Programmes, preuves et fonctions

le ménage à trois de Curry-Howard

Lionel Vaux

Institut de Mathématiques de Marseille
CNRS & Université d'Aix-Marseille

Passage à Marseille des étudiants d'info. de l'ÉNS Paris-Saclay
23 novembre 2017

Première partie

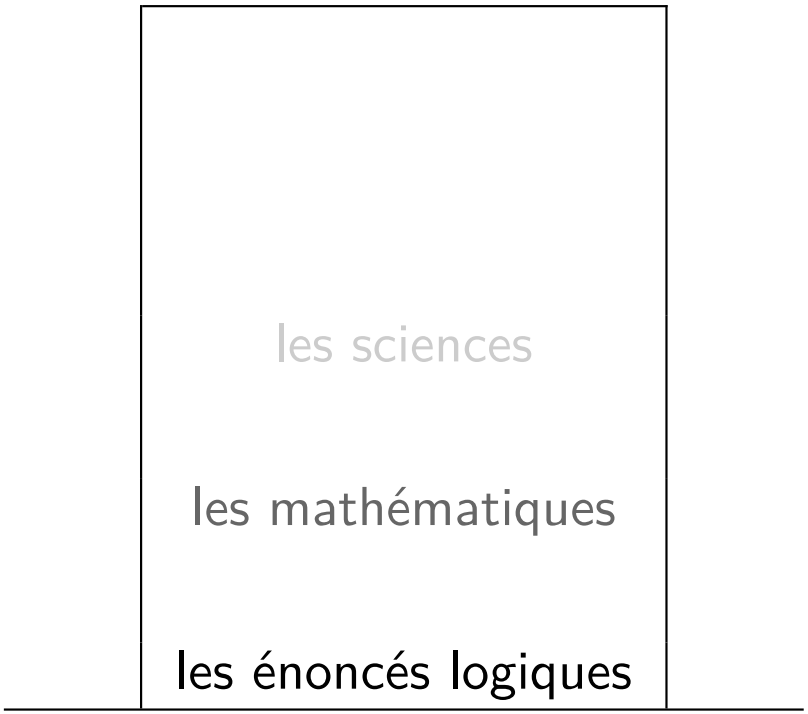
Logique

Valeurs de vérité

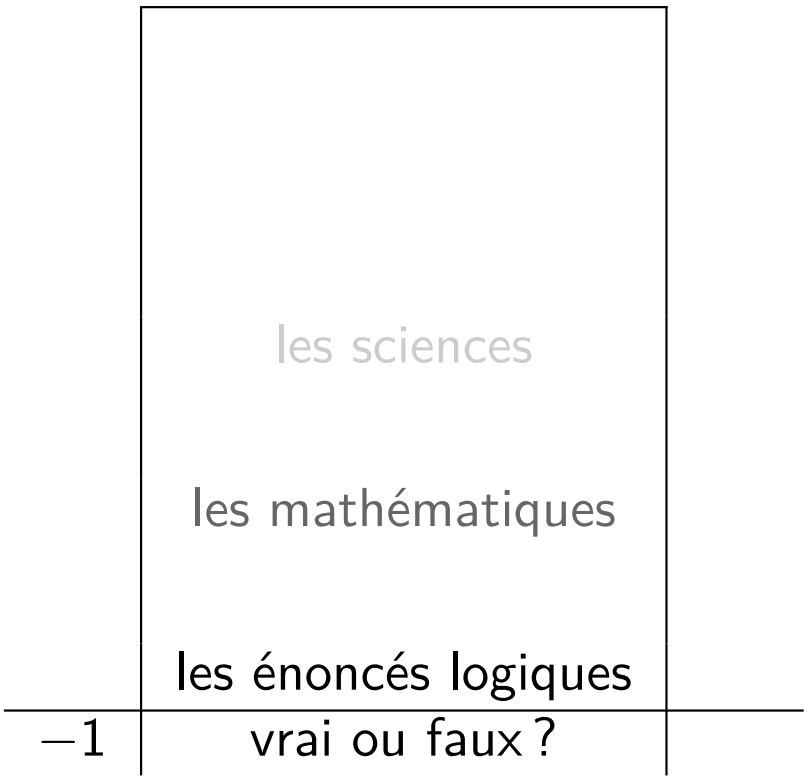
Déduction naturelle

Intuitionnisme

Les sous-sols de la logique

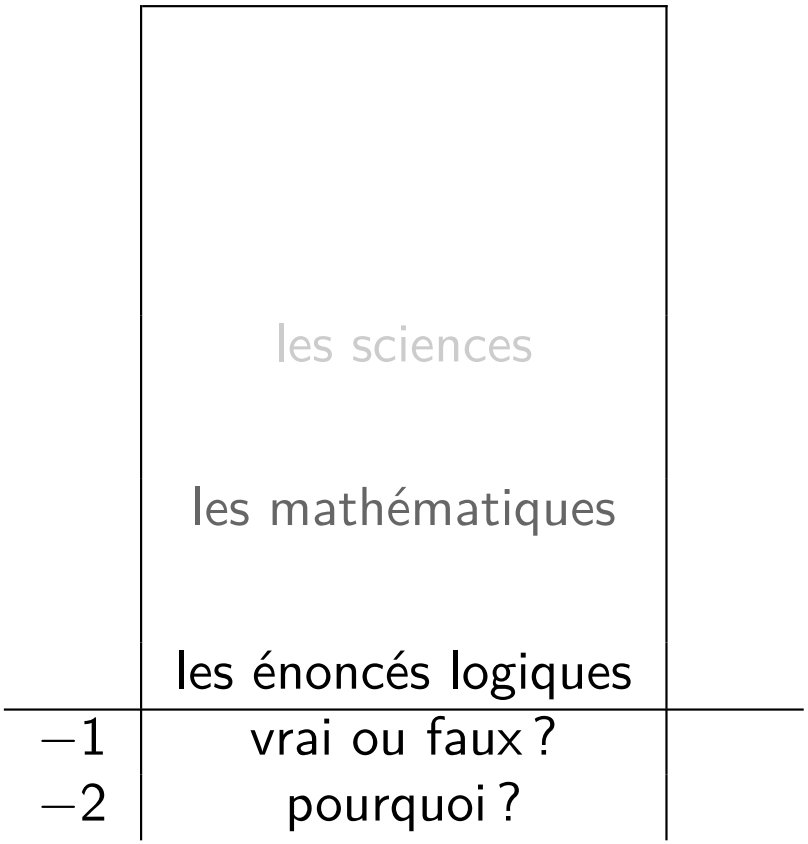


Les sous-sols de la logique



valeurs de vérité, modèles

Les sous-sols de la logique



valeurs de vérité, modèles
démonstrations

Formules logiques

On se donne un langage de formules :

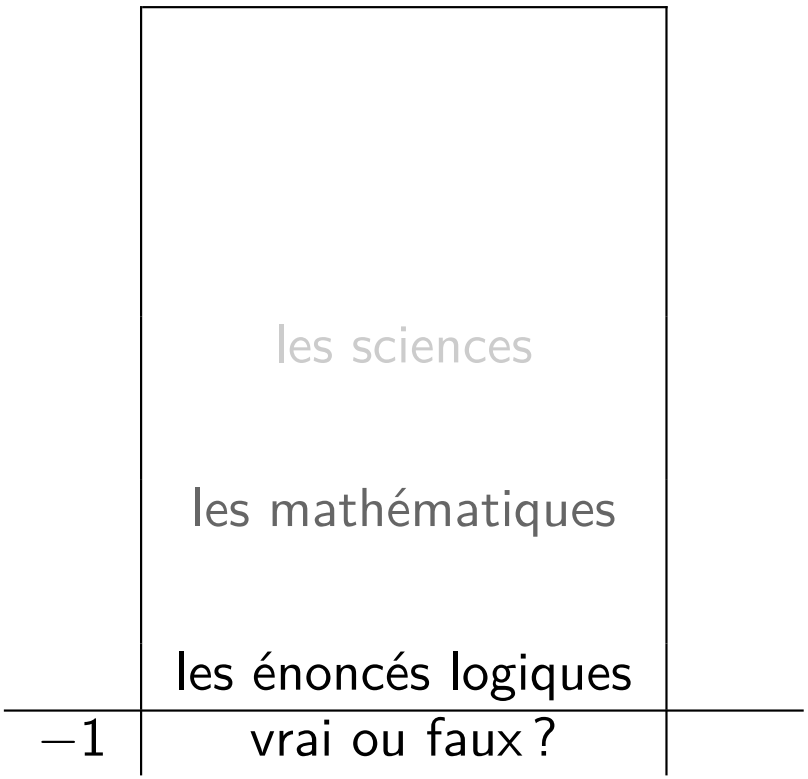
$A, B :=$	α	énoncé atomique
	$A \to B$	implication
	$A \wedge B$	conjonction
	$A \vee B$	disjonction
	$\neg A$	négation
	$A \leftrightarrow B$	équivalence
	$\top$	vrai
	$\bot$	faux
	$\forall x.A$	quantification universelle
	$\exists x.A$	quantification existentielle

Formules logiques

On se donne un langage de formules propositionnelles :

$$
\begin{array}{lll}
A, B := & X & \text{variable propositionnelle} \\
& A \to B & \text{implication} \\
& A \wedge B & \text{conjonction} \\
& A \vee B & \text{disjonction} \\
& \neg A & \text{négation} \\
& A \leftrightarrow B & \text{équivalence} \\
& \top & \text{vrai} \\
& \bot & \text{faux}
\end{array}
$$

Les sous-sols de la logique



valeurs de vérité, modèles

Les tables de la vérité (vraie)

Connecteurs

Les connecteurs sont des fonctions booléennes :

A	B	$A \wedge B$	$A \vee B$	$A \rightarrow B$	$A \leftrightarrow B$	$\neg A$	$\bot$	$\top$
0	0	0	0	1	1	1	0	1
0	1	0	1	1	0	1	0	1
1	0	0	1	0	0	0	0	1
1	1	1	1	1	1	0	0	1

Les tables de la vérité (vraie)

Connecteurs

Les connecteurs sont des fonctions booléennes :

A	B	$A \wedge B$	$A \vee B$	$A \to B$	$A \leftrightarrow B$	$\neg A$	$\bot$	$\top$
0	0	0	0	1	1	1	0	1
0	1	0	1	1	0	1	0	1
1	0	0	1	0	0	0	0	1
1	1	1	1	1	1	0	0	1

Les tables de la vérité (vraie)

Formules

On en déduit les valeurs des formules :

A	B	$A \to \neg B$	
0	0	1	
0	1	1	
1	0	1	
1	1	0	

A	B	$A \wedge B$	$A \vee B$	$A \to B$	$A \leftrightarrow B$	$\neg A$	$\bot$	$\top$
0	0	0	0	1	1	1	0	1
0	1	0	1	1	0	1	0	1
1	0	0	1	0	0	0	0	1
1	1	1	1	1	1	0	0	1

Les tables de la vérité (vraie)

Formules

On en déduit les valeurs des formules :

A	B	$A \to \neg B$	$B \wedge (A \to \neg B)$	
0	0	1	0	
0	1	1	1	
1	0	1	0	
1	1	0	0	

A	B	$A \wedge B$	$A \vee B$	$A \to B$	$A \leftrightarrow B$	$\neg A$	$\bot$	$\top$
0	0	0	0	1	1	1	0	1
0	1	0	1	1	0	1	0	1
1	0	0	1	0	0	0	0	1
1	1	1	1	1	1	0	0	1

Les tables de la vérité (vraie)

Formules, équivalences

On en déduit les valeurs des formules :

A	B	$A \to \neg B$	$B \wedge (A \to \neg B)$	$B \wedge \neg A$	
0	0	1	0	0	
0	1	1	1	1	
1	0	1	0	0	
1	1	0	0	0	

- Deux formules sont *équivalentes* si elles ont les mêmes valeurs.

A	B	$A \wedge B$	$A \vee B$	$A \to B$	$A \leftrightarrow B$	$\neg A$	$\bot$	$\top$
0	0	0	0	1	1	1	0	1
0	1	0	1	1	0	1	0	1
1	0	0	1	0	0	0	0	1
1	1	1	1	1	1	0	0	1

Les tables de la vérité (vraie)

Formules, équivalences, tautologies

On en déduit les valeurs des formules :

A	B	$A \to \neg B$	$B \wedge (A \to \neg B)$	$B \wedge \neg A$	$A \vee (A \to \neg B)$	
0	0	1	0	0	1	
0	1	1	1	1	1	
1	0	1	0	0	1	
1	1	0	0	0	1	

- Deux formules sont *équivalentes* si elles ont les mêmes valeurs.
- Une *tautologie* est une formule vraie (pour toute valeur des sous-formules)

A	B	$A \wedge B$	$A \vee B$	$A \to B$	$A \leftrightarrow B$	$\neg A$	$\bot$	$\top$
0	0	0	0	1	1	1	0	1
0	1	0	1	1	0	1	0	1
1	0	0	1	0	0	0	0	1
1	1	1	1	1	1	0	0	1

Les tables de la vérité (vraie)

Formules, équivalences, tautologies

On en déduit les valeurs des formules :

A	B	$A \to \neg B$	$B \wedge (A \to \neg B)$	$B \wedge \neg A$	$A \vee (A \to \neg B)$	$\cdots$
0	0	1	0	0	1	$\cdots$
0	1	1	1	1	1	$\cdots$
1	0	1	0	0	1	$\cdots$
1	1	0	0	0	1	$\cdots$

- Deux formules sont *équivalentes* si elles ont les mêmes valeurs.
- Une *tautologie* est une formule vraie (pour toute valeur des sous-formules)

A	B	$A \wedge B$	$A \vee B$	$A \to B$	$A \leftrightarrow B$	$\neg A$	$\bot$	$\top$
0	0	0	0	1	1	1	0	1
0	1	0	1	1	0	1	0	1
1	0	0	1	0	0	0	0	1
1	1	1	1	1	1	0	0	1

Codages

On peut coder par exemple :

$$A \leftrightarrow B = (A \wedge B) \vee \neg(A \vee B) \quad A \to B = \neg A \vee B \quad A \vee B = \neg(\neg A \wedge \neg B)$$

ou encore

$$A \leftrightarrow B = (A \to B) \wedge (B \to A) \quad A \wedge B = \neg(A \to \neg B) \quad \neg A = A \to \bot$$

Jeux complets de connecteurs

Par exemple $\{\wedge, \vee, \neg\}$, $\{\wedge, \neg\}$, $\{\to, \neg\}$, $\{\to, \bot\}$, ...

Les tables de la vérité (vraie)

C'est bien, mais pas top

- les valeurs de vérité formalisent l'intuition usuelle
- c'est simple (on se ramène à de l'algèbre sur $\{0, 1\}$)

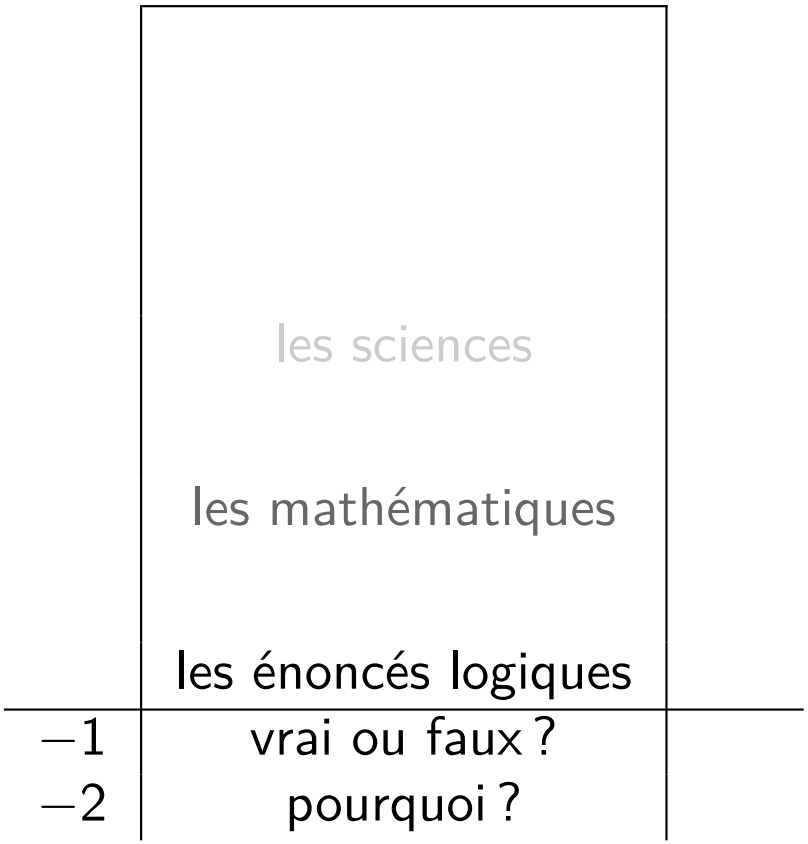
Les tables de la vérité (vraie)

C'est bien, mais pas top

- les valeurs de vérité formalisent l'intuition usuelle
- c'est simple (on se ramène à de l'algèbre sur $\{0, 1\}$)

- on n'apprend rien
- ça n'a rien à voir avec la démonstration mathématique

Les sous-sols de la logique



valeurs de vérité, modèles
démonstrations

Une suite d'affirmations justifiées

1. $2 = 1 + 1$ [calcul]
2. 2 est pair [d'après 1]
3. $2^2 = 4$ [calcul]
4. si n est pair alors n^2 est pair [lemme]
5. 4 est pair [d'après 2, 3 et 4]

Une suite d'affirmations justifiées, avec un contexte

1. supposons n pair [hypothèse P]
   - 1.1 supposons $n = p + p$ [hypothèse H]
     - 1.1.1 $n^2 = 2p^2 + 2p^2$ [calcul d'après H]
     - 1.1.2 n^2 est pair [d'après 1.1.1]
   - 1.2 n^2 est pair [d'après P et 1.1]
2. si n est pair alors n^2 est pair [d'après 1]

Qu'est-ce qu'une démonstration ?

Arbres de déduction : idées

- l'important c'est la justification (déduction)
- un « état » de démonstration =
  les hypothèses sous lesquelles on travaille + la formule démontrée

On considère donc des arbres dont les nœuds sont les états de démonstration, justifiés par les sous-arbres correspondants.

Qu'est-ce qu'une démonstration ?

Arbres de déduction : idées

- l'important c'est la justification (déduction)
- un « état » de démonstration =
  les hypothèses sous lesquelles on travaille + la formule démontrée

On considère donc des arbres dont les nœuds sont les états de démonstration, justifiés par les sous-arbres correspondants.

Définition (Séquent)

$A_1, \ldots, A_n \vdash A$: on démontre A sous les hypothèses $A_1, \ldots, A_n$.

Qu'est-ce qu'une démonstration ?

Arbres de déduction : idées

- l'important c'est la justification (déduction)
- un « état » de démonstration =
  les hypothèses sous lesquelles on travaille + la formule démontrée

On considère donc des arbres dont les nœuds sont les états de démonstration, justifiés par les sous-arbres correspondants.

Définition (Séquent)

$A_1, \ldots, A_n \vdash A$: on démontre A sous les hypothèses $A_1, \ldots, A_n$.

Définition (Déduction)

$$\frac{\Gamma_1 \vdash A_1 \quad \cdots \quad \Gamma_n \vdash A_n}{\Delta \vdash B}$$

Qu'est-ce qu'une démonstration ?

Arbres de déduction : idées

- l'important c'est la justification (déduction)
- un « état » de démonstration =
  les hypothèses sous lesquelles on travaille + la formule démontrée

On considère donc des arbres dont les nœuds sont les états de démonstration, justifiés par les sous-arbres correspondants.

Définition (Séquent)

$A_1, \ldots, A_n \vdash A$: on démontre A sous les hypothèses $A_1, \ldots, A_n$.

Définition (Déduction)

$$\frac{\Gamma_1 \vdash A_1 \quad \cdots \quad \Gamma_n \vdash A_n}{\Delta \vdash B}$$

($\Downarrow$) si on montre $\Gamma_1 \vdash A_1, \ldots, \Gamma_n \vdash A_n$ alors on peut déduire $\Delta \vdash B$

Qu'est-ce qu'une démonstration ?

Arbres de déduction : idées

- l'important c'est la justification (déduction)
- un « état » de démonstration =
  les hypothèses sous lesquelles on travaille + la formule démontrée

On considère donc des arbres dont les nœuds sont les états de démonstration, justifiés par les sous-arbres correspondants.

Définition (Séquent)

$A_1, \ldots, A_n \vdash A$: on démontre A sous les hypothèses $A_1, \ldots, A_n$.

Définition (Déduction)

$$\frac{\Gamma_1 \vdash A_1 \quad \cdots \quad \Gamma_n \vdash A_n}{\Delta \vdash B}$$

($\Uparrow$) pour montrer $\Delta \vdash B$ il suffit de montrer (séparément) $\Gamma_1 \vdash A_1$, $\ldots$, $\Gamma_n \vdash A_n$

Qu'est-ce qu'une démonstration ?

Arbres de déduction : un exemple

$$\frac{}{\vdash B \wedge (A \to \neg B) \to \neg A}$$

Qu'est-ce qu'une démonstration ?

Arbres de déduction : un exemple

$$\dfrac{\dfrac{}{B \wedge (A \to \neg B) \vdash \neg A}}{\vdash B \wedge (A \to \neg B) \to \neg A}$$

Qu'est-ce qu'une démonstration ?

Arbres de déduction : un exemple

$$\dfrac{\dfrac{\dfrac{}{B \wedge (A \to \neg B), A \vdash \bot}}{B \wedge (A \to \neg B) \vdash \neg A}}{\vdash B \wedge (A \to \neg B) \to \neg A}$$

Qu'est-ce qu'une démonstration ?

Arbres de déduction : un exemple

$$\dfrac{\dfrac{\dfrac{}{B, A \to \neg B, A \vdash \bot}}{B \wedge (A \to \neg B), A \vdash \bot}}{B \wedge (A \to \neg B) \vdash \neg A}$$
$$\vdash B \wedge (A \to \neg B) \to \neg A$$

Qu'est-ce qu'une démonstration ?

Arbres de déduction : un exemple

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{B, A \to \neg B, A \vdash \neg B} \quad \dfrac{}{B, A \to \neg B, A \vdash B}}{B, A \to \neg B, A \vdash \bot}}{B \wedge (A \to \neg B), A \vdash \bot}}{B \wedge (A \to \neg B) \vdash \neg A}}{\vdash B \wedge (A \to \neg B) \to \neg A}$$

Qu'est-ce qu'une démonstration ?

Arbres de déduction : un exemple

$$\dfrac{\dfrac{\dfrac{\dfrac{}{B, A \to \neg B, A \vdash \neg B} \quad \dfrac{}{B, A \to \neg B, A \vdash B}}{B, A \to \neg B, A \vdash \bot}}{B \wedge (A \to \neg B), A \vdash \bot}}{B \wedge (A \to \neg B) \vdash \neg A}$$
$$\vdash B \wedge (A \to \neg B) \to \neg A$$

- Quelles déductions sont valides ?

Qu'est-ce qu'une démonstration ?

Arbres de déduction : un exemple

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{B, A \to \neg B, A \vdash \neg B} \quad \dfrac{}{B, A \to \neg B, A \vdash B}}{B, A \to \neg B, A \vdash \bot}}{B \wedge (A \to \neg B), A \vdash \bot}}{B \wedge (A \to \neg B) \vdash \neg A}}{\vdash B \wedge (A \to \neg B) \to \neg A}$$

- Quelles déductions sont valides ?
- Il faut des règles.

Exemples

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B} \qquad \frac{\Gamma, A, B \vdash C}{\Gamma A \wedge B \vdash C} \qquad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A}$$

$$\frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \wedge B} \qquad \frac{\Gamma, A, A \vdash B}{\Gamma, A \vdash B} \qquad \frac{\Gamma \vdash B}{\Gamma, A \vdash B}$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B} \qquad \frac{\Gamma \vdash A \quad A \vdash B}{\Gamma \vdash B} \qquad \frac{}{\vdash A \to A}$$

Exemples

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B} \qquad \frac{\Gamma, A, B \vdash C}{\Gamma A \wedge B \vdash C} \qquad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A}$$

$$\frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \wedge B} \qquad \frac{\Gamma, A, A \vdash B}{\Gamma, A \vdash B} \qquad \frac{\Gamma \vdash B}{\Gamma, A \vdash B}$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B} \qquad \frac{\Gamma \vdash A \quad A \vdash B}{\Gamma \vdash B} \qquad \frac{}{\vdash A \to A}$$

- il faut qu'elles soient correctes (pour la vérité booléenne)
- on a intérêt à en prendre assez peu

(on ne considère que $\to$ et $\bot$: ça suffit du point de vue booléen)

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\ \langle\to\rangle \qquad \frac{\Gamma \vdash A \to B \quad \Gamma \vdash A}{\Gamma \vdash B}\ [\to]$$

$$\frac{}{\Gamma, A \vdash A} \qquad \frac{\Gamma \vdash \bot}{\Gamma \vdash A}$$

Partis pris

- on se concentre sur les conclusions (déductions « naturelles »)
- on gère les contextes de manière *additive*

Déduction naturelle

Exemple de démonstration

$$\frac{}{\vdash (A \wedge \neg A) \to B}\ \langle \to \rangle$$

Déduction naturelle

Exemple de démonstration

$$\frac{}{\vdash \neg(A \to \neg\neg A) \to B}\ \langle\to\rangle$$

Déduction naturelle

Exemple de démonstration

$$\dfrac{\dfrac{}{\neg(A \to \neg\neg A) \vdash B}\ [\bot]}{\vdash \neg(A \to \neg\neg A) \to B}\ \langle\to\rangle$$

Déduction naturelle

Exemple de démonstration

$$\dfrac{\dfrac{\dfrac{}{\neg(A \to \neg\neg A) \vdash \bot}\,[\to]}{\neg(A \to \neg\neg A) \vdash B}\,[\bot]}{\vdash \neg(A \to \neg\neg A) \to B}\,\langle\to\rangle$$

Déduction naturelle

Exemple de démonstration

$$\dfrac{\dfrac{\dfrac{}{\neg(A \to \neg\neg A) \vdash A \to \neg\neg A}\ \langle\to\rangle \qquad \dfrac{}{\neg(A \to \neg\neg A) \vdash \neg(A \to \neg\neg A)}\ (ax)}{\dfrac{\neg(A \to \neg\neg A) \vdash \bot}{\neg(A \to \neg\neg A) \vdash B}\ [\bot]}\ [\to]}{\vdash \neg(A \to \neg\neg A) \to B}\ \langle\to\rangle$$

Déduction naturelle

Exemple de démonstration

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\neg(A \to \neg\neg A), A \vdash \neg\neg A}\,\langle\to\rangle}{\neg(A \to \neg\neg A) \vdash A \to \neg\neg A}\,\langle\to\rangle \qquad \dfrac{}{\neg(A \to \neg\neg A) \vdash \neg(A \to \neg\neg A)}\,(ax)}{\neg(A \to \neg\neg A) \vdash \bot}\,[\to]}{\neg(A \to \neg\neg A) \vdash B}\,[\bot]}{\vdash \neg(A \to \neg\neg A) \to B}\,\langle\to\rangle$$

Déduction naturelle

Exemple de démonstration

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma \vdash \bot}\,[\to]}{\neg(A \to \neg\neg A), A \vdash \neg\neg A}\,\langle\to\rangle}{\neg(A \to \neg\neg A) \vdash A \to \neg\neg A}\,\langle\to\rangle \qquad \dfrac{}{\neg(A \to \neg\neg A) \vdash \neg(A \to \neg\neg A)}\,(ax)}{\dfrac{\dfrac{\neg(A \to \neg\neg A) \vdash \bot}{\neg(A \to \neg\neg A) \vdash B}\,[\bot]}{\vdash \neg(A \to \neg\neg A) \to B}\,\langle\to\rangle}\,[\to]$$

avec $\Gamma = \neg(A \to \neg\neg A), A, \neg A$.

Déduction naturelle

Exemple de démonstration

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma \vdash \neg A}\,(ax) \quad \dfrac{}{\Gamma \vdash A}\,(ax)}{\Gamma \vdash \bot}\,[\to]}{\neg(A \to \neg\neg A), A \vdash \neg\neg A}\,\langle\to\rangle}{\neg(A \to \neg\neg A) \vdash A \to \neg\neg A}\,\langle\to\rangle \quad \dfrac{}{\neg(A \to \neg\neg A) \vdash \neg(A \to \neg\neg A)}\,(ax)}{\neg(A \to \neg\neg A) \vdash \bot}\,[\to]}{\neg(A \to \neg\neg A) \vdash B}\,[\bot]}{\vdash \neg(A \to \neg\neg A) \to B}\,\langle\to\rangle$$

avec $\Gamma = \neg(A \to \neg\neg A), A, \neg A$.

Déduction naturelle

Exemple de démonstration

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma \vdash \neg A}\,(ax) \quad \dfrac{}{\Gamma \vdash A}\,(ax)}{\Gamma \vdash \bot}\,[\to]}{\neg(A \to \neg\neg A), A \vdash \neg\neg A}\,\langle\to\rangle}{\neg(A \to \neg\neg A) \vdash A \to \neg\neg A}\,\langle\to\rangle \quad \dfrac{}{\neg(A \to \neg\neg A) \vdash \neg(A \to \neg\neg A)}\,(ax)}{\neg(A \to \neg\neg A) \vdash \bot}\,[\to]}{\neg(A \to \neg\neg A) \vdash B}\,[\bot]}{\vdash \neg(A \to \neg\neg A) \to B}\,\langle\to\rangle$$

avec $\Gamma = \neg(A \to \neg\neg A), A, \neg A$.

- Hrumpfff !

Déduction naturelle

Exemple de démonstration

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma \vdash \neg A}\,(ax) \quad \dfrac{}{\Gamma \vdash A}\,(ax)}{\Gamma \vdash \bot}\,[\to]}{\neg(A \to \neg\neg A), A \vdash \neg\neg A}\,\langle\to\rangle}{\neg(A \to \neg\neg A) \vdash A \to \neg\neg A}\,\langle\to\rangle \quad \dfrac{}{\neg(A \to \neg\neg A) \vdash \neg(A \to \neg\neg A)}\,(ax)}{\neg(A \to \neg\neg A) \vdash \bot}\,[\to]}{\neg(A \to \neg\neg A) \vdash B}\,[\bot]}{\vdash \neg(A \to \neg\neg A) \to B}\,\langle\to\rangle$$

avec $\Gamma = \neg(A \to \neg\neg A), A, \neg A$.

- Hrumpfff !
- Et $\neg\neg A \to A$?

$$\dfrac{\dfrac{}{\neg\neg A \vdash A}\,?}{\vdash \neg\neg A \to A}\ \langle\to\rangle$$

$$\dfrac{\dfrac{\dfrac{?}{\neg\neg A \vdash \bot}}{\neg\neg A \vdash A}\,[\bot]}{\vdash \neg\neg A \to A}\,\langle\to\rangle$$

Déduction naturelle

Raisonner sur la vérité

$$\dfrac{\dfrac{\neg\neg A \vdash ? \quad \neg\neg A \vdash ? \to A}{\neg\neg A \vdash A}\,[\to]}{\vdash \neg\neg A \to A}\,\langle\to\rangle$$

$$\dfrac{\dfrac{?}{\neg\neg A \vdash A}}{\vdash \neg\neg A \to A}\ \langle\to\rangle$$

- On ne voit pas trop (sauf si par exemple $A = \neg B$)

$$\dfrac{\dfrac{?}{\neg\neg A \vdash A}}{\vdash \neg\neg A \to A}\ \langle\to\rangle$$

- On ne voit pas trop (sauf si par exemple $A = \neg B$)
- On a oublié quelque chose pour raisonner sur la vérité

$$\dfrac{\dfrac{?}{\neg\neg A \vdash A}}{\vdash \neg\neg A \to A}\ \langle\to\rangle$$

- On ne voit pas trop (sauf si par exemple $A = \neg B$)
- On a oublié quelque chose pour raisonner sur la vérité
- Par exemple, le raisonnement par l'absurde :

$$\dfrac{\Gamma, \neg A \vdash \bot}{\Gamma \vdash A}\ (\bot)$$

$$\dfrac{\overset{?}{\neg\neg A \vdash A}}{\vdash \neg\neg A \to A}\ \langle\to\rangle$$

- On ne voit pas trop (sauf si par exemple $A = \neg B$)
- On a oublié quelque chose pour raisonner sur la vérité
- Par exemple, le raisonnement par l'absurde :

$$\dfrac{\Gamma, \neg A \vdash \bot}{\Gamma \vdash A}\ (\bot)$$

Attention, c'est bien différent de $\dfrac{\Gamma, A \vdash \bot}{\Gamma \vdash \neg A}\ \langle\to\rangle$

$$\dfrac{\dfrac{\dfrac{}{\neg\neg A, \neg A \vdash \neg A}\,(ax) \quad \dfrac{}{\neg\neg A, \neg A \vdash \neg\neg A}\,(ax)}{\neg\neg A, \neg A \vdash \bot}\,[\to]}{\neg\neg A \vdash A}\,(\bot)}{\vdash \neg\neg A \to A}\,\langle\to\rangle$$

- On ne voit pas trop (sauf si par exemple $A = \neg B$)
- On a oublié quelque chose pour raisonner sur la vérité
- Par exemple, le raisonnement par l'absurde :

$$\dfrac{\Gamma, \neg A \vdash \bot}{\Gamma \vdash A}\,(\bot)$$

Attention, c'est bien différent de $\dfrac{\Gamma, A \vdash \bot}{\Gamma \vdash \neg A}\,\langle\to\rangle$

Déduction naturelle

Et les gagnants sont. . .

$$\frac{}{\Gamma, A \vdash A}\ (ax) \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\ \langle\to\rangle \qquad \frac{\Gamma \vdash A \to B \qquad \Gamma \vdash A}{\Gamma \vdash B}\ [\to] \qquad \frac{\Gamma, A \to \bot \vdash \bot}{\Gamma \vdash A}\ (\bot)$$

Déduction naturelle

Et les gagnants sont. . .

$$\frac{}{\Gamma, A \vdash A}\ (ax) \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\ \langle\to\rangle \qquad \frac{\Gamma \vdash A \to B \qquad \Gamma \vdash A}{\Gamma \vdash B}\ [\to] \qquad \frac{\Gamma, A \to \bot \vdash \bot}{\Gamma \vdash A}\ (\bot)$$

A-t-on encore oublié quelque chose ?

Déduction naturelle

Et les gagnants sont. . .

$$\frac{}{\Gamma, A \vdash A}\ (ax) \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\ \langle\to\rangle \qquad \frac{\Gamma \vdash A \to B \qquad \Gamma \vdash A}{\Gamma \vdash B}\ [\to] \qquad \frac{\Gamma, A \to \bot \vdash \bot}{\Gamma \vdash A}\ (\bot)$$

A-t-on encore oublié quelque chose ? Non :

Théorème (Complétude)

Si A est une tautologie, alors il existe une preuve de $\vdash A$ en déduction naturelle classique.

Déduction naturelle

Et les gagnants sont. . .

$$\frac{}{\Gamma, A \vdash A}\ (ax) \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\ \langle\to\rangle \qquad \frac{\Gamma \vdash A \to B \qquad \Gamma \vdash A}{\Gamma \vdash B}\ [\to] \qquad \frac{\Gamma, A \to \bot \vdash \bot}{\Gamma \vdash A}\ (\bot)$$

A-t-on encore oublié quelque chose ? Non :

Théorème (Complétude)

Si A est une tautologie, alors il existe une preuve de $\vdash A$ en déduction naturelle classique.

- Facile. . .

Déduction naturelle

Et les gagnants sont. . .

$$\frac{}{\Gamma, A \vdash A}\ (ax) \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\ \langle\to\rangle \qquad \frac{\Gamma \vdash A \to B \qquad \Gamma \vdash A}{\Gamma \vdash B}\ [\to] \qquad \frac{\Gamma, A \to \bot \vdash \bot}{\Gamma \vdash A}\ (\bot)$$

A-t-on encore oublié quelque chose ? Non :

Théorème (Complétude)

Si A est une tautologie, alors il existe une preuve de $\vdash A$ en déduction naturelle classique.

- Facile. . .
- On aurait pu faire pareil avec d'autres règles ou d'autres jeux de connecteurs.

Intuitionnisme

Sans raisonnement par l'absurde

$$\frac{}{\Gamma, A \vdash A}\ (ax) \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\ \langle\to\rangle \qquad \frac{\Gamma \vdash A \to B \qquad \Gamma \vdash A}{\Gamma \vdash B}\ [\to]$$

Intuitionnisme

Sans raisonnement par l'absurde

$$\frac{}{\Gamma, A \vdash A}\ (ax) \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\ \langle\to\rangle \qquad \frac{\Gamma \vdash A \to B \qquad \Gamma \vdash A}{\Gamma \vdash B}\ [\to]$$

- ça reste correct
- mais c'est incomplet

Intuitionnisme

Sans raisonnement par l'absurde

$$\frac{}{\Gamma, A \vdash A}\ (ax) \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\ \langle\to\rangle \qquad \frac{\Gamma \vdash A \to B \qquad \Gamma \vdash A}{\Gamma \vdash B}\ [\to]$$

- ça reste correct
- mais c'est incomplet

Pourquoi s'embêter avec ça ?

- en 1900 : parce que
- en 2000 : parce qu'on peut dire quelque chose d'intéressant dans ce cas

Une explication « procédurale » des preuves

- une preuve de $A \wedge B$ doit fournir une preuve de A et une preuve de B ;
- une preuve de $A \vee B$ doit fournir une preuve de A (et le bit « gauche ») ou une preuve de B (et le bit « droit ») ;
- une preuve de $A \to B$ doit fournir une preuve de B quand on lui fournit une preuve de A ;
- il n'y a pas de preuve de $\bot$.

Une explication « procédurale » des preuves

- une preuve de $A \wedge B$ doit fournir une preuve de A et une preuve de B ;
- une preuve de $A \vee B$ doit fournir une preuve de A (et le bit « gauche ») ou une preuve de B (et le bit « droit ») ;
- une preuve de $A \to B$ doit fournir une preuve de B quand on lui fournit une preuve de A ;
- il n'y a pas de preuve de $\bot$.

Si on la prend au sens littéral (un connecteur = une opération ensembliste) cette idée revient au modèle booléen dès qu'il y a une négation :

- si A a une preuve $\neg A$ n'en a pas ;
- si A n'a pas de preuve $\neg A$ en a exactement une.

Quand même. . .

- une preuve de $A \vdash B$ est une « fonction » qui transforme une preuve de A en une preuve de B

$$\dfrac{\overset{f}{A \vdash B} \qquad \dfrac{\overset{g}{B \vdash C}}{\vdash B \to C}\ \langle\to\rangle}{A \vdash C}\ [\to]$$

Quand même. . .

- une preuve de $A \vdash B$ est une « fonction » qui transforme une preuve de A en une preuve de B

$$\dfrac{x : A \overset{f}{\vdash} f(x) : B \qquad \dfrac{y : B \overset{g}{\vdash} g(y) : C}{\vdash (y \mapsto g(y)) : B \to C}\ \langle\to\rangle}{x : A \vdash g(f(x)) : C}\ [\to]$$

Quand même. . .

- une preuve de $A \vdash B$ est une « fonction » qui transforme une preuve de A en une preuve de B

$$\dfrac{x : A \overset{f}{\vdash} f(x) : B \qquad \dfrac{y : B \overset{g}{\vdash} g(y) : C}{\vdash (y \mapsto g(y)) : B \to C}\ \langle\to\rangle}{x : A \vdash g(f(x)) : C}\ [\to]$$

- ça *calcule* la composition des fonctions !

Quand même. . .

- une preuve de $A \vdash B$ est une « fonction » qui transforme une preuve de A en une preuve de B

$$\dfrac{x : A \overset{f}{\vdash} f(x) : B \qquad \dfrac{y : B \overset{g}{\vdash} g(y) : C}{\vdash (y \mapsto g(y)) : B \to C}\ \langle\to\rangle}{x : A \vdash g(f(x)) : C}\ [\to]$$

- ça *calcule* la composition des fonctions !
- mais de quoi parle-t-on ?

Deuxième partie

Calcul

Le λ-calcul
- Le langage et les axiomes
- Cohérence du système
- Expressivité du calcul

Typage

La correspondance
- L'élimination des coupures
- Les preuves comme programmes

La démarche axiomatique

Qu'est-ce qu'une fonction ?

La démarche axiomatique

Qu'est-ce qu'une fonction ?

La réponse à cette question a varié au cours du temps.

avant le XVIIème siècle une quoi ?

Leibniz (XVIIème) une courbe

Euler (XVIIIème) une formule, un calcul

Dirichlet (XIXème) une relation

Mêmes questions pour les réels, les entiers, les ensembles...
les démonstrations, les calculs...

En fin de compte, pour fonder le sens des démonstrations sur la notion de fonction, on cherche une axiomatique pure des fonctions.

λ-calcul : le langage

Le langage :

$$
\begin{array}{rll}
M, N := & \lambda x.M & \text{fonction qui à } x \text{ associe } M \\
& (M)N & \text{application de la fonction } M \text{ à l'argument } N \\
& x & \text{utilisation d'une variable}
\end{array}
$$

Par exemple :

- la fonction identité : $\lambda x.x$ $\qquad x \mapsto x$
- composition de fonctions : $\lambda x.(f)(g)x$ $\qquad x \mapsto f(g(x))$
- composition avec f : $\lambda g.\lambda x.(g)(f)x$ $\qquad g \mapsto (x \mapsto g(f(x)))$

Paramètres et indéterminées

Une analogie :

- Dans $\mathbb{R}[X, Y]$, les polynômes $X^2 + 5$ et $Y^2 + 5$ sont distincts.
- Les fonctions $x \mapsto x^2 + 5$ et $y \mapsto y^2 + 5$ sont identiques.

Substitution et évaluation

Considérons deux termes :

$$M := \lambda g.\lambda x.(g)(f)x$$
$$N := \lambda y.((h)y)y$$

$(M)N$ est l'application de la fonction M à l'argument N donc :

$$\begin{aligned}(M)N &= \big(\lambda g.\lambda x.(g)(f)x\big)\,N \\ &= \lambda x.(N)(f)x\end{aligned}$$

Substitution et évaluation

Considérons deux termes :

$$M := \lambda g.\lambda x.(g)(f)x$$
$$N := \lambda y.((h)y)y$$

$(M)N$ est l'application de la fonction M à l'argument N donc :

$$\begin{aligned}(M)N &= \big(\lambda g.\lambda x.(g)(f)x\big)\,N \\ &= \lambda x.(N)(f)x \\ \ldots \text{ et on peut continuer } &\ldots \\ &= \lambda x.(\lambda y.((h)y)y)(f)x \\ &= \lambda x.((h)(f)x)(f)x\end{aligned}$$

Substitution et évaluation

Considérons deux termes :

$$M := \lambda g.\lambda x.(g)(f)x$$
$$N := \lambda y.((h)y)y$$

$(M)N$ est l'application de la fonction M à l'argument N donc :

$$\begin{aligned}(M)N &= (\lambda g.\lambda x.(g)(f)x)\,N \\ &= \lambda x.(N)(f)x \\ \ldots \text{ et on peut continuer } &\ldots \\ &= \lambda x.(\lambda y.((h)y)y)(f)x \\ &= \lambda x.((h)(f)x)(f)x\end{aligned}$$

- β-équivalence : l'évaluation des fonctions

$$(\lambda x.M)N =_\beta M[N/x]$$

Quid de la cohérence ?

On oriente la β-équivalence :

$$(\lambda x.M)N \rightsquigarrow M[N/x] \qquad \beta\text{-réduction}$$

Théorème (Church-Rosser)

Pour tous λ-termes M et N tels que $M =_\beta N$, il existe un λ-terme R tel que $M \rightsquigarrow \cdots \rightsquigarrow R$ et $N \rightsquigarrow \cdots \rightsquigarrow R$.

C'est-à-dire qu'il suffit de réduire pour établir une équivalence.

Corollaire

Si M et N sont deux λ-termes irréductibles, alors $M =_\beta N$ si et seulement si $M = N$.

Les termes irréductibles sont donc appelés formes normales.

Normalisation

Les formes normales sont donc des représentants canoniques de classes de β-équivalence, il suffit de réduire autant qu'on peut pour **calculer** la forme normale.

Normalisation

Les formes normales sont donc des représentants canoniques de classes de β-équivalence, il suffit de réduire autant qu'on peut pour **calculer** la forme normale.

Posons $\delta := \lambda x.(x)x$ et $\Omega := (\delta)\delta$, alors $\Omega \rightsquigarrow \Omega$.
On n'atteint pas de forme normale !

Normalisation

Les formes normales sont donc des représentants canoniques de classes de β-équivalence, il suffit de réduire autant qu'on peut pour **calculer** la forme normale.

Posons $\delta := \lambda x.(x)x$ et $\Omega := (\delta)\delta$, alors $\Omega \rightsquigarrow \Omega$.
On n'atteint pas de forme normale !

Théorème

Le problème de l'existence de forme normale pour un terme donné est indécidable.

Programmer en λ-calcul

- Un booléen est une façon de choisir entre deux possibilités :

$$\begin{aligned} \text{true} &:= \lambda x.\lambda y.x \\ \text{false} &:= \lambda x.\lambda y.y \\ \text{if } P \text{ then } Q \text{ else } C &:= ((P)Q)R \end{aligned}$$

Programmer en λ-calcul

- Un booléen est une façon de choisir entre deux possibilités :

$$\begin{aligned} \text{true} &:= \lambda x.\lambda y.x \\ \text{false} &:= \lambda x.\lambda y.y \\ \text{if } P \text{ then } Q \text{ else } C &:= ((P)Q)R \end{aligned}$$

- Un entier n est un opérateur qui compose n fois une fonction :

$$\underline{n} := \lambda f.\lambda x.\underbrace{(f)(f)\cdots(f)}_{n \text{ fois}}x$$

$$\begin{aligned} \text{succ} &:= \lambda n.\lambda f.\lambda x.(f)(n)fx \\ \text{add} &:= \lambda m.\lambda n.\lambda f.\lambda x.(m)f(n)fx \quad \text{ou } \lambda m.\lambda n.((m)\text{succ})n \\ \text{mul} &:= \lambda m.\lambda n.\lambda f.\lambda x.((m)(n)f)x \quad \text{ou } \lambda m.\lambda n.((m)(\text{add})n)\underline{0} \\ \text{isZero} &:= \lambda n.n(\lambda b.\text{false})\text{true} \end{aligned}$$

Expressivité

Théorème

Une fonction partielle $f : \mathbb{N} \to \mathbb{N}$ est calculable si et seulement si il existe un λ-terme F tel que

- *si $f(n)$ est défini alors $(F)\underline{n} =_\beta \underline{f(n)}$,*
- *si $f(n)$ n'est pas défini alors $(F)\underline{n}$ n'a pas de forme normale.*

Le typage

Pour des termes qui ont un sens, on utilise des types :

$$\begin{array}{rll} A, B := & \alpha & \text{type de base (donné)} \\ & A \to B & \text{fonction de } A \text{ dans } B \end{array}$$

Moralement, un type est un ensemble de termes, on l'interprète comme un ensemble de « valeurs » possibles.

Un jugement de typage est de la forme

$$x_1 : A_1, \ldots, x_n : A_n \vdash M : B$$

Si chaque x_i prend une valeur dans le type A_i correspondant, alors M prendra une valeur dans le type B.

Règles de typage

Le λ-calcul simplement typé

- construction d'une fonction

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.M : A \to B}$$

- application d'une fonction

$$\frac{\Gamma \vdash M : A \to B \qquad \Gamma \vdash N : A}{\Gamma \vdash (M)N : B}$$

- utilisation d'une variable

$$\frac{}{\Gamma, x : A \vdash x : A}$$

Théorème (normalisation forte)

Un terme typable n'a pas de suite infinie de β-réductions.

Règles de typage

Le λ-calcul simplement typé

- construction d'une fonction

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.M : A \to B}$$

- application d'une fonction

$$\frac{\Gamma \vdash M : A \to B \qquad \Gamma \vdash N : A}{\Gamma \vdash (M)N : B}$$

- utilisation d'une variable

$$\frac{}{\Gamma, x : A \vdash x : A}$$

Théorème (normalisation forte)

Un terme typable n'a pas de suite infinie de β-réductions.

- avec des types de base et la récursion : (à peu près) `PCF`
  (et le théorème devient faux, à cause des points fixes)
- avec en plus le polymorphisme : (à peu près) `Caml`

Règles de typage

Le λ-calcul simplement typé

- construction d'une fonction

$$\frac{\Gamma, \quad A \vdash \quad B}{\Gamma \vdash \quad A \to B}$$

- application d'une fonction

$$\frac{\Gamma \vdash \quad A \to B \quad \Gamma \vdash \quad A}{\Gamma \vdash \quad B}$$

- utilisation d'une variable

$$\frac{}{\Gamma, \quad A \vdash \quad A}$$

Théorème (normalisation forte)

Un terme typable n'a pas de suite infinie de β-réductions.

- avec des types de base et la récursion : (à peu près) `PCF`
  (et le théorème devient faux, à cause des points fixes)
- avec en plus le polymorphisme : (à peu près) `Caml`

Typage et réduction

Les règles de typage sont les mêmes que les règles de la déduction naturelle, décorées avec des λ-termes.

$$\dfrac{\dfrac{\dfrac{}{\Gamma, x : A \vdash x : A}}{\vdots}}{\dfrac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.M : A \to B} \qquad \Gamma \vdash N : A}{\Gamma \vdash (\lambda x.M)N : B} \quad \rightsquigarrow \quad \dfrac{\Gamma \vdash N : A}{\vdots}{\Gamma \vdash M[N/x] : B}$$

Que signifie cette réduction du point de vue de la déduction naturelle ?

Typage et réduction

Les règles de typage sont les mêmes que les règles de la déduction naturelle, décorées avec des λ-termes.

$$\dfrac{\dfrac{\dfrac{}{\Gamma, A \vdash A}}{\vdots}}{\dfrac{\Gamma, A \vdash B}{\Gamma \vdash A \to B} \qquad \Gamma \vdash A} \quad \rightsquigarrow \quad \begin{array}{c} \Gamma \vdash A \\ \vdots \\ \Gamma \vdash B \end{array}$$

C'est une réécriture des démonstrations : l'élimination des coupures.

Élimination des coupures

Définition

En déduction naturelle, une *coupure* est l'enchaînement d'une règle d'introduction et d'une règle d'élimination de la même formule.

Théorème

Un séquent $\Gamma \vdash A$ est démontrable en déduction naturelle si et seulement si il est démontrable avec une démonstration sans coupure.

Interpréter la démonstration comme un λ-terme typé, le normaliser, le typage de la forme normale est une démonstration sans coupure.

Élimination des coupures et cohérence logique

Théorème

Le séquent $\vdash \neg\neg\alpha \to \alpha$ *n'est pas démontrable en logique minimale.*

Élimination des coupures et cohérence logique

Théorème

Le séquent $\vdash \neg\neg\alpha \to \alpha$ *n'est pas démontrable en logique minimale.*

Cherchons une démonstration *sans coupure* :

$$\frac{?}{\vdash M : \neg\neg\alpha \to \alpha}$$

Élimination des coupures et cohérence logique

Théorème

Le séquent $\vdash \neg\neg\alpha \to \alpha$ *n'est pas démontrable en logique minimale.*

Cherchons une démonstration *sans coupure* : avec $\Gamma = x : \neg\neg\alpha$

$$\dfrac{\dfrac{?}{\Gamma \vdash M : \alpha}}{\vdash \lambda x.M : \neg\neg\alpha \to \alpha}$$

Élimination des coupures et cohérence logique

Théorème

Le séquent $\vdash \neg\neg\alpha \to \alpha$ *n'est pas démontrable en logique minimale.*

Cherchons une démonstration *sans coupure* : avec $\Gamma = x : \neg\neg\alpha$

$$\dfrac{\dfrac{\dfrac{?}{\Gamma \vdash M : A_1 \to \alpha} \quad \dfrac{\vdots}{\Gamma \vdash N_1 : A_1}}{\Gamma \vdash (M)N_1 : \alpha}}{\vdash \lambda x.(M)N_1 : \neg\neg\alpha \to \alpha}$$

Élimination des coupures et cohérence logique

Théorème

Le séquent $\vdash \neg\neg\alpha \to \alpha$ *n'est pas démontrable en logique minimale.*

Cherchons une démonstration *sans coupure* : avec $\Gamma = x : \neg\neg\alpha$

$$\dfrac{\dfrac{\dfrac{?}{\Gamma \vdash M : A_2 \to A_1 \to \alpha} \quad \dfrac{\vdots}{\Gamma \vdash N_2 : A_2}}{\Gamma \vdash (M)N_2 : A_1 \to \alpha} \quad \dfrac{\vdots}{\Gamma \vdash N_1 : A_1}}{\Gamma \vdash ((M)N_2)N_1 : \alpha}$$
$$\dfrac{}{\vdash \lambda x.((M)N_2)N_1 : \neg\neg\alpha \to \alpha}$$

Élimination des coupures et cohérence logique

Théorème

Le séquent $\vdash \neg\neg\alpha \to \alpha$ *n'est pas démontrable en logique minimale.*

Cherchons une démonstration *sans coupure* : avec $\Gamma = x : \neg\neg\alpha$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{?}{\Gamma \vdash M : A_k \to \cdots A_2 \to A_1 \to \alpha}}{\vdots}}{\Gamma \vdash \cdots(M)N_k\cdots : A_2 \to A_1 \to \alpha} \quad \dfrac{\vdots}{\Gamma \vdash N_2 : A_2}}{\Gamma \vdash (\cdots(M)N_k\cdots)N_2 : A_1 \to \alpha} \quad \dfrac{\vdots}{\Gamma \vdash N_1 : A_1}}{\Gamma \vdash ((\cdots(M)N_k\cdots)N_2)N_1 : \alpha}}{\vdash \lambda x.((\cdots(M)N_k\cdots)N_2)N_1 : \neg\neg\alpha \to \alpha}$$

Élimination des coupures et cohérence logique

Théorème

Le séquent $\vdash \neg\neg\alpha \to \alpha$ *n'est pas démontrable en logique minimale.*

Cherchons une démonstration *sans coupure* : avec $\Gamma = x : \neg\neg\alpha$

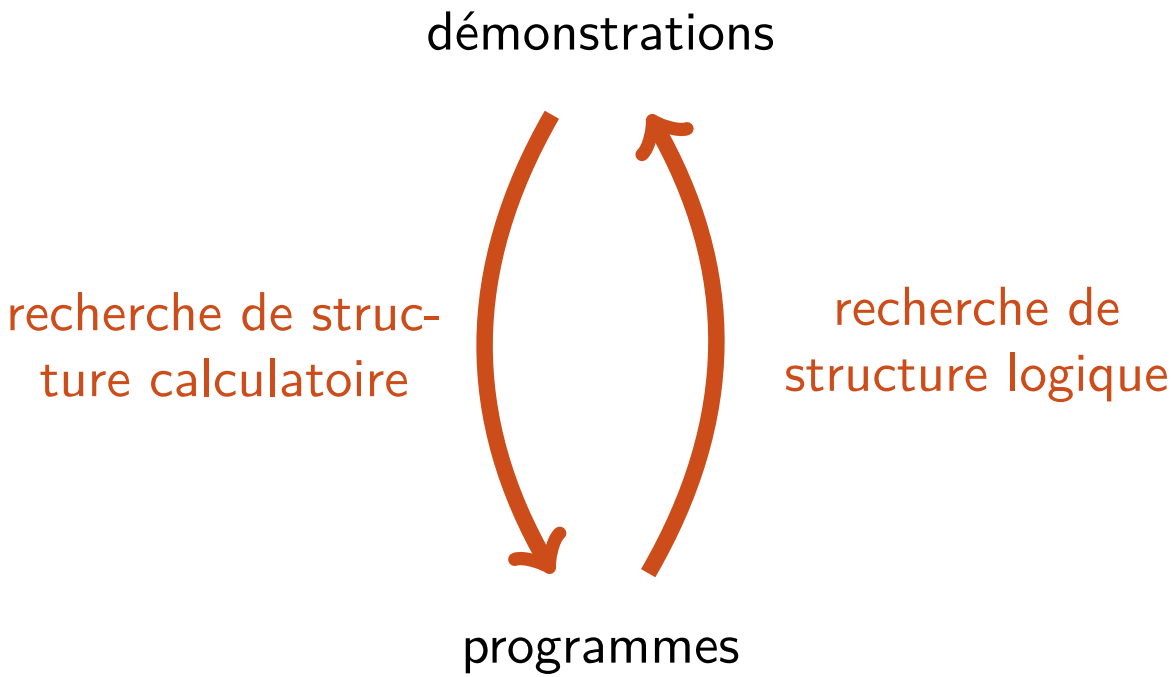
$$\dfrac{\dfrac{\dfrac{\dfrac{?}{\Gamma \vdash M : A_k \to \cdots A_2 \to A_1 \to \alpha}}{\vdots}}{\Gamma \vdash \cdots(M)N_k\cdots : A_2 \to A_1 \to \alpha} \quad \dfrac{\vdots}{\Gamma \vdash N_2 : A_2}}{\Gamma \vdash (\cdots(M)N_k\cdots)N_2 : A_1 \to \alpha} \quad \dfrac{\vdots}{\Gamma \vdash N_1 : A_1}$$
$$\dfrac{\Gamma \vdash ((\cdots(M)N_k\cdots)N_2)N_1 : \alpha}{\vdash \lambda x.((\cdots(M)N_k\cdots)N_2)N_1 : \neg\neg\alpha \to \alpha}$$

Impossible !

La correspondance

logique	**informatique**
formule	type
démonstration	programme
élimination des coupures	évaluation
règle de déduction	instruction élémentaire
lemme, proposition	sous-programme
théorie	interface de programmation
modèle	bibliothèque, environnement
traductions	compilations

Win-win partnership



Troisième partie

Sémantique dénotationnelle

(et logique linéaire)

Les preuves comme fonctions

Modèle relationnel

Pour aller plus loin

Retour à l'interprétation fonctionnelle

Curry–Howard

une preuve $=$ un λ-terme typé $=$ un programme fonctionnel

Retour à l'interprétation fonctionnelle

Curry–Howard

une preuve $=$ un λ-terme typé $=$ un programme fonctionnel

Et les « vraies » fonctions ?

Sémantique dénotationnelle

Principe

- $[\![\text{un type/une formule}]\!] = \text{un } \textit{espace}$;
- $[\![\text{un terme/une preuve}]\!] = \text{un } \textit{morphisme}$, *i.e.*

$$[\![\Gamma \vdash M : A]\!] : [\![\Gamma]\!] \to [\![A]\!];$$

- $M \rightsquigarrow N$ implique $[\![M]\!] = [\![N]\!]$.

Sémantique dénotationnelle

Principe

- $[\![\text{un type/une formule}]\!] = \text{un } \textit{espace}$;
- $[\![\text{un terme/une preuve}]\!] = \text{un } \textit{morphisme}$, *i.e.*

$$[\![\Gamma \vdash M : A]\!] : [\![\Gamma]\!] \to [\![A]\!];$$

- $M \rightsquigarrow N$ implique $[\![M]\!] = [\![N]\!]$.

- On déplace l'*infini* : de la dynamique vers les objets.

Sémantique dénotationnelle

Principe

- $[\![\text{un type/une formule}]\!] = \text{un } \textit{espace}$;
- $[\![\text{un terme/une preuve}]\!] = \text{un } \textit{morphisme}$, *i.e.*

$$[\![\Gamma \vdash M : A]\!] : [\![\Gamma]\!] \to [\![A]\!];$$

- $M \rightsquigarrow N$ implique $[\![M]\!] = [\![N]\!]$.

- On déplace l'*infini* : de la dynamique vers les objets.
- Avec un peu de chance, on apprend quelque chose.

Sémantique dénotationnelle

Principe

- $[\![\text{un type/une formule}]\!] = \text{un } \textit{espace}$;
- $[\![\text{un terme/une preuve}]\!] = \text{un } \textit{morphisme}$, *i.e.*

$$[\![\Gamma \vdash M : A]\!] : [\![\Gamma]\!] \to [\![A]\!];$$

- $M \rightsquigarrow N$ implique $[\![M]\!] = [\![N]\!]$.

- On déplace l'*infini* : de la dynamique vers les objets.
- Avec un peu de chance, on apprend quelque chose.

modèle dénotationnel	$\neq$	modèle de vérité
interprétation des preuves (niveau -2)	$\neq$	interprétation des formules (niveau -1)

Continuité (Scott)

Idée

Pour produire un bit de résultat, un programme consulte une quantité finie d'information.

Continuité (Scott)

Idée

Pour produire un bit de résultat, un programme consulte une quantité finie d'information.

Le comportement d'un programme est entièrement spécifié par ce qu'il fait sur des approximations finies de l'entrée (en temps fini, on ne lit qu'une partie finie de l'entrée).

Continuité (Scott)

Idée

Pour produire un bit de résultat, un programme consulte une quantité finie d'information.

Le comportement d'un programme est entièrement spécifié par ce qu'il fait sur des approximations finies de l'entrée (en temps fini, on ne lit qu'une partie finie de l'entrée).

Exemple (de truc pas continu)

Le test à zéro sur les flux binaires.

Continuité (Scott)

Idée

Pour produire un bit de résultat, un programme consulte une quantité finie d'information.

Le comportement d'un programme est entièrement spécifié par ce qu'il fait sur des approximations finies de l'entrée (en temps fini, on ne lit qu'une partie finie de l'entrée).

Exemple (de truc pas continu)

Le test à zéro sur les flux binaires.

Originellement formalisé à partir d'une notion d'ordre :
les domaines (des CPOs particuliers).

Stabilité (Berry)

Idée

Lorsqu'un programme produit un bit de résultat, on peut identifier une approximation minimale l'entrée qui correspond à ce bit.

Stabilité (Berry)

Idée

Lorsqu'un programme produit un bit de résultat, on peut identifier une approximation minimale l'entrée qui correspond à ce bit.

C'est ce que le programme a regardé de l'entrée avant de répondre (séquentialité).

Stabilité (Berry)

Idée

Lorsqu'un programme produit un bit de résultat, on peut identifier une approximation minimale l'entrée qui correspond à ce bit.

C'est ce que le programme a regardé de l'entrée avant de répondre (séquentialité).

Exemple (de truc pas stable)

Le OU parallèle.

Cohérence (Girard)

Grandes lignes

Reprend la condition de stabilité pour des domaines très particuliers : espaces = graphes, points = cliques ordonnées par inclusion.

Cohérence (Girard)

Grandes lignes

Reprend la condition de stabilité pour des domaines très particuliers :
espaces = graphes, points = cliques ordonnées par inclusion.

Les fonctions stables entre espaces de cohérence sont caractérisées par leur. . .

. . . trace

$$\mathrm{tr}\, f = \{(a, \beta) \mid a \subset_f |A| \text{ minimal tel que } \beta \in f(a)\}.$$

Cohérence (Girard)

Grandes lignes

Reprend la condition de stabilité pour des domaines très particuliers :
espaces = graphes, points = cliques ordonnées par inclusion.

Les fonctions stables entre espaces de cohérence sont caractérisées par leur. . .

. . . trace

$$\mathrm{tr}\, f = \{(a, \beta) \mid a \subset_f |A| \text{ minimal tel que } \beta \in f(a)\}.$$

Exemple : $\lambda a.a : \mathcal{C}(A) \to \mathcal{C}(A)$

$$\mathrm{tr}(\lambda a.a) = \{(\{\alpha\}, \alpha) \mid \alpha \in a \subset |A|\}$$

Cohérence (Girard)

Grandes lignes

Reprend la condition de stabilité pour des domaines très particuliers : espaces = graphes, points = cliques ordonnées par inclusion.

Les fonctions stables entre espaces de cohérence sont caractérisées par leur. . .

. . . trace

$$\mathrm{tr}\, f = \{(a, \beta) \mid a \subset_f |A| \text{ minimal tel que } \beta \in f(a)\}.$$

Exemple : $\lambda a.a : \mathcal{C}(A) \to \mathcal{C}(A)$

$$\mathrm{tr}(\lambda a.a) = \{(\{\alpha\}, \alpha) \mid \alpha \in a \subset |A|\}$$

Et les traces forment un espace de cohérence !

Cohérence (Girard)

Grandes lignes

Reprend la condition de stabilité pour des domaines très particuliers : espaces = graphes, points = cliques ordonnées par inclusion.

Les fonctions stables entre espaces de cohérence sont caractérisées par leur. . .

. . . trace

$$\mathrm{tr}\, f = \{(a, \beta) \mid a \subset_f |A| \text{ minimal tel que } \beta \in f(a)\}.$$

Exemple : $\lambda a.a : \mathcal{C}(A) \to \mathcal{C}(A)$

$$\mathrm{tr}(\lambda a.a) = \{(\{\alpha\}, \alpha) \mid \alpha \in a \subset |A|\}$$

Et les traces forment un espace de cohérence !

- Un modèle de la logique linéaire : $A \to B = !A \multimap B$.

Cohérence (Girard)

Grandes lignes

Reprend la condition de stabilité pour des domaines très particuliers : espaces = graphes, points = cliques ordonnées par inclusion.

Les fonctions stables entre espaces de cohérence sont caractérisées par leur. . .

. . . trace

$$\mathrm{tr}\, f = \{(a, \beta) \mid a \subset_f |A| \text{ minimal tel que } \beta \in f(a)\}.$$

Exemple : $\lambda a.a : \mathcal{C}(A) \to \mathcal{C}(A)$

$$\mathrm{tr}(\lambda a.a) = \{(\{\alpha\}, \alpha) \mid \alpha \in a \subset |A|\}$$

Et les traces forment un espace de cohérence !

- Un modèle de la logique linéaire : $A \to B = !A \multimap B$.
- On va faire plus simple (ou pas, suivant le temps).

Modèle relationnel

En bref

Principe

- un type $A \rightsquigarrow$ un ensemble $[\![A]\!]$
- on s'en tient aux traces : $s : A \rightsquigarrow [\![s]\!] \subset [\![A]\!]$
- on « compte » les appels à l'argument : $[\![A \to B]\!] = \mathcal{M}_f([\![A]\!]) \times B$

Modèle relationnel

En bref

Principe

- un type $A \rightsquigarrow$ un ensemble $[\![A]\!]$
- on s'en tient aux traces : $s : A \rightsquigarrow [\![s]\!] \subset [\![A]\!]$
- on « compte » les appels à l'argument : $[\![A \to B]\!] = \mathcal{M}_f([\![A]\!]) \times B$

$\mathcal{M}_f(A)$: multiensembles finis (listes modulo permutations) sur A

Modèle relationnel

En bref

Principe

- un type $A \rightsquigarrow$ un ensemble $[\![A]\!]$
- on s'en tient aux traces : $s : A \rightsquigarrow [\![s]\!] \subset [\![A]\!]$
- on « compte » les appels à l'argument : $[\![A \to B]\!] = \mathcal{M}_f([\![A]\!]) \times B$

$\mathcal{M}_f(A)$: multiensembles finis (listes modulo permutations) sur A

Intuition

Pour $s : A \to B$,

$$([\alpha_1, \ldots, \alpha_n], \beta) \in [\![s]\!]$$

se lit :

« s produit β en consommant $\alpha_1, \ldots, \alpha_n$ »

(on compte les multiplicités : notion de ressource).

Types

- on fixe un ensemble $[\![X]\!]$ pour chaque variable propositionnelle X
- $[\![A \to B]\!] = \mathcal{M}_f([\![A]\!]) \times B$

Types

- on fixe un ensemble $[\![X]\!]$ pour chaque variable propositionnelle X
- $[\![A \to B]\!] = \mathcal{M}_f([\![A]\!]) \times B$

Séquents

$$[\![A_1, \ldots, A_n \vdash B]\!] = \mathcal{M}_f([\![A_1]\!]) \times \cdots \times \mathcal{M}_f([\![A_n]\!]) \times [\![B]\!]$$

Types

- on fixe un ensemble $[\![X]\!]$ pour chaque variable propositionnelle X
- $[\![A \to B]\!] = \mathcal{M}_f([\![A]\!]) \times B$

Séquents

$$[\![A_1, \ldots, A_n \vdash B]\!] = \mathcal{M}_f([\![A_1]\!]) \times \cdots \times \mathcal{M}_f([\![A_n]\!]) \times [\![B]\!]$$

Termes

Si $\Gamma \vdash s : A$, on veut $[\![s]\!] \subset [\![\Gamma \vdash A]\!]$.
On définit un système d'inférence :

$$(\overline{\alpha}_1, \ldots, \overline{\alpha}_n, \beta) \in [\![s]\!] \quad \text{ssi} \quad x_1^{\overline{\alpha}_1} : A_1, \ldots, x_n^{\overline{\alpha}_n} : A_n \vdash s^{\beta} : B$$

$$\frac{}{\Gamma^{[]}, x^{[\alpha]} : A \vdash x^{\alpha} : A}$$

$$\frac{\Gamma^{\overline{\gamma}}, x^{\overline{\alpha}} : A \vdash s^{\beta} : B}{\Gamma^{\overline{\gamma}} \vdash \lambda x.s^{(\overline{\alpha},\beta)} : A \to B}$$

$$\frac{\Gamma^{\overline{\gamma}_0} \vdash s^{([\alpha_1,\ldots,\alpha_k],\beta)} : A \to B \quad \Gamma^{\overline{\gamma}_1} \vdash t^{\alpha_1} : A \quad \cdots \quad \Gamma^{\overline{\gamma}_k} \vdash t^{\alpha_k} : A}{\Gamma^{\sum_{j=0}^{k} \overline{\gamma}_j} \vdash (s)t^{\beta} : B}$$

Modèle relationnel

C'est une sémantique dénotationnelle

Théorème

Si $\Gamma \vdash s : B$ *et* $s =_\beta t$ *alors* $[\![s]\!] = [\![t]\!]$.

Cas principal de la démonstration.

Si $\Gamma, x : A \vdash s : B$ et $\Gamma \vdash t : A$, alors $[\![(\lambda x.s)t]\!] = [\![s[t/x]]\!]$. □

Modèle relationnel pour le λ-calcul pur

Idée

Si on résout l'équation $D = D \to D = \mathcal{M}_f(D) \times D$, on peut typer tous les λ-termes avec D.

Modèle relationnel pour le λ-calcul pur

Idée

Si on résout l'équation $D = D \to D = \mathcal{M}_f(D) \times D$, on peut typer tous les λ-termes avec D.

Il y a des solutions.

Modèle relationnel pour le λ-calcul pur

Idée

Si on résout l'équation $D = D \to D = \mathcal{M}_f(D) \times D$, on peut typer tous les λ-termes avec D.

Il y a des solutions.

Théorème (Normalisation)

- $[\![s]\!] \neq \emptyset$ *ssi* s *a une forme normale de tête*
- s *a une forme normale ssi il y a un élément de* $[\![s]\!]$ *qui « ne triche pas »*

On peut même parler de complexité.

On peut décomposer

$$\frac{\Gamma^{\overline{\gamma}_0} \vdash s^{([\alpha_1,\ldots,\alpha_k],\beta)} : A \to B \qquad \Gamma^{\overline{\gamma}_1} \vdash t^{\alpha_1} : A \qquad \cdots \qquad \Gamma^{\overline{\gamma}_k} \vdash t^{\alpha_k} : A}{\Gamma^{\sum_{j=0}^{k} \overline{\gamma}_j} \vdash (s)t^{\beta} : B}$$

en

$$\frac{\Gamma^{\overline{\gamma}_1} \vdash t^{\alpha_1} : A \qquad \cdots \qquad \Gamma^{\overline{\gamma}_k} \vdash t^{\alpha_k} : A}{\Gamma^{\sum_{j=1}^{k} \overline{\gamma}_j} \vdash t^{[\alpha_1,\ldots,\alpha_k]} : !A}$$

et

$$\frac{\Gamma^{\overline{\gamma}} \vdash s^{(\overline{\alpha},\beta)} : A \to B \qquad \Gamma^{\overline{\gamma}'} \vdash t^{\overline{\alpha}} : !A}{\Gamma^{\overline{\gamma}+\overline{\gamma}'} \vdash (s)t^{\beta} : B}$$

On peut décomposer

$$\frac{\Gamma^{\overline{\gamma}_0} \vdash s^{([\alpha_1,\ldots,\alpha_k],\beta)} : A \to B \qquad \Gamma^{\overline{\gamma}_1} \vdash t^{\alpha_1} : A \qquad \cdots \qquad \Gamma^{\overline{\gamma}_k} \vdash t^{\alpha_k} : A}{\Gamma^{\sum_{j=0}^{k} \overline{\gamma}_j} \vdash (s)t^{\beta} : B}$$

en

$$\frac{\Gamma^{\overline{\gamma}_1} \vdash t^{\alpha_1} : A \qquad \cdots \qquad \Gamma^{\overline{\gamma}_k} \vdash t^{\alpha_k} : A}{\Gamma^{\sum_{j=1}^{k} \overline{\gamma}_j} \vdash t^{[\alpha_1,\ldots,\alpha_k]} : !A}$$

et

$$\frac{\Gamma^{\overline{\gamma}} \vdash s^{(\overline{\alpha},\beta)} : A \to B \qquad \Gamma^{\overline{\gamma}'} \vdash t^{\overline{\alpha}} : !A}{\Gamma^{\overline{\gamma}+\overline{\gamma}'} \vdash (s)t^{\beta} : B}$$

On sépare la génération des ressources (modalité !) de la partie purement implicative :

$$A \to B = !A \multimap B$$

On peut décomposer

$$\frac{\Gamma^{\overline{\gamma}_0} \vdash s^{([\alpha_1,\ldots,\alpha_k],\beta)} : A \to B \qquad \Gamma^{\overline{\gamma}_1} \vdash t^{\alpha_1} : A \qquad \cdots \qquad \Gamma^{\overline{\gamma}_k} \vdash t^{\alpha_k} : A}{\Gamma^{\sum_{j=0}^{k} \overline{\gamma}_j} \vdash (s)t^{\beta} : B}$$

en

$$\frac{\Gamma^{\overline{\gamma}_1} \vdash t^{\alpha_1} : A \qquad \cdots \qquad \Gamma^{\overline{\gamma}_k} \vdash t^{\alpha_k} : A}{\Gamma^{\sum_{j=1}^{k} \overline{\gamma}_j} \vdash t^{[\alpha_1,\ldots,\alpha_k]} : !A}$$

et

$$\frac{\Gamma^{\overline{\gamma}} \vdash s^{(\overline{\alpha},\beta)} : A \to B \qquad \Gamma^{\overline{\gamma}'} \vdash t^{\overline{\alpha}} : !A}{\Gamma^{\overline{\gamma}+\overline{\gamma}'} \vdash (s)t^{\beta} : B}$$

On sépare la génération des ressources (modalité !) de la partie purement implicative :

$$A \to B = !A \multimap B$$

Nous voilà dans les années (19)80...

Sémantique quantitative

Un autre regard sur la « continuité » : fonctions analytiques.

On interprète les types/formules comme des espaces vectoriels et les termes/preuves par des séries entières : si $t : A \to B$ et $u : A$,

$$((t)u)_\beta = \sum_{\overline{a}} t_{(\overline{a},\beta)} \cdot u^{\overline{a}}$$

où $u^{[\alpha_1,\ldots,\alpha_n]} = \prod_i u_{\alpha_i}$.

Version quantitative de $\beta \in tu$ ssi $\exists \overline{a} \in \mathcal{M}_f(u),\ (\overline{a},\beta) \in t$.

Il faut un truc en plus pour que la somme converge :

- de gros coefficients (façon Girard, $\sim$1985) ;
- de la topologie (façon Ehrhard, $\sim$2000) ;
- ...

Quatrième partie

Pour aller plus loin

Au delà du calcul fonctionnel

Dans sa plus simple expression, Curry-Howard établit une correspondance entre programmation fonctionnelle pure et calcul propositionnel intuitionniste. Mais...

En programmation il y a

- des effets de bord (variables globales, références...)
- des structures de contrôle (exceptions, continuations...)
- des entrées-sorties

Dans le raisonnement mathématique, il y a

- des objets, des quantificateurs, des axiomes
- du raisonnement par l'absurde

Si la correspondance n'est pas fortuite, on doit pouvoir l'étendre en faisant le lien entre ces choses-là.

Au delà du calcul fonctionnel

Dans sa plus simple expression, Curry-Howard établit une correspondance entre programmation fonctionnelle pure et calcul propositionnel intuitionniste. Mais...

En programmation il y a

- des effets de bord (variables globales, références...)
- des structures de contrôle (exceptions, continuations...)
- des entrées-sorties

Dans le raisonnement mathématique, il y a

- des objets, des quantificateurs, des axiomes
- du raisonnement par l'absurde

Si la correspondance n'est pas fortuite, on doit pouvoir l'étendre en faisant le lien entre ces choses-là.
$\rightsquigarrow$ (par exemple) le programme de la *réalisabilité classique* de Krivine.

Et bien d'autres choses...

en particulier à l'I2M : *Logique de la Programmation*

- de nouveaux modèles de ZFC grâce à la réalisabilité classique
- des liens entre séries génératrices en combinatoire et développement de Taylor des λ-termes, grâce à la sémantique quantitative (à l'origine de la logique linéaire)
- Curry-Howard pour le calcul concurrent, *via* les réseaux de preuve (un sous-produit de la logique linéaire)
- une analyse des dialogues en linguistique *via* la Ludique (un autre sous-produit de la logique linéaire)
- ...