

---

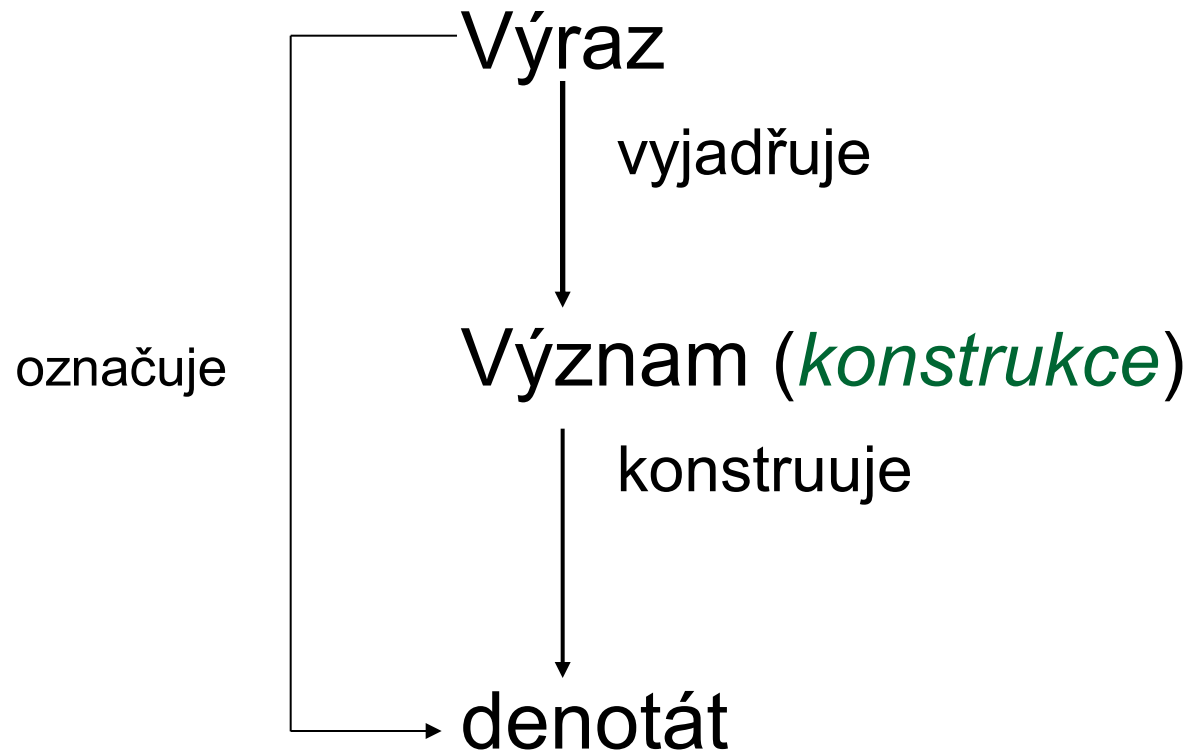
TIL (13.10. 2016)

---

Marie Duží

<http://www.cs.vsb.cz/duzi/>

# Sémantické schéma



Ontologie TIL: rozvětvená hierarchie typů

# Základní pojmy

## ■ Konstrukce

- Proměnné  $x, y, p, w, t, \dots$  v-konstruují
- Trivializace  ${}^0X$  dodá objekt  $X$
- Kompozice  $[F A_1 \dots A_n]$  aplikace funkce
- Uzávěr  $[\lambda x_1 \dots x_n X]$  konstrukce funkce
- Provedení  ${}^1X$ , Dvojí provedení  ${}^2X$

## ■ Jednoduchá teorie typů

- Báze:  $\{0, 1, \tau, \omega\}$
- Funkcionální typy:  $(\alpha\beta_1\dots\beta_n)$ 
  - *Parciální funkce*  $(\beta_1 \times \dots \times \beta_n) \rightarrow \alpha$

# Základní pojmy

- Denotát chápeme vždy jako (extensionální) funkci, včetně nulárních funkcí (atomické entity jako čísla, individua)
- Denotát může být:
  - (PWS-)intense:  $(\alpha\omega)$ , nejčastěji  $((\alpha\tau)\omega)$ , zkráceně  $\alpha_{\tau\omega}$
  - Extense: funkce, jejíž doména *není*  $\omega$
  - Konstrukce
  - **nic** (parcialita)
- Typické *extense*:
  - Množiny jsou extense typu  $(o\alpha)$
  - Relace jsou extense typu  $(o\alpha\beta)$
- Typické *intense*:
  - Vlastnosti individuí /  $(o1)_{\tau\omega}$ , individuové úřady (role) /  $1_{\tau\omega}$ , propozice /  $o_{\tau\omega}$ , vztahy mezi individui /  $(o11)_{\tau\omega}$ , ...

# Základní pojmy, notace

- **Výrokově-logické spojky** implikace ( $\supset$ ), konjunkce ( $\wedge$ ), disjunkce ( $\vee$ ) a ekvivalence ( $\equiv$ ) jsou funkce typu **(ooo)**, negace ( $\neg$ ) je typu **(oo)**.
  - Pro spojky užíváme infixní notaci bez Trivializace.
  - Tak např. místo  $[^0\wedge [^0\supset p q] [^0\neg q]]$  budeme psát  $[[p \supset q] \wedge \neg q]$ .
- Pro  **$\alpha$ -identity**, tj. relace  $=_{\alpha}/(o\alpha\alpha)$ , budeme rovněž používat infixní notaci bez vyznačení indexu typu  $\alpha$ .
  - Např. necht'  $=_1/(o11)$  je identita individuí,  $=_{((o\tau)\omega)}/(oo_{\tau\omega}o_{\tau\omega})$  identita propozic;  $a, b \rightarrow_v 1$ ,  $P \rightarrow_v (o1)_{\tau\omega}$ . Pak místo
  - $[^0\supset [^0=_1 a b] [^0=_{((o\tau)\omega)} [\lambda w\lambda t [P_{wt} a]] [\lambda w\lambda t [P_{wt} b]]]]$  budeme většinou psát prostě  $[[a = b] \supset [\lambda w\lambda t [P_{wt} a] = \lambda w\lambda t [P_{wt} b]]]$ .

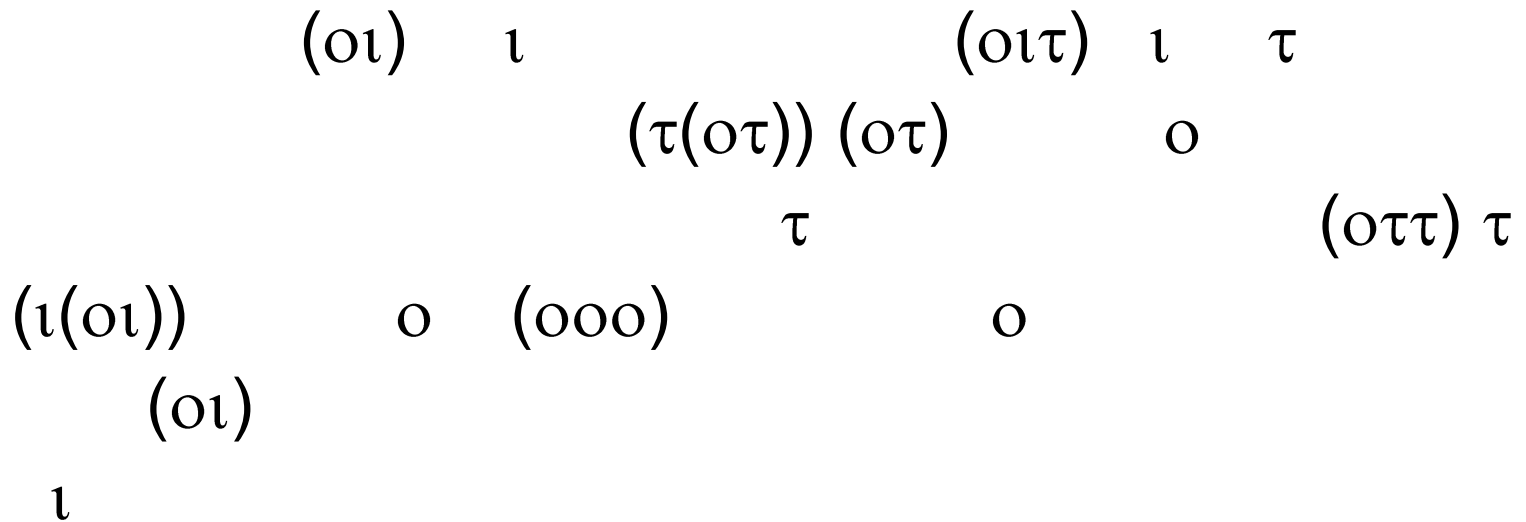
# Základní pojmy, notace

- **Kvantifikátory** (totální funkce)  $\forall^\alpha, \exists^\alpha / (\mathbf{o}(\mathbf{o}\alpha))$ .
  - Je-li  $x \rightarrow_v \alpha, B \rightarrow_v \mathbf{o}$ , tedy  $\lambda x B(x) \rightarrow_v (\mathbf{o}\alpha)$ , pak
  - $[{}^0\forall^\alpha \lambda x B(x)]$  konstruuje **T**, pokud  $\lambda x B(x)$  konstruuje celý typ  $\alpha$ , jinak **F**,
  - $[{}^0\exists^\alpha \lambda x B(x)]$  konstruuje **T**, pokud  $\lambda x B(x)$  konstruuje neprázdnou množinu prvků typu  $\alpha$ , jinak **F**.
  - Notace:  $\forall x B(x), \exists x B(x)$
- **Singularizátory** (parciální funkce)  $\mathbf{I}^\alpha / (\alpha(\mathbf{o}\alpha))$ .
  - $[{}^0\mathbf{I}^\alpha \lambda x B(x)]$  konstruuje **jediný  $\alpha$ -prvek** množiny konstruované  $\lambda x B(x)$ , pokud je to jednoprvková množina (singleton), jinak **nedefinován**.
  - Notace:  $\iota x B(x)$  čteme „to jediné  $x$ , že  $B$ “

# Příklad

- „ten jediný člověk, který zaběhl 100 m pod 9 s“
  - $Man/(o1)_{\tau\omega}$ ,  $Time/(\tau(o\tau))$ : délka časového intervalu,  
 $Run/(o1\tau)_{\tau\omega}$ ,  $I/(1(o1))$ : ten jediný..., celý výraz ozn.  $l_{\tau\omega}$ .

$\lambda w \lambda t \ [^0 I \ \lambda x \ [[^0 Man_{wt} \ x] \wedge \ [^0 Time \ \lambda t \ [^0 Run_{wt} \ x \ ^0 100] < \ ^0 9]]]$



$l_{\tau\omega}$

# Rozvětvená teorie typů

**$T_1$  (typy řádu 1)** – neprocedurální objekty, jednoduchá teorie typů

**$C_n$  (konstrukce řádu  $n$ )**

Nechť  $x$  je proměnná, která  $v$ -konstruuje objekty typu řádu  $n$ . Pak  $x$  je *konstrukce řádu  $n$  nad  $B$* .

- Nechť  $X$  je prvek typu řádu  $n$ . Pak  ${}^0X, {}^1X, {}^2X$  jsou *konstrukce řádu  $n$  nad  $B$* .
- Nechť  $X, X_1, \dots, X_m$  ( $m > 0$ ) jsou konstrukce řádu  $n$  nad  $B$ . Pak  $[X X_1 \dots X_m]$  je *konstrukce řádu  $n$  nad  $B$* .
- Nechť  $x_1, \dots, x_m, X$  ( $m > 0$ ) jsou konstrukce řádu  $n$  nad  $B$ . Pak  $[\lambda x_1 \dots x_m X]$  je *konstrukce řádu  $n$  nad  $B$* .
- Nic jiného ...

**$T_{n+1}$  (typy řádu  $n + 1$ )**

Nechť  $*_n$  je kolekce všech konstrukcí řádu  $n$  nad  $B$ .

- $*_n$  a každý typ řádu  $n$  jsou *typy řádu  $n + 1$  nad  $B$* .
- Jsou-li  $\alpha, \beta_1, \dots, \beta_m$  ( $m > 0$ ) typy řádu  $n + 1$  nad  $B$ , pak  $(\alpha \beta_1 \dots \beta_m)$ , tj. kolekce parciálních funkcí, je *typ řádu  $n + 1$  nad  $B$* .
- Nic jiného ...



# Příklady, notace: $C/\alpha \rightarrow_v \beta$

- $0+/*_1 \rightarrow (\tau\tau\tau), X/*_1 \rightarrow_v \tau$
- $[0+ X^0 1]/*_1 \rightarrow_v \tau$
- $\lambda X [0+ X^0 1]/*_1 \rightarrow_v (\tau\tau)$  *funkce následníka*
- $[\lambda X [0+ X^0 1]^0 5]/*_1 \rightarrow_v \tau$  *číslo 6*
- $[0: X^0 0]/*_1 \rightarrow_v \tau$  *nic*
- $\lambda X [0: X^0 0]/*_1 \rightarrow_v (\tau\tau)$  *degenerovaná funkce*
- Necht' *Improper* je množina konstrukcí řádu 1, které jsou  $v$ -nevlastní pro každou valuaci  $v$ . Tedy *Improper* je extenzionální objekt typu  $(0*_1)$ , což je typ řádu 2.
  - Pak  $[^0 Improper^0 [0: X^0 0]]/*_2 \rightarrow 0$  je prvek  $*_2$ , což je typ řádu 3, i když konstruuje pravdivostní hodnotu **P**, což je objekt typu řádu 1.

## Příklady, notace: $C/\alpha \rightarrow_v \beta$

- Necht' *Aritmetic* je množina unárních aritmetických funkcí definovaných na přirozených číslech, tedy *Aritmetic* je objekt typu  $(o(vv))$ , a necht' proměnná  $x \rightarrow_v v$ , kde  $v$  je typ přirozených čísel.
- Pak Kompozice  $[^0Aritmetic [\lambda x [^0+ x ^01]]]$  patří do  $*_1$ , což je typ řádu 2, a konstruuje **P**, protože Uzávěr  $[\lambda x [^0+ x ^01]]$  konstruuje unární funkci následníka, což je aritmetická funkce.

# Příklady, notace: $C/\alpha \rightarrow_v \beta$

- Kompozice  $[^0\textit{Aritmetic } ^2c]/*_3 \rightarrow_v \circ$   $v$ -konstruuje pravdivostní hodnotu **P**, pokud proměnná  $c/*_2 \rightarrow_v *_1$   $v$ -konstruuje např. Uzávěr  $[\lambda x [^0+ x ^01]]$ .
  - Dvojí Provedení  $^2c$   $v$ -konstruuje to, co je  $v$ -konstruováno tímto Uzávěrem, a to je aritmetická funkce následníka.
  - Kompozice  $[^0\textit{Aritmetic } ^2c]$  je objekt patřící do  $*_3$ , což je typ řádu 4;
  - proměnná  $c$   $v$ -konstruuje Uzávěr  $[\lambda x [^0+ x ^01]]$  typu  $*_1$ , a proto  $c$  patří do  $*_2$ , což je typ řádu 3.
  - Dvojí provedení zvyšuje řád konstrukce, proto  $^2c$  je prvek typu  $*_3$ , což je typ řádu 4. Tedy celá Kompozice  $[^0\textit{Aritmetic } ^2c]$  patří do  $*_3$ , typu řádu 4.

# Empirické výrazy

- Empirické výrazy označují *netriviální (nekonstantní) intence*
- „francouzský král“, „prezident ČR“, „nejvyšší hora na světě“ *označují úřady typu  $\iota_{\tau\omega}$* , a *aktuálně referují* (v daném  $w$  a  $t$ ) po řadě k ničemu, k Miloši Zemanovi a k Mount Everest.
- Predikáty jako „být studentem“, „být vysoký“, „být veselý“, „být 60 let starý“ *označují vlastnosti typu  $(\text{oi})_{\tau\omega}$*  a *referují* ke své populaci, tj. množině individuí, kteří jsou aktuálně studenty, vysokí, veselí a staří 60 let.
- Oznamovací věty jako „Praha je větší než Brno“ *označují* propozice typu  $\text{o}_{\tau\omega}$  a *referují* k pravdivostní hodnotě (daná věta k **P**).
- *Tedy uvedené výrazy jsou empirické v tom smyslu, že to, k čemu referují v daném stavu světa, je již mimo oblast logické analýzy a může být zjišťováno pouze empirickým zkoumáním stavu světa v daném čase.*

---

# Analytické výrazy

- Analytické výrazy označují *extense* nebo *triviální (konstantní) intense*
  - Jejich extensi můžeme určit *pouze na základě porozumění*, bez zkoumání stavu světa.
  - Matematické a logické výrazy jsou analytické, označují *extense*
  - Výrazy, které obsahují empirické podvýrazy, jsou analytické, pokud označují *konstantní intense*. Tedy referovaný objekt je *analyticky nutně*, tj. ve všech *w* a *t* jeden a tentýž.
  - „*Žádný starý mládenec není ženatý*“
  - „*Velryby jsou savci*“
-

# Omezené kvantifikátory

- Všichni studenti jsou chytrí.
- Někteří studenti jsou líní.

$\lambda w \lambda t \forall x [[{}^0\text{Student}_{wt} x] \supset [{}^0\text{Chytrý}_{wt} x]]$

$\lambda w \lambda t \exists x [[{}^0\text{Student}_{wt} x] \wedge [{}^0\text{Líný}_{wt} x]]$

Toto nejsou doslovné analýzy, odporují tzv. „Parmenidovu principu“ a naší metodě analýzy: věty nezmiňují  $\forall$ ,  $\exists$ ,  $\supset$ ,  $\wedge$ .

- *All, Some, No, Most, ... / ((o(o1))(o1))*
- $[{}^0\text{All } M]$ , kde  $M \rightarrow_v (o1)$ ,  $v$ -konstruuje množinu všech nadmnožin množiny  $M$
- $[{}^0\text{Some } M]$ , kde  $M \rightarrow_v (o1)$ ,  $v$ -konstruuje množinu všech těch množin, které mají neprázdný průnik s  $M$
- $[{}^0\text{No } M]$ , kde  $M \rightarrow_v (o1)$ ,  $v$ -konstruuje množinu všech množin, které jsou disjunktní s  $M$

$\lambda w \lambda t [[{}^0\text{All } {}^0\text{Student}_{wt}] {}^0\text{Chytrý}_{wt}]$

$\lambda w \lambda t [[{}^0\text{Some } {}^0\text{Student}_{wt}] {}^0\text{Líný}_{wt} x]$

# Analytické výrazy

“No bachelor is married”  $\rightarrow$  TRUE

- Bachelor, Married/(o1)<sub>τω</sub>; No/((o(o1))(o1))

$\lambda w \lambda t \lambda x [ [^0\text{No } ^0\text{Bachelor}_{wt} ] ^0\text{Married}_{wt} ]$

Definujeme a zjemňujeme:

- $m, n / *_{\tau} \rightarrow_{\nu} (o1), x \rightarrow_{\nu} \iota$ :  
 $^0\text{No} = \lambda m \lambda n \neg \exists x [[m x] \wedge [n x]], [[^0\text{No } m] n] = \neg \exists x [[m x] \wedge [n x]].$
- $^0\text{Bachelor} = \lambda w \lambda t \lambda x [\neg [^0\text{Married}_{wt} x] \wedge [^0\text{Man}_{wt} x]].$ 
  - (být neženatý a být muž jsou *rekvizity* vlastnosti být starý mládenec)
- $[ [^0\text{No } ^0\text{Bachelor}_{wt} ] ^0\text{Married}_{wt} ] =$   
 $\neg \exists x [ [^0\text{Bachelor}_{wt} x] \wedge [^0\text{Married}_{wt} x] ] =$   
 $\neg \exists x [ \neg [^0\text{Married}_{wt} x] \wedge [^0\text{Man}_{wt} x] \wedge [^0\text{Married}_{wt} x] ].$ 
  - $v$ -konstruuje **T** pro libovolnou valuaci  $v$ , proto můžeme generalizovat:
- $\forall w \forall t \neg \exists x [ \neg [^0\text{Married}_{wt} x] \wedge [^0\text{Man}_{wt} x] \wedge [^0\text{Married}_{wt} x] ].$

# Analytické vs. empirické výrazy

Nutně,  $8 > 5$

Počet planet = 8

---

Nutně, počet planet  $> 5$

???

$\forall w \forall t [{}^0 > {}^0 8 {}^0 5]$

(analyticky nutně)

$\lambda w \lambda t [[{}^0 \text{Počet} {}^0 \text{Planeta}_{wt}] = {}^0 8]$

(empiricky, náhodně)

---

$\lambda w \lambda t [{}^0 > [{}^0 \text{Počet} {}^0 \text{Planeta}_{wt}] {}^0 5]$

(empiricky, náhodně)

$\text{Počet}/(\tau(o_1))$ : počet prvků množiny,  $\text{Planeta}/(o_1)_{\tau\omega}$ ,  $>$ ,  $=/(\circ\tau\tau)$

Dk.:

1)  $[{}^0 > {}^0 8 {}^0 5]$

1. premisa,  $\forall E$

2)  $[[{}^0 \text{Počet}_{wt} {}^0 \text{Planeta}_{wt}] = {}^0 8]$

2. premisa,  $\lambda E$

3)  $[{}^0 > [{}^0 \text{Počet}_{wt} {}^0 \text{Planeta}_{wt}] {}^0 5]$

1, 2 Leibniz

4)  $\lambda w \lambda t [{}^0 > [{}^0 \text{Počet}_{wt} {}^0 \text{Planeta}_{wt}] {}^0 5]$

3,  $\lambda I$



# Práce s parcialitou, $v$ -nevlastní konstrukce

## ■ Chybné typování

- Jestliže  $X$  není konstrukce řádu  $n$  ( $n \geq 1$ ), pak  ${}^1X$  je nevlastní;
- Jestliže  $X$  není konstrukce řádu  $n$  ( $n \geq 2$ ), pak  ${}^2X$  je nevlastní;
- Jestliže  $X, X_1, \dots, X_n$  nejsou konstrukce vyhovující typově Definici, pak  $[X X_1 \dots X_n]$  je nevlastní, nekonstruuje nic.
- Příklad:  ${}^1Tom, {}^15, \lambda w \lambda t [{}^0Student_{wt} {}^05], {}^2[\lambda w \lambda t [{}^0Student_{wt} {}^0Tom]]$

## ■ Aplikace funkce $f$ na argument $a$ takový, že $f$ je nedefinována na $a$

- $[{}^0: x {}^00]$  je  $v$ -nevlastní pro každou valuaci  $v$
- $\lambda x [{}^0: x {}^00]$  není nevlastní, konstruuje degenerovanou funkci
- ${}^0[{}^0: x {}^00]$  není nevlastní, konstruuje  $[{}^0: x {}^00]$
- $[{}^0Improper {}^0[{}^0: x {}^00]]$  konstruuje  $\mathbf{T}$ 
  - $Improper/({}_0*_1)$ : třída konstrukcí  $v$ -nevlastních pro každou valuaci  $v$ .

# Parcialita a kompozicionalita

- „Jestliže pět děleno nulou je pět, pak Tom je papež“
  - označuje degenerovanou propozici, která je všude *nedefinována*!
- $\lambda w \lambda t \text{ } [[ [^0: \text{}^05 \text{}^00] = \text{}^05 ] \supset [^0Tom = \text{}^0Papež_{wt} ] ] \rightarrow o_{\tau\omega}$ .
  - $0, 5/\tau; :/(\tau\tau\tau); Tom/\iota; Papež/\iota_{\tau\omega}; [^0: \text{}^05 \text{}^00] \rightarrow \tau; [[^0: \text{}^05 \text{}^00] = \text{}^05] \rightarrow o; \text{}^0Tom \rightarrow \iota; \text{}^0Papež_{wt} \rightarrow_v \iota; [^0Tom = \text{}^0Papež_{wt}] \rightarrow_v o; [[ [^0: \text{}^05 \text{}^00] = \text{}^05 ] \supset [^0Tom = \text{}^0Papež_{wt} ] ] \rightarrow_v o$ .
  - Relaci = není na co aplikovat, tedy spojka implikace neobdrží první argument, proto je celá Kompozice nevlastní, nekonstruuje nic. *Parcialita je propagována nahoru*. Uzávěr – degenerovaná propozice

# Parcialita a kompozicionalita

- Pokud intuitivně cítíme, že danou větou chtěl mluvčí naznačit to, že není pravda, že pět děleno nulou je pět, tedy že věta by mohla být pravdivá, pak musíme analyzovat tuto větu:

„Jestliže je pravda, že pět děleno nulou je pět, pak Tom je papež“.

- $True^*(o*_n)$ : třída těch konstrukcí, které  $v$ -konstruuje  $T$  pro všechny valuace  $v$ .
- $[^0True^* \ ^0C]$  konstruuje  $T$ , právě když  $C$   $v$ -konstruuje  $T$  pro libovolnou valuaci  $v$ , jinak  $F$ .

$\lambda w \lambda t \ [[^0True^* \ ^0[[^0: \ ^05 \ ^00] = \ ^05]] \supset \ [^0Tom = \ ^0Papež_{wt}]]$

$\rightarrow o_{\tau\omega}$  - konstruuje propozici TRUE.

Věta je analyticky pravdivá

# Parcialita a kompozicionalita

- $False^*/(o*_n)$  a  $Improper^*/(o*_n)$  jsou třídy konstrukcí  $v$ -konstruujících  $\mathbf{F}$  resp.  $v$ -nevlastních pro všechny valuace  $v$ :
- $[^0True^* \ ^0C] = \neg[^0False^* \ ^0C] \wedge \neg[^0Improper^* \ ^0C]$
- $[^0False^* \ ^0C] = \neg[^0True^* \ ^0C] \wedge \neg[^0Improper^* \ ^0C]$
- $[^0Improper^* \ ^0C] = \neg[^0True^* \ ^0C] \wedge \neg[^0False^* \ ^0C]$

Podobně budeme často potřebovat *vlastnosti* propozic **True**, **False**, **Undefl**( $oo_{\tau\omega}$ ) $_{\tau\omega}$

# Pravidlo $\beta$ -transformace

- Základní výpočtové pravidlo  $\lambda$ -kalkulů a funkcionálních programovacích jazyků
- určuje, jak provést operaci *aplikaci funkce*  $f$  na argument  $A$  za účelem získání hodnoty funkce  $f$  na  $A$ .

*Př.:*  $[\lambda x [^0+ x ^0 1] ^0 3]$  – chci hodnotu funkce následníka na čísle 3:

$\beta$ -redukce (někdy také  $\lambda$ -redukce) „jménem“:

$$[\lambda x [^0+ x ^0 1] ^0 3] \Rightarrow [^0+ ^0 3 ^0 1] \quad (= ^0 4)$$

$\beta$ -rozvinutí (nebo také  $\lambda$ -rozvinutí):

$$[^0+ ^0 3 ^0 1] \Rightarrow [\lambda x [^0+ x ^0 1] ^0 3]$$

- Redukce obecně:  $[[\lambda x_1 \dots x_m Y] D_1 \dots D_m] \vdash Y(D_i/x_i)$

# $\beta$ -conversion: $[\lambda x C(x) A] \mid \text{---} C(A/x)$

- *Procedure* of applying the function presented by  $\lambda x C(x)$  to an argument presented by  $A$ .
- The fundamental computational rule of  $\lambda$ -calculi and functional programming languages
- The fundamental inference rule of HOL

‘**by name**’; the *procedure*  $A$  is substituted for all the occurrences of  $x$

□ **not operationally equivalent**

‘**by value**’; the *value* presented by  $A$  is substituted for all the occurrences of  $x$

# $\beta$ -conversion: $[\lambda x C(x) A] \mid \text{---} C(A/x)$

- In programming languages the difference between ‘by value’ and ‘by name’ revolves around the programmer’s choice of *evaluation strategy*.
  - Algol’60: “call-by-value” and “call-by-name”
  - Java: manipulates objects “by name”, however, procedures are called “by-value”
  - Clean and Haskell: “call-by-name”
- Similar work has been done since the early 1970s; for instance, Plotkin (1975) proved that the two strategies are not operationally equivalent.
- *Chang & Felleisen (2012)’s call-by-need reduction by value*. But their work is couched in an untyped  $\lambda$ -calculus.

---

$[\lambda x C(x) A] \mid - C(A/x)$

- *Conversion by name*  $\rightarrow$  three problems.
    1. conversion of this kind is *not guaranteed to be an equivalent transformation* as soon as partial functions are involved.
    2. even in those cases when  $\beta$ -reduction is an equivalent transformation, it can yield a *loss of analytic information* of which function has been applied to which argument
    3. In practice *less efficient* than ‘by value’
-



# Problems with $\beta$ -reduction 'by name'

## 1) non-equivalence

$$[\lambda x [\lambda y [^0+ x y]] [^0Cotg \ ^0\pi]]$$

is an **improper** construction; it does not construct anything, because there is no value of the cotangent function at  $\pi$

but its  $\beta$ -reduced Composition

$$[\lambda y [^0+ [^0Cotg \ ^0\pi] y]]$$

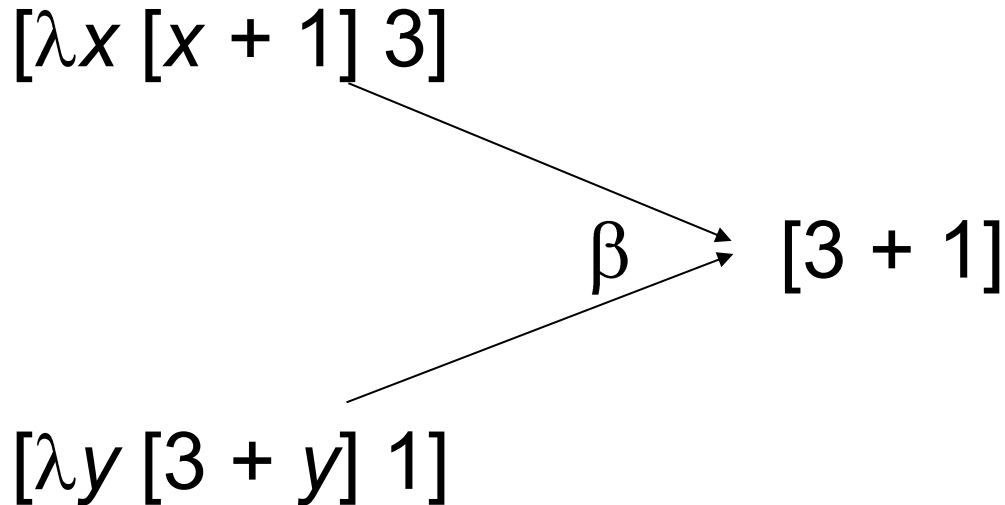
**constructs** a degenerate function

- The improper construction  $[^0Cotg \ ^0\pi]$  has been drawn into the intensional context of the Closure  $[\lambda y [^0+ x y]]$ .

---

$\beta$ -conversion by name:

2) loss of analytic information



which function has been applied to which argument?

No 'backward path'. Does it matter?

---

# Problems with $\beta$ -reduction

## 2) Loss of analytic information

- “John loves his wife, and so does Peter”  
→ **exemplary husbands** (sloppy reading)
- “loving one’s own wife” vs. “loving John’s wife”

$L^{own}(\text{John})$ :  $\lambda w \lambda t [\lambda x [{}^0\text{Love}_{wt} x [{}^0\text{Wife\_of}_{wt} \mathbf{x}]] {}^0\text{John}]$

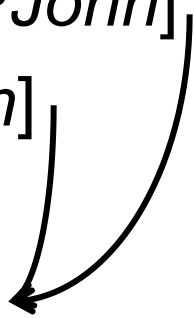
$L^{John}(\text{John})$ :  $\lambda w \lambda t [\lambda x [{}^0\text{Love}_{wt} x [{}^0\text{Wife\_of}_{wt} {}^0\mathbf{John}]] {}^0\text{John}]$

Both  $\beta$ -reduce to  $L^{John}(\text{John})$ :

$\lambda w \lambda t [{}^0\text{Love}_{wt} {}^0\text{John} [{}^0\text{Wife\_of}_{wt} {}^0\mathbf{John}]]$

- “so does Peter”
- Peter loves *John’s* wife → **trouble on the horizon**

# $\beta$ -conversion by name: loss of info

- (1)  $\lambda w \lambda t [\lambda x [{}^0\text{Love}_{wt} x [{}^0\text{Wife\_of}_{wt} {}^0\text{John}]] {}^0\text{John}]$
- (2)  $\lambda w \lambda t [\lambda x [{}^0\text{Love}_{wt} x [{}^0\text{Wife\_of}_{wt} x]] {}^0\text{John}]$
- (3)  $\lambda w \lambda t [{}^0\text{Love}_{wt} {}^0\text{John} [{}^0\text{Wife\_of}_{wt} {}^0\text{John}]]$
- 

It is uncontroversial that the contractum (3) can be equivalently expanded back both to (1) and (2).

The problem is, of course, that there is no way to reconstruct *which* of (1), (2) would be the correct redex

---

## *Does it matter?*

- HOL tools are broadly used in automatic theorem checking and applied as interactive proof assistants.
  - The underlying logic is usually a version of *simply typed*  $\lambda$ -calculus of *total functions*.
  - However, there is another application  $\rightarrow$  *natural language processing*  $\rightarrow$  hyperintensional logic is needed so that the underlying *inference machine is neither over-inferring* (that yields inconsistencies) *nor under-inferring* (that causes lack of knowledge).
  - agents' attitudes like knowing, believing, seeking, solving, designing, etc., because attitudinal sentences are part and parcel of our everyday vernacular.
-

# Hyperintensionality

- was born out of a negative need, to block invalid inferences
  - Carnap (1947, §§13ff); there are contexts that are neither extensional nor intensional (attitudes)
  - Cresswell; any context in which substitution of necessary equivalent terms fails is hyperintensional
- Yet, which inferences are valid in hyperintensional contexts?
- How hyper are hyperintensions? → procedural isomorphism
- Which contexts are hyperintensional?
- *TIL definition is positive*: a context is *hyperintensional* if the very meaning *procedure* is an object of predication; TIL is a ***hyperintensional, partial typed  $\lambda$ -calculus***

# $\beta$ -reduction by value

$$[\lambda x C(x) A] \mapsto C(A/x)$$

underspecified:

- How to execute  $C(A/x)$ ?
- a) 'by name': *construction*  $A$  is substituted for  $x \rightarrow$  problems
- b) 'by value': execute  $A$  first, and only if it does not fail, substitute the produced *value* for  $x$   
– *substitution method*  $\rightarrow$  bingo, no problems !!! 😊

## *Substitution 'by value'*

$$[\lambda x F(x) A] = {}^2[{}^0\text{Sub } [{}^0\text{Tr } A] \ {}^0x \ {}^0F(x)]$$

1. *A*: execute *A* in order to obtain the value *a*; if *A* is *v-improper*, then the whole Composition is *v-improper* (stop); else:
  2.  $[{}^0\text{Tr } A]$ : obtain Trivialization of ("pointer at") the argument *a*
  3.  $[{}^0\text{Sub } [{}^0\text{Tr } A] \ {}^0x \ {}^0F]$ : substitute this Trivialization for *x* into 'the body' *F*
  4.  ${}^2[{}^0\text{Sub } [{}^0\text{Tr } A] \ {}^0x \ {}^0F]$ : execute the result
-



# *Substitution 'by value'*

*Subl*(\*<sub>n</sub>\*<sub>n</sub>\*<sub>n</sub>\*<sub>n</sub>) operuje na konstrukcích takto:

[<sup>0</sup>Sub C<sub>1</sub> C<sub>2</sub> C<sub>3</sub>]

co za\_co kam

Nechť C<sub>1</sub> v-konstruuje konstrukci D<sub>1</sub>,

C<sub>2</sub> v-konstruuje konstrukci D<sub>2</sub>,

C<sub>3</sub> v-konstruuje konstrukci D<sub>3</sub>,

konstruuje konstrukci D, která vznikne korektní substitucí D<sub>1</sub> za D<sub>2</sub> do D<sub>3</sub>

*Trl*(\*<sub>n</sub> α) konstruuje Trivializaci α-objektu

[<sup>0</sup>Tr x] v-konstruuje Trivializaci objektu v-konstruovaného proměnnou x, x je **volná**

<sup>0</sup>x konstruuje x bez ohledu na valuaci, proměnná x je **o-vázaná**

---

# *Substitution 'by value'*

## *Příklad*

$[^0\text{Sub } [^0\text{Tr } ^0\pi] \ ^0x \ ^0[^0\text{Sin } x]]$

konstruuje *konstrukci*  $[^0\text{Sin } ^0\pi]$

$^2[^0\text{Sub } [^0\text{Tr } ^0\pi] \ ^0x \ ^0[^0\text{Sin } x]]$

konstruuje *hodnotu* funkce *Sinus* na  $\pi$ ,  
tj. číslo 0

$[^0\text{Sub } [^0\text{Tr } y] \ ^0x \ ^0[^0\text{Sin } x]]$

$v(\pi/y)$ -konstruuje *konstrukci*  $[^0\text{Sin } ^0\pi]$

---

---

## Substitution method; *broadly applied*

- Application of a function to an argument ( $\beta$ -reduction by value)
  - Existential quantification into hyperintensional contexts
  - Hyperintensional attitudes *de re*
  - Anaphoric preprocessing
  - Topic/focus articulation; presuppositions; active vs. passive
-

---

Substitution method; *broadly applied*

*de re* attitudes

*Tilman believes **of** the Pope that **he** is wise*

$\lambda w \lambda t [{}^0 \text{Believe}_{wt} {}^0 \text{Tilman} {}^2 [{}^0 \text{Of} [{}^0 \text{Tr} {}^0 \text{Pope}_{wt}] {}^0 \text{he}$   
 ${}^0 [\lambda w^* \lambda t^* [{}^0 \text{Wise}_{w^* t^*} \text{he}]]]$

*Of = Sub* operates on the (hyper)intensional context of “that he is wise”

---

# Substitution method; *broadly applied*

*Quantifying into ...*

- Tom is seeking the last decimal of  $\pi$
- 

There is a number such that Tom is seeking its last decimal

- $\lambda w \lambda t [{}^0\text{Seek}_{wt}^* {}^0\text{Tom } {}^0[{}^0\text{Last\_Dec } {}^0\pi]]$
- 

$\lambda w \lambda t [{}^0\exists \lambda x [{}^0\text{Seek}_{wt}^* {}^0\text{Tom } [{}^0\text{Sub } [{}^0\text{Tr } x] {}^0y$   
 ${}^0[{}^0\text{Last\_Dec } y]]]]$

---

---

How hyper are hyperintensions →

procedural isomorphism

- Maybe that it is philosophically wise to adopt several notions of procedural isomorphism.
  - It is not improbable that several degrees of hyperintensional individuation are called for, depending on which sort of discourse happens to be analysed.
  - What appears to be synonymous in an ordinary vernacular might not be synonymous in a professional language like the language of logic, mathematics, computer science or physics.
-

# Procedural isomorphism

- Ordinary vernacular – no variables →  
(A1'''):  $\alpha$ -conversion +  $\beta$ -conversion by value  
+ *restricted*  $\beta$ -conversion by name;  
 $[\lambda x [^0+ x ^00] y] \rightarrow [^0+ y ^00]$   
+ pairs of simple synonyms
- Programming language – variables matter →  
(A0'):  $\alpha$ -conversion + pairs of simple synonyms