Internaliser l'Arithmétique de Presburger dans le Calcul des Constr. Congruentes Inductives

avec Frédéric Blanqui, Jean-Pierre Jouannaud

Pierre-Yves Strub

Project LogiCal, Pôle Commun de Recherche en Informatique du PLateau de Saclay,
CNRS, Ecole Polytechnique, INRIA, Université Paris-Sud

Février 2007

Le Calcul des Constructions Congruentes avec $\mathbb{N}$

- Motivations
- Définition du calcul
- La règle de conversion

Etude métathéorique (en accéléré)

- Cohérence
- Décidabilité

And next...

Le Calcul des Constructions Congruentes avec $\mathbb{N}$

Motivations

Définition du calcul

La règle de conversion

Etude métathéorique (en accéléré)

Cohérence

Décidabilité

And next...

Motivations (1/2)

Plus de preuves par réflexion

$$\begin{aligned}\Gamma = {} & [x\, y\, t :^{\mathrm{u}} \mathsf{nat}], [f :^{\mathrm{r}} \mathsf{nat} \to \mathsf{nat}],\\ & [p_1 :^{\mathrm{r}} t \doteq 2], [p_2 :^{\mathrm{r}} (f\,(x+3)) \doteq x + 2],\\ & [p_3 :^{\mathrm{r}} (f\,(y+t)) + 2 \doteq y], [p_4 :^{\mathrm{r}} y + 1 = x]\end{aligned}$$

Sous Γ, une preuve par *reflexivité* de $0 \doteq 0$ est également une preuve de $0 \doteq \mathsf{S}\, t$

Motivations (2/2)

Plus de termes typables

Par exemple : *si l_1 et l_2 sont des palindromes, alors $l_1 l_2 l_2 l_1$ est un palindrome* avec des listes dépendantes.

- $\mathrm{reverse} : \forall(n : \mathsf{nat}), \mathsf{list}\, n \to \mathsf{list}\, n$
- $isAPal := \lambda[n : \mathsf{nat}, l : \mathsf{list}\, n].\, reverse(l) \doteq l$

On veut prouver : $\mathrm{reverse}(l_1 l_2 l_2 l_1) = l_1 l_2 l_2 l_1$ en utilisant $\mathrm{reverse}(ll') = \mathrm{reverse}(l')\, \mathrm{reverse}(l)$... qui n'est pas typable.

- $\mathrm{reverse}(ll') : \ldots \mathsf{list}(n + n')$
- $\mathrm{reverse}(l')\, \mathrm{reverse}(l) : \ldots \mathsf{list}(n' + n)$

Motivations (2/2)

Plus de termes typables

Par exemple : *si* l_1 *et* l_2 *sont des palindromes, alors* $l_1 l_2 l_2 l_1$ *est un palindrome* avec des listes dépendantes.

- $\mathrm{reverse} : \forall(n : \mathsf{nat}), \mathsf{list}\, n \rightarrow \mathsf{list}\, n$
- $isAPal := \lambda[n : \mathsf{nat}, l : \mathsf{list}\, n].\, reverse(l) \doteq l$

On veut prouver : $\mathrm{reverse}(l_1 l_2 l_2 l_1) = l_1 l_2 l_2 l_1$ en utilisant $\mathrm{reverse}(ll') = \mathrm{reverse}(l')\,\mathrm{reverse}(l)$... qui n'est pas typable.

- $\mathrm{reverse}(ll') : \ldots \mathsf{list}(n + n')$
- $\mathrm{reverse}(l')\,\mathrm{reverse}(l) : \ldots \mathsf{list}(n' + n)$

Definition (Notations)

- Les sortes usuelles du calcul des constructions $\mathcal{S} = \{\star, \square\, \triangle\}$.
- Un ensemble de *variables* $\mathcal{X}^\star$ (resp. $\mathcal{X}^\square$) de sorte $\star$ (resp. $\square$) $\mathcal{X} = \mathcal{X}^\star \uplus \mathcal{X}^\square$.
- Deux annotations *mystères* $\mathcal{A} = \{\mathrm{r}, \mathrm{u}\}$, ordonnées par $\mathrm{r} \prec_{\mathcal{A}} \mathrm{u}$.

Definition

On note $\mathcal{E}$ la théorie de Presburger.

Definition (Notations)

- Les sortes usuelles du calcul des constructions $\mathcal{S} = \{\star, \square\, \triangle\}$.
- Un ensemble de *variables* $\mathcal{X}^\star$ (resp. $\mathcal{X}^\square$) de sorte $\star$ (resp. $\square$) $\mathcal{X} = \mathcal{X}^\star \uplus \mathcal{X}^\square$.
- Deux annotations *mystères* $\mathcal{A} = \{\mathrm{r}, \mathrm{u}\}$, ordonnées par $\mathrm{r} \prec_{\mathcal{A}} \mathrm{u}$.

Definition

On note $\mathcal{E}$ la théorie de Presburger.

Definition (Pseudo-termes)

Les termes usuels de CC, plus ce qu'il faut pour faire de l'arithmétique linéaire :

$$\begin{array}{rcl} t \in \mathcal{T} & ::= & s \in \mathcal{S} \mid f \in \mathcal{F} \mid x \in \mathcal{X} \mid tt \mid (\forall x : t)t \mid [\lambda x : t]t \\ & \mid & \mathsf{nat} \mid \doteq \mid \mathsf{0} \mid \mathsf{S} \mid + \mid \mathrm{Eq}(t) \mid \mathsf{Rec}_{\mathbb{N}}^{\mathcal{W}}(t, T)\{t_0, t_S\} \end{array}$$

Definition (Règles de typage (1/2))

Les règles usuelles de CC, ainsi que

Des règles pour nat

$$[\text{0-Intro}]\ \frac{}{\vdash 0 : \mathsf{nat}} \qquad [\text{S-Intro}]\ \frac{}{\vdash \mathsf{S} : \mathsf{nat} \to \mathsf{nat}}$$

$$[\text{Eq-Intro}]\ \frac{\Gamma \vdash t_1 : T \quad \Gamma \vdash t_2 : T \qquad \Gamma \vdash p : \forall(P : T \to \star).\, P\, t_1 \to P\, t_2}{\Gamma \vdash \mathrm{Eq}(p) : t_1 \doteq_T t_2}$$

$$[\iota\text{-Elim}]\ \frac{\Gamma \vdash t : \mathsf{nat} \quad \Gamma \vdash Q : \mathsf{nat} \to s \in \star \quad \Gamma \vdash f_0 : \mathsf{nat} \qquad \Gamma \vdash f_S : \forall(n :^u \mathsf{nat}).\, Q\, n \to Q\,(\mathsf{S}\, n)}{\Gamma \vdash \mathsf{Rec}_{\mathbb{N}}^{\mathcal{W}}(t, Q)\{f_0, f_S\} : Q\, t}$$

Definition (Règles de typage (2/2))

Les règles usuelles de CC, ainsi que

Une nouvelle règle de conversion

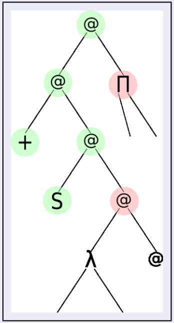
$$[\textsc{Conv}]\ \frac{\Gamma \vdash t : T \quad \Gamma \vdash T' : s' \quad T \sim_\Gamma T'}{\Gamma \vdash t : T'}$$

La règle de conversion est dépendante du contexte via la relation $\sim_\Gamma$ (qui reste à définir !)

Définition de $\sim_\Gamma$ (Essai 1)

Definition

On note $\mathcal{Y}$ un nouvel ensemble de variables fraiches.



Les positions en vert font parties du *chapeau*.
Les position en rouge sont les positions *alien*.

$\mathrm{APos}(t) \rightarrow$ positions *alien* de t.
$\mathrm{CPos}(t) \rightarrow$ positions *chapeau* de t.

$\pi(t) \rightarrow$ fonction qui choisit un représentant dans $\mathcal{Y}$ pour les classes de $\mathcal{T}$ modulo α-équivalence.

$$\mathrm{cap}(t) = t[p \leftarrow \pi(t_{\downarrow p})]_{p \in \mathrm{APos}(t)}.$$

Définition de $\sim_\Gamma$ (Essai 1)

On peut désormais définir $\sim_\Gamma$...

Definition

$u \sim_\Gamma v$ si :

- $\mathcal{E} \models E_\Gamma \Rightarrow \mathrm{cap}(u') = \mathrm{cap}(v')$
- Avec $E_\Gamma = \{\mathrm{cap}(u) = \mathrm{cap}(v) \mid [x : u \doteq v] \in \Gamma\}$
- et $u \to_\beta^* u'$ et $v \to_\beta^* v'$

Oui mais...

- $\Gamma = x, y : \text{nat}, p : T[x], \ldots$
- $\Gamma \vdash [\lambda q : x \doteq y]p : (\forall q : x \doteq y)T[y]$

Si de plus on peut prouver $x = y$ sous Γ : $\Gamma \vdash w : x \doteq y$

Alors : $\Gamma \vdash ([\lambda q : x \doteq y]p)\, w : T[y]$

Soit, après conversion : $\Gamma \vdash p : T[y]$.

Mais $T[x]$ et $T[y]$ ne sont pas a priori convertibles sous Γ.

En fait, le calcul ne possède pas la propriété
de substitutivité car des équations peuvent disparaitre.

Oui mais...

- $\Gamma = x, y : \text{nat}, p : T[x], \ldots$
- $\Gamma \vdash [\lambda q : x \doteq y]p : (\forall q : x \doteq y)T[y]$

Si de plus on peut prouver $x = y$ sous Γ : $\Gamma \vdash w : x \doteq y$

Alors : $\Gamma \vdash ([\lambda q : x \doteq y]p)\, w : T[y]$

Soit, après conversion : $\Gamma \vdash p : T[y]$.

Mais $T[x]$ et $T[y]$ ne sont pas a priori convertibles sous Γ.

En fait, le calcul ne possède pas la propriété
de substitutivité car des équations peuvent disparaitre.

Définition de $\sim_\Gamma$ (Essai 2)

Les annotations *mystères* entrent en jeu.

Les termes sont modifiés comme suit

$$t \in \mathcal{T} ::= s \in \mathcal{S} \mid f \in \mathcal{F} \mid x \in \mathcal{X} \mid tt \mid (\forall x :^{a} t)t \mid [\lambda x :^{a} t]t \qquad (a \in \mathcal{A})$$

Les environnements sont modifiés comme suit

$$\Gamma := \epsilon \mid [x :^{a} t]\,\Gamma \qquad (a \in \mathcal{A})$$

Définition de $\sim_\Gamma$ (Essai 2)

La règle APP de CC est modifiée :

$$[\text{APP}]\ \frac{\Gamma \vdash t : (\forall x :^a U)V \quad \Gamma \vdash u : U}{\Gamma \vdash tu : V\{x \mapsto u\}}$$

avec la restriction que si i) $a = \mathrm{r}$ et ii) $U \equiv v_1 \doteq v_2$ alors $v_1 \sim_\Gamma v_2$.

Definition

$u \sim_\Gamma v$ si :

- $\mathcal{E} \models E_\Gamma \Rightarrow \mathrm{cap}(u') = \mathrm{cap}(v')$
- Avec $E_\Gamma = \{\mathrm{cap}(u) = \mathrm{cap}(v) \mid [x :^{\mathrm{r}} u \doteq v] \in \Gamma\}$
- et $u \to_\beta^* u'$ et $v \to_\beta^* v'$

Définition de $\sim_\Gamma$ (Essai 2)

La règle APP de CC est modifiée :

$$[\text{APP}]\ \frac{\Gamma \vdash t : (\forall x :^{a} U)V \quad \Gamma \vdash u : U}{\Gamma \vdash tu : V\{x \mapsto u\}}$$

avec la restriction que si i) $a = \mathrm{r}$ et ii) $U \equiv v_1 \doteq v_2$ alors $v_1 \sim_\Gamma v_2$.

Definition

$u \sim_\Gamma v$ si :

- $\mathcal{E} \vDash E_\Gamma \Rightarrow \mathrm{cap}(u') = \mathrm{cap}(v')$
- Avec $E_\Gamma = \{\mathrm{cap}(u) = \mathrm{cap}(v) \mid [x :^{\mathrm{r}} u \doteq v] \in \Gamma\}$
- et $u \rightarrow^*_\beta u'$ et $v \rightarrow^*_\beta v'$

Oui, mais, il y a un mais...

Des équations peuvent encore disparaitre, notamment par conversion à cause de β.

E.g. $([\lambda x :^{\mathrm{u}} \_]x)e$ où e est une équation.

Il est donc encore possible d'avoir des termes possédant 2 types non convertibles.

Définition de $\sim_\Gamma$ (Au final)

Intuitivement, il faut que l'ensemble des équations restent stables...

$\Gamma \vdash \circ : (\forall x :^{\mathrm{r}} T)u \rightsquigarrow \Gamma[x :^{\mathrm{r}} T] \vdash \circ : u$

$\sim_\Gamma$ doit anticiper que le typage de $(\forall x :^{\mathrm{r}} T)u$ va faire "apparaitre" une équation lors du typage de u.

i.e. si $u \sim_{\Gamma[x:^{\mathrm{r}}T]} v$ alors $(\forall x :^{\mathrm{r}} T)u \sim_\Gamma (\forall x :^{\mathrm{r}} T)v$

Les équations doivent être extraites modulo $\sim_{[\cdot]}$ et non plus syntaxiquement.

i.e. si $T \sim_{\Gamma_1} u \doteq v$ alors $u = v$ est une équation de $\Gamma_1, [x :^{\mathrm{r}} T], \Gamma_2$.

Définition de $\sim_\Gamma$ (Au final)

Intuitivement, il faut que l'ensemble des équations restent stables...

$\Gamma \vdash \circ : (\forall x :^{\mathrm{r}} T)u \rightsquigarrow \Gamma[x :^{\mathrm{r}} T] \vdash \circ : u$

$\sim_\Gamma$ doit anticiper que le typage de $(\forall x :^{\mathrm{r}} T)u$ va faire “apparaitre” une équation lors du typage de u.

i.e. si $u \sim_{\Gamma[x:^{\mathrm{r}}T]} v$ alors $(\forall x :^{\mathrm{r}} T)u \sim_\Gamma (\forall x :^{\mathrm{r}} T)v$

Les équations doivent être extraites modulo $\sim_{[\cdot]}$ et non plus syntaxiquement.

i.e. si $T \sim_{\Gamma_1} u \doteq v$ alors $u = v$ est une équation de $\Gamma_1, [x :^{\mathrm{r}} T], \Gamma_2$.

Définition de $\sim_\Gamma$ (Au final)

Intuitivement, il faut que l'ensemble des équations restent stables...

$\Gamma \vdash \circ : (\forall x :^{\mathrm{r}} T)u \rightsquigarrow \Gamma[x :^{\mathrm{r}} T] \vdash \circ : u$

$\sim_\Gamma$ doit anticiper que le typage de $(\forall x :^{\mathrm{r}} T)u$ va faire "apparaitre" une équation lors du typage de u.

i.e. si $u \sim_{\Gamma[x:^{\mathrm{r}}T]} v$ alors $(\forall x :^{\mathrm{r}} T)u \sim_\Gamma (\forall x :^{\mathrm{r}} T)v$

Les équations doivent être extraites modulo $\sim_{[\cdot]}$ et non plus syntaxiquement.

i.e. si $T \sim_{\Gamma_1} u \doteq v$ alors $u = v$ est une équation de $\Gamma_1, [x :^{\mathrm{r}} T], \Gamma_2$.

Définition de $\sim_\Gamma$ (Au final)

Intuitivement, il faut que l'ensemble des équations restent stables...

$\pi(\cdot)$ ne doit pas choisir un représentant
modulo α-équivalence, mais modulo $\sim_{[\cdot]}$.

- $\pi(\cdot)$ devient paramétré par une relation d'équivalence $\mathcal{R}$
  Il choisit un représentant dans $\mathcal{Y}$ pour les classes modulo $\mathcal{R}$.
- $\mathrm{cap}(\cdot)$ devient paramétré lui aussi par une relation d'équivalance $\mathcal{R}$
  $\mathrm{cap}_{\mathcal{R}}(t) = t[p \leftarrow \pi_{\mathcal{R}}(t_{\downarrow p})]_{p \in \mathrm{APos}(t)}$.

Intuitivement, on va commencer avec $\mathcal{R} = (=_{\beta\iota})$ et *saturer* $\mathcal{R}$ avec toutes les équations de l'environnement.

Définition de $\sim_\Gamma$ (Au final)

$\sim_\Gamma$ est défini par un système d'inférence

Definition ($\sim_\Gamma$)

$$[\beta\iota]\ \frac{t =_{\beta\iota} u}{t \sim_\Gamma u} \qquad [\textsc{Eq}]\ \frac{\Gamma = \Gamma_1, [z :^{\mathrm{r}} W], \Gamma_2 \quad W \sim_{\Gamma_1} u \doteq v}{u \sim_\Gamma v}$$

$$[\textsc{Ded}]\ \frac{\mathcal{E}, \{\mathrm{cap}_{\sim_\Gamma}(u) = \mathrm{cap}_{\sim_\Gamma}(v) \,|\, u \sim_\Gamma v\} \vDash \mathrm{cap}_{\sim_\Gamma}(s) = \mathrm{cap}_{\sim_\Gamma}(t)}{s \sim_\Gamma t}$$

$$[\textsc{Prod}]\ \frac{t_1 \sim_\Gamma u_1 \quad t_2 \sim_{\Gamma, [x :^a t_1]} u_2}{(\forall x :^b t_1) t_2 \sim_\Gamma (\forall x :^b u_1) u_2}\ [x \notin \mathrm{dom}(\Gamma), a \preceq b]$$

+ la même règle pour l'abstraction

Définition de $\sim_\Gamma$ (Au final)

$\sim_\Gamma$ est défini par un système d'inférence

Plus les règles :

- de transitivité
- de compatibilité par contexte

Reste à donner la version finale de la règle APP

$$[\text{APP}]\ \frac{\Gamma \vdash t : (\forall x :^a U)V \quad \Gamma \vdash u : U}{\Gamma \vdash tu : V\{x \mapsto u\}}$$

avec la restriction que si i) $a = \mathrm{r}$ et ii) $U \sim_\Gamma v_1 \doteq v_2$

Définition de $\sim_\Gamma$ (Au final)

$\sim_\Gamma$ est défini par un système d'inférence

Plus les règles :

- de transitivité
- de compatibilité par contexte

Reste à donner la version finale de la règle App

$$[\textsc{App}]\ \frac{\Gamma \vdash t : (\forall x :^a U)V \quad \Gamma \vdash u : U}{\Gamma \vdash tu : V\{x \mapsto u\}}$$

avec la restriction que si i) $a = \mathrm{r}$ et ii) $U \sim_\Gamma v_1 \doteq v_2$

Schéma de la preuve de cohérence

- Les propriétés métathéoriques de base
  - Substitutivité
  - Inversion
  - Subject reduction
  - ...
- La normalisation de $\beta\iota$ par la proof irrevelance de Barthe[1]

[1] *The relevance of proof irrelevance*, Proceedings of the 25th International Colloquium on Automata, Languages and Programming

Le Calcul des Constructions Congruentes avec $\mathbb{N}$
- Motivations
- Définition du calcul
- La règle de conversion

Etude métathéorique (en accéléré)
- Cohérence
- Décidabilité

And next...

Idée générale

- Test de satisfiabilité d'un ensemble de littéraux :

  $t \sim_\Gamma u$ si $\mathrm{eq}(\Gamma) \cup \{t \neq u\}$ est insatisfiable où
  $\mathrm{eq}(\Gamma) = \{w_1 = w_2 \mid [x :^{\mathrm{r}} w_1 \doteq w_2] \in \Gamma\}$.

- Purification des littéraux
- Saturation à la Nelson-Oppen

$[E, N, A]$

- E : un ensemble de littéraux
- A : littéraux algébriques purs
- N : système de récriture de la forme $x \mapsto t$ où t est non-algébrique et de profondeur au plus 1

Idée générale

- Test de satisfiabilité d'un ensemble de littéraux :

  $t \sim_\Gamma u$ si $\mathrm{eq}(\Gamma) \cup \{t \neq u\}$ est insatisfiable où
  $\mathrm{eq}(\Gamma) = \{w_1 = w_2 \mid [x :^{\mathrm{r}} w_1 \doteq w_2] \in \Gamma\}$.

- Purification des littéraux
- Saturation à la Nelson-Oppen

$[E, N, A]$

- E : un ensemble de littéraux
- A : littéraux algébriques purs
- N : système de récriture de la forme $x \mapsto t$ où t est non-algébrique et de profondeur au plus 1

Décider $\sim_\Gamma$ - Purification (1/4)

$$\frac{[E \uplus \{c_1 \bowtie_C c_2\}, A, N]}{[E, A \cup \{c_1 \bowtie_C c_2\}, N]}$$

$$\frac{[E \uplus \{t \bowtie_C C_{\mathcal{A}}[u_1, \ldots, u_n]\}, A, N] \quad c, c_1, \ldots, c_n \text{ fresh}}{[E \cup \{t \bowtie_C c, u_i =_C c_i\}, A \cup \{c =_C C_{\mathcal{A}}[c_1, \ldots, c_n]\}, N]}$$

where | $C_{\mathcal{A}}$ is an algebraic context

$$\frac{[E \uplus \{t \bowtie_C u_1\, u_2\}, A, N] \quad u_1\, u_2 \notin \textbf{Alg} \quad c, c_1, c_2 \text{ fresh}}{[E \cup \{t \bowtie_C c, c_i =_C u_i\}, A, N \cup \{c \rightarrow_C c_1\, c_2\}]}$$

$$\frac{[E \uplus \{t \bowtie_C \lambda[x :^a T]t\}, A, N] \quad c, c_T, c_t \text{ fresh}}{[E \cup E', A, N \cup \{c \rightarrow_C \lambda[x :^a c_T]c_t\}]}$$

where | $C' =_C c_T \quad E' = \{t \bowtie_C c, c_T =_C T, c_t =_{C'} t\}$

Décider $\sim_\Gamma$ (2/4)

$$\frac{[A, N]}{[A, N\{c_1 \rightarrow_C c_2\} \cup \{c_1 \rightarrow_C N(c_2)\}]}$$

where $\left| \begin{array}{ll} A_{|C} \vDash_{\mathcal{T}} c_1 = c_2 & (c_1 \rightarrow_C N(c_2)) \notin N \\ c_1, c_2 \in \mathrm{Var}(A) \cup \mathrm{Var}(N) & c_1 \succ_{\mathcal{C}} N(c_2) \end{array} \right.$

$$\frac{[A, N \supseteq \{c_1 \rightarrow_C t_1, c_2 \rightarrow_C t_2\}]}{[A \cup \{c_1 =_C c_2\}, N]}$$

where $\left| \begin{array}{l} c_1 =_C c_2 \notin A \quad A_{|C} \text{ is } \mathcal{T}\text{-satisfiable} \\ N \vdash \{t_1 \equiv_C t_2\} \Rightarrow^* \top \end{array} \right.$

$$\frac{[A, N \supseteq \{c_1 \rightarrow_C t_1, c_2 \rightarrow_C t_2\}]}{[A \cup \{c_1 =_C c_2\}, N]}$$

where $\left| \begin{array}{l} c_1 =_C c_2 \notin A \quad A_{|C} \text{ is } \mathcal{T}\text{-unsatisfiable} \\ ||t_1|| =_\beta ||t_2|| \end{array} \right.$

Décider $\sim_\Gamma$ (2/4)

$$\frac{[A, N \uplus \{c \to_C \mathsf{Rec}_{\mathbb{N}}^{\mathcal{W}}(c_\iota, c_Q)\{c_0, c_S\}\}]}{[A', N']}$$

where $\left| \begin{array}{l} A_{|C} \text{ is satisfiable} \\ A_{|C} \wedge c_\iota \neq 0 \text{ is unsatisfiable} \\ (A', N') = \text{purify}(\{c =_C c_0\}, A, N) \end{array} \right.$

$$\frac{[A, N \supseteq \{c \to_C (d_1 \doteq d_2)\}]}{[A \cup \{d_1 =_C d_2\}, N]}$$

where $\left| \; d_1 =_C d_2 \notin A \right.$

$$\frac{[A, N \uplus \{c \to_C A\, B, A \to_C \lambda[x :^a T]D\}]}{[A\{x \mapsto B\}, N\{x \mapsto B\} \cup \{c \to_C B\}]} \qquad \frac{[A, N]}{\perp} \; [A_{|\emptyset} \text{ is } \mathcal{T}\text{-unsat.}]$$

Décider $\sim_\Gamma$ - Matching (4/4)

$$\frac{N \vdash E \uplus \{d \equiv_C d\}}{N \vdash E} \qquad \frac{N \vdash \emptyset}{\top} \qquad \frac{N \vdash E \uplus \{d_1 \equiv_C d_2\}}{N \vdash E \cup \{N(d_1) \equiv_C N(d_2)\}}$$

where $\mid$ $N(d_1) \neq d_1$ or $N(d_2) \neq d_2$

$$\frac{N \vdash E \uplus \{A\,B \equiv_C A'\,B'\}}{N \vdash E \cup \{A \equiv_C B, A' \equiv_C B'\}}$$

$$\frac{N \vdash E \uplus \{\lambda[x :^a T_1]D_1 \equiv_C \lambda[x :^a T_2]D_2\}}{N \vdash E\{x \mapsto y\} \cup \{T_1 \equiv_C T_2, D_1 \equiv_{C,T_1} D_2\}}$$

And next...

- Les récurseurs forts
- Extraire plus d'information
- Utiliser des procédures de décision plus puissantes, autres théories.