$\boxed{\text{Algorithme d'unification}}$

Réf : DNR : Introduction à la logique
(Baader, Nipkow : Term rewriting and all that)

Def/Algo : Soit E un ensemble d'équations. On définit, par récurrence, une suite (E_n, σ_n) de la manière suivante : $E_0 = E$ et $\sigma_0 = Id$.

1. Si $E_n = E' \uplus \{f(u_1, \dots, u_q) \sim g(v_1, \dots, v_p)\}$
   - Si $f = g$ (et alors $q = p$) : on pose $E_{n+1} = E' \cup \{u_1 \sim v_1, \dots, u_p \sim v_p\}$, $\sigma_{n+1} = \sigma_n$
   - Sinon il y a échec de l'unification par "clash"
2. Si $E_n = E' \uplus \{x \sim x\}$: on pose $E_{n+1} = E'$ et $\sigma_{n+1} = \sigma_n$
3. Si $E_n = E' \uplus \{x \sim u\}$ ou $E_n = E' \uplus \{u \sim x\}$ avec $u \neq x$:
   - Si la variable x n'apparait pas dans u : on pose $\sigma_{n+1} = [x := u] \circ \sigma_n$ et $E_{n+1} = E'[x := u]$
   - Sinon il y a échec de l'unification par "occur-check"
4. Si $E_n = \emptyset$, l'algorithme s'arrête.

Prop : L'algorithme termine toujours : soit par un échec, soit avec $E_n = \emptyset$.

Demo : • On note $f(n) = (a_n, b_n, c_n)$ où a_n est le nombre de variable, b_n le nombre de symbole de fonction et c_n le nombre d'équations dans E_n.

• Il suffit de montrer que pour tout n où $E_n \neq \emptyset$, $f(n+1) < f(n)$, où $<$ est l'ordre lexicographique sur $\mathbb{N}^3$. Examinons chacun des cas de l'algo :

[1] on a $a_{n+1} = a_n$ et $b_{n+1} = b_n - 2 < b_n$

[2] on a $a_{n+1} \leqslant a_n$, $b_{n+1} = b_n$ et $c_{n+1} = c_n - 1 < c_n$

[3] on a $a_{n+1} = a_n - 1 < a_n$

□

Th : Si l'algorithme termine avec $E_n = \emptyset$, alors la substitution σ est l'unificateur le plus général du système d'équations donné. S'il échoue (par clash ou occur-check) alors le système d'équations n'a pas d'unificateur.

$\boxed{\text{Démo}}$ • On montre un résultat plus général qui est un invariant de boucle : " σ unifie E ssi il existe σ' qui unifie E_n et telle que $\sigma = \sigma' \circ \sigma_n$ " $\equiv H_n$

$\sigma = \sigma' \circ \sigma_n$

• On suppose qu'on a H_n et on veut montrer qu'on a H_{n+1} après le passage dans la boucle, si on échoue pas :

1] On a $E_{n+1} = E' \cup \{u_1 \sim v_1, \dots, u_n \sim v_n\}$ et $\sigma_{n+1} = \sigma_n$. Il est donc clair que σ' unifie E_{n+1} ssi σ' unifie E_n.

2] On a $E_{n+1} = E'$ et $\sigma_{n+1} = \sigma_n$ et le résultat est immédiat.

3] $(E_n = E' \cup \{x \sim u\})$ alors $\sigma_{n+1} = [x := u] \circ \sigma_n$ et $E_{n+1} = E'[x := u]$.

- Si σ'' unifie E_{n+1} et $\sigma = \sigma'' \circ \sigma_{n+1}$, on prends $\sigma' = \sigma'' \circ [x := u]$ qui unifie bien E_n et qui vérifie $\sigma' \circ \sigma_n = \sigma'' \circ [x := u] \circ \sigma_n = \sigma'' \circ \sigma_{n+1} = \sigma$

- Si σ' unifie E_n et $\sigma = \sigma' \circ \sigma_n$, alors $x[\sigma'] = u[\sigma']$, alors on a $\sigma' = \sigma' \circ [x := u]$ (lemme), ce qui donne le résultat voulu.
$(\sigma'' = \sigma'$ et $\sigma'' \circ \sigma_{n+1} = \sigma' \circ [x := u] \circ \sigma_n = \sigma' \circ \sigma_n = \sigma)$

Ainsi, on a prouvé (il existe σ' qui unifie E_n et telle que $\sigma = \sigma' \circ \sigma_n$) ssi (il existe σ'' qui unifie E_{n+1} et telle que $\sigma = \sigma'' \circ \sigma_{n+1}$), ce qui montre $H_n \Rightarrow H_{n+1}$.

• On examine les différents cas de sortie de l'algorithme :

* Si on termine avec $E_n = \emptyset$: Id unifie E_n et donc par H_n, on unifie E.
De même, par H_n, σ unifie E ssi il existe σ' tq $\sigma = \sigma' \circ \sigma_n$, d'où on est bien le mgu de E.

* Si l'algorithme échoue par "clash" : E_n n'a pas d'unificateur, et donc par H_n, E n'en a pas non plus.

* Si l'algorithme échoue par "occur-check" : E_n contient une équation $x \sim u$ avec x dans u (et $x \neq u$). Pour toute substitution σ, la taille de $x[\sigma]$ est strict inférieure à celle de $u[\sigma]$, donc $x[\sigma]$ ne peut jamais être égal à $u[\sigma]$. Ainsi E_n n'a pas d'unif et donc E non plus. $\square$