Logique du premier ordre
3ème leçon

Jean-Pierre Jouannaud
École Polytechnique
91400 Palaiseau, France

email: jouannaud@lix.polytechnique.fr
http: //w^3.lix.polytechnique.fr/Labo/Jean-Pierre.Jouannaud

Project LogiCal, Pôle Commun de Recherche en Informatique du Plateau de Saclay, CNRS, École Polytechnique, INRIA, Université Paris-Sud.

March 7, 2005

Outline

Feuille de route

- Problèmes d'unification
- Formes résolues
- Régles d'unification
- Correction des règles
- Terminaison des règles
- Complétude des règles

Treillis des termes

Les termes avec variables servent à représenter l'ensemble de toutes leurs *instances closes*. Par exemple, si les entiers naturels sont représentés en unaire à l'aide des symboles 0 et s (successeur), le terme $s(x)$ représente l'ensemble infini $\{s(0), s(s(0)), \ldots\}$. Certaines opérations sur les termes sont alors fondamentales. Par exemple, l'intersection des ensembles d'instances de deux termes est obtenue grâce à l'*unification*. L'opération d'unification correspond également à la borne supérieure de deux termes (modulo similarité) pour le préordre d'inclusion des instances.

Problèmes d'unification

Definition

Un *problème d'unification* P est ou bien la formule $\bot$ ou bien une formule

$$t_1 = u_1 \wedge \ldots \wedge t_n = u_n$$

où $t_1, \ldots, t_n, u_1, \ldots, u_n$ sont des termes. Le signe "$=$" est supposé symétrique, et P est par convention la formule $\top$ si $n = 0$.
$\mathcal{V}ar(P)$ désigne l'ensemble des variables de P.
On supposera qu'il existe une constante dans $\mathcal{F}$.

Solution

Definition

Étant donné un problème d'unification
$t_1 = u_1 \wedge \ldots \wedge t_n = u_n$,

- une *solution* est une substitution close σ telle que $\sigma(t_i) = \sigma(u_i)$ pour tout i ;
- un *unificateur* est une substitution σ telle que toute substitution close soit solution du problme d'unification
  $t_1\sigma = u_1\sigma \wedge \ldots \wedge t_i\sigma = u_i\sigma$.

Toute substitution close est solution de $\top$ et aucune substitution close n'est solution de $\bot$.
Deux problèmes d'unification sont *équivalents* s'ils ont mêmes ensembles de solutions.

Unificateur

On distingue donc les solutions d'un problème d'unification des unificateurs qui les représentent : un unificateur sert à décrire (en utilisant des variables) des ensembles potentiellement infinis de solutions.

Exercice : Quels sont

1. Les solutions du problème d'unification $f(x, g(x)) = f(h(y, z), y)$?
2. Les solutions du problème d'unification $f(x, g(x)) = f(y, x)$?
3. Les solutions du problème d'unification $f(x, g(x)) = f(x, y)$?
4. Les unificateurs du problème d'unification $f(x, g(x)) = f(x, y)$?

Unificateur principal

Definition

On dit que le terme s est plus général que le terme t, noté $s \preceq t$, s'il existe une substitution λ telle que $t = s\lambda$.

Lemma

$\preceq$ est un préordre bien-fondé dont l'équivalence $\stackrel{\bullet}{=}$, est le renommage de variables ou similarité.

Definition

Étant donné un problème d'unification P, on dit que l'unificateur σ est plus général que l'unificateur τ, noté $\sigma \preceq \tau$ s'il existe une substitution λ telle que $\sigma = \tau\lambda$.

Unicité de l'unificateur principal

Definition

Un unificateur de P est dit *principal* (ou plus général) s'il est minimal pour l'ordre $\underline{\preccurlyeq}$.

Lemma

$\underline{\preccurlyeq}$ est un préordre bien-fondé dont l'équivalence $\underline{\overset{\bullet}{=}}$, est le renommage de variables ou similarité. Si σ et τ sont deux substitutions principales de Γ, alors elles sont similaires, c'est-à-dire s'échangent par renommage de leur variables.

L'unificateur principal de deux termes, lorsqu'il existe, est donc unique modulo $\overset{\bullet}{=}$.

Formes résolues

Les *formes résolues* sont des problèmes d'unification simples pour lesquels le calcul des solutions comme des unificateurs (qui sont des ensembles non-vides) est immédiat.

Definition

Un problème d'unification est une *arbre-forme résolue* (ou forme résolue tout court) si c'est $\bot$ ou $\top$ ou s'il peut s'écrire:

$$x_1 = t_1 \wedge \ldots \wedge x_n = t_n$$

où $x_1, \ldots, x_n$ sont des variables distinctes qui n'ont pas d'autre occurrence dans le problème ($\bot$ est une abbréviation pour le cas $n = 0$).

Lemma

Les solutions d'une arbre-forme résolue $x_1 = t_1 \wedge \ldots \wedge x_n = t_n$ *sont les instances closes d'une substitution* $\sigma = \{x_1 \rightarrow t_1 \ldots, x_n \rightarrow t_n\}$ *qui en est l'*unificateur principal*.*

On vérifie que σ est un unificateur, car $x_i\sigma = t_i$ et $t_i\sigma = t_i$ d'aprèss la condition sur les variables. Si θ est un unificateur arbitraire, $x_i\theta = t_i\theta = (x_i\sigma)\theta$, d'où $\theta = \sigma\theta$, ce qui montre que θ est une instance de σ qui est donc principale. Comme toute solution est un unificateur, et toute instance close d'un unificateur est une solution, on en déduit le résultat

Formes dag-résolues

Definition

Un problème d'unification est une DAG-forme résolue si c'est $\top$, $\bot$ ou s'il peut s'écrire:

$$x_1 = t_1 \wedge \ldots \wedge x_n = t_n$$

où $x_1, \ldots, x_n$ sont des variables distinctes telles que, pour tout $i \leq j$, $x_i \notin \mathit{Var}(t_j)$.

Lemma

Si $x_1 = t_1 \wedge \ldots \wedge x_n = t_n$ est une DAG-forme résolue équivalente à $s = t$, alors $\sigma = \sigma_1\sigma_2 \ldots \sigma_n$, où $\sigma_i = \{x_i \rightarrow t_i\}_{i \in [1..n]}$, est un plus général unificateur de s et t.

Preuve

$x_i\sigma = x_i\sigma_i \ldots \sigma_n = t_i\sigma_{i+1} \ldots \sigma_n = t_i\sigma$ par la condition sur les variables, donc σ unifie.
On montre par récurrence sur n que σ est principale parmi l'ensemble des unificateurs. Le cas $n = 0$ est trivial. Si $n > 0$, $\tau = \sigma_1\sigma_2 \ldots \sigma_{n-1}$ est unificateur principal de la forme résolue $P' = x_1 = t_1 \wedge \ldots \wedge x_{n-1} = t_{n-1}$. Mais tout unificateur γ de $s = t$ est en particulier unificateur de P', et donc $\gamma = \tau\theta$. Comme γ unifie l'équation $x_n = t_n$, on a $x_n\tau\theta = t_n\tau\theta$, d'où $x_n\theta = t_n\theta$ par la condition sur les variables de la forme arbre-résolue, et donc θ est une instance de la substitution $\{x_n \mapsto t_n\}$ qui est unificateur principal de cette équation par le lemme 9.

Exercices

Question : où a été utilisée l'hypothèse d'existence d'une constante ?

Les formes arbre-résolues permettent de représenter un plus général unificateur de façon compacte :

Exercice : donner les solutions et les unificateurs en forme arbre-résolue, et les unificateurs en forme dag-résolue du problème d'unification suivant:

$$f(x_1, f(x_2, f(x_3, x_4)))$$
$$=$$
$$f(f(x_2, x_2), f(f(x_3, x_3), f(f(x_4, x_4), f(a, a))))$$

Régles d'unification

Identité : $s = s \longrightarrow \top$

Decompose :

$$f(s_1, \ldots, s_n) = f(t_1, \ldots, t_n) \longrightarrow s_1 = t_1 \wedge \ldots \wedge s_n = t_n$$

Conflit : Si $f \neq g$

$$f(s_1, \ldots, s_n) = g(t_1, \ldots, t_m) \longrightarrow \bot$$

Coalescence : Si $x, y \in Var(P),\ x \neq y$

$$x = y \wedge P \longrightarrow x = y \wedge P\{x \rightarrow y\}$$

Elimination : Si $x \in Var(P), x \notin Var(s)$ et $s \notin \mathcal{X}$

$$x = s \wedge P \longrightarrow x = s \wedge P\{x \rightarrow s\}$$

Fusion : Si $x \in X$ et $0 < |s| \leq |t|$

$$x = s \wedge x = t \longrightarrow x = s \wedge s = t$$

Test d'occurrence : Si $p_1 \cdot \ldots \cdot p_n \neq \Lambda$

$$x_1 = t_1[x_2]_{p_1} \wedge \ldots \wedge x_n = t_n[x_1]_{p_n} \longrightarrow \bot$$

Correction des règles

Correction

Lemma

Chacune des règles d'unification transforme un problème d'unification en un problème équivalent.

Exercice : faire la preuve. On utilisera le fait que l'égalité est une congruence dans le domaine de Herbrand.

Preuve de correction

Les règles d'élimination ou de remplacement de variables, d'élimination des équations triviales et de fusion sont des conséquences des axiomes de l'égalité, ou, si l'on préfère, ce sont des conséquence du fait que l'égalité des termes est une congruence.
Les règles de décomposition et d'incompatibilité sont une conséquence des axiomes des termes.
Enfin, le test d'occurrence exprime qu'un terme fini ne peut être égal à un de ses sous-termes stricts. En effet, si c'était le cas, par propriété de congruence de l'égalité, on obtiendrait par récurrence un arbre infini.

Terminaison des règles

Les règles ne précisent pas de stratégie d'application : on peut fabriquer de nombreux algorithmes d'unification en spécifiant dans quel ordre et à quelle sous-problème appliquer les règles. Nous allons montrer que, quelle que soit la stratégie d'utilisation des règles, l'algorithme obtenu termine toujours. Ceci permet de factoriser les preuves de terminaison de différents algorithmes d'unification.

Theorem

Le système de règles d'unification définit un ordre strict bien fondé sur les problèmes d'unification.

Preuve de terminaison

- La variable x est *résolue* dans le problème P si $P \equiv x = s \wedge Q$, $x \notin Var(s)$ et $x \notin Var(Q)$. $\phi_1(P)$ est le nombre de variables non résolues de P.
- Si $P \equiv s_1 = t_1 \wedge \ldots \wedge s_n = t_n$, alors $\phi_2(P)$ est le multi-ensemble $\{m_1, \ldots, m_n\}$ où $m_i = m(s_i = t_i)$ et $m(s = t) = \max(|s|, |t|)$. Par convention, on pose $m(\bot) = m(\top) = 0$
- $\phi_3(P)$ est le nombre d'équations de P dont l'un des membres au moins est une variable.

$\phi(P)$ est le triplet $(\phi_1(P), \phi_2(P), \phi_3(P))$.

suite

Les triplets sont comparés lexicographiquement avec :

1. L'ordre habituel sur les entiers naturels
2. L'extension multi-ensemble de l'ordre sur les entiers naturels
3. L'ordre habituel sur les entiers naturels

ce qui constitue un ordre bien-fondé.

On montre que si $P \longrightarrow Q$ alors $\phi(P) > \phi(Q)$.
Le sens de variations des fonctions d'interprétation est résumé ci-dessous:

	ϕ_1	ϕ_2	ϕ_3
équations triviales	$\geq$	$\geq$	
Decompose	$\geq$	$\geq$	
Incompatibilité	$\geq$	$>$	
Élimination de variable	$>$		
Fusion	$\geq$	$=$	$>$
Test d'occurrence	$\geq$	$>$	

Vérification

On vérifie qu'aucune règle ne crée de nouvelle variable non-résolue. Les conditions d'application des deux règles d'élimination et de remplacement de variables garantissent par ailleurs qu'une variable non résolue (x dans la formulation des règles doit apparaitre dans P) devient résolue après leur application.

La règle de décomposition fait décroître ϕ_2 car $m = \max(|f(s_1, \ldots, s_n)|, |f(t_1, \ldots, t_n)|)$ est remplacé par n entiers strictement inférieurs.

La règle de fusion conserve ϕ_2 à cause de la condition $|s| \leq |t|$: $\max(|x|, |t|) = \max(|s|, |t|)$.

ϕ_3 est décroissante par fusion car les conditions $s \notin x$ et $|s| \leq |t|$ garantissent que $s, t \notin \mathcal{X}$.

Complétude des règles

Lemma

P est une arbre-forme résolue si aucune règle ne lui est applicable.

Comme les règles d'incompatibilité et de décomposition ne s'appliquent pas, les équations ont une variable x dans un de leurs membres. Comme le test d'occurrence et l'identité ne s'applique pas, x n'est pas une variable de $x = t \in P$. Comme la règle d'élimination de variable ne s'applique pas, x ou t doit de plus être une variable résolue, ce qui conduit au résultat souhaité.

Règles de fusion et de coalescence sont

Complétude des règles

Lemma

P est une DAG-forme résolue si la seule règle applicable est l'élimination de variables.

Si la seule règle applicable à P est l'élimination de variable, alors toutes les équations de P sont de la forme $x = t$ où $x \notin Var(t)$. Considérons alors la relation *d'occurrence* $\geq_{occ}$ sur les variables de P définie comme la plus petite relation de préordre telle que, si $x = t[y]_p$ est dans P, alors $x \geq_{occ} y$.

Complétude des règles

Comme le test d'occurrence ne s'applique pas, la relation d'équivalence associée à ce préordre est la plus petite relation d'équivalence $=_S$ qui contient $x =_S y$ si $x = y$ est dans P. Mais, si $x =_S y$, x ou y est une variable résolue puisque la règle de remplacement de variables ne s'applique pas. On peut donc construire un ordre $\geq$ sur les variables de P tel que: les variables résolues sont minimales et $x >_{occ} y \Rightarrow x > y$. En complétant $\geq$ en un ordre total, on obtient le résultat souhaité.

Complétude des règles

Lemma

Il existe s et t dont la forme arbre-résolue est exponentiellement plus grande que s et t.
La forme DAG-résolue de deux termes s et t est de taille linéaire en la taille de s et t.

La preuve est à faire en exercice.

Lemma

La forme DAG-résolue de deux termes s et t quelconque peut être calculée en temps $\mathcal{O}(|s| + |t|)$.

Il s'agit de trouver un ordre d'application des règles qui assure la linéarité du calcul.

Exemple

$$
\begin{aligned}
& x = f(f(x)) \wedge x = f(x) \\
\longrightarrow_{\text{Fusion}} \quad & x = f(x) \wedge f(f(x)) = f(x) \\
\longrightarrow_{\text{Décomposition}} \quad & x = f(x) \wedge x = f(x) \\
\longrightarrow_{\text{Fusion}} \quad & x = f(x) \wedge f(x) = f(x) \\
\longrightarrow_{\text{équations triviales}} \quad & x = f(x) \\
\longrightarrow_{\textit{Testd'occurrence}} \quad & \bot
\end{aligned}
$$

Bibliographie essentielle

Jean-Pierre Jouannaud and Claude Kirchner.
Solving equations in abstract algebras: A rule-based survey of unification.
In Jean-Louis Lassez and Gordon Plotkin, editors, *Computational Logic: Essays in Honor of Alan Robinson*, pages 257–321. MIT-Press, 1991.

Alain Colmerauer.
Equations and inequations on finite and infinite trees.
In *FGCS'84 Proceedings*, pages 85–99, November 1984.