Master 2 - LP Année 2019

TD de Typage n° 3

Inférence de types monomorphes

I) Les substitutions et les unificateurs

Exercice 1 [Les substitutions]

1. Soit $t = p(x, y)$. On considère les substitutions $\sigma_1 = \{x/f(a)\}$ et $\sigma_2 = \{y/f(x)\}$.
   - Calculer: $\sigma_1 \circ \sigma_2$, $\sigma_2 \circ \sigma_1$, $\sigma_1(t)$, $\sigma_2(t)$, $\sigma_1 \circ \sigma_2(t)$ et $\sigma_2 \circ \sigma_1(t)$.
   - Est-il vrai que $\sigma_1 \circ \sigma_2(t) = \sigma_1(\sigma_2(t))$?
   - Est-il vrai que $\sigma_2 \circ \sigma_1(t) = \sigma_2(\sigma_1(t))$?
2. Soit $\sigma_1 = \{x/y\}$ et soit $\sigma_2 = \{y/x\}$. Calculer $\sigma_1 \circ \sigma_2$.
3. Soit s le terme $r(x, y, z)$ et les substitutions

   $$\begin{array}{rcl} \sigma_1 & = & \{x/f(a), y/f(x), z/b\} \\ \sigma_2 & = & \{x/f(z), y/f(b), z/b\} \\ \sigma_3 & = & \{w/z, z/b\} \circ \{x/f(w), y/a\} \end{array}$$

   Calculer $\sigma_1(s)$, $\sigma_2(s)$ et $\sigma_3(s)$.
4. Montrer que le composition de substitutions est associative.
5. Définir l'opération $\sigma|_V$ comme la restriction de la substitution σ à l'ensemble de variables V. Montrer que pour tout terme t on a $\sigma(t) = \tau(t)$ ssi $\sigma|_{Var(t)} = \tau|_{Var(t)}$.
6. Montrer que $t \neq \sigma t$ si $\exists\ x$ t.q. $x \in Var(t)\ \&\ x \in Dom(\sigma)$ (réciproquement, si $Var(t) \cap Dom(\sigma) = \emptyset$ alors $\sigma t = t$).
7. Une substitution est *idempotente* ssi $\sigma \circ \sigma = \sigma$. Montrer que σ est idempotente ssi $Codom(\sigma) \cap Dom(\sigma) = \emptyset$.
8. Montrer que $t\{x/u\}\{y/v\} = t\{y/v\}\{x/u\{y/v\}\}$ si $x \neq y$ et $x \notin Var(v)$.

Exercice 2 [Les unificateurs]

1. Expliquer pourquoi on a besoin de la restriction "$Dom(\sigma) \subseteq VI(\mathcal{P})$" pour obtenir l'unicité de l'unificateur principal d'un problème $\mathcal{P}$.
2. Montrer que si x n'apparait pas dans t, alors $\{x/t\}$ est un unificateur principal et idempotent de l'équation $x \doteq t$.

II) L'algorithme d'unification

Exercice 3 [L'algorithme d'unification (i)] Les lettres p, q, a, b, f, g, h, k sont des symboles de fonction, les autres sont des variables. Appliquer l'algorithme d'unification aux problèmes suivants :

1. $p(a, x, f(g(y))) \doteq p(z, f(z), f(u))$
2. $q(f(a), g(x)) \doteq q(y, y)$
3. $p(x, f(y, z)) \doteq p(x, g(h(k(x))))$
4. $p(x, f(u, x)) \doteq p(f(y, a), f(z, f(b, z)))$
5. $p(x, f(x), g(f(x), x)) \doteq p(z, f(f(a)), g(f(g(a, z)), v))$
6. $p(f(g(x, y)), g(v, w), y) \doteq p(f(z), x, f(x))$
7. $p(x, f(x), f(f(x))) \doteq p(f(f(y)), y, f(y))$
8. $p(f(y), f(z), f(t), f(x)) \doteq p(g(z), g(x), g(y), g(z))$

Exercice 4 [L'algorithme d'unification (ii)]
Exhiber une équation t.q. l'algorithme d'unification utilise exactement une seule fois chaque règle de transformation.

III) Le type principal d'une expression

Exercice 5 [Le type principal d'une expression] Donner le système d'équations associé à chaque terme, et exhiber plusieurs solutions pour ce système.

1. $M \equiv \textbf{let } x = 3 \textbf{ in } x + 1.$
2. $M \equiv \textbf{let } x = \langle x_1, x_2 \rangle \textbf{ in } (\textbf{let } y = +\langle snd\ x, 1\rangle \textbf{ in } (\lambda z.y)\ (fst\ x)))$
3. $M \equiv \textbf{let } f = \lambda x. + \langle x, 1\rangle \textbf{ in}$
   $(\textbf{let } g = \lambda y. + \langle y, 4\rangle \textbf{ in } * \langle f\ 3, g\ 3\rangle)$
4. $M \equiv \lambda f.\lambda g.f\ g$
5. $M \equiv \textbf{let } f = \lambda x.x \textbf{ in } f(f\ z)$
6. $M \equiv \textbf{let } x = y\ z \textbf{ in } x\ w$

IV) Extensions de l'algorithme de typage

Exercice 6 [Extensions de l'algorithme de typage] Donner une extension de l'algorithme de typage à la Curry aux cas des listes.

Exercice 7 [Implémentation] En utilisant votre implémentation de l'algorithme d'unification, implémenter l'algorithme de typage pour les types monomorphes à la Curry.