20.02.2023

FLP

curs 1

Organizare: min 5.00

1 + notă laborator + parțial + examen

notă laborator → lucru ef. la lab. $\boxed{2 \text{ puncte}}$

parțial → săpt 7, 30 min./15 min. $\boxed{3 \text{ puncte}}$ grilă

examen → $\boxed{4 \text{ puncte}}$ - 1 oră, în sesiune - exerciții cu materiale

Lambda calcul:

extensional egale → $f(x) = x^2 - 1$, $g(x) = (x+1)(x-1)$

↳ limbaj de progr.

teorie a funcțiilor ca formule

3 reguli

un fel de fct. anonime

f: $x \mapsto x^2$, $A = f(5)$ scris și $A = (\lambda x . x^2)\, 5$

$f \circ f : \lambda x . f(f(x))$

$f \mapsto f \circ f : \lambda f . \lambda x . f(f(x))$

$f \circ f$ $(X \to X) \to (X \to X)$

$x \xrightarrow{f} y \xrightarrow{g} z$, $f g$

$f = \lambda x . x$

$f(5) \cong (\lambda x . x)(5) \cong 5$

$f(f) \cong (\lambda x . x)(\lambda x . x) \cong \lambda x . x$ $f(f) \cong f$

Obs! xx înc. $x(x)$

$w = \lambda x . xx$

$w(\lambda y . y) \cong (\lambda x . xx)(\lambda y . y) \cong (\lambda y . y)(\lambda y . y) \cong$

$\lambda y . y$

$w(w) \cong (\lambda x . xxx)(\lambda x . xxx) \cong (\lambda x . xxx)$ = Ac. fct.

$\lambda x . \cancel{xxxx}$

- o fucție $f : \mathbb{N} \to \mathbb{N}$ este calculabilă $\Leftrightarrow$ există o metodă "pe foaie" care să permită să calc. $f(n)$ pentru orice n. → pen and paper (nu e ușor de formalizat)

"unei pers. cu experiență" :)

- fct. calculabilă $\Leftrightarrow$ recursivă

- ~~un~~ un ob. $\exists \Leftrightarrow$ îl poți construi

Quiz

1 → 4 ✓

2 → 2 ✓

3 → ~~4~~ 3 ✓

27.02.2023

FLP

curs 2

$M ::= x \mid (M\ N) \mid \lambda x.M$

variabilă aplicare abstractizare

Lambda termeni:

simplă

λ(var.)

ex: x, y, z ... variabile

$(x\ y), (x\ z) \dots (x\ (x y))$

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

nu se poate

$\cancel{\lambda(x\ y).x}$

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

... Această mulțime este infinită.

ex: $(f(f\ x))$ 2 aplicări

- $x, y, z \dots$
- $(\ ,\ )\ ,\ \lambda\ ,\ .$ } → Alfabet

Convenții:

1. elimin () ext.
2. aplicarea → asoc. la stg. $MNP \equiv (M\ N)\ P$; $f\ x\ y\ z = ((f\ x)\ y)\ z$
3. corpul abstractiz. (.→) se ext. la dr. cât se poate

! $\lambda x.M\ N$ $\quad \lambda x.(M\ N)$

4. λ se comprimă $\lambda x.\lambda y.\lambda z.M \equiv \lambda xyz.M$

Exerciții:

I 1. $(\lambda x.(\lambda y.(\lambda z.((x z)(y z)))) =$

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

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

2. $((a\,b)\,(c\,d))\ ((e\,f)\,(g\,h))$

$a b\ (cd)\ (ef\ (gh))$

II

1. $x\ x\ x\ x = (((x\,x)\ x)\ x)$

2. $\lambda x.\, x\ \lambda y.\, y = \cancel{((\lambda x.x)\ (\lambda y.y))}$

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

$\lambda x.\, N \rightarrow x$ legată (altfel este liberă)

Term nu are var. libere → s.n. "închis"

$M = (\lambda x.\, xy)\ (\lambda y.\, yz)$

x legat

z liber

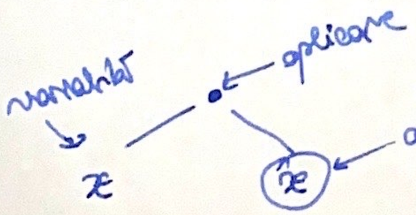
y e întâi liber, apoi legat

mulț. var. libere $FV\{z, y\}$ FV = free variables

$FV(x) = \{x\}$

$FV(MN) = FV(M) \cup FV(N)$

$FV(\lambda x.M) = FV(M) \setminus \{x\}$



Def: α echivalență

2 λ-termeni → identif. var. legate → numele lor contr. oricum nu

le redenumim și ajungem la o altă var α echivalentă.

= cea mai mică rel de ~~echivalență~~ congruență pe λ term a.î. pt. orice M și orice var y care nu apare în M avem

$$\lambda x. M =_\alpha \lambda y. (M \langle y / x \rangle)$$

REGULI: $\frac{\text{ipoteză}}{\text{concluzie}}$

- $\overline{M = M}$
- $\dfrac{M = N}{N = M}$
- $\dfrac{M = N \quad N = P}{M = P}$
- $\dfrac{M = M' \quad N = N'}{MN = M'N'}$ → congruență
- ξ • $\dfrac{M = M'}{\lambda x. M = \lambda x. M'}$
- α • $\dfrac{y \notin M}{\lambda x. M = \lambda y. (M \{y / x\})}$ x înloc. cu y

Def: Substituție $M[N / x]$ → înloc x cu N în M

→ substit. doar var. libere

→ nu legăm neintenț. val. libere

ex: $M = \lambda x. y x$; $N = \lambda z. x z$ | substit. y cu N în M

$M[N / y] \approx \lambda x. (\lambda z. x z) x$ — x nu mai e liber ⇒ NU e același x, îl redenumim. (pt. c legat)

$\Rightarrow \lambda x'. (\lambda z. x z) x'$

! x e în M și N ⇒ îl redenumim unde era legat.

Semantica nu se sch. dc. redenum. var. legate.

! $(\lambda x.M)[N/x] = \lambda x.M$

Ex:

pe x îl înloc. cu y

1. $(\lambda z.x)[y/x] = \lambda z.y$
2. $(\lambda y.x)[y/x] = \lambda y'.y$! NU $\lambda y\, y$
3. $(\lambda y.x)[(\lambda z.zw)/x] = \lambda y.(\lambda z.zw) = \lambda y z.zw$

Quiz:

1. termen închis $\to \lambda x.xxy$ (3)
2. var. libere din $\lambda x.xxy \to y$ (3)

Def: 3. combinator = termen închis

Def: β reducție = procesul de a evalua a termenilor prin "pasarea de arg. funcțiilor".

$(\lambda x.xy)z \xrightarrow{\beta} \boxed{zy}$

Redex = $\underbrace{(\lambda x.M)}_{\text{abstracție}}N$ + apl. | Redusul unui redex: $M[N/x]$

Formă normală = λ term. fără redexuri

$(\lambda x.y)((\lambda z.zz)(\lambda w.w)) \xrightarrow{\beta} (\lambda x.y)((zz)[\lambda w.w/z]) \equiv$
$(\lambda x.y)((\lambda w.w)(\lambda w.w)) \equiv (\lambda x.y)((\lambda w.w)[\lambda w.w/w])$

$\xrightarrow{\beta} \lambda x.y(\lambda w.w) \to y[\lambda w.w/x] \equiv y \to y$ formă normală

Confluența β-reducției: pe orice drum am merge, tot acolo ajungem.

06.03.2023

FLP

curs 3

Lambda calcul, β-reducții

Def: β-red = proc. de a evalua lambda termi prin "pasarea de arg. funcțiilor"

$$\boxed{(\lambda x.M)\,N \rightarrow_\beta M[N/x]}$$

- Redex: $(\lambda x.M)\,N$
- redusul său este $M[N/x]$

x este substit. cu N

- formă normală = λ-termen fără redexuri

fct. constantă

$$(\lambda x.y)((\lambda z.zz)(\lambda w.w)) \rightarrow_\beta (\lambda x.y)\,(\lambda w.w)(\lambda w.w)$$

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

$$\rightarrow y$$

! $(\lambda x.y)\,(\_\_\_) \rightarrow y\,[\lambda \text{---}] \equiv y$

↳ NE VA RET MEREU y

Evaluare infinită:

$$(\lambda x.\,x\,x)\,(\lambda x.\,x\,x) \rightarrow_\beta (\lambda x.\,x\,x)\,(\lambda x.\,x\,x)$$

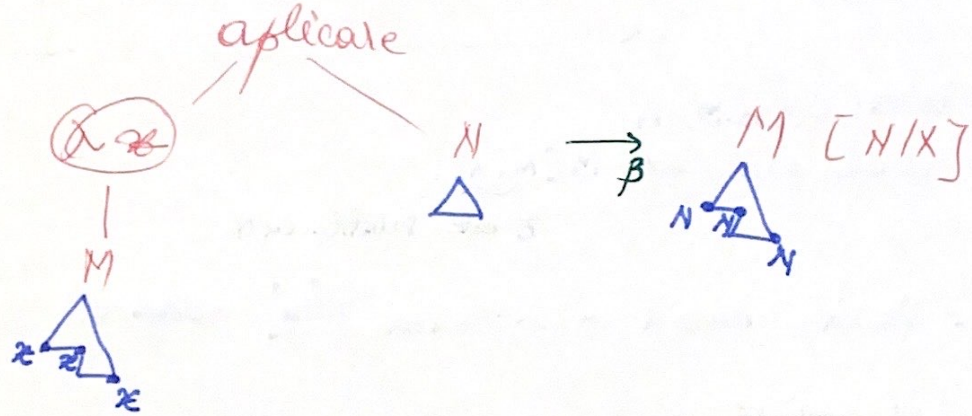
=> lung. unui termen nu trebuie să scadă neapărat λ-term. care deși se pot reduce, posib. nu vor aj. nic. la f. normală

$$(\lambda xy.y)\,(\overbrace{(\lambda x.xx)\,(\lambda x.xx)}^{\text{cel }\omega})\,(\lambda z.z) \to_\beta (\lambda y.y)(\lambda z.z) \to_\beta \lambda z.z$$

$$\downarrow$$

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

~~app λx N aplicare~~



Căutăm x în M și îl înlocuim cu N.

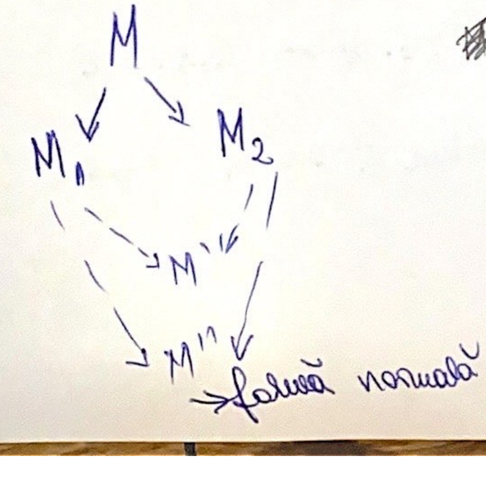
Nu putem ... app ...

app
app ceva
vom lua ac. app

Normalizare:

- slab, putem aj. la o formă norm., și se pot întâlni bucle
- puternic: ~~~~~~ , și nu se pot întâlni bucle

Un λ ~~termel~~ termen are cel mult o β-formă normală (modulo α-echivalenței)



$M \to_\beta M_1$

$M \to_\beta M_2 \to \exists\, M'$ a.î. $M_1 \twoheadrightarrow_\beta M'$, $M_2 \twoheadrightarrow_\beta M'$

Exerciții

1. ~~λx z y. z~~

~~(λ z y z) M N~~ $\to_\beta$ x[M | x] → M

(λ z. z) M $\to_\beta$ z [M | x] $\to_\beta$ M

2. (λz y. z) M N $\to_\beta$ ~~MN~~

(λz. (λy. z)) M N → λy. z) [x/M] N = (λy. M) N

= M [N | y] = M

3. (λz. z z) (λy. y y y) → (λy. y y y) (λy. y y y) ..

→ nu are formă normală

Strategii de evaluare

Normală: cel mai din stg ext și apoi cel mai din ext.

Dacă un tl. are o formă normală, atunci strateg. normală va converge la ea.

Aplicativă: ——— ~ ——— și apoi cel mai din int.

. CBN : Haskell
call by name → normală, fără a face red. în corpul unei abstractizări

. CBV : Alte limbaje
call by value → aplicativă ....

Haskell → $\boxed{\text{lazy evaluation}}$ pt că face call by name

Quiz

1. β-redex → λ term $(\lambda x.M)\,N$ (2)
2. f-normală → fără redexuri (4)
3. str. normală → sg. ext. (1)

DATA VIITOARE

Expresivitatea λ calculului → elem. booleene

bool $T \overset{\Delta}{=} \lambda xy.x$ (true)

$F \overset{\Delta}{=} \lambda xy.y$ (false)

$if \overset{\Delta}{=} \lambda btf.\ btf$

Codare peano:

λfx | de n ori peste o → nr. n

13.03.2023

FLP

curs 4

Expresivitatea λ calculului

<u>Booleeni:</u>

$T \stackrel{\Delta}{=} \lambda xy.\, x$ (true) → proiecția pe prima componentă

$F \stackrel{\Delta}{=} \lambda xy.\, y$ (false)

$if = \lambda b\, t\, f \begin{cases} t, & \text{if } b = true \\ f, & \text{if } b = false \end{cases}$

$\begin{cases} T\,t\,f \twoheadrightarrow_\beta t \\ F\,t\,f \twoheadrightarrow f \end{cases}$ $\boxed{if \stackrel{\Delta}{=} \lambda\, btf.\, b\, t\, f}$

$if\ T\, t_1 t_2 = if\ (\lambda xy.x)\, t_1 t_2$

<u>explicate:</u> $T t_1 t_2 = (\lambda xy.x)\, t_1 t_2 = (\lambda x.(\lambda y.\, x))\, t_1\, t_2$

$\rightarrow_\beta (\lambda y.\, t_1)\, t_2 \rightarrow_\beta t_1$

$F t_1 t_2 = (\lambda x.\, (\lambda y.y))\, t_1 t_2 \rightarrow_\beta (\lambda y.y)\, t_2 \rightarrow_\beta t_2$

<u>exemplu:</u> $if\ (\lambda x.x)\, t_1 t_2 \rightarrow_\beta (\lambda x.x)\, t_1 t_2 \rightarrow_\beta t_1 t_2$

<u>Operații:</u>

$and \stackrel{\Delta}{=} \lambda b_1 b_2 .\ if\ b_1 b_2 F$

$or \stackrel{\Delta}{=} \lambda b_1 b_2 .\ if\ b_1 T b_2$

$not \stackrel{\Delta}{=} \lambda b_1 .\ if\ b_1 F T$

and	F	T
b_1 F		
T	F	T

$b_1 T \Rightarrow ret\ B_2$

$b_1 T \Rightarrow ret\ T$

$b_1 T \rightarrow ret\ F$

$b_1 F \rightarrow ret\ T$

SLIDE 6 → toate def.

Exercițiu:

and TF $= (\lambda b_1 b_2 . \text{if } b_1 b_2 F)\ T\ F$

$\to_\beta (\lambda b_2 . \text{if } T b_2 F) F \to_\beta \text{if } T F F = (\lambda\ b\ t_1 t_2 . b\ t_1 t_2)\ T F F$

$\to_\beta TFF = (\lambda xy.x)\ FF \to_\beta F$

or FT $= (\lambda b_1 b_2 . \text{if } b_1 T b_2)\ F T \to_\beta$

$\to_\beta (\lambda b_2 . \text{if } F T b_2) T$

$\to_\beta \text{if } F T T = (\lambda b\ t_1 t_2 . b\ t_1 t_2)\ F T T \to_\beta$

$\to_\beta FTT = (\lambda xy.y)\ TT \to_\beta T$

not T $\triangleq (\lambda b_1 . \text{if } b_1 F T)\ T \to_\beta \text{if } TFT =$

$= (\lambda\ b\ t_1 t_2 . b\ t_1 t_2)\ TFT =$

$\to_\beta TFT = (\lambda xy.x)\ FT = F$

Numere naturale:

$\overline{n} \triangleq \lambda fx. f^n x = \lambda fx (f(f(\underbrace{\dots(f(x))\dots)\dots)}_{n}$

aplicare repetată

tot ce mă duce mai departe de unde plec

$\overline{0} \triangleq \lambda fx. f^0 x = \lambda fx.x$

$\overline{1} = \lambda fx. fx$

$\overline{2} = \lambda fx. f(fx)$

$\overline{3} = \lambda fx. f(f(fx))$

$\vdots$

$$\boxed{\text{Succ} \triangleq \lambda nfx. f(nfx)}$$

$\text{Succ } \bar{2} = (\lambda n f x . f(\underbrace{n f x}_{\text{de } n \text{ ori}})) \bar{2} =$ f de x practic.

$\rightarrow_\beta \lambda f x . f(\bar{2} f x) =$

$\lambda f x . f((\lambda f' x' . f'(f'(x'))) f x)$

$\rightarrow_\beta \lambda f x . f(f f(x))$

$= \bar{3}$

$\bar{2} = \lambda f' x . f'(f'(x))$ → sau pentru orice n oarecare

$$\boxed{\text{add} \triangleq \lambda m n f x . m f (\underbrace{n f x}_{\text{de } n \text{ ori } f x})}$$

de m ori

$$\boxed{\begin{array}{l}\text{mul} \triangleq \lambda m n . m (\text{add } \bar{n}\ \bar{0}) \\ \text{exp} \triangleq \lambda m n . m (\text{mul } n)\ \bar{1}\end{array}}$$

$\text{iszero} \triangleq \lambda n x y . n (\lambda z . y)\ x$ ⇄ 0 – true; ~~$\neq 0$~~ – false

$\text{iszero } \bar{0} \twoheadrightarrow_\beta \lambda x y . x$

$\text{iszero } \bar{n} \twoheadrightarrow_\beta \lambda x y . y$

Factorial: $0! = 1$

$n! = n \cdot (n-1)!$, dacă $n \neq 0$

Puncte fixe: $f(x) = x$

$f(x) = x^2$ are 2 pct. fixe, 0 și 1

$f(x) = x+1$ NU are

β echivalență:

$M =_\beta M'$

ex: $(\lambda y . y\, v)\, z \to_\beta z\, v \xleftarrow[\beta]{} (\lambda x . z . x)\, v$

$=_\beta$

$\Rightarrow$ M pct fix al lui F dacă $FM =_\beta M$ $\quad FM \twoheadrightarrow_\beta M$, $M \twoheadrightarrow_\beta FM$

Teoremă. In λ calcul fără tipuri, orice termen are 1 pct fix.

+dem

fact n: $\boxed{fact = F\, fact}$ $\to$ din th știm că pb. are sol.

$\Rightarrow$ ΘF, YF $\rangle$ Soluții.

Quiz

1. $T = \lambda x y . x$ $\to 1$
2. codare Church $\to 3$
3. $FM =_\beta M$ $\to 2$

.23bob 13/5 → 10.03.23 20.03.2023

FLP

curs 5

λ-lambda calcul cu tipuri:

$\mathbb{T}$ = mulț. tuturor tipurilor simple

$\mathbb{T} = V \mid T \to T$

- tipul variabilă
- tipul săgeată ("funcții")

Nat → Real = toate fcț. de la nat la real

(Nat → Int) → (Int → Nat)

! la tipurile săgeată, avem asoc. la dr.

$\alpha_1 \to \alpha_2 \to \alpha_3 \to \alpha_4$

$(\alpha_1 \to (\alpha_2 \to (\alpha_3 \to \alpha_4)))$

! var. sunt asoc la stg

$(((x_1\ x_2)\ x_3)\ x_4)$

Ce ms. că M are tip σ?

$M : \sigma$

VAR: $x : \sigma$

- var. legate → dist.
- $x : \sigma$ și $x : \tau$ ⇒ $\sigma = \tau$

APL:

- $M : \sigma \to \tau$, $N : \sigma$ ⇒ $MN : \tau$

ABSTR

- $M : \tau$ $\lambda x . M$

$x : \sigma$ și $M : \tau$ ⇒ $\lambda x . M : \sigma \to \tau$

Typeable :

- $\lambda x . x : \sigma \to \sigma$
- yx

  $y : \sigma \to \tau$

  x se potr. cu dom σ $\Rightarrow$ $yx : \tau$

- xx NU este typeable

  σ trb $\sigma \to -$ $\Rightarrow$ nu poate exista.

  ptc orice variabilă are un unic tip

$$\boxed{\begin{array}{l} f : \rho \to \sigma \to \tau \\ f x y : \tau \end{array}}$$

Asociere

- implicată · curry typing = pp. că nu știu nimic despre var.

  → RĂSP. FINAL VA FI MAI PRECIS

- explicată Church Typing = punem tipuri

  ex: $x : \alpha \to \alpha$, $y : (\alpha \to \alpha) \to \beta$

  $yx : \beta$

  $z : \beta$, $u : \gamma$, $\lambda z u . z : \beta \to \gamma \to \beta$

  $(\lambda z u . z)(yx) : \gamma \to \beta$

↳ exemplu curry typing

$M = (\lambda z u . z)(y x)$

$\lambda z u . z : A \to B$ $\Longrightarrow$ $\lambda z u . z : F \to G \to F$

$z : F$

$u : G$

$\boxed{yx : A}$ (1) $\Longrightarrow$ $y : C \to D$

$x : C$

$\boxed{yx : D}$ (2) $\Rightarrow$ $A \equiv D \equiv F$

$B \equiv G \to F$

$x : C$

$\Rightarrow$ $y : C \to A$

$x : A$

lambda termen ⊢ termen

$$\Lambda_T = x \mid \Lambda_T \Lambda_T \mid \lambda x : T . \Lambda_T$$

afirmație: $M : \sigma$ — subiect, tip

declarație: subiect = variabilă

context: decl. cu sub. diferiți

judecată: $\underbrace{\Gamma}_{\text{context}} \vdash \underbrace{M : \sigma}_{\text{afirmație}}$ (gama)

Cum deducem o judecată?

premise căci — axiomă

$$\frac{}{\Gamma \vdash x : \sigma} \text{ (var)}$$ dacă $x : \sigma \in \Gamma$

$$\frac{\Gamma \vdash M : \sigma \to \tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash MN : \tau} \text{ (app)}$$

$$\frac{\Gamma, x : \sigma \vdash M : \tau}{\Gamma \vdash (\lambda x : \sigma . M) : \sigma \to \tau} \text{ (abs)}$$

Arbore de derivare:

Exemplu: $\lambda y : \alpha \to \beta . \lambda z : \alpha . yz$ are tipul $(\alpha \to \beta) \to \alpha \to \beta$

pubem premise

$$\frac{\dfrac{\dfrac{\dfrac{}{y : \alpha \to \beta, z : \alpha \vdash y : \alpha \to \beta} \text{ (var)} \quad \dfrac{}{y : \alpha \to \beta, z : \alpha \vdash z : \alpha} \text{ (var)}}{y : \alpha \to \beta, z : \alpha \vdash yz : \beta} \text{ (app)}}{y : \alpha \to \beta \vdash (\lambda z : \alpha . yz) : \alpha \to \beta} \text{ (abs)}}{\emptyset \vdash (\lambda y : \alpha \to \beta . \lambda z : \alpha . yz) : (\alpha \to \beta) \to \alpha \to \beta} \text{ (abs)}$$

context vid

Quiz:

1. typeable, (1)
2. ___ $x(xy)$ (3) $xxy =$ ~~sau~~ $(xx)y$ NU
3. judecată $\Gamma \vdash$ ___ (3)

27.05.2023

FLP

curs 6

PARȚIAL:

- 15 grile } 4 var → 1 corectă
- data viitoare

- tipul variabilă
- tipul săgeată
- context Γ
- judecată $\Gamma \vdash M : \sigma$

$$\boxed{\Lambda_{\Pi} = x \mid \Lambda_{\Pi}\Lambda_{\Pi} \mid \lambda x : \Pi . \Lambda_{\Pi}}$$

context $\vdash$ term : type

→ xx NU ESTE TYPEABLE

$(\lambda f. \text{if } f(T)(f3)(f5))(\lambda x . x)$ ⟶ NU ARE TIP

let $f = (\lambda x \to x)$ in if $(fT)(f3)(f5)$ ⟹ 3

Putem adăuga tipuri: ad. Unit

$\Pi = V \mid \Pi \to \Pi \mid \text{Unit} \mid \text{Void}$

$\Gamma \vdash$

ad. Void

ad. produs : $\Pi \times \Pi$

$\Lambda_{\Pi} = x \mid \Lambda_{\Pi}\Lambda_{\Pi} \mid \lambda x : \Pi . \Lambda_{\Pi} \mid \text{unit} \mid \langle \Lambda_{\Pi}, \Lambda_{\Pi} \rangle \mid \text{fst}\,\Lambda_{\Pi} \mid \text{snd}\,\Lambda_{\Pi}$

$$\frac{\Gamma \vdash M : \sigma \times \tau}{\Gamma \vdash \text{fst}\, M : \sigma}\ (\times E_1) \qquad \frac{\Gamma \vdash M : \sigma \times \tau}{\Gamma \vdash \text{snd}\, M : \tau}\ (\times E_2)$$

$$\to \quad \frac{\Gamma \vdash M : \sigma \qquad \Gamma \vdash N : \tau}{\Gamma \vdash \langle M, N \rangle : \sigma \times \tau}\ (\times I)$$

$\lambda x.\lambda y.\langle x,y\rangle :: \sigma \to \tau \to \sigma \times \tau$ } nu avem v. libere → nu ne trb. CONTEXT

$\lambda x.\lambda y.x :: \sigma \to \tau \to \sigma$

ip / concl

$$\frac{\Gamma \cup \{x:\sigma\} \vdash M : \tau}{\Gamma \vdash (\lambda x.M) : \sigma \to \tau}\ (\to_I)$$

var → am terminat — var → au term.

$$\frac{\dfrac{\dfrac{x:\sigma, y:\tau \vdash x:\sigma \qquad x:\sigma, y:\tau \vdash y:\tau}{x:\sigma, y:\tau \vdash \langle x,y\rangle :: \sigma\times\tau}\ (\times_I)}{x:\sigma \vdash \lambda y.\langle x,y\rangle :: \tau \to \sigma\times\tau}\ (\to_I)}{\vdash \lambda x.\lambda y.\langle x,y\rangle :: \sigma\to\tau\to\sigma\times\tau}\ (\to_I)$$

⇒ mutam y: la context.

→ dos

⇒ mutam $x:\sigma$ la context

$\varphi \to (\tau \to \gamma)$ → nu

Modus tones

$\sigma \to \tau \qquad \sigma$

Dacă afară e lt, atunci porcii zboară — Afară e lunecuc

Porcii zboară

τ

$x>0 \to x+3>0 \qquad x>0$

$x+3>0$

$M : \sigma + \tau$

$\text{left } M : \sigma$

$M_1 : \sigma \to \gamma$

$M_1(\text{left } M) : \gamma$

fals impl. orice

Curry-Howard: Propositions are types. / Proofs are Terms!

Orice tip din λ calcul este o prop. egala!

dem. sunt elemen

$$\frac{\dfrac{\{\sigma,\tau\} \vdash \sigma}{\{\sigma\} \vdash (\tau \supset \sigma)}}{\vdash \sigma \supset (\tau \supset \sigma)}\ (\supset_I)$$

pt c σ apare in context

Quiz:
1. type check → SE DAU TOATE : 3
2. nu e demo → TOATE SUNT : 4
3. void → nu are nimic : 3