Lambda Calculus and category theory

Valeran MAYTIE

Contents

1 Introduction

Boole :

- If you consider propositions (no quantifiers) of <u>classical logic</u> : $A ::= P|A \wedge B|\neg A|A \wedge B|\top|\bot$
- <u>Ordered</u> by logical implication $A \leq B \Leftrightarrow A \Rightarrow B$, A implies B or $A \vdash B$

<u>Observation</u> $A \wedge B \leq A, A \wedge B \leq B$. moreover if $C \leq A$ and $C \leq B$ then $C \leq A \wedge B$ (for all proprieties) Which means that $A \wedge B$ define a infimum of A and B (greatest lower bound, or <u>glb</u>)

Definition – $A \Rightarrow B = (\neg A) \vee B = \neg(A \wedge \neg B)$.

Observation :

- $A \wedge (A \Rightarrow B) \leq B$
- $A \vee \neg A \leq$ true
- $A \wedge \neg A \geq$ false

Frege Ideography (first proof system)

The idea that a mathematical proof is a mathematical object. In particular there may be different proofs of a proposition A formula.

$$\begin{tikzcd} B \\ A \arrow[u, bend left, "\pi_1"] \arrow[u, bend right, "\pi_2"'] \arrow[u, phantom, "\neq"] \end{tikzcd} \qquad\qquad \begin{tikzcd} B \\ A \arrow[u, dash, "\leq"'] \end{tikzcd}$$

Lambek Lambek

Lambek understood connection between:

$$\begin{tikzcd} \text{Intuitionists minimal propositional logic} \arrow[rr, leftrightarrow, "\text{Curry, Howard, Lambek}"] \arrow[dr, leftrightarrow, "\text{Curry, Howard}"'] & & \text{Cartesian closed category} \arrow[dl, leftrightarrow, "\text{Lambek}"] \\ & \text{Simply typed lambda calculus} & \end{tikzcd}$$

Definition – A monoid $(M, \bullet, e)$ is a set M equipped with a binary operation $\bullet : M \times M \to M$ with a neutral element $e \in M_e : M^0 \to M$ satisfying two equations :

- (associativity) $\forall x, y, z \in M, x \bullet (y \bullet z) = (x \bullet y) \bullet z$
- (neutrality) $\forall x, \in M, x \bullet e = x = e \bullet x$

Example – $(\mathbb{N}, +, 0), (\mathbb{Z}, +, 0), (\mathbb{N}, \times, 1)$ and any group.
Free monoid on a set (=alphabet) A. A^* contains finite sequences of element A $w = [a_1 \ldots a_n]$

- Binary operation is concatenation.
- Neutral element is the empty word.

2 Categories

Definition – A category $\mathcal{C}$ is a graph

- Whose nodes are called objects
- Whose edges are called morphism/maps/arrow.

The objects of $\mathcal{C}$ form a class of objects.
Every pair of object A, B comes with a set $Hom(A, B)$ of morphisms $A \xrightarrow{f} B$, $f \in Hom(A, B)$
The graph is equipped with:

- A morphism $id_A \in Hom(A, A)$ for all object A of $\mathcal{C}$
- A composition defined as a function $\circ_{A,B,C} : Hom(B, C) \times Hom(A, B) \to Hom(A, C)$ for every objects A, B, C of $\mathcal{C}$

  It satisfying the following equation :

  – associativity :

$$\begin{tikzcd} & B \arrow[r, "g"] & C \arrow[dr, "h"] & \\ A \arrow[ur, "f"] \arrow[urr, "g \circ f"'] \arrow[rrr, "h \circ g \circ f"'] & & & D \end{tikzcd} \qquad \begin{tikzcd} & B \arrow[r, "g"] \arrow[drr, "h \circ g"'] & C \arrow[dr, "h"] & \\ A \arrow[ur, "f"] \arrow[rrr, "h \circ g \circ f"'] & & & D \end{tikzcd}$$

$$h \circ (g \circ f) = (h \circ g) \circ f$$

  – neutrality :

$$\begin{tikzcd} A \arrow[loop above, "Id_A"] \arrow[r, "f"'] & B \arrow[loop above, "Id_B"] \end{tikzcd}$$

$$Id_B \circ f = f = f \circ Id_A$$

Definition – A small category is a category whose class of object is a set. What we defined as a category is called "locally small category".

Example – Ordered Set: Every ordered set A defines a category.

- Objects: elements of A
- Morphisms : $a \to b \Leftrightarrow a \leq b$

$$Hom(a, b) = \begin{cases} \text{singleton} & a \leq b \\ \emptyset \end{cases}$$

  The composition is defined by transitivity:

$$\begin{tikzcd}
a \arrow[r, "a \leq b"'] & b \arrow[r, "b \leq c"'] & c \\
a & \leq & c \\
a \arrow[rr] & & b
\end{tikzcd}$$

Definition – An ordered category $\mathcal{C}$ is a category where $Hom(A, B)$ is a singleton for all object A, B of $\mathcal{C}$.

Observation – An ordered category is the same thing as a pre-order (= trans, refl).

Example – <u>Monoid</u>

- A category with one object $*$, $M = Hom(*, *)$ define a monoid.
  - $\circ : Hom(*, *) \times Hom(*, *) \to Hom(*, *)$
  - $id_* \in M = Hom(*, *)$ define the neutral element
- Conversely every monoid $M = (M, \bullet, e)$ defines a category $\mathcal{B}M$ or ΣM with:
  - One object $*$
  - $Hom(*, *) = M$
  - Composition defined by $y \circ x = y \bullet x$ with e, the neutral element.

$$\begin{tikzcd}
 & 2 = 1 \circ 1 \\
\mathcal{B}(\mathbb{N}, +, 0) & * \arrow[loop above, "0"] \arrow[loop below, "1"] \\
 & 3 = 2 \circ 1
\end{tikzcd}$$

3 Functors

A functor $F : \mathcal{A} \to \mathcal{B}$ between category $\mathcal{A}$ and $\mathcal{B}$ is a graph homomorphism :

- F associates an object FA in $\mathcal{B}$ to every object A in $\mathcal{A}$
- F associates a morphism $FA \xrightarrow{Ff} FA'$ in $\mathcal{B}$ to every morphism $A \xrightarrow{f} A'$ in $\mathcal{A}$.

It preserves composition and identity

$$\begin{tikzcd}
A' \arrow[rr, mapsto] & & FA' \\
f \arrow[rr, mapsto] & & Ff \\
A \arrow[rr, mapsto] & & FA
\end{tikzcd}$$

$$\begin{tikzcd}
A & FA \\
A \arrow[u, "Id_A"'] & FA \arrow[u, "F(Id_A) = Id_{FA}"']
\end{tikzcd}
\qquad
\begin{tikzcd}
A'' & FA'' \\
A' \arrow[u, "g"] & FA' \arrow[u, "Fg"'] \\
A \arrow[u, "f"] \arrow[uu, bend right, "g \circ_A f"'] & FA \arrow[u, "Ff"'] \arrow[uu, bend right, "Fg \circ_b Ff = F(g \circ_A f)"']
\end{tikzcd}$$

Example – List of different application :

- A Functor $F : \mathcal{A} \to \mathcal{B}$ between order category is a same thing as a monotone (=order preserving) function. The reason is that the preservation of composition and identity holds in $\mathcal{B}$.

- A functor $F : \mathcal{B}M \to \mathcal{B}N$ between categories with one object is a function $H \to M$ which defined monoid homomorphism.

$$F(m \circ_M n) = Fm \circ_N Fn$$

- functor $F : \mathcal{B}M \to \mathit{Set}$ is a same thing as a $\mathit{Set}\ X = F(*)$ equipped with a family of functions $F_m : X \to X$ satisfying the equation :
  - $Fm \circ Fn = F(m \bullet n)$
  - $F(e) = Id_X$

  $\hookrightarrow$ Equivalently a function

$$\begin{aligned} M \times X &\mapsto X \\ (m, x) &\to m \bullet x \end{aligned}$$

  Where we write $F_m(x) = m \bullet x$ and <u>action</u> of Mon X satisfying the equations of $(m \bullet_M n) \bullet x = m \bullet (m \bullet x)$ and $e \bullet x$.

  Example – Given a finite set A (alphabet) of letters. We construct the monoid A^* of finite words on A equipped with concatenation as composition and empty element as neutral element. We write $[a_1 \dots a_n] \in A^*$ and ϵ : the empty word noted $[]$. A functor $\mathcal{B}A^* \xrightarrow{F} \mathit{Set}$ is a set $Q = F(*)$ equipped with an action that it a function :

$$\begin{aligned} A^* \times Q &\to Q \\ ([w], q) &\to [w] \bullet q \end{aligned}$$

  Remark – this action $A \times Q \to Q$ is equivalent to the data of a family $\delta_a : Q \to Q$ of function called transition function of a deterministic total automata with Set of Q of states.

  Exercise – Given a functor $\mathcal{B}M \to \mathit{Set}$ construct a category $\int F$ (Grothendieck construction) whose object are the elements of $F(*)$, whose morphisms $x \xrightarrow{m} y$ are of the form $x \mapsto m \bullet x$. Show that $\int F$ comes with a functors $\pi : \int F \to \mathcal{B}M$.

4 Transformation

Suppose given functors $F, G : \mathcal{A} \to \mathcal{B}$ we want to "compare" F and G in the same way as we compare two monotone functions $f, g : (A, \leq_A) \to (B \leq_g)$ between ordered set.

<u>In ordered Set we have</u> :

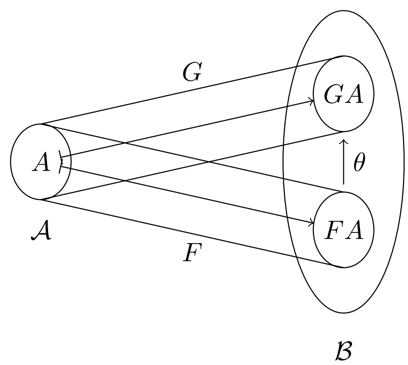
$$\begin{tikzcd} A \arrow[r, bend left, "f"] \arrow[r, bend right, "g"'] \arrow[r, phantom, "\leq"] & B \end{tikzcd}$$

$$f \leq_A g \Leftrightarrow \forall a \in A, f(a) \leq_B g(a)$$

Definition – A transformation $\theta : F \Rightarrow G : \mathcal{A} \to \mathcal{B}$ is a family of morphisms $\theta : FA \to GA$ in $\mathcal{B}$ parametrised by the object A of $\mathcal{A}$

Notation – We write :

$$\begin{tikzcd} \mathcal{A} \arrow[r, bend left, "F"] \arrow[r, bend right, "G"'] \arrow[r, phantom, "\Uparrow\theta"] & \mathcal{B} \end{tikzcd}$$

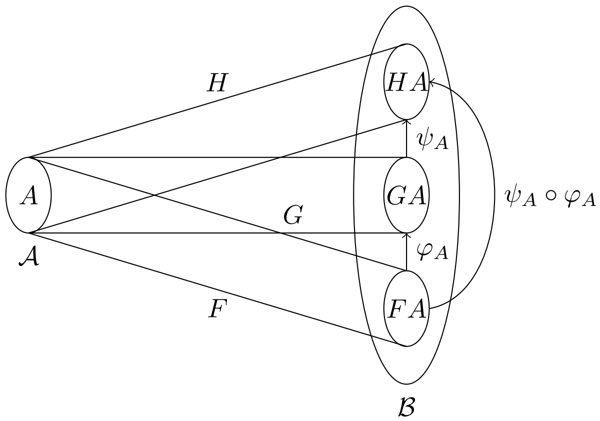


Fact – Every pair of categories $\mathcal{A}, \mathcal{B}$ defined a category $\mathit{Trans}(\mathcal{A}, \mathcal{B})$ object are functors $F : A \to B$ and morphism are transformation $\theta : F \Rightarrow G$.

$$\begin{aligned} \psi &: G \Rightarrow H \\ \varphi &: F \Rightarrow G \end{aligned} \qquad \begin{tikzcd} \mathcal{A} \arrow[r, bend left=60, "H"] \arrow[r, "G" description] \arrow[r, bend right=60, "F"'] \arrow[r, phantom, shift left=3ex, "\Uparrow\psi"] \arrow[r, phantom, shift right=3ex, "\Uparrow\varphi"] & \mathcal{B} \end{tikzcd}$$

The composite transposition it's easy to define $(\psi \circ \varphi)_A = \psi_A \circ \varphi_A$ and the identity is $Id_F : F \Rightarrow F$ $(Id_F)_A = Id_{FA}$.

We can write this :



4.1 Post-composition

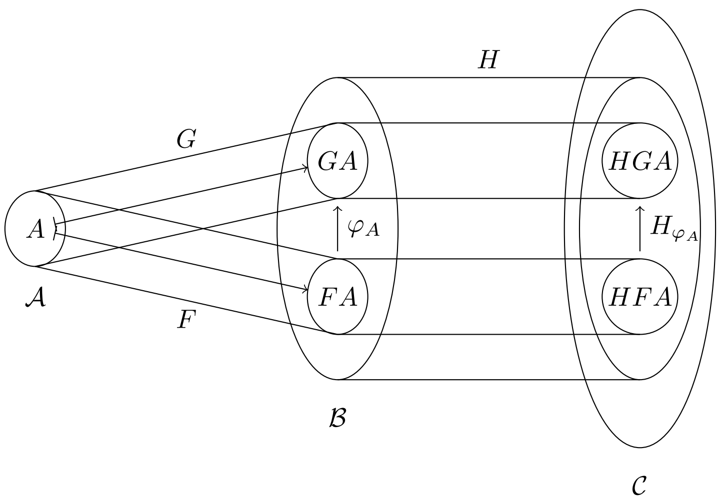
Suppose – Given a transformation $\varphi : F \Rightarrow G : \mathcal{A} \to \mathcal{B}$ and a functor $H : \mathcal{B} \to \mathcal{C}$

$$\begin{tikzcd} \mathcal{A} \arrow[r, bend left=50, "F"] \arrow[r, bend right=50, "G"'] \arrow[r, phantom, "\Uparrow\varphi"] & \mathcal{B} \arrow[r, "H"] & \mathcal{C} \end{tikzcd}$$

We define the transformation

$$\begin{aligned} H \circ_l \varphi_A : \mathcal{A} &\mapsto \varphi \\ HFA &\to HGA \end{aligned}$$

We can represent this transformation like this :



4.2 Pre-composition