D + 5))$ [x := 5]

$\rightarrow_\delta\ +\ (*\ 10\ 10)\ (D * (D + 5))$ [+ 5 5 = 10]

# Rešenje zadatka 8a – normalni redosled (2/2)

$+\ (*\ 10\ 10)\ (D * (D + 5))$

$\rightarrow_{\delta}\ +\ 100\ (D * (D + 5))$ [$*\ 10\ 10 = 100$]

$\rightarrow_{\beta}\ +\ 100\ ((\lambda x.\ *\ x\ x)\ (D + 5))$ [redeks D $*$; f := $*$]

$\rightarrow_{\beta}\ +\ 100\ (*\ (D + 5)\ (D + 5))$ [x := D + 5]

$\rightarrow_{\beta}\ +\ 100\ (*\ ((\lambda x.\ +\ x\ x)\ 5)\ (D + 5))$ [redeks D +; f := +]

$\rightarrow_{\beta}\ +\ 100\ (*\ (+\ 5\ 5)\ (D + 5))$ [x := 5]

$\rightarrow_{\delta}\ +\ 100\ (*\ 10\ (D + 5))$

$\rightarrow_{\beta}\ +\ 100\ (*\ 10\ ((\lambda x.\ +\ x\ x)\ 5))$ [redeks D +; f := +]

$\rightarrow_{\beta}\ +\ 100\ (*\ 10\ (+\ 5\ 5))$ [x := 5]

$\rightarrow_{\delta}\ +\ 100\ (*\ 10\ 10)$

$\rightarrow_{\delta}\ +\ 100\ 100$

$\rightarrow_{\delta}\ 200$ [ukupno 22 koraka; isti posao je urađen dva puta]

# Rešenje zadatka 8b – aplikativni redosled

$(\lambda g.\ g + (g * (g + 5)))\ D$

$\rightarrow_\beta\ D + (D * (D + 5))$ [jedini redeks; g := D]

$\rightarrow_\beta\ (\lambda x.\ +\ x\ x)\ (D * (D + 5))$ [najlevlji unutrašnji: D +; f := +]

$\rightarrow_\beta\ (\lambda x.\ +\ x\ x)\ ((\lambda x.\ *\ x\ x)\ (D + 5))$ [D *; f := *]

$\rightarrow_\beta\ (\lambda x.\ +\ x\ x)\ ((\lambda x.\ *\ x\ x)\ ((\lambda x.\ +\ x\ x)\ 5))$ [D +; f := +]

$\rightarrow_\beta\ (\lambda x.\ +\ x\ x)\ ((\lambda x.\ *\ x\ x)\ (+\ 5\ 5))$ [x := 5]

$\rightarrow_\delta\ (\lambda x.\ +\ x\ x)\ ((\lambda x.\ *\ x\ x)\ 10)$

$\rightarrow_\beta\ (\lambda x.\ +\ x\ x)\ (*\ 10\ 10)$ [x := 10]

$\rightarrow_\delta\ (\lambda x.\ +\ x\ x)\ 100$

$\rightarrow_\beta\ +\ 100\ 100$ [x := 100]

$\rightarrow_\delta\ 200$ [10 koraka (normalnim: 22)]

# De Bruijnovi indeksi

- Ideja: izrazi koji se razlikuju samo po imenima vezanih promenljivih treba da imaju isti zapis;
- Imena vezanih promenljivih se uklanjaju: svako pojavljivanje zamenjuje se brojem λ-ova između njega i λ koje ga vezuje (brojimo od 0).
- Primeri:

| Standard | de Bruijn |
|---|---|
| $\lambda x.x$ | $\lambda.0$ |
| $\lambda z.z$ | $\lambda.0$ |
| $\lambda x.\lambda y.x$ | $\lambda.\lambda.1$ |
| $\lambda x.\lambda y.\lambda s.\lambda z.x\ s\ (y\ s\ z)$ | $\lambda.\lambda.\lambda.\lambda.3\ 1\ (2\ 1\ 0)$ |
| $(\lambda x.x\ x)\ (\lambda x.x\ x)$ | $(\lambda.0\ 0)\ (\lambda.0\ 0)$ |
| $(\lambda x.\lambda x.x)\ (\lambda y.y)$ | $(\lambda.\lambda.0)\ (\lambda.0)$ |

# Primer: kako se računa indeks

λx.λy.λs.λz. x s (y s z)

- x: između njega i λx su λy, λs, λz → 3
- s: između je λz → 1; y: λs, λz → 2; z → 0

⇒ λ.λ.λ.λ. 3 1 (2 1 0)

- α-ekvivalentni izrazi imaju isti zapis:

λx.λy. x y i λa.λb. a b ⇒ oba su λ.λ. 1 0

# Zadaci

9. Zapisati funkciju λn. + n 1 u de Bruijnovom zapisu.
10. Zapisati sledeći lambda izraz u de Bruijnovom zapisu.

$$\lambda x.\lambda y.\lambda f.f\ (\lambda x.x)(+\ x\ y)$$

# Rešenje zadataka 9 i 10

- 9.

λn. + n 1 ⇒ λ. + 0 1 [n je vezano najbližim λ: indeks 0]

- Pažnja: 1 je ovde konstanta (broj jedan), a ne indeks
- (Čerčov sledbenik λn.λf.λx. n f (f x) ⇒ λ.λ.λ. 2 1 (1 0))
- 10.

λx.λy.λf. f (λx. x) (+ x y)

- f → 0 (vezano najbližim λf)
- unutrašnje λx. x → λ. 0 (vezano svojim λx)
- x u (+ x y): između su λy i λf → 2; y: između je λf → 1

⇒ λ.λ.λ. 0 (λ. 0) (+ 2 1)

# LAMBDA RAČUN KAO PROGRAMSKI JEZIK

# Predstavljanje podataka u lambda računu

- Notacija:

$$s \stackrel{\triangle}{=} s' \qquad\qquad s =_{\text{def}} s'$$

jednaki po definiciji;

- Uslovni izraz; if E then E1 else E2 zapisujemo i kao COND E E1 E2
- Promenljive sa leve strane definicije apstrahujemo: fst p ≜ p true znači fst ≜ λp. p true.

$$\text{fst}\ p \stackrel{\triangle}{=} p\ \text{true} \qquad\qquad \text{fst} \stackrel{\triangle}{=} \lambda p.\ p\ \text{true}$$

# Logičke vrednosti i operatori

- Logičke vrednosti:

$$\begin{aligned} \text{true} &\triangleq \lambda x\ y.\ x \\ \text{false} &\triangleq \lambda x\ y.\ y \end{aligned}$$

- Uslovni izraz (drugi zapis: COND E E1 E2):

$$\text{if } E \text{ then } E_1 \text{ else } E_2 \triangleq E\ E_1\ E_2$$

# Logičke vrednosti i operatori

$$
\begin{aligned}
\text{if true then } E_1 \text{ else } E_2 &= \text{true } E_1\ E_2 \\
&= (\lambda x\ y.\ x)\ E_1\ E_2 \\
&= E_1
\end{aligned}
$$

$$
\begin{aligned}
\text{if false then } E_1 \text{ else } E_2 &= \text{false } E_1\ E_2 \\
&= (\lambda x\ y.\ y)\ E_1\ E_2 \\
&= E_2
\end{aligned}
$$

# Isto, korak po korak

if true then E1 else E2

$=$ true E1 E2 [def. if-then-else]

$=$ $(\lambda x.\lambda y.\ x)$ E1 E2 [def. true]

$\rightarrow_\beta$ $(\lambda y.\ E1)$ E2 [x := E1 (ako je $y \in FV(E1)$, prvo $\lambda y \equiv_\alpha \lambda z$)]

$\rightarrow_\beta$ E1 [y := E2; y nije slobodno u E1]

if false then E1 else E2

$=$ false E1 E2 [def. if-then-else]

$=$ $(\lambda x.\lambda y.\ y)$ E1 E2 [def. false]

$\rightarrow_\beta$ $(\lambda y.\ y)$ E2 [x := E1; x se ne javlja u telu, E1 se odbacuje]

$\rightarrow_\beta$ E2 [y := E2]

# Logičke vrednosti i operatori

$$
\begin{aligned}
\text{not } p &\triangleq \text{if } p \text{ then false else true} \\
p \text{ and } q &\triangleq \text{if } p \text{ then } q \text{ else false} \\
p \text{ or } q &\triangleq \text{if } p \text{ then true else } q
\end{aligned}
$$

# not, and, or kao lambda izrazi

not p ≜ if p then false else true = p false true [def. if]

⇒ not ≜ λp. p false true

p and q ≜ if p then q else false = p q false [def. if]

⇒ and ≜ λp.λq. p q false

p or q ≜ if p then true else q = p true q [def. if]

⇒ or ≜ λp.λq. p true q

- Sa ovim izrazima rešavamo zadatke 11–13 (sledeći slajdovi)

# Zadaci

11. Izračunati:
    a) not true
    b) not false

12. Izračunati:
    a) and true true
    b) and false true

13. Izračunati:
    a) or true true
    b) or false true

# Rešenje zadatka 11

not true $=$ $(\lambda p.\ p\ \text{false}\ \text{true})\ \text{true}$ [def. not]

$\rightarrow_\beta$ $\text{true}\ \text{false}\ \text{true}$ [p := true]

$=$ $(\lambda x.\lambda y.\ x)\ \text{false}\ \text{true}$ [def. true]

$\rightarrow_\beta$ $(\lambda y.\ \text{false})\ \text{true}$ [x := false]

$\rightarrow_\beta$ false [y := true]

not false $=$ $(\lambda p.\ p\ \text{false}\ \text{true})\ \text{false}$ [def. not]

$\rightarrow_\beta$ $\text{false}\ \text{false}\ \text{true}$ [p := false]

$=$ $(\lambda x.\lambda y.\ y)\ \text{false}\ \text{true}$ [def. false]

$\rightarrow_\beta$ $(\lambda y.\ y)\ \text{true}$ [x := false; x se ne javlja u telu]

$\rightarrow_\beta$ true [y := true]

# Rešenje zadatka 12

and true true = (λp.λq. p q false) true true [def. and]

$\rightarrow_\beta$ (λq. true q false) true [p := true]

$\rightarrow_\beta$ true true false [q := true]

= (λx.λy. x) true false [def. true]

$\rightarrow_\beta$ (λy. true) false [x := true]

$\rightarrow_\beta$ true [y := false]

and false true = (λp.λq. p q false) false true [def. and]

$\rightarrow_\beta$ (λq. false q false) true [p := false]

$\rightarrow_\beta$ false true false [q := true]

= (λx.λy. y) true false [def. false]

$\rightarrow_\beta$ (λy. y) false [x := true]

$\rightarrow_\beta$ false [y := false]

# Rešenje zadatka 13

or true true $=$ $(\lambda p.\lambda q.\ p\ \text{true}\ q)\ \text{true}\ \text{true}$ [def. or]

$\rightarrow_\beta$ $(\lambda q.\ \text{true}\ \text{true}\ q)\ \text{true}$ [p := true]

$\rightarrow_\beta$ true true true [q := true]

$=$ $(\lambda x.\lambda y.\ x)\ \text{true}\ \text{true}$ [def. true]

$\rightarrow_\beta$ $(\lambda y.\ \text{true})\ \text{true}$ [x := true]

$\rightarrow_\beta$ true [y := true]

or false true $=$ $(\lambda p.\lambda q.\ p\ \text{true}\ q)\ \text{false}\ \text{true}$ [def. or]

$\rightarrow_\beta$ $(\lambda q.\ \text{false}\ \text{true}\ q)\ \text{true}$ [p := false]

$\rightarrow_\beta$ false true true [q := true]

$=$ $(\lambda x.\lambda y.\ y)\ \text{true}\ \text{true}$ [def. false]

$\rightarrow_\beta$ $(\lambda y.\ y)\ \text{true}$ [x := true]

$\rightarrow_\beta$ true [y := true]

# Parovi i torke

- Uređeni par

$$(E_1, E_2) \triangleq \lambda f.\ f\ E_1\ E_2$$

- Prvi, drugi element para:

$$\begin{aligned} \text{fst}\ p &\triangleq p\ \text{true} \\ \text{snd}\ p &\triangleq p\ \text{false} \end{aligned}$$

- Pokazati da je definicija ispravna: fst (E1, E2) = E1 i snd (E1, E2) = E2 (dokaz na sledeća dva slajda).

# Dokaz (1/2): fst (E1, E2) = E1

fst (E1, E2)

$=$ $(\lambda p.\ p\ \text{true})\ (\lambda f.\ f\ E1\ E2)$ [def. fst (fst $\triangleq \lambda p.\ p\ \text{true}$) i def. para]

$\rightarrow_\beta$ $(\lambda f.\ f\ E1\ E2)\ \text{true}$ [$p := \lambda f.\ f\ E1\ E2$]

$\rightarrow_\beta$ $\text{true}\ E1\ E2$ [$f := \text{true}$]

$=$ $(\lambda x.\lambda y.\ x)\ E1\ E2$ [def. true]

$\rightarrow_\beta$ $(\lambda y.\ E1)\ E2$ [$x := E1$ (ako je $y \in FV(E1)$, prvo $\lambda y \equiv_\alpha \lambda z$)]

$\rightarrow_\beta$ $E1$ [$y := E2$; y nije slobodno u E1]

- Pretpostavka: f nije slobodno u E1 i E2 (inače bi ga par „uhvatio”)

# Dokaz (2/2): snd (E1, E2) = E2

snd (E1, E2)

$=$ $(\lambda p.\ p\ \text{false})\ (\lambda f.\ f\ E1\ E2)$ [def. snd (snd $\triangleq \lambda p.\ p$ false) i def. para]

$\rightarrow_\beta$ $(\lambda f.\ f\ E1\ E2)$ false [p := $\lambda f.\ f\ E1\ E2$]

$\rightarrow_\beta$ false E1 E2 [f := false]

$=$ $(\lambda x.\lambda y.\ y)\ E1\ E2$ [def. false]

$\rightarrow_\beta$ $(\lambda y.\ y)$ E2 [x := E1; x se ne javlja u telu, E1 se odbacuje]

$\rightarrow_\beta$ E2 [y := E2]

# Parovi i torke

- n-torka

$$(\mathrm{E1}, \mathrm{E2}, ..., \mathrm{En}) \triangleq (\mathrm{E1}, (\mathrm{E2}, ..., \mathrm{En}))$$

- Uređena četvorka

$$\begin{aligned}
(p, q, r, s) &= (p, (q, (r, s))) \\
&= \lambda f.\ f\ p\ (q, (r, s)) \\
&= \lambda f.\ f\ p\ (\lambda f.\ f\ q\ (r, s)) \\
&= \lambda f.\ f\ p\ (\lambda f.\ f\ q\ (\lambda f.\ f\ r\ s)) \\
&= \lambda f.\ f\ p\ (\lambda g.\ g\ q\ (\lambda h.\ h\ r\ s))
\end{aligned}$$

# Četvorka, korak po korak

$(p, q, r, s)$

$= (p, (q, (r, s)))$ [def. n-torke]

$= \lambda f.\ f\ p\ (q, (r, s))$ [def. para za spoljašnji par]

$= \lambda f.\ f\ p\ (\lambda f.\ f\ q\ (r, s))$ [def. para (q, (r, s))]

$= \lambda f.\ f\ p\ (\lambda f.\ f\ q\ (\lambda f.\ f\ r\ s))$ [def. para (r, s)]

$\equiv_\alpha \lambda f.\ f\ p\ (\lambda g.\ g\ q\ (\lambda h.\ h\ r\ s))$ [dva unutrašnja f preimenovana u g i h]

- Preimenovanje nije neophodno – samo olakšava čitanje

# Primer: treći element četvorke

t ≡ (p, q, r, s) = (p, (q, (r, s)))

fst (snd (snd t))

= fst (snd (q, (r, s))) [snd (E1, E2) = E2; E1 := p]

= fst (r, s) [snd (E1, E2) = E2; E1 := q]

= r [fst (E1, E2) = E1]

- svaki snd odbacuje prvi element, fst uzima prvi

# Prirodni brojevi (Čerčovi brojevi)

- Nula, sledbenik nule, sledbenik sledbenika nule...
- $0 \triangleq \lambda f.\lambda x.\ x$
- $1 \triangleq \lambda f.\lambda x.\ f\ x$
- $2 \triangleq \lambda f.\lambda x.\ f\ (f\ x)$
- $3 \triangleq \lambda f.\lambda x.\ f\ (f\ (f\ x))$
- ...
- $6 \triangleq ?$
- Zašto ovakav način definisanja brojeva?

$$n \stackrel{\triangle}{=} \lambda f\ x.\ f^n\ x$$

# Odgovor: 6 i ideja Čerčovih brojeva

$6 \triangleq \lambda f.\lambda x.\ f\ (f\ (f\ (f\ (f\ (f\ x)))))$ $\quad (= \lambda f.\lambda x.\ f^6\ x)$

- Broj n znači „primeni f n puta”: n f x = $f^n$ x = f (f (... (f x)...))
- Primer: broj 2 primenjen na „dodaj 1” i na 0:

$2\ (\lambda y.\ +\ y\ 1)\ 0$

$=\ (\lambda f.\lambda x.\ f\ (f\ x))\ (\lambda y.\ +\ y\ 1)\ 0$ [def. 2]

$\rightarrow_\beta\ (\lambda x.\ (\lambda y.\ +\ y\ 1)\ ((\lambda y.\ +\ y\ 1)\ x))\ 0$ [f := λy. + y 1]

$\rightarrow_\beta\ (\lambda y.\ +\ y\ 1)\ ((\lambda y.\ +\ y\ 1)\ 0)$ [x := 0]

$\rightarrow_\beta\ +\ ((\lambda y.\ +\ y\ 1)\ 0)\ 1$ [y := (λy. + y 1) 0]

$\rightarrow_\beta\ +\ (+\ 0\ 1)\ 1$ [y := 0]

$\rightarrow_\delta\ +\ 1\ 1$

$\rightarrow_\delta\ 2$

- Zato se sabiranje, množenje i ISZERO lako definišu

# Funkcija sledbenik

- Funkcija sledbenik

$$\mathrm{SUC} \triangleq \lambda n\ f\ x.\ n\ f\ (f\ x)$$

- Odredimo sledbenik broja n

$$\begin{aligned}
\mathrm{SUC}\ n &= (\lambda n\ f\ x.\ n\ f\ (f\ x))(\lambda f\ x.\ f^n\ x) \\
&= \lambda f\ x.\ (\lambda f\ x.\ f^n\ x) f\ (f\ x) \\
&= \lambda f\ x.\ (\lambda x.\ f^n\ x)(f\ x) \\
&= \lambda f\ x.\ f^n\ (f\ x) \\
&= \lambda f\ x.\ f^{n+1}\ x \\
&= n+1
\end{aligned}$$

# SUC n, korak po korak

SUC n

$=$ $(\lambda n.\lambda f.\lambda x.\ n\ f\ (f\ x))\ (\lambda f.\lambda x.\ f^n\ x)$ [def. SUC i broja n]

$\rightarrow_\beta$ $\lambda f.\lambda x.\ (\lambda f.\lambda x.\ f^n\ x)\ f\ (f\ x)$ [n := $\lambda f.\lambda x.\ f^n\ x$ (zatvoren izraz)]

$\rightarrow_\beta$ $\lambda f.\lambda x.\ (\lambda x.\ f^n\ x)\ (f\ x)$ [unutrašnje f := f]

$\rightarrow_\beta$ $\lambda f.\lambda x.\ f^n\ (f\ x)$ [x := f x]

$=$ $\lambda f.\lambda x.\ f^{n+1}\ x$ [$f^n$ (f x) = f primenjeno n + 1 puta]

$=$ $n + 1$ [def. broja n + 1]

# Zadaci

14. Izračunati sledbenik broja 0 (SUC 0).
15. Izračunati sledbenik broja 1 (SUC 1).

# Rešenje zadataka 14 i 15

SUC 0 = $(\lambda n.\lambda f.\lambda x.\ n\ f\ (f\ x))\ (\lambda f.\lambda x.\ x)$ [def. SUC i 0]

$\rightarrow_\beta\ \lambda f.\lambda x.\ (\lambda f.\lambda x.\ x)\ f\ (f\ x)$ [n := $\lambda f.\lambda x.\ x$]

$\rightarrow_\beta\ \lambda f.\lambda x.\ (\lambda x.\ x)\ (f\ x)$ [unutrašnje f := f; f se ne javlja u telu]

$\rightarrow_\beta\ \lambda f.\lambda x.\ f\ x$ = 1 [x := f x]

SUC 1 = $(\lambda n.\lambda f.\lambda x.\ n\ f\ (f\ x))\ (\lambda f.\lambda x.\ f\ x)$ [def. SUC i 1]

$\rightarrow_\beta\ \lambda f.\lambda x.\ (\lambda f.\lambda x.\ f\ x)\ f\ (f\ x)$ [n := $\lambda f.\lambda x.\ f\ x$]

$\rightarrow_\beta\ \lambda f.\lambda x.\ (\lambda x.\ f\ x)\ (f\ x)$ [unutrašnje f := f]

$\rightarrow_\beta\ \lambda f.\lambda x.\ f\ (f\ x)$ = 2 [x := f x]

- Isti imena f, x u SUC i u broju ne smetaju – to su različite promenljive

# Funkcija koja proverava da li je broj jednak nuli

$$\text{ISZERO}\ n \triangleq n\ (\lambda x.\ \text{false})\ \text{true}$$

$$\text{ISZERO}\ 0 = (\lambda f\ x.\ x)(\lambda x.\ \text{false})\ \text{true} = \text{true}$$

$$\begin{aligned}
\text{ISZERO}\ (n+1) &= (\lambda f\ x.\ f^{n+1}x)(\lambda x.\ \text{false})\text{true} \\
&= (\lambda x.\ \text{false})^{n+1}\ \text{true} \\
&= (\lambda x.\ \text{false})((\lambda x.\ \text{false})^{n}\ \text{true}) \\
&= \text{false}
\end{aligned}$$

# ISZERO 0 i ISZERO 1, korak po korak

ISZERO $\triangleq$ $\lambda n.\ n\ (\lambda y.\ \text{false})\ \text{true}$ $\quad$ $(\lambda x.\ \text{false} \equiv_\alpha \lambda y.\ \text{false})$

ISZERO 0 $=$ $(\lambda n.\ n\ (\lambda y.\ \text{false})\ \text{true})\ (\lambda f.\lambda x.\ x)$ [def. ISZERO i 0]

$\rightarrow_\beta$ $(\lambda f.\lambda x.\ x)\ (\lambda y.\ \text{false})\ \text{true}$ [n := $\lambda f.\lambda x.\ x$]

$\rightarrow_\beta$ $(\lambda x.\ x)\ \text{true}$ [f := $\lambda y.$ false; f se ne javlja u telu]

$\rightarrow_\beta$ true [x := true]

ISZERO 1 $=$ $(\lambda n.\ n\ (\lambda y.\ \text{false})\ \text{true})\ (\lambda f.\lambda x.\ f\ x)$ [def. ISZERO i 1]

$\rightarrow_\beta$ $(\lambda f.\lambda x.\ f\ x)\ (\lambda y.\ \text{false})\ \text{true}$ [n := $\lambda f.\lambda x.\ f\ x$]

$\rightarrow_\beta$ $(\lambda x.\ (\lambda y.\ \text{false})\ x)\ \text{true}$ [f := $\lambda y.$ false]

$\rightarrow_\beta$ $(\lambda y.\ \text{false})\ \text{true}$ [x := true]

$\rightarrow_\beta$ false [y := true; y se ne javlja u telu]

# Sabiranje, množenje

$$
\begin{aligned}
m + n &\triangleq \lambda f\ x.\ m\ f\ (n\ f\ x) \\
m * n &\triangleq \lambda f\ x.\ m\ (n\ f)\ x
\end{aligned}
$$

$$
\begin{aligned}
m + n &= \lambda f\ x.\ m\ f\ (n\ f\ x) \\
&= \lambda f\ x.\ (\lambda f\ x.\ f^m\ x)\ f\ (n\ f\ x) \\
&= \lambda f\ x.\ (\lambda x.\ f^m\ x)\ (n\ f\ x) \\
&= \lambda f\ x.\ f^m\ (n\ f\ x) \\
&= \lambda f\ x.\ f^m\ ((\lambda f\ x.\ f^n\ x)\ f\ x) \\
&= \lambda f\ x.\ f^m\ ((\lambda x.\ f^n\ x)\ x) \\
&= \lambda f\ x.\ f^m\ (f^n\ x) \\
&= \lambda f\ x.\ f^{m+n} x
\end{aligned}
$$

# m + n, korak po korak

$m + n$

$= \lambda f.\lambda x.\ m\ f\ (n\ f\ x)$ [def. +]

$= \lambda f.\lambda x.\ (\lambda f.\lambda x.\ f^m\ x)\ f\ (n\ f\ x)$ [def. broja m]

$\rightarrow_\beta \lambda f.\lambda x.\ (\lambda x.\ f^m\ x)\ (n\ f\ x)$ [unutrašnje f := f]

$\rightarrow_\beta \lambda f.\lambda x.\ f^m\ (n\ f\ x)$ [x := n f x]

$= \lambda f.\lambda x.\ f^m\ ((\lambda f.\lambda x.\ f^n\ x)\ f\ x)$ [def. broja n]

$\rightarrow_\beta \lambda f.\lambda x.\ f^m\ ((\lambda x.\ f^n\ x)\ x)$ [f := f]

$\rightarrow_\beta \lambda f.\lambda x.\ f^m\ (f^n\ x) = \lambda f.\lambda x.\ f^{m+n}\ x$ [x := x]

- Ovde + i $*$ nisu ugrađene konstante, već lambda izrazi:

PLUS $\triangleq \lambda m.\lambda n.\lambda f.\lambda x.\ m\ f\ (n\ f\ x)$ MULT $\triangleq \lambda m.\lambda n.\lambda f.\lambda x.\ m\ (n\ f)\ x$

# Zadaci

16. Izračunati zbir brojeva 2 i 3 u lambda notaciji.

# Rešenje zadatka 16

$2 + 3$

$= \lambda f.\lambda x.\ 2\ f\ (3\ f\ x)$ [def. +; m := 2, n := 3]

$= \lambda f.\lambda x.\ (\lambda f.\lambda x.\ f\ (f\ x))\ f\ (3\ f\ x)$ [def. 2]

$\rightarrow_\beta \lambda f.\lambda x.\ (\lambda x.\ f\ (f\ x))\ (3\ f\ x)$ [unutrašnje f := f]

$\rightarrow_\beta \lambda f.\lambda x.\ f\ (f\ (3\ f\ x))$ [x := 3 f x]

$= \lambda f.\lambda x.\ f\ (f\ ((\lambda f.\lambda x.\ f\ (f\ (f\ x)))\ f\ x))$ [def. 3]

$\rightarrow_\beta \lambda f.\lambda x.\ f\ (f\ ((\lambda x.\ f\ (f\ (f\ x)))\ x))$ [f := f]

$\rightarrow_\beta \lambda f.\lambda x.\ f\ (f\ (f\ (f\ (f\ x))))$ [x := x]

$= 5$

# Sabiranje, množenje

$$
\begin{aligned}
m + n &\triangleq \lambda f\ x.\ m\ f\ (n\ f\ x) \\
m * n &\triangleq \lambda f\ x.\ m\ (n\ f)\ x
\end{aligned}
$$

$$
\begin{aligned}
m * n &= \lambda f\ x.\ m\ (n\ f)\ x \\
&= \lambda f\ x.\ (\lambda f\ x.\ f^m\ x)\ (n\ f)\ x \\
&= \lambda f\ x.\ (\lambda x.\ (n\ f)^m\ x)\ x \\
&= \lambda f\ x.\ (n\ f)^m\ x \\
&= \lambda f\ x.\ ((\lambda f\ x.\ f^n\ x)\ f)^m\ x \\
&= \lambda f\ x.\ ((\lambda x.\ f^n\ x)^m\ x \\
&= \lambda f\ x.\ (f^n)^m\ x \\
&= \lambda f\ x.\ f^{mn} x
\end{aligned}
$$

# m ∗ n, korak po korak

m ∗ n

$= \lambda f.\lambda x.\ m\ (n\ f)\ x$ [def. ∗]

$= \lambda f.\lambda x.\ (\lambda f.\lambda x.\ f^m\ x)\ (n\ f)\ x$ [def. broja m]

$\rightarrow_\beta \lambda f.\lambda x.\ (\lambda x.\ (n\ f)^m\ x)\ x$ [f := n f]

$\rightarrow_\beta \lambda f.\lambda x.\ (n\ f)^m\ x$ [x := x]

$= \lambda f.\lambda x.\ ((\lambda f.\lambda x.\ f^n\ x)\ f)^m\ x$ [def. broja n]

$\rightarrow_\beta^* \lambda f.\lambda x.\ (\lambda x.\ f^n\ x)^m\ x$ [f := f, u svakoj od m kopija]

$\rightarrow_\beta^* \lambda f.\lambda x.\ f^{nm}\ x$ [m puta: svaka primena dodaje n primena f]

= m ∗ n (broj)

# Zadaci

17. Izračunati proizvod brojeva 4 i 3 u lambda notaciji.

# Rešenje zadatka 17 (1/2)

$4 * 3$

$= \lambda f.\lambda x.\ 4\ (3\ f)\ x$ [def. $*$]

$= \lambda f.\lambda x.\ (\lambda f.\lambda x.\ f\ (f\ (f\ (f\ x))))\ (3\ f)\ x$ [def. 4]

$\rightarrow_\beta \lambda f.\lambda x.\ (\lambda x.\ 3\ f\ (3\ f\ (3\ f\ (3\ f\ x))))\ x$ [f := 3 f]

$\rightarrow_\beta \lambda f.\lambda x.\ 3\ f\ (3\ f\ (3\ f\ (3\ f\ x)))$ [x := x]

- $3\ f = (\lambda g.\lambda y.\ g\ (g\ (g\ y)))\ f \rightarrow_\beta \lambda y.\ f\ (f\ (f\ y)) \equiv F$
- (u broju 3 su f, x preimenovani u g, y; $f^3\ y$ znači $f\ (f\ (f\ y))$)

$\rightarrow_\beta \lambda f.\lambda x.\ F\ (3\ f\ (3\ f\ (3\ f\ x)))$ [prvo 3 f: g := f]

$\rightarrow_\beta \lambda f.\lambda x.\ F\ (F\ (3\ f\ (3\ f\ x)))$ [drugo 3 f]

$\rightarrow_\beta \lambda f.\lambda x.\ F\ (F\ (F\ (3\ f\ x)))$ [treće 3 f]

$\rightarrow_\beta \lambda f.\lambda x.\ F\ (F\ (F\ (F\ x)))$ [četvrto 3 f]

# Rešenje zadatka 17 (2/2)

$\lambda f.\lambda x.\ F\ (F\ (F\ (F\ x)))$ $\quad F \equiv \lambda y.\ f^3\ y$

$\rightarrow_\beta\ \lambda f.\lambda x.\ F\ (F\ (F\ (f^3\ x)))$ [najunutrašnjije F x: y := x]

$\rightarrow_\beta\ \lambda f.\lambda x.\ F\ (F\ (f^6\ x))$ [y := $f^3$ x]

$\rightarrow_\beta\ \lambda f.\lambda x.\ F\ (f^9\ x)$ [y := $f^6$ x]

$\rightarrow_\beta\ \lambda f.\lambda x.\ f^{12}\ x$ [y := $f^9$ x]

$=\ 12$ [def. broja 12]

# Stepenovanje

- $a^b$ = b a  (a, b su Čerčovi brojevi: broj b se primenjuje na broj a)
- Primer: $3^2$ = 2 3 (sledeći slajd).

# Primer: $3^2 = 2\ 3$

$2\ 3$

$= (\lambda f.\lambda x.\ f\ (f\ x))\ 3$ [def. 2]

$\rightarrow_\beta \lambda x.\ 3\ (3\ x)$ [f := 3]

$= \lambda x.\ 3\ ((\lambda g.\lambda y.\ g\ (g\ (g\ y)))\ x)$ [def. 3, α-preimenovano]

$\rightarrow_\beta \lambda x.\ 3\ (\lambda y.\ x^3\ y)$ [g := x; $x^3\ y = x\ (x\ (x\ y))$]

$= \lambda x.\ (\lambda g.\lambda z.\ g\ (g\ (g\ z)))\ (\lambda y.\ x^3\ y)$ [def. 3, α-preimenovano]

$\rightarrow_\beta \lambda x.\lambda z.\ (\lambda y.\ x^3\ y)\ ((\lambda y.\ x^3\ y)\ ((\lambda y.\ x^3\ y)\ z))$ [$g := \lambda y.\ x^3\ y$]

$\rightarrow_\beta \lambda x.\lambda z.\ (\lambda y.\ x^3\ y)\ ((\lambda y.\ x^3\ y)\ (x^3\ z))$ [y := z]

$\rightarrow_\beta \lambda x.\lambda z.\ (\lambda y.\ x^3\ y)\ (x^6\ z)$ [$y := x^3\ z$]

$\rightarrow_\beta \lambda x.\lambda z.\ x^9\ z$ [$y := x^6\ z$]

$\equiv_\alpha \lambda f.\lambda x.\ f^9\ x\ =\ 9$ [x preimenovano u f, z u x]

# KOMBINATORI

# Kombinatori

- Kombinator je lambda izraz bez slobodnih promenljivih (zatvoren izraz) – samostalan „program”;
- Zašto su važni: S i K su dovoljni da se izrazi svaki lambda izraz, pa se vezane promenljive i λ mogu eliminisati (Šenfinkel 1924, Kari);
  - Na tome se zasnivaju neke tehnike prevođenja funkcionalnih jezika (Terner: prevođenje u S, K, I); kombinator Y daje rekurziju bez imena (deo o rekurziji);

$$
\begin{aligned}
I &= \lambda x.\ x \\
K &= \lambda x\ y.\ x \\
S &= \lambda f\ g\ x.\ (f\ x)(g\ x)
\end{aligned}
$$

$I$ – funkcija identiteta;
K – K x je konstantna funkcija koja uvek vraća x;
S – prosleđuje isti argument dvema funkcijama: S f g x = (f x) (g x);

λx.y ← Ovo nije kombinator jer je y slobodna promenljiva.

# Primeri: I, K i S na argumentima

$I\ a = (\lambda x.\ x)\ a \rightarrow_\beta a$ [x := a]

$K\ a\ b = (\lambda x.\lambda y.\ x)\ a\ b$ [def. K]

$\rightarrow_\beta (\lambda y.\ a)\ b$ [x := a]

$\rightarrow_\beta a$ [y := b; y se ne javlja u telu]

$S\ p\ q\ a = (\lambda f.\lambda g.\lambda x.\ (f\ x)\ (g\ x))\ p\ q\ a$ [def. S]

$\rightarrow_\beta (\lambda g.\lambda x.\ (p\ x)\ (g\ x))\ q\ a$ [f := p]

$\rightarrow_\beta (\lambda x.\ (p\ x)\ (q\ x))\ a$ [g := q]

$\rightarrow_\beta (p\ a)\ (q\ a)$ [x := a]

- K odbacuje drugi argument; S prosleđuje a i funkciji p i funkciji q

# Primer: S K K a $\rightarrow_*$ a

S K K a

$=$ $(\lambda f.\lambda g.\lambda x.\ (f\ x)\ (g\ x))$ K K a [def. S]

$\rightarrow_\beta$ $(\lambda g.\lambda x.\ (K\ x)\ (g\ x))$ K a [f := K]

$\rightarrow_\beta$ $(\lambda x.\ (K\ x)\ (K\ x))$ a [g := K]

$\rightarrow_\beta$ (K a) (K a) [x := a]

$=$ $(\lambda x.\lambda y.\ x)$ a (K a) [def. K (prvo K)]

$\rightarrow_\beta$ $(\lambda y.\ a)$ (K a) [x := a]

$\rightarrow_\beta$ a [y := K a; y se ne javlja u telu]

- Za svaki argument a važi S K K a = a = I a

# Primer: S K K $\to_*$ I (bez argumenta)

S K K

$=$ $(\lambda f.\lambda g.\lambda x.\ (f\ x)\ (g\ x))\ K\ K$ [def. S]

$\to_\beta$ $(\lambda g.\lambda x.\ (K\ x)\ (g\ x))\ K$ [f := K]

$\to_\beta$ $\lambda x.\ (K\ x)\ (K\ x)$ [g := K]

$=$ $\lambda x.\ ((\lambda x.\lambda y.\ x)\ x)\ (K\ x)$ [def. K (prvo K)]

$\to_\beta$ $\lambda x.\ (\lambda y.\ x)\ (K\ x)$ [x := x]

$\to_\beta$ $\lambda x.\ x\ =\ I$ [y := K x; y se ne javlja u telu]

# Primer: kombinator B kao lambda izraz

- B je kompozicija funkcija: B f g x = f (g x)

$B \triangleq \lambda f.\lambda g.\lambda x.\ f\ (g\ x)$

$B\ p\ q\ a$

$= (\lambda f.\lambda g.\lambda x.\ f\ (g\ x))\ p\ q\ a$ [def. B]

$\rightarrow_\beta (\lambda g.\lambda x.\ p\ (g\ x))\ q\ a$ [f := p]

$\rightarrow_\beta (\lambda x.\ p\ (q\ x))\ a$ [g := q]

$\rightarrow_\beta p\ (q\ a)$ [x := a]

- Primer:

$B\ (+\ 1)\ (*\ 2)\ 5$

$\rightarrow_\beta^* (+\ 1)\ ((*\ 2)\ 5) = +\ 1\ (*\ 2\ 5)$ [kao gore: p := + 1, q := * 2, a := 5]

$\rightarrow_\delta +\ 1\ 10$

$\rightarrow_\delta 11$

# LAMBDA RAČUN KAO PROGRAMSKI JEZIK

REKURZIVNE FUNKCIJE

# Rekurzivne funkcije

- Definicija rekurzivne funkcije;
- Predstavljanje rekurzivnih funkcija u lambda računu?
- Rekurzivna definicija funkcije sum (koja sabira brojeve od 0 do $n$);

$$\text{sum n} = \text{if n} = 0 \text{ then } 0 \text{ else n} + \text{sum } (\text{n} - 1)$$

- Funkcija sum kao lambda apstrakcija sa dodatnim parametrom;

$$\text{SUM} \triangleq \lambda \text{s}.\lambda \text{n}.\ \text{if } (\text{eq n } 0)\ 0\ (+\ \text{n } (\text{s } (-\ \text{n } 1)))$$

# Zašto „dodatni parametar” s?

- Lambda izraz nema ime, pa ne može da pozove sam sebe
- Zato rekurzivni poziv postaje parametar s: SUM s n koristi s umesto „sum”
- Tražimo funkciju sum takvu da je sum = SUM sum: fiksnu tačku od SUM
- Takvu fiksnu tačku daje kombinator Y (sledeći slajdovi)
- δ-pravila za konstante u primerima: if true a b → a, if false a b → b
- eq 0 0 → true, eq 2 0 → false, − 2 1 → 1, + 2 1 → 3, ...

# Rekurzivne funkcije

- Y – kombinator fiksne tačke;
- $Y\ f\ =\ f\ (Y\ f)$
- Primena na lambda definiciju funkcije sum;

```
Y SUM = Y (λ s.λ n. if (eq n  0) 0 (+  n (s (- n 1))))
      = (λ s.λ n. if (eq n  0) 0 (+  n (s (- n 1)))) (Y SUM)
      = (λ n. if (eq n  0) 0 (+ n ((Y SUM) (- n 1))))
```

$$= \lambda n.\ \text{if}\ (\text{eq}\ n\ 0)$$
$$0$$
$$(+\ n\ ((\lambda n.\ \text{if}\ (\text{eq}\ n\ 0)\ 0\ (+\ n\ ((\text{Y SUM})\ (-\ n\ 1))))\ (-\ n\ 1)))$$

# Y SUM, korak po korak

Y SUM

$=$ SUM (Y SUM) [Y f = f (Y f) – dokaz posle definicije Y]

$=$ $(\lambda s.\lambda n.$ if (eq n 0) 0 (+ n (s (− n 1)))) (Y SUM) [def. SUM]

$\rightarrow_\beta$ $\lambda n.$ if (eq n 0) 0 (+ n ((Y SUM) (− n 1))) [s := Y SUM]

- Y SUM se ponaša kao sum: za n ≠ 0 vraća n + (Y SUM) (n – 1)

# Definicija Y

- Y kao lambda izraz:

$$\lambda h.(\lambda x.h\ (x\ x))(\lambda x.h\ (x\ x))$$

- Pokazati da je Y h = h (Y h) za proizvoljnu funkciju h (dokaz na sledećem slajdu).

# Dokaz: Y h = h (Y h)

Y h

$=$ $(\lambda h.\ (\lambda x.\ h\ (x\ x))\ (\lambda x.\ h\ (x\ x)))\ h$ [def. Y]

$\rightarrow_\beta$ $(\lambda x.\ h\ (x\ x))\ (\lambda x.\ h\ (x\ x))$ [h := h]

$\rightarrow_\beta$ $h\ ((\lambda x.\ h\ (x\ x))\ (\lambda x.\ h\ (x\ x)))$ [x := λx. h (x x)]

h (Y h) $=$ $h\ ((\lambda h.\ (\lambda x.\ h\ (x\ x))\ (\lambda x.\ h\ (x\ x)))\ h)$ [def. Y]

$\rightarrow_\beta$ $h\ ((\lambda x.\ h\ (x\ x))\ (\lambda x.\ h\ (x\ x)))$ [h := h, unutar h (...)]

- Oba izraza se svode na isti izraz, pa je Y h = h (Y h)
- Dakle Y h je fiksna tačka funkcije h – za svaku funkciju h
- Aplikativnim redosledom: h (h (h ...)) – računanje se ne završava

# Primer: (Y SUM) 2 (1/3)

$G \equiv \lambda n.\ \text{if}\ (\text{eq}\ n\ 0)\ 0\ (+\ n\ ((Y\ \text{SUM})\ (-\ n\ 1)))$ $(\text{SUM}\ (Y\ \text{SUM}) \rightarrow_\beta G)$

$(Y\ \text{SUM})\ 2$

$=\ \text{SUM}\ (Y\ \text{SUM})\ 2$ [Y f = f (Y f)]

$\rightarrow_\beta\ G\ 2$ [s := Y SUM]

$\rightarrow_\beta\ \text{if}\ (\text{eq}\ 2\ 0)\ 0\ (+\ 2\ ((Y\ \text{SUM})\ (-\ 2\ 1)))$ [n := 2]

$\rightarrow_\delta\ \text{if false}\ 0\ (+\ 2\ ((Y\ \text{SUM})\ (-\ 2\ 1)))$ [eq 2 0 → false]

$\rightarrow_\delta\ +\ 2\ ((Y\ \text{SUM})\ (-\ 2\ 1))$ [if false a b → b]

$\rightarrow_\delta\ +\ 2\ ((Y\ \text{SUM})\ 1)$ [− 2 1 → 1]

# Primer: (Y SUM) 2 (2/3)

$+\ 2\ ((\mathrm{Y}\ \mathrm{SUM})\ 1)$

$=\ +\ 2\ (\mathrm{SUM}\ (\mathrm{Y}\ \mathrm{SUM})\ 1)$ [Y f = f (Y f)]

$\rightarrow_{\beta}\ +\ 2\ (\mathrm{G}\ 1)$ [s := Y SUM]

$\rightarrow_{\beta}\ +\ 2\ (\mathrm{if}\ (\mathrm{eq}\ 1\ 0)\ 0\ (+\ 1\ ((\mathrm{Y}\ \mathrm{SUM})\ (-\ 1\ 1))))$ [n := 1]

$\rightarrow_{\delta}\ +\ 2\ (\mathrm{if\ false}\ 0\ (+\ 1\ ((\mathrm{Y}\ \mathrm{SUM})\ (-\ 1\ 1))))$ [eq 1 0 → false]

$\rightarrow_{\delta}\ +\ 2\ (+\ 1\ ((\mathrm{Y}\ \mathrm{SUM})\ (-\ 1\ 1)))$ [if false a b → b]

$\rightarrow_{\delta}\ +\ 2\ (+\ 1\ ((\mathrm{Y}\ \mathrm{SUM})\ 0))$ [− 1 1 → 0]

# Primer: (Y SUM) 2 (3/3)

$+\ 2\ (+\ 1\ ((\mathrm{Y}\ \mathrm{SUM})\ 0))$

$=\ +\ 2\ (+\ 1\ (\mathrm{SUM}\ (\mathrm{Y}\ \mathrm{SUM})\ 0))$ [Y f = f (Y f)]

$\rightarrow_{\beta}\ +\ 2\ (+\ 1\ (\mathrm{G}\ 0))$ [s := Y SUM]

$\rightarrow_{\beta}\ +\ 2\ (+\ 1\ (\mathrm{if}\ (\mathrm{eq}\ 0\ 0)\ 0\ (+\ 0\ ((\mathrm{Y}\ \mathrm{SUM})\ (-\ 0\ 1)))))$ [n := 0]

$\rightarrow_{\delta}\ +\ 2\ (+\ 1\ (\mathrm{if\ true}\ 0\ (+\ 0\ ((\mathrm{Y}\ \mathrm{SUM})\ (-\ 0\ 1)))))$ [eq 0 0 → true]

$\rightarrow_{\delta}\ +\ 2\ (+\ 1\ 0)$ [if true a b → a: rekurzija se zaustavlja]

$\rightarrow_{\delta}\ +\ 2\ 1$ [+ 1 0 → 1]

$\rightarrow_{\delta}\ 3$ [+ 2 1 → 3]

- Grana sa (– 0 1) se nikad ne računa – zato se rekurzija zaustavlja
- Aplikativnim redosledom razvijanje Y SUM se nikad ne bi završilo

# KARIJEVE FUNKCIJE

# Karijeve funkcije

- Funkcije sa više argumenata moguće je definisati koristeći funkciju kao povratnu vrednost:

add' uzima kao argument ceo broj x i vraća kao rezultat funkciju add' x. Ta funkcija uzima ceo broj y i vraća rezultat x + y.

```
add'     :: Int → (Int → Int)
add' x y = x+y
```

# add’ kao lambda izraz

add’ ≜ λx.λy. + x y

add’ 3 4

= ((λx.λy. + x y) 3) 4 [def. add’; primena je levo asocijativna]

$\rightarrow_\beta$ (λy. + 3 y) 4 [x := 3 – ovo je funkcija add’ 3]

$\rightarrow_\beta$ + 3 4 [y := 4]

$\rightarrow_\delta$ 7

- Parcijalna primena: add’ 1 $\rightarrow_\beta$ λy. + 1 y – funkcija „dodaj 1”

# Karijeve funkcije

```
add  :: (Int,Int) → Int

add' :: Int → (Int → Int)
```

- add i add' imaju isti konačan rezultat, ali add uzima oba argumenta istovremeno, dok add' uzima jedan po jedan argument.
- Funkcije koje uzimaju jedan po jedan argument zovu se Karijeve funkcije, po Haskelu Kariju (Haskell Curry).

# Karijeve funkcije

- Funkcije sa više od dva argumenta mogu se definisati kao Karijeve korišćenjem ugnežđenih funkcija kao povratnih vrednosti:

mult uzima argument $x$ i vraća funkciju mult $x$, koja slično uzima argument $y$ i vraća funkciju mult $x\ y$, koja na kraju uzima argument $z$ i vraća rezultat $x * y * z$.

```
mult       :: Int → (Int → (Int → Int))
mult x y z = x*y*z
```

# mult kao lambda izraz

$\text{mult} \triangleq \lambda x.\lambda y.\lambda z.\ *\ (*\ x\ y)\ z$

mult 2 3 4

$=\ ((\lambda x.\lambda y.\lambda z.\ *\ (*\ x\ y)\ z)\ 2)\ 3\ 4$ [def. mult; primena je levo asocijativna]

$\rightarrow_\beta\ (\lambda y.\lambda z.\ *\ (*\ 2\ y)\ z)\ 3\ 4$ [x := 2 – mult 2 :: Int → (Int → Int)]

$\rightarrow_\beta\ (\lambda z.\ *\ (*\ 2\ 3)\ z)\ 4$ [y := 3 – mult 2 3 :: Int → Int]

$\rightarrow_\beta\ *\ (*\ 2\ 3)\ 4$ [z := 4]

$\rightarrow_\delta\ *\ 6\ 4$

$\rightarrow_\delta\ 24$

# Karijeve funkcije

- Zašto su Karijeve funkcije korisne?
  - Karijeve funkcije su fleksibilnije od funkcija nad torkama jer se parcijalnom primenom Karijevih funkcija dobijaju nove korisne funkcije (npr. add' 1, mult 2).

- Konvencije u zapisu Karijevih funkcija
  - Strelica → je desno asocijativna.

Int → (Int → (Int → Int))

```
Int → Int → Int → Int
```

  - Primena funkcija je levo asocijativna.

((mult x) y) z

```
mult x y z
```

# TIPOVI

# Tipovi

- Tipovi – različite vrste podataka (prirodni brojevi, logičke vrednosti, funkcije);
- Svrha?
- Zašto dodati tipove i u lambda račun?
  - Logički aspekt: sprečavanje besmislenih izraza (npr. samoprimene x x); primena funkcija samo na podatke iz domena;
  - Programerski aspekt: Generisanje efikasnijeg koda; Lakše otkrivanje grešaka; netipizirani jezici; slabo tipizirani jezici; dinamički tipizirani jezici; statički tipizirani jezici;
- Statički tipiziran jezik i fleksibilnost koda?
  - Polimorfizam;
  - Određivanje tipa izraza;

# Odgovori na pitanja sa prethodnog slajda

- Svrha tipova: greške se otkrivaju pre izvršavanja programa
- npr.  primena broja kao funkcije (3 5) otkriva se pri prevođenju
- Tip je i dokumentacija, a prevodilac pravi efikasniji kod
- Statički tipiziran jezik može biti fleksibilan – polimorfizam:
- jedna definicija radi za mnogo tipova, npr.

$\lambda x.\, x : \alpha \to \alpha$  (radi i kao $\text{int} \to \text{int}$ i kao $\text{bool} \to \text{bool}$)

- Zaključivanje tipa: tip ne mora da se piše, izvodi se (Haskell, ML)

# Tipizirani lambda račun

- U tipiziranom lambda računu svaki ispravno formiran izraz ima tip.
- Izraz s može da se primeni na izraz t ako je s tipa $\sigma \rightarrow \tau$, a t tipa $\sigma$; tada je s t tipa $\tau$.
  - Primer: sledeći slajd.
- Strogo tipiziran jezik;
  - Da li je C strogo tipiziran?
- Notacija
  - $t : \sigma$ u značenju da je $t$ tipa $\sigma$;
  - $f : \sigma \rightarrow \tau$
  - Tipovi su skupovi kojima pripada određen objekat; $t : \sigma$ tj. $t \in \sigma$ .

# Primer: tipovi i primena

$\text{suc} : \text{int} \to \text{int}, \quad 3 : \text{int} \quad \Rightarrow \quad \text{suc}\ 3 : \text{int}$

- suc true nema tip: true : bool, a suc očekuje int
- Da li je C strogo tipiziran? Ne sasvim:
- konverzije (int $\leftrightarrow$ double), kastovanje i void* propuštaju neke greške tipa
- Haskell je strogo i statički tipiziran: not 3 je greška pri prevođenju

# Vrste tipova

- Šta su tačno tipovi?
- Prosti tipovi (bool, int);
- Složeni tipovi (*composite type*) formiramo na osnovu konstruktora tipova (*type constructor*);
- C – skup prostih tipova; skup svih tipova $\mathrm{Ty}_C$ gradi se nad C pravilima:
  - Primer: sledeći slajd.

$$\frac{\sigma \in C}{\sigma \in Ty_C}$$

$$\frac{\sigma \in Ty_C \quad \tau \in Ty_C}{\sigma \to \tau \in Ty_C}$$

# Primer: izvođenje tipova iz $\mathrm{Ty}_C$

$C = \{\mathrm{int}, \mathrm{bool}\}$

$\mathrm{int} \in C \Rightarrow \mathrm{int} \in \mathrm{Ty}_C$ [prvo pravilo]

$\mathrm{bool} \in C \Rightarrow \mathrm{bool} \in \mathrm{Ty}_C$ [prvo pravilo]

$\mathrm{int} \in \mathrm{Ty}_C, \mathrm{bool} \in \mathrm{Ty}_C \Rightarrow \mathrm{int} \to \mathrm{bool} \in \mathrm{Ty}_C$ [drugo pravilo]

$\mathrm{int} \to \mathrm{bool} \in \mathrm{Ty}_C, \mathrm{int} \in \mathrm{Ty}_C \Rightarrow (\mathrm{int} \to \mathrm{bool}) \to \mathrm{int} \in \mathrm{Ty}_C$ [drugo pravilo]

- Strelica je desno asocijativna: $\mathrm{int} \to \mathrm{int} \to \mathrm{int} = \mathrm{int} \to (\mathrm{int} \to \mathrm{int})$

# Vrste tipova

- Tipske promenljive (α, β, ...) – stoje umesto proizvoljnog tipa;
- Konstruktori tipova (ne samo →);
  - Npr.

$$\frac{\sigma \in Ty_C \quad \tau \in Ty_C}{\sigma \times \tau \in Ty_C}$$

- Proizvoljan skup konstruktora proizvoljne arnosti, npr. (int)list – u Haskell-u [Int];

$$(\alpha_1, \ldots, \alpha_n)con$$

- Tip nikada nije jednak svom pravom delu: ne postoji σ takav da je σ = σ → τ (zato x x nema tip).

# Primer: proizvod tipova i samoprimena

$\text{int} \in \text{Ty}_C, \text{bool} \in \text{Ty}_C \Rightarrow \text{int} \times \text{bool} \in \text{Ty}_C$ [pravilo za ×]

- npr. par (3, true) je tipa int × bool
- Zašto x x nema tip:
- u x x promenljiva x mora biti funkcija tipa $\sigma \to \tau$, a i argument tipa $\sigma$
- x ima jedan tip, pa bi bilo $\sigma = \sigma \to \tau$: tip jednak svom delu – nemoguće
- Zato ni $\Omega = (\lambda x.\ x\ x)\ (\lambda x.\ x\ x)$ (beskonačno računanje) nema tip

# Čerčovo i Karijevo tipiziranje

- Čerčovo tipiziranje – eksplicitno
  - Svaki izraz ima tačno jedan tip (tip promenljive je zadat uz nju).

$v : \sigma$ (promenljiva nosi svoj tip)

Konstanta c je tipa $\sigma \Rightarrow c : \sigma$

$s : \sigma \to \tau,\ t : \sigma \Rightarrow s\ t : \tau$

$v : \sigma,\ t : \tau \Rightarrow \lambda v.\ t : \sigma \to \tau$

# Čerčovo i Karijevo tipiziranje

- Karijevo tipiziranje – implicitno
  - Izraz ne mora da ima tip; ako ga ima, može da ima više tipova (npr. λx. x : int → int i λx. x : bool → bool).
  - Polimorfizam;
  - Notacija: u kontekstu Γ (pretpostavke o tipovima promenljivih, npr. Γ = {x : int}) izraz t ima tip σ:

$$\Gamma \vdash t : \sigma$$

# Formalna pravila tipiziranosti

$v : \sigma \in \Gamma \Rightarrow \Gamma \vdash v : \sigma$

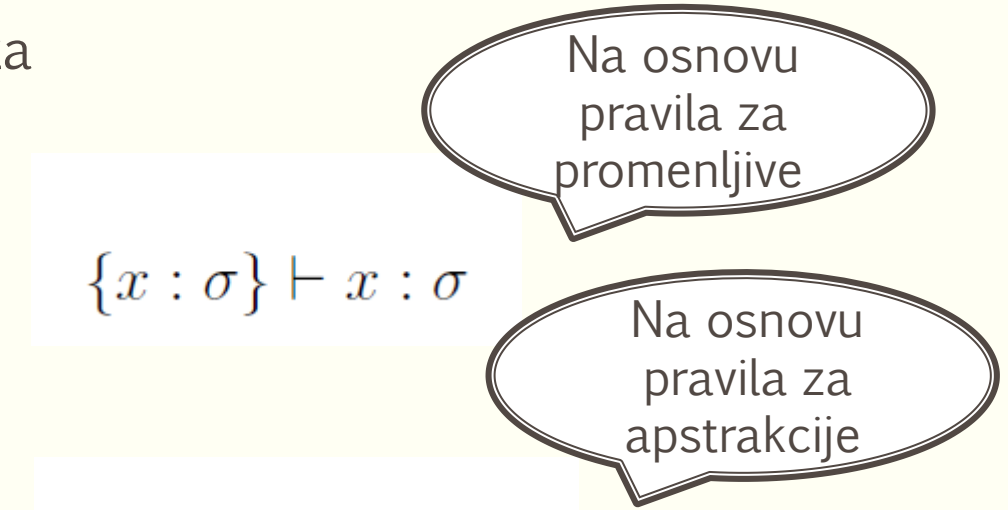
Konstanta c je tipa $\sigma \Rightarrow \Gamma \vdash c : \sigma$

$\Gamma \vdash s : \sigma \to \tau,\ \Gamma \vdash t : \sigma \Rightarrow \Gamma \vdash s\ t : \tau$

$\Gamma \cup \{v : \sigma\} \vdash t : \tau \Rightarrow \Gamma \vdash \lambda v.\ t : \sigma \to \tau$

# Formalna pravila tipiziranosti

- Izraz je datog tipa ako se to može izvesti iz prethodnih pravila.
- Na primer, funkcija identiteta




- Zašto je potreban kontekst u Karijevom tipiziranju?

# Primer: tip izraza λx.λy. x (kombinator K)

1. $\{x : \sigma, y : \tau\} \vdash x : \sigma$ [pravilo za promenljive]
2. $\{x : \sigma\} \vdash \lambda y.\, x : \tau \to \sigma$ [pravilo za apstrakciju, iz 1]
3. $\emptyset \vdash \lambda x.\lambda y.\, x : \sigma \to \tau \to \sigma$ [pravilo za apstrakciju, iz 2]

- Kontekst je potreban jer promenljive nemaju unapred zadat tip:
- tip slobodne promenljive dolazi iz pretpostavki (iz Γ)
- $\emptyset \vdash + x\ 1 : \text{int}$ ne može da se izvede, ali $\{x : \text{int}\} \vdash + x\ 1 : \text{int}$ može

# Primer: tip Čerčovog broja 2

- Neka je $\Gamma = \{f : \alpha \rightarrow \alpha, x : \alpha\}$

1. $\Gamma \vdash f : \alpha \rightarrow \alpha$ [promenljiva]
2. $\Gamma \vdash x : \alpha$ [promenljiva]
3. $\Gamma \vdash f\ x : \alpha$ [primena, 1 i 2]
4. $\Gamma \vdash f\ (f\ x) : \alpha$ [primena, 1 i 3]
5. $\{f : \alpha \rightarrow \alpha\} \vdash \lambda x.\ f\ (f\ x) : \alpha \rightarrow \alpha$ [apstrakcija, 4]
6. $\emptyset \vdash \lambda f.\lambda x.\ f\ (f\ x) : (\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha$ [apstrakcija, 5]

# Polimorfizam

- Polimorfizam vs. preopterećivanje (overloading) vs. podtipovi (subtyping)?

$$\text{if } (\lambda x.\ x) \text{ true then } (\lambda x.\ x)\ 1 \text{ else } 0$$




$$\dfrac{\dfrac{\dfrac{\{x : bool\} \vdash x : bool}{\vdash (\lambda x.\ x) : bool \to bool} \quad \dfrac{}{\vdash \text{true} : bool}}{\vdash (\lambda x.\ x) \text{ true} : bool} \quad \dfrac{\dfrac{\dfrac{\{x : int\} \vdash x : int}{\vdash (\lambda x.\ x) : int \to int} \quad \dfrac{}{\vdash 1 : int}}{\vdash (\lambda x.\ x)\ 1 : int} \quad \dfrac{}{\vdash 0 : int}}{\vdash \text{if } (\lambda x.\ x) \text{ true then } (\lambda x.\ x)\ 1 \text{ else } 0 : int}$$

# Isto izvođenje, red po red

1. $\{x : bool\} \vdash x : bool$ [promenljiva]
2. $\vdash \lambda x.\ x : bool \to bool$ [apstrakcija, 1]
3. $\vdash (\lambda x.\ x)\ true : bool$ [primena, 2 i true : bool]
4. $\{x : int\} \vdash x : int$ [promenljiva]
5. $\vdash \lambda x.\ x : int \to int$ [apstrakcija, 4]
6. $\vdash (\lambda x.\ x)\ 1 : int$ [primena, 5 i 1 : int]

- Konstanta COND (if-then-else) je ovde tipa $bool \to int \to int \to int$

7. $\vdash$ if $(\lambda x.\ x)$ true then $(\lambda x.\ x)$ 1 else 0 : int [COND; 3, 6 i 0 : int]

- Funkcija $\lambda x.\ x$ dobila je dva tipa – polimorfizam; najopštiji je $\alpha \to \alpha$
- Preopterećivanje: isto ime za različite funkcije (+ za int i za double)
- Podtipovi: vrednost podtipa može da se koristi gde se očekuje nadtip

# Opštiji tipovi

- Izraz može imati mnogo tipova: λx. x ima tipove int → int, bool → bool, (int → int) → (int → int), ...; svi se dobijaju zamenom α u α → α – to je njegov najopštiji tip.
- Tipovi se grade primenom konstruktora tipova na tipske promenljive (α, β, ...) i tipske konstante (int, bool).
- Zamene – npr. $(\sigma \to bool)[(\sigma \to \tau)/\sigma] = (\sigma \to \tau) \to bool.$
- Formalno

$$\alpha i[\tau 1/\alpha 1, ..., \tau k/\alpha k] = \tau i, \quad \text{za } 1 \leq i \leq k$$
$$\beta[\tau 1/\alpha 1, ..., \tau k/\alpha k] = \beta, \quad \text{ako je } \beta \neq \alpha i \text{ za sve } 1 \leq i \leq k$$
$$(\sigma 1, ..., \sigma n)con[\theta] = (\sigma 1[\theta], ..., \sigma n[\theta])con$$

# Primer: najopštiji tip izraza λf.λx. f x

- Neka je $\Gamma = \{f : \alpha \rightarrow \beta, x : \alpha\}$

1. $\Gamma \vdash f : \alpha \rightarrow \beta$ [promenljiva]
2. $\Gamma \vdash x : \alpha$ [promenljiva]
3. $\Gamma \vdash f\ x : \beta$ [primena, 1 i 2]
4. $\{f : \alpha \rightarrow \beta\} \vdash \lambda x.\ f\ x : \alpha \rightarrow \beta$ [apstrakcija, 3]
5. $\emptyset \vdash \lambda f.\lambda x.\ f\ x : (\alpha \rightarrow \beta) \rightarrow \alpha \rightarrow \beta$ [apstrakcija, 4]

- To je najopštiji tip; ostali tipovi su njegove instance:

$((\alpha \rightarrow \beta) \rightarrow \alpha \rightarrow \beta)[\text{int}/\alpha, \text{bool}/\beta] = (\text{int} \rightarrow \text{bool}) \rightarrow \text{int} \rightarrow \text{bool}$

$((\alpha \rightarrow \beta) \rightarrow \alpha \rightarrow \beta)[\alpha/\beta] = (\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha$

- Zamena se primenjuje istovremeno na sva pojavljivanja promenljive

# Opštiji tipovi

- $\sigma$ je opštiji tip od $\sigma'$, u oznaci $\sigma \preccurlyeq \sigma'$, ako postoji zamena $\theta$ takva da je $\sigma' = \sigma[\theta]$.
  - Relacija je refleksivna i tranzitivna;
- Na primer:

$$\begin{aligned} \alpha &\preceq \sigma \\ \alpha \to \alpha &\preceq \beta \to \beta \\ \alpha \to bool &\preceq (\beta \to \beta) \to bool \end{aligned}$$

$$\begin{aligned} \beta \to \alpha &\preceq \alpha \to \beta \\ \alpha \to \alpha &\not\preceq (\beta \to \beta) \to \beta \end{aligned}$$

# Zamene za primere sa prethodnog slajda

$\alpha \preccurlyeq \sigma \quad [\theta = \sigma/\alpha]$

$\alpha \to \alpha \preccurlyeq \beta \to \beta \quad [\theta = \beta/\alpha]$

$\alpha \to \text{bool} \preccurlyeq (\beta \to \beta) \to \text{bool} \quad [\theta = (\beta \to \beta)/\alpha]$

$\beta \to \alpha \preccurlyeq \alpha \to \beta \quad [\theta = \alpha/\beta, \beta/\alpha$ – istovremena zamena$]$

$\alpha \to \alpha \not\preccurlyeq (\beta \to \beta) \to \beta \quad [\alpha$ bi moralo biti i $\beta \to \beta$ i $\beta$ – nemoguće$]$