Plan

1. Le système F et le calcul des prédicats du second ordre
   - Les formules
   - Le typage et la réduction
   - AF_2, Lien avec le système F

2. La normalisation, l'expressivité.
   - Normalisation forte du système F
   - Quelques remarques sur les liens avec la cohérence
   - Les fonctions exprimables dans le système F

Types du système F

$F := X \mid \forall X F \mid F \to G$

$F \alpha G$:

* $F = X$ $F\alpha G \Leftrightarrow G = X$
* $F = A \to B$ $F\alpha G \Leftrightarrow G = A' \to B'$, $A\alpha A'$, $B\alpha B'$.
* $F = \forall X G$ $F\alpha G \Leftrightarrow G = \forall Y B$, $A < Z/X > \alpha B < Z/Y >$ pour "presque" tous les Z.

La définition de l'alpha-équivalence permet de définir une substitution "propre" sur les types, que l'on notera $A[B/X]$.

Formules en logique d'ordre 2

$$F := X \mid x \mid f(F, ..., F) \mid R(F, ...F) \mid F \Rightarrow G \mid \forall X F \mid \forall x F$$

$F \alpha G$:

* $F = X \; F \alpha G \Leftrightarrow G = X$
* $F = A \Rightarrow B \; F \alpha G \Leftrightarrow G = A' \Rightarrow B'$, $A \alpha A'$, $B \alpha B'$.
* $F = \forall (X, x) G \; F \alpha G \Leftrightarrow G = \forall (Y, y) B$, $A < (Z, z)/(X, x) > \alpha B < (Z, z)/(Y, y) >$ pour "presque" tous les Z.

La substitution

On définit une substitution de la façon suivante : On considère une formule F, une variable de relation X, et n variables x_1,..,x_n.

- Si A est une formule atomique de la forme $X(t1, ..., tn)$ alors $A[F/Xx_1, ..., x_n] = F[t_1/x_1, ..., t_n/x_n]$.
- si A est une autre formule atomique, $A[F/Xx_1, ..., x_n]$ est A.
- Si A est $B \to C$, $A[F/Xx_1, ..., x_n]$ est $B[F/Xx_1, ..., x_n] \to C[F/Xx_1, ..., x_n]$.
- Si A est $\forall X \; B$, $A[F/Xx_1, ..., x_n]$ est A
- Si A est $\forall \mu \; B$, $\mu \neq X$, $A[F/Xx_1, ..., x_n]$ est $\forall \mu \; B[F/Xx_1, ..., x_n]$.

Règles de typage du système F

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

$$\frac{\Gamma \vdash t : A, \; \Gamma \vdash u : A \to B}{\Gamma \vdash (u)(t) : B}$$

$$\frac{\Gamma \vdash t : A}{\Gamma \vdash t : \forall X.A}$$

Si X n'est libre dans aucun des types de Γ

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

$$\frac{\Gamma \vdash t : \forall X.A}{\Gamma \vdash t : A[B/X]}$$

Pour tout type B

Quelques propriétés des types

1. Si $\Gamma \vdash t : \forall X_i A$, alors $\Gamma \vdash t : [B_i/X_i]$.
2. Si $\Gamma, x : F[A_i/X_i] \vdash t : B$, alors $\Gamma, x : \forall X_i F \vdash t : B$
3. Si $\Gamma \vdash t : A$, et si $t \beta t'$, alors $\Gamma \vdash t' : A$.

Déduction en logique d'ordre 2

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

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B}$$

$$\frac{\Gamma \vdash A, \ \Gamma \vdash A \Rightarrow B}{\Gamma \vdash B}$$

$$\frac{\Gamma \vdash \forall x A}{\Gamma \vdash A[u/x]}$$

Pour tout terme u

$$\frac{\Gamma \vdash A}{\Gamma \vdash \forall x A}$$

Si x n'est pas libre dans Γ.

$$\frac{\Gamma \vdash \forall X.A}{\Gamma \vdash A[F/Xx_1 ... x_n]}$$

Pour tout F formule

$$\frac{\Gamma \vdash A}{\Gamma \vdash \forall X.A}$$

Si X n'est pas libre dans Γ

L'arithmétique Fonctionnelle

Principe : λ-calcul typé dont les Types sont des formules d'un langage du second ordre. Les règles de typage sont la traduction exacte des règles de déduction de la logique d'ordre 2.

Pour l'arithmétique fonctionnelle, l'isomorphisme de Curry-Howard est évident : un jugement de typage est exactement une démonstration intuitionniste.

Schéma de compréhension

(SC) $\forall X A \to A[F/X_{x_1 \ldots x_n}]$
où A et F formules quelconques,
X variable de relation n-aire,

$x_1 \ldots x_n$ variables d'individu

(SC') $\exists Y \forall x_1 \ldots \forall x_n [Y(x_1 \ldots x_n) \leftrightarrow F]$
où Y variable de relation n-aire,

F formule quelconque

(SC) $\Leftrightarrow$ (SC')

Retour au système F

Theorème. *Pour qu'un terme soit typable par AF_2, il faut et il suffit qu'il soit typable dans le système F.*

$A \to A^-$

- $X(t_1, ..., t_n) \to X$
- $(A \Rightarrow B) \to (A^- \to B^-)$
- $\forall x B \to B^-$
- $\forall X B \to \forall X B^-$

Une dérivation du type $x_1 : A_1, ..., x_n : A_n \vdash t : A$ dans AF_2 peut alors s'interpréter comme une dérivation de $x_1 : A_1^-, ..., x_n : A_n^- \vdash t : A^-$

Le théorème de normalisation forte

Tout terme typable dans le système F est fortement normalisable, c'est à dire :
Si $\Gamma \vdash u : F$, alors toutes les chaines de β-réductions issues se u terminent sur un même terme normal.

Candidat de réducibilité

Un ensemble R de termes de type U est un candidat de réducibilité s'il vérifie :

(P1) : Si $t \in R$, t est fortement normalisant.
(P2) : Si $t \in R$ et $t \to t'$ alors $t' \in R$.
(P3) : Si t n'est pas une abstraction et que toute réduction d'un rédex de t donne un terme $t' \in R$, alors $t \in R$.

Définition de RED

Si T type de variables libres $\underline{X}$, $\underline{U}$ séquence de types, $\underline{R}$ candidats de réducibilité pour ces types, on définit $RED_T[\underline{X} := \underline{R}]$, ensemble de termes de type $T[\underline{U}/\underline{X}]$ par :

- $RED_{X_i}[\underline{X} := \underline{R}] = R_i$
- $RED_{F \to G}[\underline{X} := \underline{R}] = \{t | \forall u \in F, tu \in G\}$
- $RED_{\forall Y F}[\underline{X} := \underline{R}] = \{t | \forall V, \forall S$ candidat de réducibilité de V, $tV \in RED_F[\underline{X} := \underline{R}, Y := S]\}$

Une percée fulgurante

Lemme 1. $RED_T[\underline{X} := \underline{R}]$ *est un candidat de réducibilité de type* T

Lemme 2. $RED_{T[V/Y]}[\underline{X} := \underline{R}] = RED_T[\underline{X} := \underline{R}, Y := RED_V[\underline{X} := \underline{R}]]$

Lemme 3. *Si pour tout* V*, et* S *candidat de réducibilité de* V*,* $u[Y := V] \in RED_W[\underline{X} := \underline{R}, Y := S]$*, alors* $u \in RED_{\forall Y W}[\underline{X} := \underline{R}]$

Lemme 4. *Si* $t \in RED_{\forall Y W}[\underline{X} := \underline{R}]$*, alors* $tV \in RED_{W[Y:=V]}[\underline{X} := \underline{R}]$ *pour tout type* V*.*

Objectif atteint

Nouvelle définition : t de type T est réductible si $t \in RED_T[\underline{X} := \underline{SN}]$ où SN_i est l'ensemble des termes fortement normalisants de type X_i.
Alors tous les termes typables en système F sont réductibles, donc fortement normalisants d'après (P1).

Interprétation des types

Définition.

- *Une partie* χ *de* Λ *est* ***saturée*** *si* $(u[t/x])t_1 \ldots t_n \in \chi \Rightarrow (\lambda xu)tt_1 \ldots t_n \in \chi$
- *Une interprétation est une application de l'ensemble des variables de types dans l'ensemble des parties saturées de* Λ
- *On définit l'interprétation* $|A|$ *d'un type* A *par induction :*

  1. *Si* A *est une variable de type, l'interprétation est déjà définie.*
  2. *Si* $A = B \to C$*,* $t \in |A| \Leftrightarrow (t)u \in |B| \forall u \in |C|$
  3. *Si* $A = \forall XB$*,* $t \in |A| \Leftrightarrow t \in |B|$ *quelle que soit l'interprétation de X.*

Les types de données.

Définition. *Un type A est dit* type de données *si*

1. *A est un type clos*
2. $|A| \neq \emptyset$
3. $\forall t \in |A| \; \exists \; c$, c *clos, tq* $t\beta c$.

Theorème. $\mathtt{Id} = \forall X(X \to X)$, $\mathtt{Bool} = \forall X(X \to (X \to X))$, $\mathtt{Int} = \forall X(X \to X) \to (X \to X)$ *sont des types de données.*

Codage des entiers et de l'arithmétique

Theorème. $t \in |\texttt{Ent}| \Leftrightarrow \exists$ *n tq t* $\simeq_\beta \lambda f, x f^n(x)$ *ou* $\lambda f f$.

Démonstration. Soit $t \in |\texttt{Ent}|$; f, x étant des variables non dans t. On définit $|X| = \{t$ tq $\exists k \geq 0$ tq $t \simeq_\beta (f)^k(x)\}$ $t \in |\texttt{Ent}|$ donc $t \in |(X \to X) \to (X \to X)|$ donc $(tf)x \in |X|$, et $(tf)x \simeq_\beta (f)^k(x)$ d'où $t \simeq_\beta \lambda f, x(f)^k(x)$. t a donc une forme normale t'. Il est facile de montrer que $t' = \lambda f, x(f)^k(x)$, ou $t' = \lambda f f$. $\square$

Les fonctions représentables dans le système F

Theorème. *On ne peut pas représenter l'ensemble des fonctions récursives dans le système F.*

Démonstration. Soit $t_1, ..., t_n$ une énumération des termes typés $\mathtt{Ent} \rightarrow \mathtt{Ent}$ dans le système F. Soit $\phi : n \rightarrow S((t_n)n)$. ϕ n'est pas représentable dans le système F. $\square$

Fonctions prouvablement totales

Définition. *Une fonction récursive est prouvablement totale si il existe un terme t(x) tel que* $\vdash \forall x(\mathtt{Ent}[x] \to \mathtt{Ent}[t(x)])$

Theorème. *Soit t, tel que* $\vdash t : \mathtt{Ent} \to \mathtt{Ent}$ *dans le système F, alors la fonction* ϕ *représentée par t est prouvablement totale dans une théorie contenant (SC)*

Theorème. *Soit* ϕ *une fonction prouvablement totale dans une théorie contenant (SC). Alors* $\exists t, \vdash t : \mathtt{Ent} \to \mathtt{Ent}$ *dans le système F, qui représente* ϕ