Le λ-calcul simplement typé avec équations récursives

Forte normalisation

René David et Karim Nour

Université de Savoie, Lama

Journées LAC du GDR IM

8-9 Février 2007

Outline

Outline

1. Introduction
   - Le λ-calcul simplement typé
   - Les équations récursives sur les types
2. La forte normalisation du calcul
   - Idée de la preuve
   - Plan de la preuve
3. Applications et questions ouvertes
   - Le $\lambda\mu$-calcul simplement typé
   - Un exemple
   - Questions ouvertes

Le λ-calcul simplement typé

- **Règles de typage**

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

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

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

- **Règle de réduction**

$$(\lambda x M\ N)\ \ \triangleright_\beta\ \ M[x := N]$$

Les propriétés du calcul

1. Confluence.
2. Préservation de type.
3. Forte normalisation.
4. Décidabilité de typage.
5. Caractérisation des fonctions représentables (Schwichtenberg).

Outline

Les équations récursives

Objectif : Écrire des équations de la forme $X = F[X]$.

Ceci revient à ajouter les règles de typage :

$$\frac{\Gamma \vdash M : X}{\Gamma \vdash M : F[X]}$$

$$\frac{\Gamma \vdash M : F[X]}{\Gamma \vdash M : X}$$

Attention :

- Avec $X = X \to Y$, on type $(\delta\ \delta)$.
- Avec $X = X \to X$, on type tous les termes.

Bonne équation récursive

C'est une équation $X = F[X]$ où X est positive dans F.

Exemples : $X = Y \to X$
$X = (X \to Y) \to Y$
$X = (X \to Y) \to (Y \to X)$

Généralisations :

- Autoriser plusieurs équations.
- Effectuer la substitution dans un sous-type.

Idée : Introduire une relation de congruence sur les types.

Définitions

Définition

Soit $X \in \mathcal{A}$

- $X \in \mathcal{T}^+(X)$.
- *Si* $Y \in \mathcal{A} - \{X\}$, *alors* $Y \in \mathcal{T}^+(X)$ *et* $Y \in \mathcal{T}^-(X)$.
- *Si* $U \in \mathcal{T}^-(X)$ *et* $V \in \mathcal{T}^+(X)$, *alors* $U \to V \in \mathcal{T}^+(X)$ *et* $V \to U \in \mathcal{T}^-(X)$.

Définition

Soit $\mathcal{X} \subseteq \mathcal{A}$

- $\mathcal{T}^+(\mathcal{X}) = \bigcap_{X \in \mathcal{X}} \mathcal{T}^+(X)$
- $\mathcal{T}^-(\mathcal{X}) = \bigcap_{X \in \mathcal{X}} \mathcal{T}^-(X)$

Bonne congruence sur les types

Définition

Soit $\mathcal{X} \subseteq \mathcal{A}$. Une relation de congruence $\approx$ sur les types $\mathcal{T}$ est dite $\mathcal{X}$-bonne ssi

1. *Si $X \in \mathcal{X} \cap var(U)$ et $U \approx V$, alors $X \in var(V)$.*
2. *Si $U \approx V$ et $var(U) \cap \mathcal{X} = \emptyset$, alors $U = V$.*
3. *Si $U_1 \to V_1 \approx U_2 \to V_2$, alors $U_1 \approx U_2$ et $V_1 \approx V_2$.*
4. *Si $X \in \mathcal{X}$, $U \approx V$ et $U \in \mathcal{T}^+(X)$, alors $V \in \mathcal{T}^+(X)$.*
5. *Si $X \in \mathcal{X}$, $U \approx V$ et $U \in \mathcal{T}^-(X)$, alors $V \in \mathcal{T}^-(X)$.*

Exemples

Soit $\mathcal{X} = \{X_1, X_2\}$.

- Soit $\approx$ la p.p. congruence contenant

$$X_1 \approx (X_1 \to (X_2 \to Y)) \to Y$$
$$X_2 \approx (X_2 \to (X_1 \to Y)) \to Y$$

- Soit $\approx$ la p.p. congruence contenant

$$X_1 \approx X_2 \to X_1$$
$$X_2 \approx X_1 \to X_2$$

Le système

Définition

Soient $\mathcal{X} \subseteq \mathcal{A}$ et $\approx$ une congruence $\mathcal{X}$-bonne sur les types. On ajoute la règle de typage :

$$\frac{\Gamma \vdash M : U \quad U \approx V}{\Gamma \vdash M : V}$$

Théorème

1. *Préservation de type.*
2. *Forte normalisation.*

Questions

La preuve de Mendler utilise des candidats de réductibilité avec des points fixes.

- Quelle est la puissance du système nécessaire pour formaliser cette preuve ?
- Peut-on trouver une preuve arithmétique de la forte normalisation ?

Outline

Idée de la preuve

Objectif 1. Si $M, N \in SN$, alors $M[x := N] \in SN$.

Par induction sur $type(N)$.

Sinon, $M = (x\ P)$ avec $(N\ P_1) \notin SN$ et $P_1 = P[x := N] \in SN$.

Alors $N \triangleright^* \lambda y N_1$ et $N_1[y = P_1] \notin SN$.

Si $type(N)$ est une flèche, alors $type(P_1) < type(N)$.

Donc la propriété est démontrée si $type(N) \cap \mathcal{X} = \emptyset$.

Il suffit de prouver la propriété pour $type(N) = X \in \mathcal{X}$.

Idée de la preuve (suite)

Objectif 2. Si $M, \sigma \in SN$ et $Im(\sigma) \subseteq \mathcal{T}^+$, alors $M[\sigma] \in SN$.

Par induction sur $\langle \eta(M), cxty(M) \rangle$.

On suppose que $M = (x\ P), \sigma(x) = N \triangleright^* \lambda y N_1$, $P_1 = P[\sigma] \in SN$ et $N_1[y = P_1] \notin SN$.

Alors il existe $(y\ N_2) \leq N_1$ tel que $(P_1\ N_2[y := P_1]) \notin SN$ et $N_2[y := P_1] \in SN$.

Donc P_1 doit se réduire à un λ.

Deux cas à étudier.

Idée de la preuve (suite)

1. $P \rhd^* \lambda x_1 M_1$ et $M_1[\sigma'] \notin SN$.
   Une contradiction car $\eta(M_1) < \eta(M)$.
2. $P \rhd^* (x' \overrightarrow{Q})$ avec $x' \in dom(\sigma)$.
   On obtient $type(P) = type(P_1) \in \mathcal{T}^+ \cap \mathcal{T}^-$,
   – $type(x') \in \mathcal{T}^+$
   – $M = (x\ P)$ et $type(x) \in \mathcal{T}^+$.
   Donc $type(P_1) \cap \mathcal{X} = \emptyset$ et $N_1[y = P_1] \in SN$.

Si y possède plusieurs arguments, on itére le même raisonnement.

Outline

Plan de la preuve

Lemme

Si $M, N \in SN$, $\Gamma, x : U \vdash M : V$, $\Gamma \vdash N : U$ *et* $var(U) \cap \mathcal{X} = \emptyset$, *alors* $M[x := N] \in SN$.

Lemme

Si $M, \sigma \in SN$, $dom(\sigma) = \{x_1, ..., x_n\}$, $Im(\sigma) = \{N_1, ..., N_n\}$, $\Gamma, (x_i : U_i)_i \vdash M : V$, $\Gamma \vdash N_i : U_i$ *et le* $U_i \in \mathcal{T}^+(\mathcal{X})$, *alors* $M[\sigma] \in SN$.

Plan de la preuve (suite)

Corollaire

Si $X \in \mathcal{X}$, $M, N \in SN$, $\Gamma, x : X \vdash M : V$, $\Gamma \vdash N : X$, *alors* $M[x := N] \in SN$.

Corollaire

Si $M, N \in SN$, $\Gamma, x : U \vdash M : V$, $\Gamma \vdash N : U$, *alors* $M[x := N] \in SN$.

Théorème

Tout terme typable est fortement normalisable.

Outline

1. Introduction
   - Le λ-calcul simplement typé
   - Les équations récursives sur les types
2. La forte normalisation du calcul
   - Idée de la preuve
   - Plan de la preuve
3. Applications et questions ouvertes
   - Le $\lambda\mu$-calcul simplement typé
   - Un exemple
   - Questions ouvertes

Le $\lambda\mu$-calcul simplement typé

Soit $\approx$ la p.p. congruence contenant $(X \approx \neg\neg X)_{X \in \mathcal{A}}$

Définition (Le λ-terme M_T du type $\neg\neg T \to T$)

$M_\perp = \lambda x\, (x\ I) \qquad M_X = I$

$M_{U \to V} = \lambda x \lambda y\, (M_V\ \lambda z(x\ \ \lambda t(z\ (t\ y))))$

Définition (Traduction du $\lambda\mu$-calcul dans le λ-calcul)

$x^* = x \qquad (\lambda x\, M)^* = \lambda x\, M^* \qquad (M\, N)^* = (M^*\, N^*)$

$(\alpha\, M)^* = (\alpha\, M^*) \qquad (\mu\alpha\, M)^* = (M_T\ \lambda\alpha\, M^*)$ *si* $\alpha : \neg T$

Résultats

Lemme

Si $\Gamma \vdash_{\lambda\mu} M : U$*, alors* $\Gamma \vdash_{\approx} M^* : U$*.*

Lemme

Si $M \triangleright_{\beta\mu} N$*, alors* $M^* \triangleright_{\beta}^{+} N^*$*.*

Théorème

Tout $\lambda\mu$-terme typable est fortement normalisable.

Outline

1. Introduction
   - Le λ-calcul simplement typé
   - Les équations récursives sur les types
2. La forte normalisation du calcul
   - Idée de la preuve
   - Plan de la preuve
3. Applications et questions ouvertes
   - Le $\lambda\mu$-calcul simplement typé
   - Un exemple
   - Questions ouvertes

Un exemple

$Nat = (X \to X) \to (X \to X)$ et $Bool = Y \to (Y \to Y)$

Soit $\approx$ la p.p. congruence contenant $X \approx (X \to Bool) \to Bool$

$Inf_n = \lambda x\, (x\, M\, \lambda y \mathbf{1}\, (M^{n-1}\, \lambda y \mathbf{0}))$
$M = \lambda x\, \lambda y\, (y\, x)$
$\mathbf{0} = \lambda x \lambda y\, y$ et $\mathbf{1} = \lambda x \lambda y\, x$

Lemme

- $\vdash Inf_n : Nat \to Bool$.
- $(Inf_n\, \widetilde{m}) \triangleright^*_\beta \mathbf{1}$ *si* $m \leq n$ *et* $\mathbf{0}$ *sinon.*

Outline

Questions ouvertes

1. Caractériser la classe des fonctions représentables.
2. Étudier les relations de congruence $\mathcal{X}$-bonnes.
3. Chercher des résultats de décidabilité.
4. Ajouter d'autres règles de typage et de réduction (par exemple, celles de M. Parigot).
5. Étendre le résultat à d'autres systèmes de typage (par exemple, le système D).