Homotopy and Type Theory

February 13, 2018

Part I
Untyped lambda calculus

Description of the most basic behaviour of fucntions. For Comparison. Objects of set theory: *sets*

$$1 \cup (\leq \setminus +) \cap \mathbb{R},$$

is a valid statement, in set theory. = depends on the implementation of $1, \leq, +, \mathbb{R}$.

Objects of λ-calculus: *functions only*.

Basic operations of λ-calculus: Application and abstraction

Definition. The set Λ of all λ-terms ("set" to make meta-statements on λ-calculus)

(1) (Variable) We have another set $V \ldots$ set of variables.
If $u \in V$ then $u \in \Lambda$

(2) (Application) If $M, N \in \Lambda$, then $(MN) \in \Lambda$

(3) (Abstraction) If $u \in V$, $M \in \Lambda$, then $(\lambda u.M) \in \Lambda$

These are the only ways to construct λ-terms.

Short form $\Lambda = V \mid (\Lambda\Lambda) \mid (\lambda V.\Lambda)$[1]

Notation:

- Elements of V : $a, b, c, a', a'', a_1, a_2, \ldots$
- Elements of Λ : $A, B, C, A', A'', A_1, A_2, \ldots$ (meta-variables: represent an arbitrary λ-term)

Example. $x, y, z, (xx), (x(xz)), (\lambda x.(xz)), (y(\lambda x.(xz)) \in \Lambda$

Definition (Syntactical identity $\equiv$)**.** $(xz) \equiv (xz)$ but $(xz) \not\equiv (xy)$

[1] $\mid$ stands for "or", $V = \{a, b, c, \ldots\} \ldots$variables, are also functions

Definition. Multiset[2] Sub of subterms of a λ-term

(1) (Variable): $\mathrm{Sub}(x) = \{x\}$ for all $x \in V$

(2) (Application): $\mathrm{Sub}((MN)) = \mathrm{Sub}(M) \cup \mathrm{Sub}(N) \cup \{(MN)\}$[3]

(3) (Abstraction): $\mathrm{Sub}((\lambda x.M)) = \mathrm{Sub}(M) \cup \{(\lambda x.M)\}$

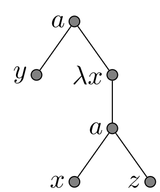
$L \in \Lambda$ is called a *subterm of* $M \in \Lambda$ if $L \in \mathrm{Sub}(M)$[4]

(1) (Reflexivity) $M \in \mathrm{Sub}(M)$

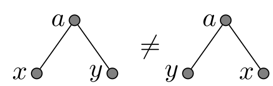
(2) (Transitivity) $L \in \mathrm{Sub}(M), M \in \mathrm{Sub}(N) \Rightarrow L \in \mathrm{Sub}(N)$

Example. "tree"[5] of subterms

$$\mathrm{Sub}((y(\lambda x.(xz)))) = \{(y(\lambda x.(xz))), y, (\lambda x.(xz)), (xz), x, z\}$$



"tree" of $y(\lambda x.(xz))$



"trees" are not commutative

Notation:

- drop outermost brackets: $MN = (MN)$[6]
- application is left associative: $MNL = ((MN)L)$
- abstraction is right associative: $\lambda xy.M = \lambda x.(\lambda y.M)$ and use only one λ
- application takes precedence over abstraction: $\lambda x.MN = \lambda x.(MN)$

H.B. Currying: $f : \underset{(x,y)\mapsto x+y}{\mathbb{R}^2 \to \mathbb{R}} \to \overline{f} : \underset{x\mapsto \left(f(x):\, \underset{y\mapsto x+y}{R\to\mathbb{R}}\right)}{\mathbb{R} \to \left(\mathbb{R}^{\mathbb{R}}\right)}$

(For-Later-Example: $(\lambda xy.x)5 \to_\beta \lambda y.5$)

Definition (free, bound, and binding of variables of λ-terms)**.** We call $\mathrm{FV}(M)$ the *set of free variables* of M for $M \in \Lambda$

(1) (Variable) $\mathrm{FV}(x) = \{x\}$

---

[2] may contain identical elements, multiple times
[3] unions of multisets, $\{x\} \cup \{x\} = \{x, x\} \neq \{x\}$
[4] $(L \neq M)$
[5] not a tree from graph theory because embedding matters
[6] = stands for "stands for" (in proper context)

(2) (Application) $\mathrm{FV}(MN) = \mathrm{FV}(M) \cup \mathrm{FV}(N)$

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

- x *free* in M if $x \in \mathrm{FV}(M)$
- x *bound* in M if $x \in \mathrm{B}(M)$
- x *binding* in M if $x \in \mathrm{B}_i\,(M)$:

(1B): $\mathrm{B}(x) = \{\}$
(2B): $\mathrm{B}(MN) = \mathrm{B}(M) \cup \mathrm{B}(N)$
(3B): $\mathrm{B}(\lambda x.M) = \mathrm{B}(M) \cup (\{x\} \cap \mathrm{FV}\,(M))$
(3B$_i$): $\mathrm{B}_i\,(x) = \{\}$
(3B$_i$): $\mathrm{B}_i(MN) = \mathrm{B}_i(M) \cup \mathrm{B}_i(N)$
(3B$_i$): $\mathrm{B}_i(\lambda x.M) = \mathrm{B}_i(M) \cup \{x\}$

Example. $\mathrm{FV}(\lambda x.xy) = \mathrm{FV}\,(xy) \setminus \{x\} = (\mathrm{FV}(x) \cup \mathrm{FV}\,(y)) \setminus \{x\} = \{x, y\} \setminus \{x\}$
$\mathrm{FV}\,(x\,(\lambda x.xy)) = \{x, y\}$ (here, the 1st x is free, the 2nd is a binding and the 3rd is a bound)

Definition (*closed* λ-term or *combinator*)**.** $M \in \Lambda$ is called *closed* if $\mathrm{FV}(M) = \emptyset$. Denote the set of all closed λ-terms by Λ°

Definition (alpha conversion, renaming, $M^{x \to y}$,$=_\alpha$)**.** For $M \in \Lambda$, $x, y \in V$ let $M^{x \to y} \in \Lambda$ denote the λ-term obtained by replacing every free occurence of x by y.

Example. $(x\,(\lambda x.xy))^{x \to y} \equiv y\,(\lambda x.xy)$

For $M \in \Lambda$, $x, y \in V$ with $y \notin \mathrm{FV}\,(M)$ and $y \notin \mathrm{B}_i(M)$ (i.e. y does not occur in M) we define the notation *renaming* by $\lambda x.M =_\alpha \lambda y.M^{x \to y}$.

We say "$\lambda x.M$ has been renamed by $\lambda y.M^{x \to y}$"

Conditions on renaming: Renaming should not change the "status" (free, bound, binding) of the variable

- If $y \in \mathrm{FV}(M)$: $(\lambda x.y)^{x \to y}$ would become $\lambda y.y$ (y would change its status from free to bound)
- If $y \in \mathrm{B}_i(M) : (\lambda xy.x)^{x \to y}$ would become $\lambda yy.y$ (while the rightmost x is bound to the first λ the rightmost y afterwards is bound to the second λ)

(For-Later-Example: $(\lambda xy.x)5 \to_\beta \lambda y.5$ while $(\lambda yy.y)5 \to_B \lambda y.y$)

Definition (λ-conversion)**.** extend $=_\lambda$

(1) (Renaming) $\lambda x.M =_\alpha \lambda y.M^{x \to y}$ if $y \notin \mathrm{FV}(M), y \notin \mathrm{B}_i(M)$

(2) (Compatability)[7] If $M =_\alpha$, then $ML =_\alpha NL$, $LM =_\alpha LN$, $\lambda z.M =_\alpha \lambda z.N$ for all $L \in \Lambda$, $z \in V$

[7] α-conversion within subterms

(3a) (Reflexivity) $M =_\alpha M$

(3b) (Symmetry) $M =_\alpha N \Rightarrow N =_\alpha M$

(3c) (Transitivity) $L =_\alpha M, M =_\alpha N \Rightarrow L =_\alpha N$

If $M =_\alpha N$ then M and N are said to be $\underline{\alpha - equivalent}$.