An efficient unification algorithm

- Definition and purposes

- Unification problem as the solution of a set of equations
  . Formalization
  . Algorithm

- Extension of the modelisation of the unification problem
  . Formalization
  . Algorithm
  . Possible improvement

Definition

- L'unification est un processus algorithmique qui, étant donnés deux termes, trouve une substitution qui appliquée aux deux termes les rend identiques

$$f(x1, h(x1), x2) = f(g(x3), x4, x3)$$

$$\{(g(x3), x1), (x3, x2), (h(g(x3), x4)\}$$

$$f(g(x3), h(g(x3)), x3) = f(g(x3), h(g(x3)), x3)$$

What for ?

- Resolution rule : la règle de résolution ou principe de résolution de Robinson (en) est une règle d'inférence en logique mathématique. Cette règle est principalement utilisée dans les systèmes de preuve automatiques
- Vérification du bon typage des langages de programmation avec des structures complexes
- Interpreteurs

==> Dans la majorité des cas où interviennent des expressions symboliques

Formalization
Constants, variables and terms

- $$A = \bigcup_{i=0,1,\ldots} A_i \qquad (A_i \cap A_j = \emptyset,\ i \neq j)$$

Alphabet contenant des constantes (A0) et des i-aire fonctions

- **V :** alphabet des variables

- **Termes** :

(1) constant symbols and variables are terms;

(2) if $t_1, \ldots, t_n$ $(n \geq 1)$ are terms and $f \in A_n$, then $f(t_1, \ldots, t_n)$ is a term.

Formalization
Unification as the solution of a set of equations

- Le problème d'unification peut être écrit sous forme d'équation :

$$t'=t''$$

- Une solution de l'équation est une substitution que rend les deux termes identiques. Une subtitution est de la forme : v={(t1,x1), (t2,x2), …, (tm,xm)}
-

$$\textit{équation} : f(x1, h(x1), x2) = f(g(x3), x4, x3)$$

$$\textit{substitution} : \{(g(x3), x1), (x3, x2), (h(g(x3), x4)\}$$

$$f(g(x3), h(g(x3)), x3) = f(g(x3), h(g(x3)), x3)$$

Formalization
Unification as the solution of a set of equations

- Cas général:

$$t_1'=t_1''$$
$$\dots$$
$$t_k'=t_k''$$

- Une solution de cet ensemble d'équations est une subtitution qui rend tous les pairs de termes (t_j', t_j'') identiques. Une subtitution est de la forme v={(t1,x1), (t2,x2), …, (tm,xm)}

Initial algorithm transformations

(1) *Term Reduction.* Let

$$f(t'_1, t'_2, \ldots, t'_n) = f(t''_1, t''_2, \ldots, t''_n), \qquad f \in A_n, \tag{1}$$

be an equation where both terms are not variables and where the two root function symbols are equal. The new set of equations is obtained by replacing this equation with the following ones:

$$\begin{aligned} t'_1 &= t''_1 \\ t'_2 &= t''_2 \\ &\vdots \\ t'_n &= t''_n. \end{aligned} \tag{2}$$

If $n = 0$, then f is a constant symbol, and the equation is simply erased.

(2) *Variable Elimination.* Let $x = t$ be an equation where x is a variable and t is any term (variable or not). The new set of equations is obtained by applying the substitution $\vartheta = \{(t, x)\}$ to both terms of all other equations in the set (without erasing $x = t$).

Initial algorithm goal

- $t_1' = t_1''$ $\qquad\qquad$ $x_1 = t_1$

  … $\qquad$ =================> $\qquad$ …

  $t_k' = t_k''$ $\qquad\qquad$ $x_k = t_k$

- Toutes variable qui apparaît à gauche dans une équation apparaît uniquement à cet endroit
- La subsitution résultante est donc *v={(t1,x1), ..., (tk,xk)}*

Initial algorithm
algorithm

Algorithm 1

Given a set of equations, repeatedly perform any of the following transformations. If no transformation applies, stop with success.

(a) Select any equation of the form

$$t = x$$

where t is not a variable and x is a variable, and rewrite it as

$$x = t.$$

(b) Select any equation of the form

$$x = x$$

where x is variable, and erase it.

(c) Select any equation of the form

$$t' = t''$$

where t' and t'' are not variables. If the two root function symbols are different, stop with failure; otherwise, apply term reduction.

(d) Select any equation of the form

$$x = t$$

where x is a variable which occurs somewhere else in the set of equations and where $t \neq x$. If x occurs in t, then stop with failure; otherwise, apply variable elimination.

Initial algorithm
Example

- Ensemble d'équations

$$g(x_2) = x_1;$$
$$f(x_1, h(x_1), x_2) = f(g(x_3), x_4, x_3).$$

- Etape 1 :

$$g(x_2) = x_1;$$
$$x_1 = g(x_3);$$
$$h(x_1) = x_4;$$
$$x_2 = x_3.$$

Initial algorithm
Example

- Etape 1 :

$$g(x_2) = x_1;$$
$$x_1 = g(x_3);$$
$$h(x_1) = x_4;$$
$$x_2 = x_3.$$

- Etape 2 :

$$g(x_2) = g(x_3);$$
$$x_1 = g(x_3);$$
$$h(g(x_3)) = x_4;$$
$$x_2 = x_3.$$

Initial algorithm
Example

- Etape 2 :

$$
\begin{aligned}
g(x_2) &= g(x_3);\\
x_1 &= g(x_3);\\
h(g(x_3)) &= x_4;\\
x_2 &= x_3.
\end{aligned}
$$

- Etape 3 :

$$
\begin{aligned}
x_2 &= x_3;\\
x_1 &= g(x_3);\\
x_4 &= h(g(x_3));\\
x_2 &= x_3.
\end{aligned}
$$

Initial algorithm
Example

- Etape 3 :

$$x_2 = x_3;$$
$$x_1 = g(x_3);$$
$$x_4 = h(g(x_3));$$
$$x_2 = x_3.$$

- Etape 4 :

$$x_2 = x_3;$$
$$x_1 = g(x_3);$$
$$x_4 = h(g(x_3)).$$

Initial algorithm
Example

- Etape 4 :

$$x_2 = x_3;$$

$$x_1 = g(x_3);$$

$$x_4 = h(g(x_3)).$$

- Resultat :

$$\vartheta = \{(g(x_3), x_1), (x_3, x_2), (h(g(x_3)), x_4)\}.$$

Initial algorithm
A nondeterministic algorithm

- Le résultat varie en fonction de l'ordre selon lequel les équations sont traitées par exemple.
- D'autre algorithme plus spécifique peuvent donc être déduit de l'algorithme précédent.

Formalization
Multi-equations

- Multiéquation :

$$S=M$$

où S est un ensemble non vide de variables et M un ensemble de termes sans variables

- Example :

$$\{x_1, x_2, x_3\} = (t_1, t_2)$$

Formalization
Multi-equations

- $\{x_1, x_2, x_3\} = (t_1, t_2)$
- $x_1 = x_2$

  $x_3 = x_1$

  $t_1 = x_1$

  $x_2 = t_2$

  $t_1 = t_2$

Formalization
Common part

- La partie commune d'un ensemble de terme M=(t1, …, tn) (multiequation S=M) est un terme qui, intuitivement, est obtenu en superposant tous les termes de M et en prenant la partie commune de tous ces termes en commençant par la racine.

- Exemple :

$$(f(x_1, g(a, f(x_5, b))), f(h(c), g(x_2, f(b, x_5))), f(h(x_4), g(x_6, x_3))),$$

the common part is

$$f(x_1, g(x_2, x_3)).$$

Formalization
Frontier

- La frontière d'un ensemble de termes est un ensemble de multiéquations où chaque multiéquation est associé à une extrémité de la partie commune et consistent en tous les sous-termes (un pour chaque terme de M) correspondant à cette extrémité.
- Exemple :

$$(f(x_1, g(a, f(x_5, b))), f(h(c), g(x_2, f(b, x_5))), f(h(x_4), g(x_6, x_3))),$$

the common part is

$$f(x_1, g(x_2, x_3)).$$

$$\begin{aligned} \{\{x_1\} &= (h(c), h(x_4)), \\ \{x_2, x_6\} &= (a), \\ \{x_3\} &= (f(x_5, b), f(b, x_5))\}. \end{aligned}$$

Transformations on multiequations
Reduction

Multiequation Reduction. Let Z be a set of multiequations containing a multiequation $S = M$ such that M is nonempty and has a common part C and a frontier F. The new set Z' of multiequations is obtained by replacing $S = M$ with the union of the multiequation $S = (C)$ and of all the multiequations of F:

$$Z' = (Z - \{S = M\}) \cup \{S = (C)\} \cup F.$$

Exemple :

Z: *{y, z} = (f(x1, g(a, f(x5, b))) , f(h(c), g(x2, f(b, x5))) , f(h(x4), g(x6, x3)))*

Z': *{y, z} = f(x1, g(x2, x3))*
{x1} = (h(c), h(x4)),
{x2, x6} = (a),
{x3} = (f(x5, b), f(b, x5))

Improved unification algorithm

Given a system $R = (T, U)$ with an empty T part, an equivalent system with an empty U part can be computed with the following algorithm.

Algorithm 2

(1) **repeat**

(1.1) Select a multiequation $S = M$ of U with $M \neq \emptyset$.

(1.2) Compute the common part C and the frontier F of M. If M has no common part, stop with failure (clash).

(1.3) If the left-hand sides of the frontier of M contain some variable of S, stop with failure (cycle).

(1.4) Transform U using multiequation reduction on the selected multiequation and compactification.

(1.5) Let $S = \{x_1, \ldots, x_n\}$. Apply the substitution $\vartheta = \{(C, x_1), \ldots, (C, x_n)\}$ to all terms in the right-hand side of the multiequations of U.

(1.6) Transfer the multiequation $S = (C)$ from U to the end of T.

until the U part of R contains only multiequations, if any, with empty right-hand sides.

(2) Transfer all the multiequations of U (all with $M = \emptyset$) to the end of T, and stop with success.

Improved unification algorithm
Example

$$t_1 = f(x_1,\ g(x_2,\ x_3),\ x_2,\ b)$$ $$t_2 = f(g(h(a,\ x_5),\ x_2),\ x_1,\ h(a,\ x_4),\ x_4)$$

$$U\colon \{\{x\} = (f(x_1, g(x_2, x_3), x_2, b), f(g(h(a, x_5), x_2), x_1, h(a, x_4), x_4)),$$
$$\{x_1\} = \varnothing, \{x_2\} = \varnothing, \{x_3\} = \varnothing, \{x_4\} = \varnothing, \{x_5\} = \varnothing\};$$
$$T\colon (\ ).$$

Premièrement on choisit une multiéquation S=M avec M non vide (un seul choix ici)

S = {x}

M= (f(x1, g(x2,x3), x2,b) , f(g(h(a,x5), x2), x1, h(a,x4), x4))

Improved unification algorithm
Example

$t_1 = f(x_1,\ g(x_2,\ x_3),\ x_2,\ b)$ $t_2 = f(g(h(a, x_5), x_2), x_1, h(a, x_4), x_4)$.

$$U\colon \{\{x\} = (f(x_1, g(x_2, x_3), x_2, b), f(g(h(a, x_5), x_2), x_1, h(a, x_4), x_4)),$$
$$\{x_1\} = \varnothing, \{x_2\} = \varnothing, \{x_3\} = \varnothing, \{x_4\} = \varnothing, \{x_5\} = \varnothing\};$$
$$T\colon (\ ).$$

On calcule ensuite la partie commune de M:

M= (f(x1, g(x2,x3), x2,b) , f(g(h(a,x5), x2), x1, h(a,x4), x4))

C= f(x1, x1, x2, x4)

Improved unification algorithm
Example

$t_1 = f(x_1,\ g(x_2,\ x_3),\ x_2,\ b)$ $t_2 = f(g(h(a, x_5), x_2), x_1, h(a, x_4), x_4)$.

$$U\colon \{\{x\} = (f(x_1, g(x_2, x_3), x_2, b), f(g(h(a, x_5), x_2), x_1, h(a, x_4), x_4)),$$
$$\{x_1\} = \varnothing, \{x_2\} = \varnothing, \{x_3\} = \varnothing, \{x_4\} = \varnothing, \{x_5\} = \varnothing\};$$
$$T\colon (\ ).$$

On calcule ensuite la frontière de M:

M= (f(x1, g(x2,x3), x2,b) , f(g(h(a,x5), x2), x1, h(a,x4), x4))

C= f(x1, x1, x2, x4)

F= {x1}= (g(h(a,x5), x2) , g(x2, x3))
{x2}= (h(a,x4))
{x4}= (b)

Improved unification algorithm
Example

$t_1 = f(x_1, g(x_2, x_3), x_2, b)$ $t_2 = f(g(h(a, x_5), x_2), x_1, h(a, x_4), x_4)$

U: $\{\{x\} = (f(x_1, g(x_2, x_3), x_2, b), f(g(h(a, x_5), x_2), x_1, h(a, x_4), x_4)),$

$\{x_1\} = \varnothing, \{x_2\} = \varnothing, \{x_3\} = \varnothing, \{x_4\} = \varnothing, \{x_5\} = \varnothing\};$

T: ( ).

On applique ensuite une réduction à U:

M= (f(x1, g(x2,x3), x2,b) , f(g(h(a,x5), x2), x1, h(a,x4), x4))

C= f(x1, x1, x2, x4)

F= {x1}= (g(h(a,x5), x2) , g(x2, x3))
{x2}= (h(a,x4))
{x4}= (b)

U={ {x}= C,
{x1}= (g(h(a,x5), x2) , g(x2, x3))
{x2}= (h(a,x4))
{x4}= (b)
{x3}= null, {x5}=null
}

Improved unification algorithm
Example

$t_1 = f(x_1, g(x_2, x_3), x_2, b)$ $t_2 = f(g(h(a, x_5), x_2), x_1, h(a, x_4), x_4)$

$$U: \{\{x\} = (f(x_1, g(x_2, x_3), x_2, b), f(g(h(a, x_5), x_2), x_1, h(a, x_4), x_4)),$$
$$\{x_1\} = \varnothing, \{x_2\} = \varnothing, \{x_3\} = \varnothing, \{x_4\} = \varnothing, \{x_5\} = \varnothing\};$$
$$T: (\ ).$$

On ajoute ensuite la multiéquation contenant C dans T :

$$U: \{\{x_1\} = (g(h(a, x_5), x_2), g(x_2, x_3)),$$
$$\{x_2\} = (h(a, x_4)),$$
$$\{x_3\} = \varnothing,$$
$$\{x_4\} = (b),$$
$$\{x_5\} = \varnothing\};$$
$$T: (\{x\} = (f(x_1, x_1, x_2, x_4))).$$

U={ {x}= C,
{x1}= (g(h(a,x5), x2) , g(x2, x3))
{x2}= (h(a,x4))
{x4}= (b)
{x3}= null, {x5}=null
}

Improved unification algorithm
Example

$t_1 = f(x_1, g(x_2, x_3), x_2, b)$ $t_2 = f(g(h(a, x_5), x_2), x_1, h(a, x_4), x_4)$

$$U: \{\{x\} = (f(x_1, g(x_2, x_3), x_2, b), f(g(h(a, x_5), x_2), x_1, h(a, x_4), x_4)),$$
$$\{x_1\} = \varnothing, \{x_2\} = \varnothing, \{x_3\} = \varnothing, \{x_4\} = \varnothing, \{x_5\} = \varnothing\};$$
$$T: (\ ).$$

On répète la suite d'opérations précédente:

$$U: \varnothing;$$
$$\begin{aligned} T: (&\{x\} = (f(x_1, x_1, x_2, x_4)), \\ &\{x_2\} = (h(a, x_4)), \\ &\{x_1\} = (g(h(a, x_4), x_3)), \\ &\{x_4, x_5\} = (b), \\ &\{x_3\} = (h(a, b))). \end{aligned}$$

Conclusion

- Améliorations
- Algorithme efficace et rapide.
- Peut être utilisé en parallèle avec d'autres algorithmes pour des systèmes de preuves automatiques plus efficaces.