Cours 12

Résolution en calcul des prédicats

L'inconvénient de la méthode de Herbrand standard, sur le plan pratique, est que l'ensemble des clauses engendrées, pour la plupart des formules, est très grand. Ce qui demandera pour leur traitement un espace mémoire à l'ordinateur et un temps de calcul inacceptable. Pour surmonter ces difficultés Robinson introduit une nouvelle méthode de résolution générale en généralisant la règle de résolution décrite déjà pour le calcul propositionnel. L'idée de base consiste à rendre identiques les arguments des littéraux à résoudre en substituant des termes aux variables de manière appropriée. Par exemple si on a $\{P(y,x), \neg P(f(g(f(x),g(b))),a)\}$. On peut voir que l'ensemble est contradictoire, si on substitue à y $f(g(f(x),g(b))$ et à x le terme a. Alors qu'en générant les instances de base, il faut attendre H_3 pour avoir la contradiction, ce qui demandera beaucoup d'espace et temps de calcul. Pour pouvoir décrire cette méthode générale de résolution il est nécessaire d'introduire quelques définitions.

12.1 Substitution

Une substitution est une application σ qui associe un terme à chaque variable. On définit pour chaque terme t le résultat de la substitution dans t de toute variable x par $\sigma(x)$ que l'on note $t[\sigma]$ ou simplement $t\sigma$.

On dénote (et souvent on implémente sur ordinateur) les substitutions par des listes de couples (listes associatives), notées

$$\{x_1 \to \sigma(x_1), \ldots, x_n \to \sigma(x_n)\}, \text{ou}[\sigma(x_1)/x_1, \ldots, \sigma(x_n)/x_n].$$

Si une variable n'apparaît pas dans une liste, alors la convention est qu'elle est fixée par σ (c'est à-dire $\sigma(x) = x$).

On peut définir par induction l'action de σ sur t comme suit :

$$\begin{aligned} x\sigma &= \sigma(x), \\ f(t_1, \ldots, t_n)\sigma &= f(t_1\sigma, \ldots, t_n\sigma). \end{aligned}$$

La substitution identité(ou vide) est celle qui fixe toutes les variables. Elle est donc notée par []. La composition de deux substitutions τ et τ, notée $\tau \circ \sigma$, est la substitution définie par : $(\tau \circ \sigma)(x) = (\sigma(x))_\tau$

Définition 1. *On peut composer deux substituons σ_1 et σ_2 en une nouvelle substitution notée $\sigma_1\sigma_2$ définie par $(\sigma_1\sigma_2)(x) = \sigma_1(x)[\sigma_2]$. Pour tout terme t, on a $t[\sigma_1\sigma_2] = (t[\sigma_1])[\sigma_2]$.*

Remarque 1. *Il ne faut pas confondre la composition de deux substitutions qui correspond à deux substitutions successives avec la substitution simultanée. Ainsi si on pose $\sigma_1 = \{x \to y\}$ et $\sigma_2 = \{y \to z\}$. On a $\sigma_1\sigma_2 = \{x \to z; y \to z\}$ et $(x+y)[\sigma_1\sigma_2] = z + z$ alors que la substitution simultanée est définie par $\{x \to y; y \to z\}$ et donne le résultat $(x+y)[x \to y; y \to z] = y + z$.*

Exercice Prouvez que la composition de substitutions est associative, et que la substitution identité est son un élément neutre. Prouvez les relations suivantes : $t[] = t, \quad t(\tau \circ \sigma) = (t\sigma)_\tau$.

Définition 2. *Soient σ et τ deux substitutions. On dit que σ est plus générale que τ s'il existe une substitution ρ telle que $\tau = \rho \circ \sigma$.*

Exemple 1. *Soit*

$$\sigma = [f(w,x)/x, z/y], \qquad \tau = [f(g(y),x)/x, c/y, c/z, g(y)/w]$$

On a alors $\sigma \le \tau$, à cause de $\rho = [c/z, g(y)/w]$.

12.2 Unification

On voit que la méthode qui consiste à énumérer de manière naive l'ensemble des instances d'une formule n'est pas forcément efficace. En pratique, on préfère engendrer des instances qui vont pouvoir interagir avec d'autres parties de la formule pour conduire à une contradiction. On s'intéresse ici à résoudre des équations sur les termes avec variables. On s'intéresse à des substitutions dites à *support fini* c'est à dire telle que $\sigma(x) = x$ sauf pour un nombre fini de variables.

12.3 Algorithme d'unification

Définition 3. *Un problème d'unification est une liste* $(s_1, t_1), \ldots, (s_n, t_n)$ *avec* s_i, t_i *des termes. Une solution de ce problème—appelé unificateur de* $(s_1, t_1), \ldots, (s_n, t_n)$ *est une substitution* σ *telle que* $s_i\sigma = t_i\sigma$*, pour* $i = 1, \cdots, n$*. On notera Unif*$[(s_1, t_1), \ldots, (s_n, t_n)]$ *l'ensemble des unificateurs* $(s_1, t_1), \ldots, (s_n, t_n)$*.*

ALGORITHME D' UNIFICATION.

Le schéma général d'un algorithme d'unification est le suivant :

Entrée : un problème d'unification $(s_1, t_1), \ldots, (s_n, t_n)$
Sortie :
un `MGU` de $(s_1, t_1), \ldots, (s_n, t_n)$ si `UNIF` $[(s_1, t_1), \ldots, (s_n, t_n)] \neq \emptyset$
`ECHEC` sinon

---

1. Si $n = 0$, retourner la substitution identité
2. Sinon, on analyse le couple (s_1, t_1) :

a) si $s_1 = f(r_1, \ldots, r_k)$ et $t_1 = g(r'_1, \ldots, r'_{k'})$ alors
    si $f \neq g$, retourner `ECHEC`
    sinon /* $f = g$ implique $k = k'$ */
    retourner UNIFIER($(r1, r'_1), \ldots, (r_k, r'_k), (s_2, t_2), \ldots, (s_n, t_n)$)
b) si s_1 est la variable x, alors :
    si t_1 est aussi la variable x,
    retourner UNIFIER($(s_2, t_2), \ldots, (s_n, t_n)$)
    si $x \in VAR(t_1)$, retourner ECHEC
    sinon,
      soit τ le résultat de UNIFIER($(s_2[t_1/x], t_2[t_1/x]), \ldots, (s_n[t_1/x], t_n[t_1/x])$)
      si $\tau =$ `ECHEC`, retourner `ECHEC`
      sinon retourner $\tau \circ [t_1/x]$
c) si t_1 est la variable x, alors
    traitement comme auparavant, avec s_1 à la place de t_1

Exemple 2.

1. *La substitution* $\sigma = [g(z)/x, g(z)/y]$*. est un unificateur de* $(f(x, g(z)), f(g(z), y))$*, car* $f(x, g(z))[g(z)/x, g(z)/y] = f(g(z), g(z)) = f(g(z), y)[g(z)/x, g(z)/y]$*.*
2. *Nous avons Unif*$[(f(x, y), g(z))] = \emptyset$*. De même pour Unif*$[(x, g(x))]$ *.*

Proposition 1. *Si Unif* $[(s_1, t_1), \ldots, (s_n, t_n)] \neq \emptyset$*, alors il existe* $\sigma \in$ *Unif*$[(s_1, t_1), \ldots, (s_n, t_n)]$ *tel que* $\sigma \leq \tau$ *pour tout* $\tau \in$ *Unif*$[(s_1, t_1), \ldots, (s_n, t_n)]$*. On appelle un tel* σ *un unificateur le plus général des couples* $(s_1, t_1), \ldots, (s_n, t_n)$*. On dira aussi que* σ *un MGU des couples* $(s_1, t_1), \ldots, (s_n, t_n)$*, où MGU est un acronyme de l'anglais « Most General Unifier ».*

Exemple 3. $\tau = [g(f(w))/x, g(f(w))/y] \in$ *Unif*$[(f(x, g(z)), f(g(z), y))]$*, mais* τ *n'est pas un MGU de ce problème. En fait,* $\sigma = [g(z)/x, g(z)/y]$ *est un MGU, et on a* $\sigma \leq \tau$*, car* $\tau = \rho \circ \sigma$*, avec* $\rho = [f(w)/z]$*.*

12.4 Résolution pour le calcul des prédicats

Substitution sur les formules propositionnelles

L'action d'une substitution s'étend aisément aux formules sans quantificateurs :

- $R(t_1, \ldots, t_n)\sigma = R(t_1\sigma, \ldots, t_n\sigma)$
- $(\neg G)\sigma = \neg(G\sigma)$
- $(F\alpha G)\sigma = F\sigma\alpha G\sigma, \quad \alpha \in \{\wedge, \vee, \Rightarrow, \Leftrightarrow\}$

Lemme 1. *Soit F une formule sans quantificateur, σ une substitution. (I, e) une assignation. On a*

$$[F\sigma]_{(I,e)} = 1 \quad ssi \quad [F]_{(I,e\sigma)} = 1$$

où $e\sigma(x) = [\sigma(x)]_{(I,e)}$.

La preuve se fait par induction.

Nous pouvons généraliser la substitution à toute formule, c'est ce que nous avons fait dans le cours 10, où nous l'avons appelé instanciation.

Rappel : Un littéral est ou bien une formule atomique, ou bien la négation d'une formule atomique. Une clause universelle est la fermeture universelle d'une disjonction de littéraux.

Par exemple, nous allons considérer l'expression $R(x, y) \vee \neg Q(f(x), z)$ comme un raccourci de sa fermeture universelle : $\forall x \forall y \forall z (R(x, y) \vee \neg Q(f(x), z))$.

Lemme 2. *Pour toute substitution σ, la règle d'inférence suivante est correcte :*

$$\frac{C}{C\sigma}\sigma$$

C'est à dire si I est un modèle de C alors I est un modèle $C\sigma$.

Lemme 3. *σ est un unificateur de l_0 et l_1 ssi*
1. *$l_0 = R(s_1, \ldots, s_n), l_1 = R(t_1, \ldots, t_n)$, et $\sigma \in$ Unif $[(s_1, t_1), \ldots, (s_n, t_n)]$, ou bien*
2. *$l_0 = \neg R(s_1, \cdots, s_n), l_1 = \neg R(t_1, \ldots, t_n)$, et $\sigma \in$ Unif$[(s_1, t_1), \cdots, (s_n, t_n)]$.*

Du Lemme il en découle tout de suite que l'ensemble des unificateurs de deux littéraux, appelons encore une fois $Unif[l_0, l_1]$, ou bien il est vide, ou bien il possède un élément le plus général, qui sera appelé un MGU de l_0 et l_1.

12.4.1 Les règles du calcul de la résolution

On peut considérer le calcul de la résolution comme une généralisation de la méthode de la résolution propositionnelle. Comme dans le cas propositionnel, la méthode de résolution prend en paramètre un ensemble Γ de clauses et essaye de dériver la clause vide? depuis les clauses dans Γ. Les deux règles pour dériver des nouvelles clauses à partir de clauses déjà construites sont les suivantes :

$$\frac{C \vee A_0 \qquad C' \vee \neg A_1}{(C \vee C')_\sigma} \textbf{ Résolution}$$

où σ est un MGU des formules atomiques A_0 et A_1, et

$$\frac{C \vee l_0 \vee l_1}{(C \vee l_0)_\sigma} \textbf{ Factorisation}$$

où σ est un MGU de l_0 et l_1.

Exemple

$$\frac{\neg Q(x) \vee R(x, a) \quad \neg R(b, y) \vee P(x, y)}{\neg Q(b) \vee P(b, a)} \textbf{ Résolution}$$

Ici **Exemple**

$$\frac{\neg P(x) \vee R(x, y) \vee Q(x, a) \vee Q(b, y)}{\neg P(b) \vee R(b, a) \vee Q(b, a)} \textbf{ Factorisation}$$

Renommage

Définition 4. *Soient C une clause et σ une substitution, qui ne change que les variables de C et dont la restriction aux variables de C est une bijection entre ces variables et celles de la clause $C\sigma$. La clause $C\sigma$ est une copie de la clause C. La substitution σ est aussi appelée un renommage de C.*

Proposition 2. *Soient deux clauses copies l'une de l'autre, leurs fermetures universelles sont équivalentes.*

$P(x, y)$ et $P(u, v)$ sont copies l'une de l'autre. Les clauses sont équivalentes.

12.4.2 Théorème de correction

La méthode de résolution est correcte.

En fait, on peut penser que les règles du calcul sont obtenues comme synthèse de deux règles, l'une qui porte sur les quantificateurs universels, et l'autre étant la règle correspondante propositionnelle :

$$\frac{\dfrac{C \vee A_0}{C\sigma \vee A_0\sigma}\sigma \qquad \dfrac{C' \vee \neg A_1}{C'\sigma \vee \neg A_0\sigma}\sigma}{C\sigma \vee C'\sigma} \quad \text{Résolution propositionnelle}$$

où on a $A_0\sigma = A_1\sigma$

De façon semblable on déduit la correction de la factorisation :

$$\frac{\dfrac{C \vee l_0 \vee l_1}{C\sigma \vee l_0\sigma \vee l_0\sigma}\sigma}{C\sigma \vee l_0\sigma} \text{Factorisation propositionnelle}$$

12.4.3 Théorème de complétude

Théorème 1. *Si un ensemble de clauses Γ n'admet pas un modèle, alors il existe une preuve de $\perp$ à partir de Γ dans le calcul de la résolution.*

12.4.4 Indecidabilité.

Bien que le calcul soit correct et complet, nos résultats n'amènent pas à la construction d'un algorithme—c'est à dire une sorte de programme qui s'arrête toujours et qui donne la réponse souhaitée à la fin des calculs—pour décider si un ensemble de clauses universelles est satisfaisable ou non. Pour s'en apercevoir, il suffit de considérer l'exemple : $C := \{\neg P(x) \vee P(s(x)), P(0)\}$.

12.4.5 Exemples

Exemple 4. *Montrons par résolution que $\forall(C_1, C_2)$ n'a pas de modèle.*
$C_1 : P(x, y) \vee P(y, x)$.
$C_2 : \neg P(u, v) \vee \vee\neg P(v, u)$.
$C_3 : P(x, x)$ *Factorisation. MGU* $[y/x]$
$C_4 : \neg P(u, u)$ *Factorisation. MGU* $[v/u]$
$C_5 : \perp$ *Résolution(C_3, C_4). MGU* $[u/x]$

Cet exemple montre, a contrario, que la résolution binaire seule est incomplète, sans la factorisation, nous ne pouvons pas déduire la clause vide.

Exercice 1. *Montrer par résolution que* $\{\forall x\}$

Exercice 2. *Soient les trois clauses :*
$C_1 : \neg P(z, a) \vee \neg P(z, x) \vee \neg P(x, z)$.
$C_2 : P(z, f(z)) \vee P(z, a)$.
$C_3 : P(f(z), z) \vee P(z, a)$.

Montrer par résolution que l'ensemble des clauses $\{C_1, C_2, C_3\}$ est contradictoire.

Exercice 3. *On considère l'ensemble S de clauses :*

$$S = \{\neg P(x) \vee Q(y) \vee R(x, y), \neg P(x) \vee \neg Q(f(y)), P(f(x)) \vee R(x, y), \neg R(x, f(x))\}$$

Montrer par résolution que S est inconsistant.