β-редукция

$\forall M, N \in \Lambda$

$$(\lambda x. M)\, N \to_\beta M[x := N]$$

β-эквивалентность

$$(\lambda x. M)\, N =_\beta M[x := N]$$

$M =_\beta N \Rightarrow ML =_\beta NL$

$LM =_\beta LN$

$M =_\beta N \Rightarrow \lambda x. M =_\beta \lambda x. N$

$M =_\beta N,\ N =_\beta L \Rightarrow M =_\beta L$

F2 $\forall MNL$ $\quad FMNL =_\beta ML(NL)$

$FMN =_\beta \lambda x. Mx(Nx)$

$FM =_\beta \lambda xy. Mx(yx)$

$F =_\beta \lambda xyz. zx(yx)$

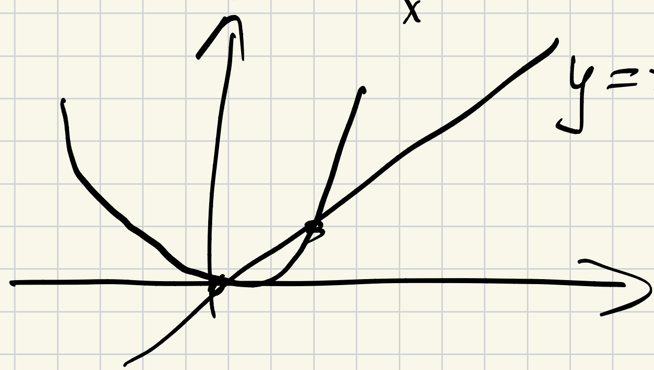
$FM = MNF$

$F =_\beta \lambda x. x\, NF$

Fixed-point

$f: D \to C$

$x \in D \quad f(x) = x$



λ, $\lambda \to$

Теорема

$\forall F \in \Lambda \quad \exists X \in \Lambda \qquad FX = X$

Док-во

$T = \lambda x.\ F(xx)$

$X = TT = (\lambda x.\ F(xx))\ \lambda x.\ F(xx) \to_\beta F((\lambda x.F(xx))\ \lambda x.\ F(xx))$

$= F(TT) = F\ X \qquad \square$

Теорема ($\exists$ комбинатор Y) $Y \in \Lambda$

$$\forall F \in \Lambda \quad \boxed{F(YF) \equiv_\beta YF}$$

Док-во

$$Y = \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$$

$$YF = \underbrace{\lambda f.(\lambda x.f(xx))(\lambda x.f(xx))\ F}_{YF} \to_\beta \underbrace{(\lambda x.F(xx))(\lambda x.F(xx))}_{YF} \to_\beta$$

$$\to_\beta F\underbrace{((\lambda x.F(xx))\lambda x.F(xx))}_{YF} = F(YF)$$

$$YF \equiv_\beta F(YF) \equiv_\beta F(F(YF))$$

$$\equiv_\beta F(F(F(YF)))$$

$$\forall F \quad \exists X \qquad FX = X$$

$$\exists Y \quad \forall F \qquad F(YF) = YF$$

if

$\mathbb{N}$

is_zero

mult

pred

$0, 1, 2 \in \mathbb{N}$

$fac : \mathbb{N} \to \mathbb{N}$

$n \times fact(n-1)$

$$fac = \lambda n.\ if\ \underbrace{(is\_zero\ n)}_{if}\ 1\ \underbrace{(mult\ n\ (fac\ (pred\ n)))}_{else}$$

$$F(YF) \equiv_\beta YF$$

$$fac \equiv_\beta \underbrace{(\lambda f n.\ if\ (is\_zero\ n)\ 1\ (mult\ n\ (f\ (pred\ n))))}_{F}\ fac$$

$$fac \equiv_\beta F\ fac$$

$$\Rightarrow \quad fac \equiv_\beta YF$$

$$fac\ 3 \equiv_\beta (YF)\ 3 \to_\beta$$

$\to_\beta$

$\twoheadrightarrow_\beta$

$$(F(YF))\ 3 =$$

$$= (\lambda f n.\ \underbrace{if\ (is\_zero\ n)\ 1\ (mult\ n\ (f\ (pred\ n)))}\ )(YF)\ 3 \twoheadrightarrow_\beta$$

$$if\ (is\_zero\ 3)\ 1\ (mult\ 3\ ((YF)\ (pred\ 3))) \to_\beta$$

$$mult\ 3\ ((YF)(pred\ 3)) \to_\beta$$

$$mult\ 3\ (F(YF)\ (pred\ 3)) \to mult\ 3\ (F(YF)\ 2) \to mult\ 3 (mu$$

$mult\ 3\ (mult\ 2\ (mult\ 1\ 1))$

Многошаговая бета-редукция $\twoheadrightarrow_\beta$

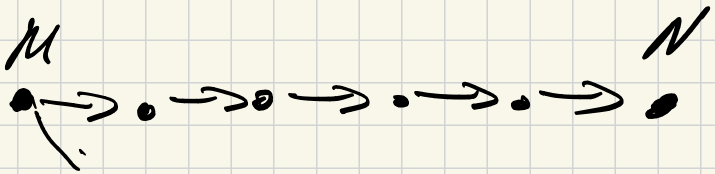
$\rightarrow_\beta$

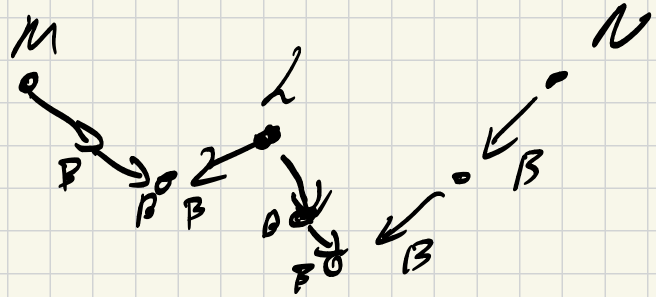
а) $\forall L \in \Lambda \quad L \twoheadrightarrow_\beta L$

б) $\forall M, N \in \Lambda \quad M \rightarrow_\beta N \Rightarrow M \twoheadrightarrow_\beta N$

в) $\forall M, N, L \in \Lambda \quad M \rightarrow_\beta N \quad N \rightarrow_\beta L \Rightarrow M \twoheadrightarrow_\beta L$

β-эквивалентность





$\forall M, M' \in \Lambda, \; M =_\beta M' \Leftrightarrow$

$\exists n \; M_i, i \in \{1, \dots, n\}, \quad \forall i \in \{1, \dots, n-1\} \quad M_i \rightarrow_\beta M_{i+1}$

$M_{i+1} \rightarrow_\beta M_i$

$$KI =_\beta II K_*$$

$$K = \lambda xy.x \qquad K_* = \lambda xy.y \qquad I = \lambda x.x$$

$$(\lambda xy.x) I \to_\beta \lambda y. I = \lambda y x. x = K_*$$

$$I I K_* \to (\lambda x.x)(\lambda x.x)\,\lambda xy.y \to_\beta (\lambda x.x) K_* \to_\beta K_*$$

$$\begin{tikzcd}
 & & IIK_* \arrow[d] \\
KI \arrow[dr] & & IK_* \arrow[dl] \\
 & K_* &
\end{tikzcd}$$

Нормальная форма

λ терм <u>находится</u> <u>β-nf</u> (БНФ), если $\nexists$ подтермов, которые явл-ся β-редексами

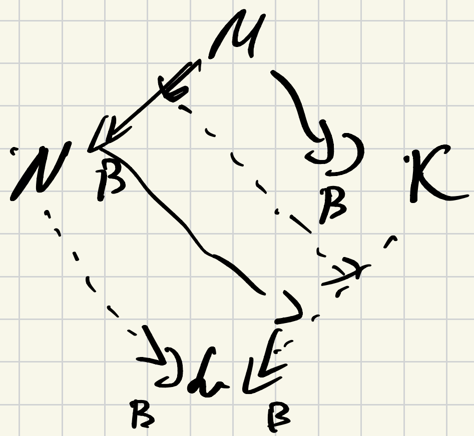
λ терм <u>имеет</u> β-nf N, если вып-ся $M =_\beta N$

$$KI\Omega = ((\lambda xy.x)\, I)\ \underbrace{((\lambda x.xx)\,(\lambda x.xx))}_{\Omega} \twoheadrightarrow_\beta I$$

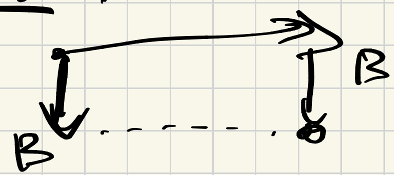
Теорема Чёрча-Россера (Confluence theorem)

$\forall\, M, N, K \in \Lambda$ если $M \twoheadrightarrow_\beta N$, $M \twoheadrightarrow_\beta K$, то тогда $\exists\, L \in \Lambda$, такой что $N \twoheadrightarrow_\beta L$, $K \twoheadrightarrow_\beta L$.

β-редукция обладает св-м ромба:



<u>Идея док-ва</u>:



Свойство редуцируемости к NF

если $M \in \Lambda$ $\exists N$-β-n.f, то тогда $M \twoheadrightarrow_\beta N$

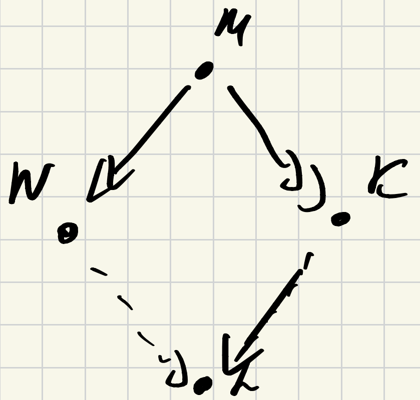
Док-во $\underline{M =_\beta N} \Rightarrow \exists L$ $M \twoheadrightarrow_\beta L$, $N \twoheadrightarrow_\beta L$

$\Downarrow$

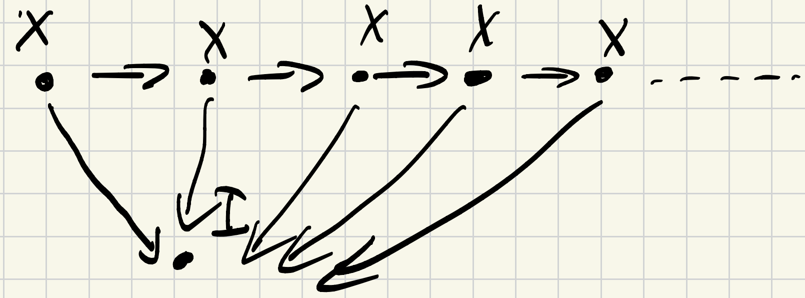
$N = L$

$\Omega = (\lambda x.xx)(\lambda x.xx)$

$\Omega \twoheadrightarrow_\beta \Omega$



$X = KI\Omega$



Стратегии редукции

v – переменная

$\lambda x.M$ — редуцируем внутри M

MN — редуцируем внутри M или N

$\underline{M}$

Нормальная стратегия — редуцирование самого левого редекса.

$$(\ldots((\lambda x.M)\, N_1)\, N_2 \ldots N_k)$$

Аппликативная стратегия — редуц-я N_i

$(\lambda x.\ xxx)\underline{N} \twoheadrightarrow_\beta \lambda x.\ N'N'N'$ and

$N \twoheadrightarrow_\beta N'$

$(\lambda x.\ xxx)N \to_\beta NNN$

Pair

pair = $\lambda abf.\ fab$

fst

snd

$\text{pair}\ ab\ \text{fst} \twoheadrightarrow_\beta a$

$\text{pair}\ ab\ \text{snd} \twoheadrightarrow_\beta b$

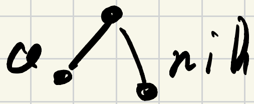
$\text{pair}\ ab = \lambda f.\ fab$

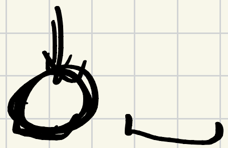
List

nil

pair(a, nil)

pair(pair(a, nil), b)



pair 

true
false

head

tail

cons

$cons = \lambda h t f.\ f\,h\,t$ $\qquad$ $\underline{pair(1, nil)}$ $\qquad$ $\underline{nil}$

$cons\ 1\ \underline{nil} \twoheadrightarrow_\beta \lambda f.\ f\ 1\ nil = l_1$

$cons\ 2\ l_1 \twoheadrightarrow_\beta \lambda f.\ f\ 2\ l_1 = \lambda f.\ f\,2\ (\lambda f.\ f\,1\ nil) = l_2$

$cons\ 3\ l_2 \twoheadrightarrow_\beta \lambda f.\ f\ 3\ l_2 = l_3$

$cons\ l_2\ l_3 \twoheadrightarrow_\beta \lambda f. f\,l_2\,l_3 = \lambda f.\ f\,(\lambda f. f\,2\ (f\,1\ nil))\ l_3 \to_\beta$

$\to \lambda f.\ f\,(l_3\ 2)\,(l_3\ 1\ nil) \to$

$\underline{head} = \lambda l.\ l\ (\lambda x y.\ x)$

$head\ l_3 = (\lambda l.\ l(\lambda x y.x))\,l_3 \to_\beta l_3\ (\lambda x y.\ x) =$

$= (\lambda f.\ f\ 3\, l_2)\ (\lambda x\,y.\ x) \twoheadrightarrow_\beta 3$

$\underline{tail} = \lambda l.\ l\ (\lambda x y.\ y)$

$is\_empty \; : \; List \Rightarrow_\beta bool$

$is\_empty \; \ell \Rightarrow_\beta false$

$is\_empty \; nil \Rightarrow_\beta true$

$\ell \; is\_empty' \Rightarrow_\beta false$

$nil \; is\_empty' \Rightarrow_\beta true$

$is\_empty = \lambda \ell. \; \ell \; is\_empty'$

$(\lambda f. f\, a\, b) \; is\_empty' \Rightarrow_\beta false$

$is\_empty' \; a \; b \Rightarrow_\beta false$

$is\_empty' = \lambda x y. \; false$

$is\_empty = \lambda \ell. \; \ell \, (\lambda x y. \, false)$

$nil \, (\lambda x y. false) \Rightarrow_\beta true$

$\lambda x. \; true$

$(\lambda \ell. \ell (\lambda x y. \; false)) \; \lambda x. true \Rightarrow_\beta (\lambda x. true) \; (\lambda x y. false) \Rightarrow_\beta true$

```
all : (list [T], F) -> bool
any : (list [T], F) -> bool
```

$F: T \to bool$

cons
tail
head
is_empty
true
false
if

or
and

```
all: def . if (is empty l) false
        (and (f (head l)) (all (tail l) f))

any  def if (is_empty l) false
        (or (f (head l)) (any (tail l) f))
```

1.10.3

λ-term $\underline{M}$ — $\underline{MX =_\beta XMX}$ LMX

$L = \lambda y x.\ x y x$ $(LM) \Rightarrow (\lambda x.\ x M x)\ \underline{L}$

$M =_\beta \underline{LM}$ F.

$\underline{(YF) =_\beta F(YF)}$ LY

$M =_\beta YF$

- $\underline{YY}$
- Fixed-point
- $\exists M \in \Lambda$ $\not\exists$ β-nf