Proofs and Programs

Phillipe Audebaud *
ENS de Lyon

Contents

*https://perso.ens-lyon.fr/philippe.audebaud/PnP/

Basis

- Lecture: Tue 8h-10h (Philippe Audebaud)
- Tutorial: We 8h-10h (Aurore)

10 Weeks of courses (3x3), which is really low.

$$Final\ mark = 50\% \cdot CC + 50\% \cdot Exam$$

No mid-time exam, but weekly homework.

Warning Presence at the courses and tutorial will have an impact on the marks.

Prerequisites

- L2.2 → Logical (Natacha P., Chapter 1 & 2):
  - Proof theory
  - Formal system for logic inference.
- λ-calculus
- Category theory

Part I
(Pure) λ-Calculus

0.1 Computing with functions ?

How do we do mathematics ?

A. Having *structures:* numbers, spaces (points, vectors, functions) → Eilenberg-Mac Lane ($\sim$ 1942) Category theory

B. Build, explore, transform structures → Church ($\sim$ 1930) λ-*Calculus*

C. Compare "stuff": *equality* → Voevoski ($\sim$ 2006) Algebraic topology → search HoTT (Hight order Type Theory)

D. Provide a framework (*rules*) to reasoning on all that! → 1st point

0.2 Church λ-calculus (informally)

$$\begin{aligned} f :& A \to B \\ & x \mapsto e \end{aligned}$$

Given $a \in A$, $f(a)$ is the "replacement of the occurrence of x in e by a"

$$f \stackrel{\text{def}}{=} \lambda x.e \qquad (\lambda\text{-abstraction})$$

$$f\ a = (\lambda a.e)\ a \qquad (\text{Application})$$

Notation

$$e\langle a/x\rangle$$

is the replacement in e of all the occurrences of a by x.

Example

1. $$\lambda x.x$$
   $$x \mapsto x$$

   is the identity function

2. $$\lambda x.y$$
   $$x \mapsto y$$

   Here x and y are variables, $x \neq y$. $(\lambda x.y)\ a$ leads to $y\langle a/x\rangle \equiv y$

   $$(\lambda x.a)\ b \to_\beta a\langle b/x\rangle$$

   $\to_\beta$ is a binary relation on lambda-terms $\Rightarrow$ idea of computation on terms.

Notion of α-equivalence

$$\lambda x.a \overset{?}{=}_\alpha \lambda y.b$$

Pick a *fresh* variable, let say z,

$$a\langle z/x\rangle =_\alpha b\langle z/y\rangle$$

All the results and proofs will be done up to α-equivalence (no difference made between $\lambda x.x$ and $\lambda y.y$).

1 A toolbox on λ-calculus

λ-calculus: Syntax and Semantics, Herk Barendregt (1977)

Let $\mathcal{X}$ be a measurable set of variables, ranged over by $x, y, z, ...$

Definition 1. *A λ-term e is generated by the grammar:*

$$a, b, e... ::= x \in \mathcal{X} \mid \lambda x.e \mid a\ b$$

The set of λ-terms is denoted Λ.

Definition 2 (Free variable)**.** *The set of free variables in e, denoted $FV(e)$ is defined inductively:*

- *if $e \equiv x \in \mathcal{X}$, $FV(x) \equiv \{x\}$*
- *if $e \equiv \lambda x.a_0$, $FV(\lambda x.a_0) \equiv FV(a_0) \setminus \{x\}$*
- *if $e \equiv a_1\ a_2$, $FV(a_1\ a_2) \equiv FV(a_1) \cup FV(a_2)$*

A term e is closed *if $FV(e) = \emptyset$*

Definition 3 (Substitution)**.** *Given $x \in \mathcal{X}$, $a \in \Lambda$, the substitution of (all the) occurrences of a in $e \in \Lambda$, denoted $e\langle a/x\rangle$ is:*

- *if* $y \in \mathcal{X} \setminus \{x\},\ y\langle a/x\rangle \equiv y$ *and* $x\langle a/x\rangle \equiv a$
- $(\lambda y.e)\langle a/x\rangle = \lambda y.e\langle a/x\rangle$
- $(e\ f)\langle a/x\rangle = (e\langle a/x\rangle)\ f\langle a/x\rangle$

Definition 4 ($\rightarrow_\beta$ reduction)**.**

$$\rightarrow_\beta \subseteq \Lambda \times \Lambda$$
$$\left\{ \big(\underbrace{(\lambda x.a)\ b}_{redex},\ \underbrace{a\langle b/x\rangle}_{contraction} \big) \middle| x \in \mathcal{X},\ a, v \in \Lambda \right\}$$

Example

1.
$$\underbrace{(\lambda x.(\lambda y.y)\ a)}_{\twoheadrightarrow_\beta (\lambda y.y)\ b}\ b) \rightarrow_\beta ((\lambda y.y)\ a)\ \langle b/x\rangle \equiv ((\lambda y.y)\ \langle b/x\rangle)a\langle b/x\rangle$$
$$\equiv (\lambda y.y)\ a\langle b/x\rangle$$

2.
$$(\lambda x.y)\ a \rightarrow_\beta y$$

3.
$$(\lambda x.x\ x)(\lambda x.x\ x) \rightarrow_\beta (x\ x)\langle \lambda x.x\ x/x\rangle \text{ or } (x\ x)\langle \lambda y.y\ y/x\rangle$$
$$(\lambda x.x\ x)(\lambda x.x\ x)$$

Russell paradox: we get an infinite β-reduction!

$$\rightarrow_\beta \subseteq\ \beta_0\ \subseteq \underbrace{\beta}_{\beta-\text{reduction}} = \beta_0^*$$

$\rightarrow_\beta^*$ is the β-reduction, noted $\twoheadrightarrow_\beta$

Definition 5 (β_0-contraction)**.** *Let* $a, b \in \Lambda$*.* $a\ \beta_0\ b$ *is defined by cases:*

- $x\ \beta_0\ x$
- $(\lambda x.u)v\ \beta_0\ u\langle v/x\rangle$
- $(\lambda x.u)\ \beta_0\ (\lambda x.v)$ *if* $u\ \beta_0\ v$
- $(u\ v)\ \beta_0\ (u'\ v)$ *if* $u\ \beta_0\ u'$
- $(u\ v)\ \beta_0\ (u\ v')$ *if* $v\ \beta_0\ v'$

Maintenant en français !

Remarque: β_0 est réflexive.

Definition 6. *La* β-réduction *est la clôture transitive de* β_0*:*

$$\beta = \beta_0^*$$

Remarque Si $a, b \in \Lambda$, alors $a\ \beta\ b$ si il existe $n \geq 0$ et $(e_k)_{0 \leq k \leq n}$ λ-termes tels que :

- $a = e_0$ et $b = e_n$
- pour tout $k < n$, $e_k\ \beta_0\ e_{k+1}$

Definition 7. *Soit $\mathcal{R}$ une relation binaire sur Λ. On dit que $\mathcal{R}$ est* λ-compatible *si elle satisfait les propriétés suivantes :*

- $x\ \mathcal{R}\ x$
- *si* $a\ \mathcal{R}\ b$ *et* $c\ \mathcal{R}\ d$ *alors* $a\ c\ \mathcal{R}\ b\ d$
- *si* $a\ \mathcal{R}\ b$ *alors* $\lambda x.a\ \mathcal{R}\ \lambda x.b$

Propriety 1. *La β-réduction est la plus petite relation λ-compatible et transitive contenant $\rightarrow_\beta$*

Proof. $\star$ On vérifie d'abord :

$$\rightarrow_\beta \subseteq \beta_0 \subseteq \beta_0^* = \beta$$

D'autre part, β_0 est λ-compatible :

- par réflexivité, $x\ \beta\ x$
- soit $a\ \beta\ b$; par définition,il existe $n \geq 0$, $(e_k)_{0 \leq k \leq n}$ tel que $a = e_0$, $b = e_n$ et pour tout $k < n$, $e_k\ \beta_0\ e_{k+1}$. Du coup, par définition de β_0, pour tout $k < n$, $\lambda x.e_k\ \beta_0\ \lambda x.e_{k+1}$.

  Ainsi, $\lambda x.a\ \beta\ \lambda x.b$.

$\star$ Soit $\mathcal{R}$ une autre relation λ-compatible et transitive contenant $\rightarrow_\beta$. Montrons que $\beta \subseteq \mathcal{R}$. Il "suffit" de vérifier que $\beta_0 \subseteq \mathcal{R}$ (laissé en exercice). □

Propriétés essentielles de la β-réduction

Remarque (Λ, β_0) est un *système de réduction abstrait*[1].

Definition 8 (Forme normale, Relation normalisante)**.** *Soit $\mathcal{R}$ une relation binaire sur Λ,*

- *On dit que a* est *une forme normale (relativement à $\mathcal{R}$) s'il n'existe pas $b \in \Lambda$ tel que $a\ \mathcal{R}\ b$.*
- *On dit que a* a *une forme normale (relativement à $\mathcal{R}$) s'il existe $b \in \Lambda$ tel que b* est *une forme normale et $a\ \mathcal{R}^*\ b$*
- *On dit que $\mathcal{R}$ est normalisante si tout $a \in \Lambda$ a une forme normale*

Exemple

- $\lambda x.x$ est une forme normale relativement à β_0
- β_0 n'est pas normalisante !

$$\Omega \equiv (\lambda x.x\ x)\ (\lambda x.x\ x)$$
$$\Omega \rightarrow_\beta \Omega$$

Definition 9 (Confluence)**.** *Soit $\mathcal{R}$ une relation binaire sur Λ. On dit que $\mathcal{R}$ est confluente si pour tout $(a, b, c) \in \Lambda^3$ tel que*

$$a\ \mathcal{R}^*\ b \text{ et } a\ \mathcal{R}^*\ c$$

alors il existe $d \in \Lambda$, tel que

$$b\ \mathcal{R}^*\ d \text{ et } c\ \mathcal{R}^*\ d$$

Theorem 1. *La β_0-réduction est confluente.*

Proof. En semaine 3 ou 4. □

Corollary 1. *Tout λ-terme admet au plus une forme normale, relativement à β_0*

[1] Cf ThPr

Notion d'égalité sur les λ-termes

Definition 10. *La β-équivalence sur Λ est la relation binaire notée $=_\beta$, définie comme la clôture réflexive symétrique transitive de β_0 :*
$a =_\beta b$ s'il existe $n \geq 0$ et $(e_k)_{0 \leq k \leq n}$ tel que $a = e_0$ et $b = e_n$ et $\forall k < n$, soit $e_k \ \beta_0 \ e_{k+1}$ soit $e_{k+1} \ \beta_0 \ e_k$

Definition 11 (λ-congruence)**.** *Une relation binaire $\mathcal{R}$ (sur Λ) est une λ-congruence si c'est une relation d'équivalence et qu'elle est λ-compatible.*

Theorem 2 (Church-Rosser)**.** *Pour tout $(a, b) \in \Lambda^2$, $a =_\beta b$ si et seulement si il existe $c \in \Lambda$ tel que $a \ \beta \ b$ et $b \ \beta \ c$*

Proof. La condition est suffisante

Réciproquement, pour la condition nécessaire, on introduit $R \subseteq \Lambda \times A$ défini par :
$a \ \mathcal{R} \ b$ s'il existe c tel que $a \ \beta \ c$ et $b \ \beta \ c$.
On remarque, par définition de $\mathcal{R}$,

- $\mathcal{R}$ est réflexive et symétrique
- $\mathcal{R}$ est transitive

De plus, $\mathcal{R}$ contient β (ou β_0). Donc, si $a =_\beta b$, alors $a \ R \ b$. □

Theorem 3. *La relation d'équivalence $=_\beta$ est la plus petite λ-congruence contenant $\rightarrow_\beta$*

Proof. En exo. □

Notation On note $\equiv$ pour une définition ($\stackrel{\text{def}}{=}$), mais aussi pour l'α-équivalence ($=_\alpha$). On peut utiliser la notation de Bruijn (cf références).

2 Calcul propositionnel et correspondance de Curry-Howard

2.1 Éléments de langage (informels)

- Théorie de la *démonstration* (prouvabilité)
- Thème des modèles

Quelques "ingrédients" :

- *énoncés* (logiques) : ici les familles du calcul propositionnel:

$$A ::= x \mid \top \mid \bot \mid A \Rightarrow B \mid A \wedge B \mid A \vee B \mid \neg A^2 \qquad (\star)$$

  La notation "A propriété" signifie que A est engendrée par la grammaire ($\star$)

- On parle de *jugements* sur ces énoncés : "A true"
- On introduit aussi des jugements hypothétiques : A_1 true, A_2 true, ... ,A_n true $\vdash B$ true

Commentaires sur les différentes règles de (NJ):

- Le vrai
- L'implication (/!\: $A \Rightarrow B \neq \neg A \vee B$ dans (NJ))
- Le faux
- La négation
- La disjonction

[2] $\neg A$ signifie en fait $A \Rightarrow \bot$

2.2 Fragments $\lambda_\to$

On peut associer des règles au typages de λ-termes en raisonnant sur $\lambda x.t : T$

Theorem 4 (Curry-Howard)**.** *Le fragment $NJ_\to$ et $\lambda_\to$ sont en correspondance via:*

1. *Si $\Delta \vdash t : T$ dans $\lambda_\to$*

$\lambda_\to$	$NJ_\to$
type	*proposition*
variable de type	proposition atomique
type flèche	implication
terme	*dèrivation*
variable de terme	hypothèse
λ-abstraction	règle d'introduction
application	règle d'élimination
β-redex	coupure
β-réduction	transformation sur les dérivations
forme normale	*dérivation sans coupure*

Figure 1: Correspondance de Curry-Howard

2.3 Interprétation BHK

L'interprétation de Brouwer-Heyting-Kolmogorov consiste à construire un témoin (une preuve) d'une proposition selon le protocole suivant :

- Un témoin pour $A \wedge B$ est une paire formée par un témoin pour A et un témoin pour B
- Il y a un témoin unique pour $\top$
- Un témoin pour $A \vee B$ est soit un témoin pour A, soit un témoin pour B
- Il n'y a pas de témoin pour $\bot$
- Un témoin pour $A \Rightarrow B$ est une application de témoins pour A vers des témoins pour B
- Un témoin pour $\neg A$ est un témoin de $A \Rightarrow \bot$

Avec A, B engendrés par la grammaire

$$A ::= X \mid A \Rightarrow A \mid A \vee A \mid A \wedge A \mid \top \mid \bot$$

Definition 12 (Produit (paire))**.** *Soit A,B. Le produit de A par B est le damier de $A \times B$, et de la propriété universelle suivante :*
Pour tout $f : D \to A$ et $g : D \to B$, il existe $h : D \to A \times B$ tel que $\pi_1 \circ h = f$ et $\pi_2 \circ h = g$[3]*.*
De plus, h est unique *et ne dépend que de f et de g,*

$$h = \langle f, g \rangle$$

[3]Ces égalité correspondent à des β-réduction dans le λ-calcul

Par ailleurs, si $e : D \to A \times B$, alors

$$\begin{cases} \pi_1 \circ e : D \to A \\ \pi_2 \circ e : D \to B \end{cases}$$

Pour ce couple, il existe $\langle \pi_1 \circ e, \pi_2 \circ e \rangle : D \to A \times B$

Du coup, par unicité, on a nécessairement

$$\langle \pi_1 \circ e, \pi_2 \circ e \rangle = e$$

Cette observation donne lieu à :

- une transformation sur les dérivations
- une autre forme de réduction sur les λ-termes

On parle alors d'η-réduction.

On rajoute alors les règles de typage du produit $(\times_i)$ et $(\times_{E,k})$ pour $k \in \{1, 2\}$.

Definition 13 (Somme (coproduit))**.** *Soit A, B. C'est la donné de $A + B$ avec la propriété universelle suivante :*
Si $f : A \to C$, et $g : B \to C$, il existe $k : A + B \to C$ unique, ne dépendant que de f et de g noté $\{f, g\}$, tel que

$$\begin{cases} k \circ in_l = f \\ k \circ in_r = g \end{cases}$$

Par ailleurs, si on se donne

$$e : A + B \to C$$

Alors

$$\begin{cases} e \circ in_l : A \to C \\ e \circ in_r : B \to C \end{cases}$$

Donc

$$\{e \circ \in_l, e \circ in_r\} = e$$

On rajoute alors trois règles : (+Ig), (+Id) et (+E)

3 λ-calcul simplement typé

Quelques Lemmes

Lemma 1. *Si $\Delta \vdash t : T$ clos, $FV(t) \subseteq FV(\Delta)$, où $FV(\emptyset) = \emptyset$, et $FV(\Delta, x : S) = FV(\Delta) \cup \{x\}$.*

Attention *: Un contexte de typage $\Delta \equiv x_1 : S, ..., x_p : S_p$ où $p \geq 0$ est valide si les variables $x_1, ..., x_p$ sont distinctes deux à deux.*

On peut rajouter des règles sur la validité de Δ en tant que contexte.

$$\frac{}{\emptyset \text{ contexte valide}}$$

$$\frac{\Delta \text{ contexte valide} \qquad T \text{ type} \qquad x \notin FV(\Delta)}{\Delta, x : T \text{ contexte valide}}$$

Et on augmente (Hyp).

$$\frac{\Delta \text{ contexte valide} \qquad x : T \in \Delta}{\Delta, x : T \text{ contexte valide}} \text{ (Hyp)}$$

Lemma 2 (Affaiblissement)**.** *Si $\Delta \vdash t : T$ et si $\Delta \subseteq \Delta'$, avec Δ' contexte valide, alors $\Delta' \vdash t : T$.*

Proof. Par induction sur la dérivation principale, c'est-à-dire $\Delta \vdash t : T$. Le seul cas "délicat" est lorsque

$$\frac{\Delta, x : U \vdash a : V}{} \to_I$$

□

Theorem 5. *Si $\Delta \vdash t : T$, alor t est fortement normalisant.*

Proof. Deux parties : poser la notation générale, puis l'adapter à $\to_\lambda$.

1. *Définition générale :* Si $e \in \Lambda$, $e \equiv \lambda \overline{x}.\Delta \overline{u}$ avec $|\overline{x}| > 0$, $|\overline{u}| > 0$ et $\Delta \in \mathcal{X}$ ou bien Δ est un β-redex
   - e est en forme normale si $\Delta \in \mathcal{X}$ et chaque u_i est en forme normale
   - e est une forme normale de tête (HNF) si $\Delta \in \mathcal{X}$
   - si e n'est pas en HNF, c'est-à-dire Δ est un β-redex, Δ est appelé *redex de tête*.

Definition 14. *$e \in \Lambda$ est fortement normalisant (SN) s'il n'existe pas de β-réduction infinie issue de e*

Exemple

- Ω n'est pas SN
- $(\lambda x.\lambda y.y\Omega$ n'est pas SN (il existe une dérivation infinie) $\to$ *attention* : la β-équivalence n'est pas compatible avec la propriété d'être fortement normalisant.

Par contre, si $a =_\beta b$ et b a une NF (resp HNF), alors a a une NF (resp HNF)
De plus :

- Si e a une NF (resp HNF), $\lambda x.e$ a une NF (resp HNF)
- Si e est SN, $\lambda x.e$ est SN

Soit $\mathcal{N}$ l'ensemble des termes SN, et $\mathcal{N}_0 \equiv \{x\overline{u} \mid x\overline{u} \in \mathcal{N}\} \subseteq \mathcal{N}$

Notation

- $e \in \Lambda$, $Succ(e) = \{e' \in \Lambda \mid e\beta_0 e'\}$, et cet ensemble est *fini* (réduction à branchements fini)
- Lemme de Koenig : si un arbre est infini et que cet arbre est a branchement fini, alors il existe un chemin infini

Si $e \in \mathcal{N}$, $\bigcup_{p \geq 0} Succ^p(e)$ est *fini*, de sorte que la définition suivante est bien fondée :

Definition 15. *Pour $e \in \mathcal{N}$, $\ell(e)$ désigne la somme des longueurs des chemins de tout réduction issue de e.*

Lemma 3. *Sont immédiats :*

- *Si $e \in \mathcal{N}$, alors $\lambda x.e \in \mathcal{N}$*
- *Si de plus $e' \in \mathcal{N}$ et $e\beta e'$, alors $e' \in \mathcal{N}$*
- *Si $e \in \Lambda$ tel que $Succ(e) \subseteq \mathcal{N}$, alors $e \in \mathcal{N}$*

Proof. Pour le troisième point, soit $e \in \Lambda$ tel que $Succ(e) \subseteq \mathcal{N}$, pour tout $e' \in Succ(e)$, $\ell(e') < \ell(e) \to$ une récurrence simple sur $\ell(e)$ permet d'établir $\mathcal{P}(e) \equiv$ "$Succ(e) \subseteq \mathcal{N}$ implique $e \in \mathcal{N}$" □

□

Part II
Polymorphisme

4 Abstraction des types

Programmation	Théorie des types	Raisonnement
λ-calcul	$\lambda_{\rightarrow}$	NJ($\Rightarrow$)
λ-calcul enrichie	$\lambda_{\rightarrow,\times,\top\bot,...}$	NJ
	λ_{μ} ($\sim$ 1990)	NK
Calcul de combinateurs (S,K,I)		Système de Hilbert (1900)
*	Système F	//
//	++	//

Figure 2: Correspondance programmation - langage de preuve

4.1 Motivations

Prenons quelques exemples:

1. L'identité :
   Dans $\lambda_{\rightarrow}$, $\vdash \lambda x.x : T \rightarrow T$ pour n'importe quel type T.
   On voudrait donner à $\lambda x.x$ un type "polymorphe"

2. Un entier de Church :
   $\bar{z} = \lambda x.\lambda f.f(fx)$. Dans $\lambda_{\rightarrow}$, $\vdash \lambda x.\lambda f.f(fx) : A \rightarrow (A \rightarrow A) \rightarrow A$
   $x : A, f : B \vdash f \underbrace{(f\ x)}_{:V} : C$

   (a) $f : b \equiv U \rightarrow V$ et $x : U$ donc $A \equiv U$
   (b) $B \equiv V \rightarrow W$ et $W \equiv C$ et $U \rightarrow V \equiv V \rightarrow W$ d'où $U \equiv V$ et $V \equiv W$
   (c) $\Delta \equiv \lambda x.x\ x$ qui nécessite $W = W \rightarrow V$.

Observations

⋆ On veut donner à un terme "plusieurs types" "d'un coup "

En C $\rightarrow$ template
En O'Caml $\rightarrow$ `fun x → x` : $'\alpha \rightarrow \alpha$

⋆ On peut avoir besoin de gérer plusieurs occurences d'une même variable

⋆ Observation de J.Reunolds ('74)

⋆ Introduit par JY Girard ('70) (cadre logique)

4.2 Le système F à la Church

Que veut-on ?

- ⋆ Une notion de généralisation (d'abstraction) sur les types

$$T ::= X \in \mathcal{V} \mid T \to T \mid \forall X.T$$

- ⋆ Une notion *d'instanciation* de type

$$\forall X.T \rightsquigarrow T\langle S/X\rangle$$

La conséquence sur les termes :

$$t ::= x \in \mathcal{X} \mid \lambda x^T.t \mid t\ t \mid \Lambda X.t \mid t\ T$$

(Hyp), $(\to_i)$[4] et $(\to_E)$ de $\lambda_\to$ plus :

$$\frac{\Delta \vdash t : T \qquad X \notin FT(\delta)}{\Delta \vdash \Lambda X.t : \forall X.T}\ (\forall_i) \qquad\qquad \frac{\Delta \vdash t : \forall X.T}{\Delta \vdash t\langle S/X\rangle : T\langle S/X\rangle}\ (\forall_e)$$

Retour sur les exemples

1. $\vdash \lambda x.x : X \to X$ devient $\Lambda X.\lambda x^X.x : \forall X\ X \to X$
2. $\bar{z} \equiv \lambda x.\lambda f.f\ (f\ x)$

$$\frac{\vdash \lambda x^X.\lambda f^{X\to X}.f\ (f\ x) : X \to (X \to X) \to X}{\vdash \Lambda X.\lambda x.\lambda f.f\ (f\ x) : \forall X.X \to (X \to X) \to X}$$

3. $\lambda x.x\ x$

   - ⋆ $x\ U : U \to U$
   - ⋆ $x\ V : V \to V$
     avec $V \to V \equiv U$

   On en déduit que $W \equiv V \to V$.

$$\frac{\dfrac{x\ U : U \to U}{x : \forall X.S \vdash x\ U : (V \to V) \to W} \qquad \dfrac{x\ V : V \to V}{x : \forall X.S \vdash x\ V : V \to V}}{\dfrac{x : \forall x : S \vdash x\ U\ (x\ V) : W^{V\to V}}{\vdash \lambda x^{\forall X.S}.x\ U\ (X\ V) : (\forall X.S) \to W}}$$

   On trouve un type $((\forall X.(X \to X) \to \forall X.(X \to X)) \to ((\forall X.(X \to X) \to \forall X.(X \to X))$

4.2.1 Système F à la Curry

- ⋆ λ-term pur
- ⋆ Même types que dans le système à la Church
- ⋆ (Hyp), $(\to_i)$, $(\to_e)$

[4] où $\lambda x^S.t$ précise le type de x

- ⋆ Plus

$$\frac{\Delta \vdash t : T \qquad X \notin FT(\Delta)}{\Delta \vdash t : \forall X.T}\ (\forall_i) \qquad\qquad \frac{\Delta \vdash t : \forall X.T}{\Delta \vdash t : T\langle S/X\rangle}\ (\forall_e)$$

4.2.2 Aspects dynamiques

a. A la Church :

- ⋆ $(\lambda x^T.a)\ b \to_\beta a\langle b/x\rangle$
- ⋆ $(\Lambda X.t)\ S \to_\beta t\langle S/X\rangle$

b. A la Cury :