TD de Maths pour l'Info n$^{\text{o}}$ 10

Calcul des prédicats : Unification et Résolution

Exercice 1 (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 $\sigma_3(s)$.

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

Calcul des prédicats : Sémantique

Exercice 3 (Modélisation) On considère la phrase P : « tous les hommes sont méchants », et les "mondes réels" M_1, dans lequel on trouve le chien *Rantanplan*, les chats *Garfield* et *Félix*, le poussin *Saturnin*, et les personnes : *Jeanne* (sympathique), aimée par *Jules* (saoûl) et *Jim* (méchant), et le monde M_2, avec seulement *Garfield* et *Jim*.

1. La phrase P est-elle vraie dans M_1 et M_2 ?
2. On considère le langage $\mathcal{L}$, où l'ensemble de symboles de fonction F est vide et où les symboles de prédicats sont :

   homme/1, *méchant*/1, *femme*/1, *chat*/1, *chien*/1, *personne*/1,
   sympa/1, *saoûl*/1, *aime*/2

   où p/n signifie que le prédicat p est d'arité n.

   On considère aussi les formules suivantes :
   - $\varphi_1 \;=\; \forall x\ (\mathit{homme}(x) \wedge \mathit{méchant}(x))$,
   - $\varphi_2 \;=\; \forall x\ (\mathit{chat}(x) \vee \mathit{méchant}(x))$,
   - $\varphi_3 \;=\; \forall x\ (\mathit{homme}(x) \rightarrow \mathit{méchant}(x))$

   (a) Modéliser M_1 et M_2 par des interprétations $\mathcal{I}_1$ et $\mathcal{I}_2$ de $\mathcal{L}$.

   (b) Evaluer φ_1, φ_2 et φ_3 dans $\mathcal{I}_1$ et $\mathcal{I}_2$.

   (c) Parmi φ_1, φ_2 et φ_3, laquelle est une modélisation correcte de la phrase P ?

Exercice 4 (Satisfiabilité) Pour chacune des formules suivantes quelles sont celles qui sont satisfiables, valides, insatisfiables :

(1) $\forall x\ (p(x) \vee \neg p(x))$	(3) $\forall x\ \forall y\ (p(x) \leftrightarrow \neg p(y))$	(5) $(\forall x\ p(x)) \rightarrow \neg p(z)$
(2) $\forall x\ (p(x) \rightarrow p(z))$	(4) $\forall x\ (p(x) \wedge \neg p(x))$	(6) $\exists x\ (p(x) \rightarrow p(a) \wedge p(b))$

Exercice 5 (Indépendance) Soient les formules :

1. $F_1 \;=\; \forall x\ p(x,x)$
2. $F_2 \;=\; \forall x\ \forall y\ (p(x,y) \rightarrow p(y,x))$
3. $F_3 \;=\; \forall x\ \forall y\ \forall z\ ((p(x,y) \wedge p(y,z)) \rightarrow p(x,z))$

Montrer qu'aucune de ces formules n'est conséquence logique des deux autres. Quel est le sens intuitif de chacune de ces formules ?

Exercice 6 (Validité) Montrer que :

1. $\exists y\ \forall x\ p(y,x) \;\models\; \forall x\ \exists y\ p(y,x)$
2. $\forall x\ \exists y\ p(y,x) \;\not\models\; \exists y\ \forall x\ p(y,x)$
3. $p(x) \;\models\; \forall x\ p(x)$.
4. $\neg(\forall x\ A) \;\equiv\; \exists x\ (\neg A)$
5. $(\exists x\ A) \vee B \;\equiv\; \exists x\ (A \vee B)$ si $x \notin FV(B)$.
6. $\forall x\ (A \vee B) \;\not\equiv\; (\forall x\ A) \vee (\forall x\ B)$, $\quad \exists x\ (A \wedge B) \;\not\equiv\; (\exists x\ A) \wedge (\exists x\ B)$

Exercice 7 (Implémentation) En Caml, on définit les types des termes et des formules du calcul des prédicats par

```
type variable = string ;;  (* variables *)
type sym_fun = string ;;   (* symboles de fonction *)
type sym_pre = string ;;   (* symboles de prédicat *)

type terme =
  | Var of variable
  | Fun of sym_fun * terme list ;;

type formule =
  | Atome of sym_pre * terme list  (* Formule atomique *)
  | Non of formule                 (* Négation *)
  | Et of formule * formule        (* Conjonction *)
  | Ou of formule * formule        (* Disjonction *)
  | Imp of formule * formule       (* Implication *)
  | Eqv of formule * formule       (* Équivalence logique *)
  | Univ of variable * formule     (* Quantification universelle *)
  | Exis of variable * formule     (* Quantification existentielle *)
```

1. Définir des fonctions

`var_list` : `terme` → `variable list`   et   `varlib_list` : `formule` → `variable list`

qui calculent respectivement l'ensemble des variables d'un terme et l'ensemble des variables libres d'une formule (ces ensembles étant représentés par des listes de variables).

On définit les notions d'interprétation (uniquement sur des domaines finis) et d'assignation de la manière suivante :

```
type valeur = string ;;

type interpretation = {
  ind_dom : valeur list ;
  int_fun : (sym_fun, valeur list -> valeur) list ;
  int_pre : (sym_pre, valeur list -> bool) list
} ;;

type assignation = (variable, valeur) list ;;
```

Ici, le champ `int_dom` contient l'énumération de toutes les valeurs du domaine, et les champs `int_fun` et `int_pre` contiennent des listes d'associations fournissant les fonctions sémantiques associées aux symboles de fonction et de prédicat. Les assignations sont représentées par des listes d'associations variable/valeur.

2. Définir des fonctions

`interp_terme` : `interpretation` → `assignation` → `valeur`
`interp_formule` : `interpretation` → `assignation` → `bool`

qui calculent respectivement l'interprétation d'un terme et d'une formule dans une interprétation et une assignation données.

Indication. On aura tout intérêt à utiliser les fonctions `List.map`, `List.for_all` et `List.exists` de la librairie standard de Caml...