ALGORITHME D'UNIFICATION

Cf. David, Nour, Raffalli p 248.

On travaille sur des termes écrit dans un langage $\mathcal{L}$ et sur l'ensemble de variables $\mathcal{V}$.

Déf

- Une substitution est une fonction qui à chaque variable associe un terme.

ex $\mathcal{L} = (s, +, 0)$ $\quad \mathcal{V} = (x, y, z)$ $\quad \sigma = \begin{matrix} x \mapsto x+y \\ y \mapsto y \\ z \mapsto y+0 \end{matrix}$

- Si u est un terme et σ une substitution, on note $u[\sigma]$ le terme obtenu en remplaçant (simultanément) chaque variable par son image par σ.

En particulier si $x \in \mathcal{V}$, $x[\sigma] = \sigma(x)$.

⚠ $u[\sigma' \circ \sigma] = (u[\sigma])[\sigma']$

ex $\mathcal{L} = (s, +, 0)$ $\quad \mathcal{V} = (x, y, z)$ $\quad \sigma = \begin{matrix} x \mapsto y \\ y \mapsto s(z) \\ z \mapsto 0 \end{matrix}$ $\quad \sigma' = \begin{matrix} x \mapsto x \\ y \mapsto 1 \\ z \mapsto 0 \end{matrix}$ $\quad u = s(x) + y$

$u[\sigma] = s(y) + s(z)$

$u[\sigma' \circ \sigma] = s(1) + s(0)$

- Deux termes u et v sont unifiables s'il existe une substitution σ telle que $u[\sigma] = v[\sigma]$. On dit alors que σ unifie u et v.

Rq $\hat{e}$ unifiables est une relation réflexive et symétrique mais non transitive ⚠ ex $g(x) \sim y$; $y \sim f(z)$ mais $g(x) \not\sim f(z)$

ex $s(x+y)$ et $s(z)+x$ ne sont pas unifiables.

$s(x+y)+y$ et $s(z)+x$ sont unifiables :

pour $\sigma = \begin{pmatrix} y \mapsto x \\ z \mapsto x+x \end{pmatrix}$ $\quad (s(x+y)+y)[\sigma] = s(x+x)+x = s(z)+x\,[\sigma]$

- Un unificateur le plus général (m.g.u "most general unifier") de u et v est une substitution σ qui unifie u et v et telle que si τ unifie u et v, alors il existe τ' tq $\tau = \tau' \circ \sigma$.

ex 4 dernière page. $u = f(x, x, y)$ $\quad v = f(f(y, y, z), f(y, y, z), a)$

$mg.u(u, v) = \begin{pmatrix} x \mapsto f(a, a, z) \\ y \mapsto a \end{pmatrix}$

⚠ Toutes ces notions sont purement syntaxiques

ex $2 + x$ et 3 ne sont pas unifiables.

$\hat{m}$ par $x := 1$ car "$2+1$" $\neq$ "3"

Ptt | Si u et v sont unifiables, alors ils admettent un m.g.u. On le notera m.g.u(u,v).

La preuve sera l'algorithme que l'on va construire et dont on va montrer qu'il termine toujours, soit par une erreur si u et v ne sont pas unifiables, soit en fournissant un m.g.u s'ils le sont. Cependant notre algorithme sera plus général, puisqu'il cherchera à unifier simultanément plusieurs couples de termes.

Algo

UNIFICATION $(\{u_k, v_k\}_{k\in[1..K]})$

$E \leftarrow \{(u_k, v_k) \mid k \in [1..K]\}$

$\sigma \leftarrow id$

Tant que $E \neq \emptyset$

Choisir $e \in E$

match e with

| (x,x) où $x \in \mathcal{V}$ → $E \leftarrow E \setminus e$

| (x,u) où $x \in \mathcal{V}$ ou (u,x) → Si x apparaît dans u
alors fail with "occur-check"
sinon $E \leftarrow E[x := u]$ (reste)
$\sigma \leftarrow [x := u] \circ \sigma$

| $(f(t_1, t_2 .., t_p), g(s_1, ..., s_q))$ → Si $f \neq g$
alors fail with "clash"
sinon $E \leftarrow E \setminus e \cup \{(t_i, s_i) \mid i \in [1..p]\}$

Retourner σ.

Terminaison

On note v resp. f resp $é$ le nombre courant de variables, resp. de symboles de fonction, respectivt d'équations dans E.

Sur les ensembles "d'équations" c-à-d de couples de termes, on peut considérer l'ordre $\leq_v$ du nombre de variables, ie $E \leq_v E'$ si E fait apparaître moins de variables que E'

Cet ordre est bien-fondé car toute chaine strictement décroissante est minorée : on ne peut avoir moins de 0 variables.

Correction

On note E_0 l'ensemble E dans son état initial puis E_i son état après l'étape i, et σ de même.

On introduit la ptié H_n : "τ unifie E ssi il existe τ' unifiant E_n tel que $\tau = \tau' \circ \sigma_n$."

où $E = \{u_k \sim v_k \mid k \in [1..K]\}$

On cherche à MQ qu'on a H_n pour n allant de 0 à la dernière étape N d'une exécution.

Montrons déjà en quoi cela permet de conclure.

↳ Si on s'arrête parce que $E_N = \emptyset$, alors id unifie $\emptyset = E_N$. Par H_N on en déduit que $\sigma_N = id \circ \sigma_n$ unifie E, et c'est même le m.g.u puisque si τ unifie E il existe τ' (unifiant $E_N = \emptyset$, donc une substitution quelconque) telle que $\tau = \tau' \circ \sigma_N$.

↳ Si on s'arrête par un clash, E_N contient une "équation" de la forme $f(t_1, \dots, t_p) \sim g(s_1, \dots s_q)$ où $f \neq g$ - qu'aucune subst. ne peut unifier, car s'il existait une substitution τ unifiant E, il en existerait, d'après H_N, une autre τ' unifiant E_N. On en déduit que E n'est pas unifiable.

↳ Si on s'arrête par un occur check, alors E_N contient une "équa°" de la forme $x \sim u$ où x apparaît dans le terme u. Comme précédemment ces deux termes ne sont pas unifiables.

[En effet x apparaît dans u mais $u \neq x$, donc u contient au moins un symbole de fonction en plus d'une occurrence de x, donc $|u[\sigma]| \geq 1 + |x[\sigma]| > |x[\sigma]|$ donc $u[\sigma] \neq x[\sigma]$ pour σ quelconque]

Donc on déduit de même de H_N que E n'est pas unifiable.

Montrons maintenant que $\forall n \in [0..N]$ on a H_n.

→ Pour $n = 0$ $E_n = E_0 = E$ et $\sigma_0 = id$ donc la propriété se réécrit en τ unifie E ssi il existe τ' unifiant E tq $\tau = \tau' \circ id$ - qui est trivialement vérifiée pour $\tau' = \tau$.

On a bien H_0.

→ Pour $n < N$ tel que H_n est vraie

On sait MQ H_{n+1} est vraie aussi, et on a trois cas à étudier.

De même on considère les ordres $\lhd_f$ et $\lhd_é$ qui sont aussi bien-fondés.
On pose alors $\lhd = \lhd_v \times \lhd_f \times \lhd_é$ l'ordre lexicographique produit de ces trois ordres.
Pour rappel cela signifie que $E \lhd E'$ si

→ E a strictement moins de variables que E'

ou → E a autant de var. que E' mais st. moins de f.

ou → E a autant de var. et autant de f que E', mais st. moins d'équa.

Comme produit d'ordre lexicographiques bien-fondés, $\lhd$ est bien-fondé
Il nous suffit donc de montrer qu'à chaque étape E decroient + petit pour $\lhd$, à moins que l'algo ne se termine par une erreur.

Dans le cas 1. $E_{\setminus(x,x)}$ a peut-être moins de variables, si enlever (x,x) a suffit à faire disparaître tous les occurences de x, mais sans cela on est tout de m̃ assuré que le nbre d'équa décroit st. (et que celui de variable ou de f ne croît pas).
On synthétise par $\boxed{v \searrow \quad \text{ou} \quad v = \quad f = \quad \lhd_é \searrow}$

Dans le cas 2, dans le cas où x n'apparaît pas dans u, tous les occurences de x disparaissant, remplacées par u : $\boxed{v \searrow}$

Dans le cas 3. dans le cas où $f = g$ $\boxed{v = f \searrow}$

Donc l'algorithme termine, soit par une erreur, soit en renvoyant une substitution.

→ Si $E_{n+1} = E_n \setminus \{e\}$ où $e = (x, x)$ (cas 1)

→ Soit τ unifiant E.
D'après H_n il existe τ' unifiant E_n tel que $\tau = \tau' \circ \sigma_n$.
Puisque $E_{n+1} \subset E_n$, τ' unifie aussi E_{n+1} et $\sigma_{n+1} = \sigma_n$ donc $\tau = \tau' \circ \sigma_{n+1}$.

→ Réciproquement si τ' unifie E_{n+1}, $x[\tau'] = x[\tau']$ donc τ' unifie aussi E_n.
Donc $\tau' \circ \sigma_{n+1} = \tau' \circ \sigma$ unifie aussi E.
On a bien H_{n+1}.

→ Si $E_{n+1} = E_n[x := u]$ et $\sigma_{n+1} = [x := u] \circ \sigma_n$ (cas 2).

→ Soit τ unifiant E.
Par H_n il existe τ' unifiant E_n tq $\tau = \tau' \circ \sigma_n$. En particulier τ' unifie x et u puisque $(x, u) \in E_n$; on a donc $x[\tau'] = u[\tau']$.
donc $\tau = \tau' \circ [x := u]$ $\left[\begin{array}{l} x[\tau' \circ [x := u]] = (x[x := u])[\tau'] \\ \qquad\qquad = u[\tau'] = x[\tau'] \\ \text{et si } y \neq x \quad y[\tau' \circ [x := u]] = (y[x := u])[\tau'] = y[\tau'] \end{array}\right]$

Pour $(a', b') \in E_{n+1}$, il existe $(a, b) \in E_n$ tq $(a', b') = (a, b)[x := u]$.
donc $a'[\tau'] = a'[x := u][\tau'] = a[\tau'] = b[\tau'] = (b'[x := u])[\tau'] = b'[\tau']$.
Donc τ' unifie aussi E_{n+1}.
Et $\tau = \tau' \circ \sigma_n = \tau' \circ [x := u] \circ \sigma_n = \tau' \circ \sigma_{n+1}$.

→ Réciproquement si τ' unifie E_{n+1}, $\tau' \circ [x := u]$ unifie E_n. puisque pour $(a, b) \in E_n$ on a $(a[x := u], b[x := u]) \in E_{n+1}$ donc $a[x := u][\tau'] = b[x := u][\tau']$.
Donc par H_n $\tau' \circ [x := u] \circ \sigma_n$ unifie E or $\tau' \circ [x := u] \circ \sigma_n = \tau' \circ \sigma_{n+1}$.
On a bien H_{n+1}.

→ Si $E_{n+1} = E_n \setminus e \cup \{(t_i, s_i) \mid i \in [1..p]\}$ où $e = f(t_1 \ldots t_p) : f(s_1 \ldots s_p)$ (cas 3) (sans erreur).

→ Soit τ unifiant E.
Par H_n il existe τ' unifiant E_n tel que $\tau = \tau' \circ \sigma_n$.
Puisque $e \in H_n$, τ' unifie $\grave{a}$ $f(t_1, \ldots, t_p)$ et $f(s_1 \ldots s_p)$ donc unifie t_i et s_i pour $i \in [1..p]$. donc τ' unifie E_{n+1} et $\sigma_{n+1} = \sigma_n$
donc $\tau = \tau' \circ \sigma_{n+1}$ où τ' unifie E_{n+1}.

→ Réciproquement si τ' unifie E_{n+1}, comme il unifie en particulier les (t_i, s_i) pour $i \in [1..p]$, il unifie aussi $f(t_1, \ldots, t_p)$ et $f(s_1 \ldots s_p)$ donc unifie E_n.
Par H_n on en déduit que $\tau' \circ \sigma_n$ unifie E. Or $\sigma_{n+1} = \sigma_n$ donc $\tau' \circ \sigma_n$ unifie E.

On a bien H_{n+1}.

CQFD.

esc

$u = f(x, x, y)$

$v = f(f(y,y,z), f(y,y,z), a)$

v : nombre de variables
f : nbre de symboles de ϕ
e : nombre d'équations

$E \leftarrow \{(f(x,x,y) \,,\, f(f(y,y,z), f(y,y,z), a))\}$ $v=3$ $f=4$ $e=1$

$\sigma \leftarrow id$.

$e = (f(x,x,y) \,,\, f(f(y,y,z), f(y,y,z), a))$
est de la forme $f(t_1, t_2, t_3) = f(s_1, s_2, s_3)$ cas 3

$E \leftarrow \{(x, f(y,y,z)) \,;\, (x, f(y,y,z)) \,;\, (y, a)\}$ $v=3$ $f=2$ $e=3$

$e = (x, f(y,y,z))$ et x n'apparaît pas dans $f(y,y,z)$ cas 2

$E \leftarrow \{(f(y,y,z), f(y,y,z)) \,;\, (y, a)\}$ $v=2$ $f=2$ $e=2$

$\sigma \leftarrow x := f(y,y,z)$

$e = (f(y,y,z), f(y,y,z))$ cas 3

$E \leftarrow \{(y,y) \,;\, (y,y) \,;\, (z,z) \,;\, (y,a)\}$ $v=2$ $f=0$ $e=4$

$e = (y,y)$ cas 1

$E \leftarrow \{(y,y), (z,z), (y,a)\}$ $v=2$ $f=0$ $e=3$

$e = (y,y)$ cas 1

$E \leftarrow \{(z,z)\ (y,a)\}$ $v=2$ $f=0$ $e=2$

$e = (z,z)$ cas 1

$E \leftarrow \{(y,a)\}$ $v=1$ $f=0$ $e=1$

$e = (y,a)$ cas 2

$\sigma \leftarrow [y := a] \circ [x := f(y,y,z)] = \begin{cases} y \mapsto a \\ x \mapsto f(a,a,z) \end{cases}$

$E \leftarrow \emptyset$ $v=0$ $f=0$ $e=0$

NB : l'algo donné est non déterministe du fait de "choisir e dans E".
Ici on a utiliser une pile pour gérer E.
C'est une exécution possible, mais pas la seule.