

Proofs and Programs

Phillipe Audebaud *

ENS de Lyon

Contents

I	(Pure) λ-Calculus	3
0.1	Computing with functions ?	3
0.2	Church λ -calculus (informally)	3
1	A toolbox on λ-calculus	4
2	Calcul propositionnel et correspondance de Curry-Howard	7
2.1	Éléments de langage (informels)	7
2.2	Fragments λ_{\rightarrow}	8
2.3	Interprétation BHK	8
3	λ-calcul simplement typé	9
3.1	Quelques Lemmes	9
3.2	Parties saturées de Λ	11
3.3	Normalisation pour λ_{\rightarrow}	12
II	Polymorphisme	13
4	Abstraction des types	13
4.1	Motivations	13
4.2	Le système F à la Church	14
4.2.1	Système F à la Curry	14
4.2.2	Aspects dynamiques	15
5	Propriétés méta du système F	15
5.1	Propriétés de F-Church	15
5.2	Propriétés de F-Curry	16
5.2.1	Système alternatif	16
III	Égalité	18
0	Usage des inductifs dans Coq	18
0.1	Les booléens	18
0.2	Les entiers de Peano	18
0.3	Le produit $A \times B$	18

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

1	Types dépendants	19
2	Egalité (prélude)	19
3	Question d'égalité	20
4	Mise en oeuvre de $Id_A(-, -)$	20
4.1	20
4.2	Transport de l'égalité	20

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 \rightarrow 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 ?

- Having *structures*: numbers, spaces (points, vectors, functions) \rightarrow Eilenberg-Mac Lane (\sim 1942) Category theory
- Build, explore, transform structures \rightarrow Church (\sim 1930) λ -Calculus
- Compare "stuff": *equality* \rightarrow Voevodski (\sim 2006) Algebraic topology \rightarrow search HoTT (Hight order Type Theory)
- Provide a framework (*rules*) to reasoning on all that! \rightarrow 1st point

0.2 Church λ -calculus (informally)

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

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

$$\begin{aligned} f &\stackrel{\text{def}}{=} \lambda x. e && (\lambda\text{-abstraction}) \\ f\ a &= (\lambda a. e)\ a && (\text{Application}) \end{aligned}$$

Notation

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

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

Example

1.

$$\begin{aligned} \lambda x.x \\ x \mapsto x \end{aligned}$$

is the identity function

2.

$$\begin{aligned} \lambda x.y \\ x \mapsto y \end{aligned}$$

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

$$(\lambda x.a) b \rightarrow_{\beta} a\langle b/x \rangle$$

\rightarrow_{β} is a binary relation on lambda-terms \Rightarrow idea of computation on terms.

Notion of α -equivalence

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

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, \dots

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

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

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_β reduction).

$$\rightarrow_\beta \subseteq \Lambda \times \Lambda$$

$$\left\{ \left(\underbrace{(\lambda x.a) b}_{\text{redex}}, \underbrace{a\langle b/x \rangle}_{\text{contraction}} \right) \mid x \in \mathcal{X}, a, v \in \Lambda \right\}$$

Example

1.

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

$$\equiv (\lambda y.y) a \langle b/x \rangle$$

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!

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

$$\rightarrow_\beta^* \text{ is the } \beta\text{-reduction, noted } \rightarrow_\beta$$

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 $\lambda x.a \mathcal{R} \lambda x.b$

Propriété 1. La β -réduction est la plus petite relation λ -compatible et transitive contenant \rightarrow_β

Proof. ★ On vérifie d'abord :

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

D'autre part, β_0 est λ -compatible :

- par réflexivité, $x \beta_0 x$
- soit $a \beta_0 b$; 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, $\lambda x.a \beta_0 \lambda x.b$.

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

Propriétés essentielles de la β -réduction

Remarque (Λ, β_0) est un système de réduction abstrait¹.

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. \square

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

¹Cf 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 $R \subseteq \Lambda \times \Lambda$ défini par :
 $a R b$ 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_β

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:

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

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

² $\neg A$ signifie en fait $A \Rightarrow \perp$

2.2 Fragments λ_{\rightarrow}

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

Theorem 4 (Curry-Howard). *Le fragment NJ_{\rightarrow} et λ_{\rightarrow} sont en correspondance via:*

1. *Si $\Delta \vdash t : T$ dans λ_{\rightarrow}*

λ_{\rightarrow}	NJ_{\rightarrow}
<i>type</i>	<i>proposition</i>
variable de type type flèche	proposition atomique implication
<i>terme</i>	<i>dérivation</i>
variable de terme λ -abstraction application β -redex β -réduction	hypothèse règle d'introduction règle d'élimination coupure transformation sur les dérivations
<i>forme normale</i>	<i>dérivation sans coupure</i>

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 \perp
- 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 $A \Rightarrow \perp$

Avec A, B engendrés par la grammaire

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

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 $f : D \rightarrow A$ et $g : D \rightarrow B$, il existe $h : D \rightarrow A \times B$ 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 ,

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

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

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

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

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

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 donnée de $A + B$ avec la propriété universelle suivante :*

Si $f : A \rightarrow C$, et $g : B \rightarrow C$, il existe $k : A + B \rightarrow C$ 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

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

Alors

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

Donc

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

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

3 λ -calcul simplement typé

3.1 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 $\Delta \equiv x_1 : S, \dots, x_p : S_p$ où $p \geq 0$ est valide si les variables x_1, \dots, x_p sont distinctes deux à deux.

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

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

Et on augmente (Hyp).

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

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

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

□

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

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

1. *Définition générale* : Si $e \in \Lambda$, $e \equiv \lambda \bar{x}. \Delta \bar{u}$ avec $|\bar{x}| > 0$, $|\bar{u}| > 0$ 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) \rightarrow *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 $\mathcal{N}_0 \equiv \{x\bar{u} \mid x\bar{u} \in \mathcal{N}\} \subseteq \mathcal{N}$

Notation

- $e \in \Lambda$, $\text{Succ}(e) = \{e' \in \Lambda \mid e \beta_0 e'\}$, 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 $e \in \mathcal{N}$, $\bigcup_{p \geq 0} \text{Succ}^p(e)$ 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 $\text{Succ}(e) \subseteq \mathcal{N}$, alors $e \in \mathcal{N}$

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

□

Soit $\mathcal{N}_0 \equiv \{ \underbrace{x \bar{u}}_{(((x \ u_1) \ u_2) \dots) \ u_n} \mid x \bar{u} \in \mathcal{N} \subseteq \mathcal{N} \}$

Lemma 4. i) Si $e \in \mathcal{N}$, alors $\lambda x.e \in \mathcal{N}$

ii) Si $e \in \mathcal{N}$, et $e \beta' e'$, alors $e' \in \mathcal{N}$

iii) Si $e \in \Lambda$ tel que $\text{Succ}(e) \subseteq \mathcal{N}$, alors $e \in \mathcal{N}$

iv) Si $e \in \Lambda$ et $x \in \mathcal{V}$, $e x \in \mathcal{N}$ implique $e \in \mathcal{N}$

Lemma 5. Si $b \in \mathcal{N}$ et $a\langle b/x \rangle \bar{u} \in \mathcal{N}$

Proof. Par récurrence sur $l(b) + l(a\langle b/x \rangle \bar{u})$ Il suffit de montrer que $\text{Succ}((\lambda x.a \ b \ \bar{u}) \subseteq \mathcal{N}$
 \downarrow
 β_0

□

3.2 Parties saturées de Λ

Definition 16. Soit $S \subseteq \Lambda$. On dit que S est saturée si elle satisfait les conditions suivantes:

$$\mathcal{N}_0 \subseteq S \subseteq \mathcal{N} \quad (\text{Sat 1})$$

$$Si \ e \in S \text{ et } e \beta_0^* e' \quad (\text{Sat 2})$$

$$Si \ e \in \Lambda \text{ et } e \text{ n'est pas une } \lambda\text{-abstraction, et si } \text{Succ}(e) \subseteq S, e \in S \quad (\text{Sat 3})$$

Propriety 2. i) \mathcal{N}_0 est saturée

ii) \mathcal{N} est saturée

iii) Si X, Y sont des parties saturées, alors $X \rightarrow Y = \{e \in \Lambda \mid \forall a \in X, e \ a \in Y\}$

Proof. i) En exercice

ii) Il suffit de vérifier (Sat 3) En clair, soit $e \in \Lambda$ qui n'est pas une λ -abstraction, et tel que $\text{Succ}(e) \in \mathcal{N}$.
On procède par induction structurelle sur e .

* $e \in \mathcal{X}$ trivial

* $e \equiv e_0 \ a$, avec $\text{Succ}(e_0 \ a) \in \mathcal{N}$

$$\left[\begin{array}{l} e \beta_0 e'_0 \ a, \text{ avec } e_0 \beta_0 e'_0 \\ e \beta_0 e_0 \ a', \text{ avec } a \beta_0 a' \\ e \beta_0 e_1 \langle a/x \rangle, \text{ si } e_0 \equiv \lambda x.e_1 \end{array} \right] \begin{array}{l} e'_0 \ a \in \mathcal{N} \\ e_0 \ a' \in \mathcal{N} \\ e_1 \langle a/x \rangle \in \mathcal{N} \end{array}$$

iii) $X, Y \in \text{Sat}(\Lambda)$, montrons que $X \rightarrow Y \in \text{Sat}(\Lambda)$

(Sat 1)

* $\mathcal{N}_0 \subseteq X \rightarrow Y$?

Si $x \bar{u} \in \mathcal{N}$, et $a \in X \subseteq \mathcal{N}$

* $X \rightarrow Y \subseteq \mathcal{N}$

Si $e \in X \rightarrow Y$, c'est-à-dire pour tout $a \in X$, $e \ a \in Y \subseteq \mathcal{N}$

(Sat 2) facile : $e \in X \rightarrow Y$ et $e \beta_0 e'$

Si $a \in X$, $(e \ a) \beta_0 (e' \ a)$ donc $e' \ a \in \text{Succ}(e \ a)$. Par hypothèse, $e \in X \rightarrow Y$, c'est-à-dire $e \ a \in Y$. Par (Sat 2) appliqué à Y , $e' \ a \in Y$.

(Sat 3) Soit e qui n'est pas un λ -abstraction et tel que $\text{Succ}(e) \subseteq X \rightarrow Y$; montrons que $e \in X \rightarrow Y$. Cela revient à établir que pour tout $a \in X$, $e \ a \in Y$. On montre ça en appliquant (Sat 3) à Y , car $e \ a$ n'est pas un λ -abstraction.

Il suffit de vérifier que $Succ(e\ a) \subseteq Y$:

$$\begin{aligned} e\ a\ \beta_0\ e'\ a & \text{ avec } e\ \beta_0\ e' \\ e\ a\ \beta_0\ e'\ a & \text{ avec } a\ \beta_0\ a' \end{aligned}$$

Et c'est tout ! On remarque qu'il suffit de faire une démonstration par récurrence sur $\ell(a) \rightarrow$ à faire ! \square

3.3 Normalisation pour λ_{\rightarrow}

Theorem 6 (SN). *Si $\Delta \vdash_{\lambda_{\rightarrow}} e : T$, alors $e \in \mathcal{N}$*

Proof. * Une “interprétation” des types comme parties saturées
 * Un lemme d’“étiquetage”
 * Le théorème apparaît comme corollaire. \square

Definition 17. *Soit $\rho \in \mathcal{V} \rightarrow Sat(\Lambda)$. On définit par induction structurelle l'interprétation d'un type T selon ρ notée $[T]_{\rho}$:*

- * Si $T \in \mathcal{V}$, $[T]_{\rho} \equiv \rho(T)$
- * Si $T \equiv U \rightarrow V$, $[T]_{\rho} \equiv [U]_{\rho} \rightarrow [V]_{\rho}$

Proof. D'après la proposition précédente :

- ρ existe
 - $[T]_{\rho} \in Suc(\Lambda)$
- \square

Remarque $[T]_{\rho}$ ne dépend que de $\rho \upharpoonright FT(T)$

Definition 18. $x_1, \dots, x_n \in \mathcal{X}$ et $t_1, \dots, t_n \in \Lambda$. La substitution $\sigma \equiv \langle t1/x_1, \dots, t_n/x_n \rangle$ donne lieu à $\sigma(t)$ où $t \in \Lambda$:

- Si $t \equiv x_i \in \{x_1, \dots, x_n\}$, alors $\sigma(t) = t_i$
- Si $t \in \mathcal{X} \setminus \{x_1, \dots, x_n\}$, $\sigma(t) \equiv t$
- Si $t \equiv \lambda x. t_0$, on peut supposer que $x \notin \{x_1, \dots, x_n\} \cup \{\cup_{i \leq n} FV(t_i)\}$ et $\sigma(t) \equiv \lambda x. \sigma(t_0)$
- Si $t \equiv a\ b$, $\sigma(t) = \sigma(a)\ \sigma(b)$

Propriety 3. *Soit ρ une interprétation des types et $\Delta \vdash_{\lambda_{\rightarrow}} t : T$*

Pour tout substitution σ de domaine $\subseteq FV(\Delta)$ telle que pour tout $x : S \in \Delta$, $\sigma(x) \in [S]_{\rho}$, on a :

$$\sigma(t) \in [T]_{\rho}$$

Proof. Par induction sur la hauteur de la dérivation, le seul cas intéressant est \rightarrow_i

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

Où $t \equiv \lambda x. a$.

Soit σ de domaine $\subseteq FV(\Delta)$; montrons que si $\forall x : S \in \Delta, \sigma(x) \in [S]_{\rho}$, alors $\sigma(\lambda x. a) \in [U \rightarrow V]_{\rho} = [U]_{\rho} \rightarrow [V]_{\rho}$. Par définition, cela revient à montrer que pour tout $b \in [U]_{\rho}$, $\sigma(\lambda x. a)\ b \in [V]_{\rho}$. On peut faire en sorte que

$$x \notin dom(\Delta) \cup \left\{ \bigcup_{y:S \in \Delta} FV(\sigma(y)) \right\} \cup FV(b)$$

Du coup, $\sigma' \equiv \langle t_1/x_1, \dots, t_n/x_n, b/x \rangle$ où $\sigma \equiv \langle t_1/x_1, \dots, t_n/x_n \rangle$ avec $\{x_1, \dots, x_n\} \subseteq \text{dom}(\Delta)$
 Et σ' satisfait :

$$\sigma(\lambda x.a) b \equiv \sigma'(a)$$

σ' satisfait les hypothèses relatives à $\Delta, x : U$ car $\sigma'(x) = b \in [U]_\rho$. Du coup, par hypothèse d'induction,

$$\sigma'(a) \in [V]_\rho$$

Finalement,

$$(\lambda x.\sigma(a) b \beta_0 \sigma'(a) \in [V]_\rho$$

En résumé, $\lambda x.\sigma(a) \equiv \sigma(\lambda x.a) \in [U \rightarrow V]_\rho$ □

Part II

Polymorphisme

4 Abstraction des types

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

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 $\lambda_{\rightarrow}, \vdash \lambda x.\lambda f.f(fx) : A \rightarrow (A \rightarrow A) \rightarrow A$

$x : A, f : B \vdash \underbrace{f(fx)}_{:V} : C$

(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 $x \rightarrow x : ' \alpha \rightarrow \alpha$

- ★ On peut avoir besoin de gérer plusieurs occurrences 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

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

- ★ 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), $(\rightarrow_i)^4$ et (\rightarrow_E) de λ_{\rightarrow} plus :

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

Retour sur les exemples

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

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

3. $\lambda x.x x$

- ★ $x U : U \rightarrow U$
- ★ $x V : V \rightarrow V$
- avec $V \rightarrow V \equiv U$

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

$$\frac{\frac{x U : U \rightarrow U}{x : \forall X.S \vdash x U : (V \rightarrow V) \rightarrow W} \quad \frac{x V : V \rightarrow V}{x : \forall X.S \vdash x V : V \rightarrow V}}{\frac{x : \forall x : S \vdash x U (x V) : W^{V \rightarrow V}}{\vdash \lambda x^{\forall X.S}.x U (X V) : (\forall X.S) \rightarrow W}}$$

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

4.2.1 Système F à la Curry

- ★ λ -term pur
- ★ Même types que dans le système à la Church
- ★ (Hyp), (\rightarrow_i) , (\rightarrow_e)

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

★ Plus

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

4.2.2 Aspects dynamiques

a. A la Church :

$$\star (\lambda x^T.a) b \rightarrow_\beta a\langle b/x \rangle$$

$$\star (\Lambda X.t) S \rightarrow_\beta t\langle S/X \rangle$$

b. A la Curry :

5 Propriétés méta du système F

	F-Church	F-Cury
Confluence (Church-Rosser)	×	Déjà fait
SR (Subject Reduction)	Facile	Plus dure
SN (Strong Normalisation)	En TD	A partir de λ_{\rightarrow}

Figure 3: Propriétés de confluence de systèmes à la Curry et à la Church

Lemma 6 (Dans les deux versions du système F). *i) Si $\Delta \vdash t : T$ et Δ' contexte de typage tel que $\Delta \upharpoonright_{FV(t)} = \Delta' \upharpoonright_{FV(t)}$, alors $\Delta' \vdash t : T$*

ii) Si $\Delta \vdash t : T$, alors pour tout S et pour tout $X \in \mathcal{V}$, $\Delta\langle S/X \rangle \vdash \underbrace{t\langle S/X \rangle}_t : T\langle S/X \rangle$

iii) Si $\Delta, x : S \vdash t : T$ et $\Delta \vdash s : S$ alors $\Delta \vdash t\langle s/x \rangle : T$

5.1 Propriétés de F-Church

Propriety 4 (Subject reduction). *Si $\Delta \vdash_{Church} t : T$ et $t \rightarrow t'$ alors $\Delta \vdash t' : T$*

Proof. En exercice.

Pour rappel, \rightarrow est l'analogue de β_0 :

$$\rightarrow_\beta \subseteq \rightarrow$$

également dans un terme. □

Propriety 5 (Strong Normalization). *Si $\Delta \vdash_{Church} t : T$, alors t est fortement normalisant.*

Proof. Fiche d'exercice (semaine 7), basée sur le résultat pour λ_{\rightarrow} □

Propriety 6 (Confluence faible). *Si $\Delta \vdash_{Church} t : T$ et si $t \rightarrow t_1$, $t \rightarrow t_2$, il existe t_3 tel que $t_1 \rightarrow^* t_3$ et $t_2 \rightarrow^* t_3$.*

Corollary 2 (Confluence). *La relation \rightarrow est confluente sur les termes bien typés.*

Proof. On s'appuie sur SN et la confluence faible. □

5.2 Propriétés de F-Curry

On veut établir le résultat suivant:

Propriety 7. Si $\Delta \vdash_{Curry} t : T$ et $t \xrightarrow[\beta_0]{\rightarrow} t'$, alors $\Delta \vdash t' : T$.

Le résultat est vrai, mais sa démonstration nécessite des détours...

5.2.1 Système alternatif

On introduit la notion de *séquence de types*.

$$\Delta \rightarrow_n T_0, \dots, T_n \text{ seq}$$

avec les règles suivantes :

$$\begin{array}{c} \text{(Ax)} \frac{}{\Delta \vdash_0 T \text{ seq}} \qquad \qquad \qquad (\rightarrow_i) \frac{\Delta, x : S \vdash_n t : T}{\Delta \vdash_0 \lambda x. t : S \rightarrow T} \\[10pt] \text{(Gen)} \frac{\Delta \rightarrow_n T_0, \dots, T_n \text{ seq} \quad X \notin FT(\Delta)}{\Delta \vdash_{n+1} T_0, \dots, T_n, \forall X. T_n \text{ seq}} \qquad \qquad \qquad (\rightarrow_e) \frac{\Delta \vdash_p e : S \rightarrow T \quad \Delta \vdash_q s : S}{\Delta \vdash_0 e s : T} \\[10pt] \text{(Inst)} \frac{\Delta \rightarrow_n T_0, \dots, T_n \text{ seq} \quad T_n \equiv \forall X. T}{\Delta \vdash_{n+1} T_0, \dots, T_n, T\langle S/X \rangle \text{ seq}} \qquad \qquad \qquad \text{(Sub)} \frac{\Delta \vdash_0 t : T \quad \Delta \vdash_n T_0, \dots, T_n \text{ seq}}{\Delta \vdash_n t : T_n} \\[10pt] \text{(Hyp)} \frac{x : T \in \Delta}{\Delta \vdash_0 x : T} \end{array}$$

Propriety 8. $\Delta \vdash_{Curry} t : T$ si et seulement si il existe $n \geq 0$ tel que $\Delta \vdash_n t : T$

Proof. Dans chacun des sens, par induction. □

Exo Si $\Delta \vdash_p t : T$ et $\Delta \vdash_p T_0, \dots, T_q \text{ seq}$ avec $T_0 \equiv T$, alors

$$\Delta \rightarrow_{p+q-1} t : T_q$$

Exo Si $\Delta \rightarrow_n T_0, \dots, T_n \text{ seq}$ avec $T_0 \equiv U \rightarrow V$ et $T_n \equiv A \rightarrow B$ alors il existe $\bar{X} \subseteq \mathcal{V}$ et \bar{S} de même longueur tel que

$$A \equiv U\langle \bar{S}/\bar{X} \rangle, B \equiv V\langle \bar{S}/\bar{X} \rangle$$

Lemma 7. i) Si $\Delta \vdash_n t : T$, alors $\Delta\langle S/X \rangle \vdash_n t : TSX$

ii) Si $\Delta, x : S \vdash_n t : T$ et $\Delta \rightarrow_r s : S$ alors il existe $n' \geq 0$ tel que

$$\Delta \vdash_{n'} t\langle s/x \rangle : T$$

Proof. i) facile

ii) Considérons le cas où t est une variable. Soit:

$$\overline{\Delta, x : S \vdash_0 t : T}$$

- Si $t \equiv x$, $S \equiv T$
Par ailleurs, $\Delta \vdash_r s : T$, du coup $\Delta \vdash_r t\langle s/x \rangle : T$
- Si $t \neq x$, $t : T \in \Delta$ et

$$(\text{Hyp}) \frac{x : T \in \Delta}{\Delta \vdash_0 x : T}$$

Soit $n \geq 0$:

$$\frac{\Delta \vdash_0 t : T_0 \quad \Delta \vdash_n T_0, \dots, T_n \text{ seq}}{\Delta \vdash_n t : T_n}$$

Par hypothèse d'induction, il existe $n'' \geq 0$ tel que $\Delta \rightarrow_{n''} t : T_0$. On en conclut $\Delta \vdash_{n''+n-1} t : T_n$ \square

\rightarrow La démonstration “séquence” est différente !

Lemma 8. *Si $\Delta \vdash_n (\lambda x.a) b : T$, alors il existe $n, \geq 0$ tel que $\Delta \vdash_{n'} a \langle b/x \rangle : T$*

Proof. On se limite ici au cas où $n = 0$:

$$\frac{\Delta \vdash_p \lambda x.a : S \rightarrow T \quad \Delta \vdash_q b : S}{\Delta \vdash_0 (\lambda x.a) b : T}$$

$$\frac{\frac{\square}{\Delta, x : U \vdash_r a : V} \quad \Delta \vdash_p F_0, \dots, F_p \text{ seq}}{\Delta \vdash_p \lambda x.a : S \rightarrow T}$$

Avec $F_0 \equiv U \rightarrow V$ et $F_p \equiv S \rightarrow T$, donc il existe \bar{X}, \bar{S} tels que $S \equiv U \langle \bar{S}/\bar{X} \rangle$ et $T \equiv \langle \bar{S}/\bar{X} \rangle$ et $\bar{X} \cap FT(\Delta) = \emptyset$.

Par ailleurs,

$$\frac{\Delta \vdash_o b : S \quad \Delta \vdash_q S_0, \dots, S_q \text{ seq}}{\Delta \vdash_q b : S \quad (S_q \equiv S)}$$

On en déduit :

* $\Delta \langle \bar{S}/\bar{X} \rangle, x : U \langle \bar{S}/\bar{X} \rangle \vdash_r a : V \langle \bar{S}/\bar{X} \rangle$
 mais, puisque $\bar{X} \cap FV(\Delta) = \emptyset$

$$\Delta, x : S \vdash_r a : T$$

* Finalement,

$$\left\{ \begin{array}{l} \Delta, x : S \vdash_r a : T \\ \Delta \vdash_q b : S \end{array} \right\} \Rightarrow \Delta \vdash_{r'} a \langle b/x \rangle : T$$

Pour un certain $r' \geq 0$ \square

Propriety 9 (SR). *Si $\Delta \vdash_n t : T$ et si $t \beta_0 t'$, alors il existe $n' \geq 0$ tel que $\Delta \vdash_{n'} t' : T$*

Proof. On a vu le cas de base:

* $\rightarrow_\beta \subseteq \beta_0$

* On regarde les autres cas qui définissent β_0 \square

Theorem 7 (SR). *Si $\Delta \vdash_{\text{Curry}} t : T$ et si $t \beta_0 t'$, alors $\Delta \vdash_{\text{Curry}} t' : T$*

Remarque Dans (SU)⁵, on utilise $\sigma \preceq \tau$ avec σ, τ types.

Part III

Égalité

0 Usage des inductifs dans Coq

0.1 Les booléens

bool $\equiv \{\mathbf{true}, \mathbf{false}\}$

En Coq,

Inductive **bool** : Type :=

| **true**

| **false**

(★)

correspond à la déclaration de **true** et **false** comme éléments de type **bool**.

Ce qui “correspond” à l’énoncé d’un principe d’induction sur **bool** :

Pour tout $P : \mathbf{bool} \rightarrow \mathbf{True}$, si $\begin{cases} (P \mathbf{true}) \text{ est habitable} \\ (P \mathbf{false}) \text{ est habitable} \end{cases}$, alors pour tout $b \mathbf{bool}$, $P b$ est habitable.

Dans Coq, (★) engendre:

- un nouveau type **bool**
- Les constructeurs **true** et **false**
- Le principe d’induction, de type $\Pi P : \mathbf{bool} \rightarrow \mathbf{True}$

0.2 Les entiers de Peano

nat : Type

0 : nat

S : nat \rightarrow nat

Inductive **nat** : Type :=

| 0 : nat

| S : nat \rightarrow nat

Ce qui correspond à l’introduction d’un entier, et au principe de récurrence “standard”.

0.3 Le produit $A \times B$

Inductive **prod** (A B : Type) : Type :=

pair : A \rightarrow B \rightarrow **prod** A B

Inductive **sum** (A B : Type) : Type :=

| **inl** : A \rightarrow **sum** A B

| **inr** : B \rightarrow **sum** A B

⁵Bouquin de référence

1 Types dépendants

$\rightarrow (\Pi x : A) B(x)$ “ $\forall x \in A. B(x)$ ”
 $\rightarrow (\Sigma x. A) B(X)$ “ $\exists x \in A. B(x)$ ”

Terminologie

- On a des constantes d'univers : $Type_i, i \in \mathbb{N}$, noté “ \mathcal{U} ” \rightarrow Ce sont des types
- Un objet T est un *type* s'il existe un univers \mathcal{U} tel que $T : \mathcal{U}$
- Un objet t est un *terme* s'il existe un type T tel que T (en particulier, on a toujours $Type_i : Type_{i+1}$, donc $\forall i \in \mathbb{N}, Type_i : \mathcal{U}$)

2 Egalité (prélude)

Dans Coq :

★ égalité de Leibniz.

$a, b : A$

“ $a = b$ ” si pour tout $P : A \rightarrow Type$, $P a \rightarrow P b$.

$$a = b \equiv (\Pi A : A \rightarrow W) \underbrace{P a}_{\mathcal{U}} \rightarrow \underbrace{P b}_{\mathcal{U}} : \mathcal{U}$$

★ Type identité (Martin-Löf)

$A : \mathcal{U}, \quad a, b : A \quad Id_A(a, b) : \mathcal{U}$

Peut être présenté à l'aide des *inductifs* de Coq :

Inductive {Id A : Type} (x, y : A) :=

| Id_reflexive : forall a : A, Id a a.

(** Ou @Id A a a pour forcer le premier argument**)

Qui fournit (entres autres):

- Le constructeur *Id_reflexive* \rightsquigarrow règle d'introduction

$$\frac{A : Type \quad a : A}{Id_reflexive a : Id_A(a, a)}$$

- Un principe de raisonnement inductif \rightsquigarrow règle d'élimination (faire “**Print Id_rect**”)

Pour tout $A : Type$, $C : \prod_{a,b:A} Id_A(a, b) \rightarrow \mathcal{U}$, si pour tout $a : A$, $C(a, a, Id_reflexive a)$ est prouvable, alors pour tout $a, b : A$, pour tout $p : Id_A(a, b)$, $C(a, b, p)$ est prouvable.

$ind_{=A}(C, c, a, b, p) : C(a, b, p)$ avec $p : Id_A(a, b)$ vérifie $ind_{=A}(C, c, x, x, Id_reflexive x) \equiv C x$ (β -règle, en fait i -règle, pour I inductif !)

Remarques Si $a \equiv b$ dans A , alors $a =_A b$ via *Id_reflexive* (i.e. $Id_A(a, b)$ prouvable) ! “ \equiv ” est l'égalité définitionnelle, et “ $=_A (Id_a(-, -))$ ” est l'égalité propositionnelle.

Exemple soit $f : \left\{ \begin{array}{l} \mathbb{N} \rightarrow \mathbb{N} \\ a \mapsto x \star x \end{array} \right\} \lambda x. x \star x$

$$f(3) \equiv (\lambda x. x \star x) 3$$

$$(\lambda x. x \star x) 3 \equiv 3 \star 3$$

Mais on n'a pas toujours $3 \star 3 \equiv 9$! Par contre, $3 \star 3 =_{\mathbb{N}} 9 ? \rightarrow Id_{\mathbb{N}}(3 \star 3, 9) ?$

3 Question d'égalité

En Logique

Si P, Q sont énoncés logiques,

$$P \Leftrightarrow Q \equiv (P \Rightarrow Q) \wedge (Q \Rightarrow P)$$

Relation d'équivalence. Est-ce une notion d'égalité ? \rightarrow pas toujours en logique intuitionniste.

Isomorphisme en math

$$\begin{aligned} A \simeq B \quad & f : A \rightarrow B \quad g : B \rightarrow A \\ g \circ f &= id_A ; f \circ g = id_B \end{aligned}$$

4 Mise en oeuvre de $Id_A(-, -)$

4.1

Lemma 9. $Id_A(-, -)$ est une relation d'équivalence.

Proof. i) Pour tout $a : A$, $Id_A(a, a)$ par application de $Id_reflexive$ "par réflexivité"

ii) Pour tout $a, b : A$, $Id_A(a, b) \rightarrow Id_A(b, a)$

$$\prod_{a, b : A} \prod_{p : Id_A(a, b)} Id_A(b, c)$$

On raisonne par induction sur le chemin $p : Id_A(a, b)$:

Il suffit de considérer le cas de base, c'est-à-dire lorsque b est a , et p est $refl_a$ (alors $Id_reflexive$).

Dans ce cas, on doit prouver

$$Id_A(a, a)$$

Il suffit de fournir $refl_a$.

iii) pour tout $a, b, c : A$, $p : Id_a(a, b)$, $q : Id_A(b, c)$ on a $Id_A(a, c)$

On raisonne par induction sur p :

Le cas de base correspond à b est a et p est $refl_a$. Dans ce cas, il faut prouver $\prod_{q : Id_a(a, c)} Id_a(a, c)$.

On raisonne par induction sur q :

Le cas de base : c est a , et q est $refl_a : refl_a : Id_a(a, a)$

□

On interprète $Id_A(a, b)$ comme un chemin de a à b : on obtient une structure de groupoïde.

Propriety 10. i) $p = p.refl_y$ et $p = refl_x.p$

ii) $p^{-1}.p = refl_x$ et $p.p^{-1} = refl_y$

iii) $(p^{-1})^{-1} = p$

iv) $p.(q.r) = (p.q).r$

v) $(refl_x)^{-1} = refl_x$

4.2 Transport de l'égalité

Lemma 10. $f : A \rightarrow B, x, y : A$ et $p : x =_A y$ ($\text{elim } \text{Id}_A(x, y)$) Alors il existe un terme

$$\text{ap}_f(p) : f(x) =_B f(y)$$

tel que pour tout $x : A$, $\text{ap}_f(\text{refl}_x) \equiv \text{refl}_{f(x)}$

Proof. Par induction sur la classe $p : x =_A y$, il suffit de considérer le cas de base : y est x et p est refl_x .
Alors $\text{refl}_{f(x)} =_B f(x)$ □