Licence Informatique, semestre 5 — 2019–20

Eléments de logique pour l'informatique (Info 315) — 9 décembre 2019

`http://www.lri.fr/~paulin/Logique`

Feuille 8 - Résolution, Calcul des Séquents et Révisions

Exercice 1 Soit les formules :
$H_1 \stackrel{\text{def}}{=} (\exists x, P(x)) \Rightarrow \forall x, P(x)$, $H_2 \stackrel{\text{def}}{=} \forall x, (P(x) \vee Q(x))$ et $C \stackrel{\text{def}}{=} (\exists x, \neg Q(x)) \Rightarrow \forall x, P(x)$.

1. Mettre en forme clausale l'ensemble de formules $\{H_1, H_2, \neg C\}$.
2. Trouver des instances closes contradictoires de ces formules et dériver la clause vide par résolution propositionnelle.
3. Faire directement une dérivation de la clause vide en utilisant la résolution dans le calcul des prédicats (c'est-à-dire les règles de renommage, factorisation et résolution binaire).

Exercice 2 On rappelle qu'une relation binaire est totale si pour tout x, il existe un y tel que x est en relation avec y.

Utiliser la méthode de résolution pour prouver que toute relation binaire symétrique, transitive et totale est réflexive.

Exercice 3 *Résolution* Soit les trois clauses

$$\begin{array}{lcl} C_1 & \stackrel{\text{def}}{=} & \neg P(z,a) \vee \neg P(z,x) \vee \neg P(x,z) \\ C_2 & \stackrel{\text{def}}{=} & P(z,f(z)) \vee P(z,a) \\ C_3 & \stackrel{\text{def}}{=} & P(f(z),z) \vee P(z,a) \end{array}$$

1. donner une dérivation de la clause $\neg P(a,a)$
2. donner une dérivation des clauses $P(a,f(a))$ et $P(f(a),a)$
3. en déduire une dérivation de $\perp$

Exercice 4 *Logique et problème de Post*

Savoir si une formule logique du calcul des prédicats est valide est un *problème indécidable* : il n'existe pas d'algorithme qui étant donnée une formule A répond vrai si la formule est valide et faux sinon. Pour montrer ce résultat, on ramène un problème connu comme indécidable à celui de la validité d'une fomule logique.

On prend comme problème indécidable la *correspondance de Post*. Soit un ensemble à deux éléments a et b appelés *caractères*, et on appelle *mot* une suite finie de caractères. On note ϵ le mot vide qui correspond à une suite sans caractères et on note par simple juxtaposition uv, la concaténation de deux mots. On se donne un ensemble fini de couples de mots $\{(u_1,v_1);\ldots;(u_n,v_n)\}$. Une *correspondance de Post* est une manière d'assembler ces paires de mots en une séquence (non vide) $i_1,\ldots,i_k$ pour que le mot $u_{i_1}\ldots u_{i_k}$ formé en regardant la première composante soit le même que le mot $v_{i_1}\ldots v_{i_k}$ formé en regardant la seconde composante.

Exemple. Soit le système $S \stackrel{\text{def}}{=} \{(a,baa);(ab,aa);(bba,bb)\}$. On utilise les mêmes notations que précedemment, $u_1 \stackrel{\text{def}}{=} a, v_1 \stackrel{\text{def}}{=} baa, u_2 \stackrel{\text{def}}{=} ab, v_2 = \stackrel{\text{def}}{=} aa, u_3 \stackrel{\text{def}}{=} bba, v_3 \stackrel{\text{def}}{=} bb$. Le mot $u_3u_2u_3u_1 = bba\,ab\,bba\,a$ est égal au mot $v_3v_2v_3v_1 = bb\,aa\,bb\,baa$. On a donc bien une correspondance de Post pour ce système. Par contre si on regarde le système $\{(ab,aa);(bba,bb)\}$ on montre qu'il n'y a pas de solution : en effet un tel mot solution ne peut pas commencer par (ab,aa) (car cela donne des mots dont la deuxième lettre diffère) ni par (bba,bb) car alors le premier mot sera toujours plus long que le second.

On peut prouver qu'un système particulier a ou n'a pas de solution mais on ne peut pas décider en général (à l'aide d'un programme) si un ensemble de couples de mots $\{(u_1,v_1);\ldots;(u_n,v_n)\}$ admet une correspondance de Post.

Nous allons maintenant montrer comment on peut ramener ce problème a un problème logique.

Pour cela on considère un langage avec une constante e pour représenter le mot vide et deux symboles de fonction unaires a et b pour représenter des mots qui commencent par la lettre a ou la lettre b. Ainsi le mot baa est représenté dans la logique par le terme $b(a(a(e)))$. Le domaine de Herbrand (ensemble des termes clos) associé à ce langage correspond à l'ensemble des mots sur l'alphabet $\{a, b\}$ dans le sens où on peut faire correspondre un terme à un mot et réciproquement.

On introduit également un prédicat binaire P et les formules suivantes :

$$\begin{array}{llll} A_0 \stackrel{\text{def}}{=} P(e,e) & & A_1 \stackrel{\text{def}}{=} \forall xy, P(x,y) \Rightarrow P(a(x), b(a(a(y)))) \\ A_2 \stackrel{\text{def}}{=} \forall xy, P(x,y) \Rightarrow P(a(b(x)), a(a(y))) & & A_3 \stackrel{\text{def}}{=} \forall xy, P(x,y) \Rightarrow P(b(b(a(x))), b(b(y))) \end{array}$$

1. On étudie d'abord le lien entre une solution au problème de Post et le fait que la formule $\exists x, P(a(x), a(x)) \vee P(b(x), b(x))$ est conséquence logique de A_0, A_1, A_2, A_3.

   Pour cela on introduit une relation binaire sur d : $u \sim v$ si et seulement si il existe $k \geq 0$ et une séquence $i_1, \ldots, i_k$ (éventuellement vide) tels que le terme u correspond au mot $u_{i_1} \ldots u_{i_k}$ et le terme v correspond au mot $v_{i_1} \ldots v_{i_k}$.

   (a) Montrer que l'interprétation de Herbrand dans laquelle $P(u, v)$ est vrai si et seulement si $u \sim v$ est un modèle des formules A_0, A_1, A_2, A_3.

   (b) Montrer que si $u \sim v$ alors $A_0, A_1, A_2, A_3 \models P(u, v)$. On pourra raisonner par récurrence sur la taille de la séquence $i_1, \ldots, i_k$.

   (c) Montrer que $A_0 \models \exists x, P(x, x)$.

   (d) Montrer que si le système S admet une solution pour le problème de correspondance de Post alors $A_0, A_1, A_2, A_3 \models \exists x, (P(a(x), a(x)) \vee P(b(x), b(x)))$.

   (e) Sans utiliser le fait que l'on connait une solution pour le problème de Post du système S, montrer réciproquement que si $A_0, A_1, A_2, A_3 \models \exists x, (P(a(x), a(x)) \vee P(b(x), b(x)))$ alors le système de Post S admet une solution.

2. Dans cette partie on utilise la résolution pour "résoudre" le problème de Post en construisant une preuve de $\exists x, (P(a(x), a(x)) \vee P(b(x), b(x)))$

   (a) Mettre les formules A_0, A_1, A_2, A_3 en forme clausale.

   (b) A l'aide de la résolution montrer que $A_0, A_1, A_2, A_3 \models \exists z, P(b(z), b(z))$

3. L'objet de cette question est de bien faire la différence entre la validité d'une formule (vraie dans tous les modèles) et la satisfiabilité (vraie dans un modèle).

   (a) Montrer qu'il existe un modèle de Herbrand qui vérifie les quatres propriétés A_0, A_1, A_2, A_3 ainsi que la formule $\forall xy, P(x, y)$.

   (b) Montrer qu'il n'y a pas de solution au problème de Post S qui commence par la lettre a et donc que pour tout x on a $a(x) \not\sim a(x)$.

   (c) A-t-on $A_0, A_1, A_2, A_3 \models \forall x, \neg P(a(x), a(x))$?

Exercice 5 On rappelle qu'une relation binaire est totale si pour tout x, il existe un y tel que x est en relation avec y.

Utiliser le calcul des séquents pour prouver que toute relation binaire symétrique, transitive et totale est réflexive.

Exercice 6 Soit un langage avec un prédicat unaire p. Utiliser le système G pour montrer les séquents suivants :

1. $\vdash (\exists x, p(x)) \vee (\forall y, \neg p(y))$
2. $\vdash (\forall x, p(x)) \Rightarrow \neg \exists y, \neg p(y)$
3. $\vdash \exists x, \forall y, (p(y) \Rightarrow p(x))$

   (on introduira une constante `a`, qui sera notre première tentative pour la valeur x)

Rappel des règles pour les quantificateurs :

	gauche	droite
$\forall$	$\dfrac{P[x \leftarrow t], (\forall x, P), \Gamma \vdash \Delta}{(\forall x, P), \Gamma \vdash \Delta}$	$\dfrac{\Gamma \vdash \Delta, P}{\Gamma \vdash \Delta, (\forall x, P)}\ x \notin \mathsf{Vl}(\Gamma, \Delta)$
$\exists$	$\dfrac{P, \Gamma \vdash \Delta}{(\exists x, P), \Gamma \vdash \Delta}\ x \notin \mathsf{Vl}(\Gamma, \Delta)$	$\dfrac{\Gamma \vdash \Delta, (\exists x, P), P[x \leftarrow t]}{\Gamma \vdash \Delta, (\exists x, P)}$

Exercice 7 *Théories*

Soit un langage avec deux prédicats unaires p et q.

1. Le séquent suivant est-il valide ?

$$(\exists x, (p(x) \Rightarrow q(x))), (\exists x, p(x)) \vdash \exists x, q(x)$$

   Comment peut-on démarrer la preuve dans le système G ? que constate-t-on ?

2. Faire la preuve dans le système G du séquent

$$(\forall x, (p(x) \Rightarrow q(x))), (\exists x, p(x)) \vdash \exists x, q(x)$$

3. En déduire $(\forall x, (p(x) \Leftrightarrow q(x))) \models (\exists x, p(x)) \Leftrightarrow (\exists x, q(x))$.
4. On dit qu'une théorie $\mathcal{Th}$ admet l'élimination des quantificateurs si pour toute formule sans quantificateurs F avec $p+1$ variables, on peut trouver une formule sans quantificateur G avec p variables telles que

$$\mathcal{Th} \models \forall x_1, \ldots, x_p, ((\exists x, F(x_1, \ldots, x_p, x)) \Leftrightarrow G(x_1, \ldots, x_p))$$

   Montrer que si une théorie $\mathcal{Th}$ admet l'élimination des quantificateurs alors pour toute formule close A il existe une formule B close et sans quantificateur telle que $\mathcal{Th} \models A$ est vrai si et seulement si $\mathcal{Th} \models B$ est vrai.

On considère un symbole de relation binaire noté $<$ de manière infixe et la théorie qui dit que

— $<$ est un ordre strict (irréflexif et transitif)
— $<$ est un ordre dense : si $x < y$ alors il existe z tel que $x < z \wedge z < y$
— $<$ est un ordre total : si $x < y$ alors pour tout z, soit $x < z$, soit $z < y$
— $<$ n'a pas d'élément maximal ou minimal (tout élément admet un élément plus petit et un élément plus grand).

1. Donner les formules logiques qui correspondent à la théorie décrite ci-dessus (notée $\mathcal{Th}$).
2. Quelles sont les formules atomiques dans cette théorie ? les littéraux.
3. Proposer un modèle de cette théorie.
4. On note $y \leq x$ la propriété $\neg(x < y)$. Montrer que cette relation est réflexive et transitive. Montrer que $\forall x\, y\, z, x < y \wedge y \leq z \Rightarrow x < z$ et que $\forall x\, y, x < y \Rightarrow x \leq y$
5. On veut montrer que cette théorie admet l'élimination des quantificateurs. On cherche donc à montrer que si on a une formule de la forme $\exists x, F(x_1, \ldots, x_n, x)$ avec F sans quantificateur alors on peut trouver une formule propositionnelle équivalente $G(x_1, \ldots, x_n)$.
   (a) Proposer des formules équivalentes sans quantificateur pour les formules suivantes (y, z, y_i, z_i sont des variables libres)
   — $\exists x, (y < x \wedge z < y)$
   — $\exists x, (y < x \wedge x < z)$
   — $\exists x, (y \leq x \wedge x < z)$
   — $\exists x, (y_1 < x \wedge y_2 < x)$
   — $\exists x, (y_1 < x \wedge y_2 < x \wedge x < z_1 \wedge x < z_2)$
   (b) Montrer qu'il suffit de regarder le cas dans lequel F est une conjonction de littéraux.
   (c) Proposer un algorithme pour éliminer un quantificateur existentiel sur une formule sans quantificateur.

(d) Si on prend une formule existentielle close, comment peut-on déterminer si elle est ou non valide dans la théorie $\mathcal{Th}$.

Exercice 8 *Preuves, examen session 1, 2018-19*

On considère une logique dont la signature est composée d'une constante e et d'un symbole de prédicat F ternaire (à trois arguments).

On introduit un ensemble de trois formules logiques A, B et C (nommé Γ dans la suite) :

— $A \stackrel{\text{def}}{=} \forall x, F(x, e, x)$

— $B \stackrel{\text{def}}{=} \forall x, \exists y, F(x, y, e) \wedge F(y, x, e)$

— $C \stackrel{\text{def}}{=} \forall x\, y\, z\, t\, u, F(x, y, t) \wedge F(y, z, u) \Rightarrow \forall v, (F(t, z, v) \Leftrightarrow F(x, u, v))$

Questions :

1. Combien d'interprétations différentes y-a-t-il sur un domaine avec un seul élément e ? Pour chacune de ces interprétations, dire si chacune des propriétés A, B et C est vraie ou fausse dans l'interprétation. Est-ce que l'ensemble Γ est satisfiable ?
2. Combien d'interprétations différentes y-a-t-il sur un domaine avec deux éléments e et a ?
   Donner un modèle de l'ensemble des formules Γ sur ce domaine qui ne soit pas l'interprétation dans laquelle F est toujours vraie.
3. Soit Γ et Δ deux ensembles de formules et A et B deux formules. Montrer que la règle suivante (notée *mp*) est correcte. Est-elle inversible ?
   $$\frac{\Gamma, B \vdash \Delta}{\Gamma, A \Rightarrow B, A \vdash \Delta}$$
4. (Bonus) Soit D la formule $\forall x\, y, F(x, y, e) \wedge F(y, x, e) \Rightarrow F(e, x, x)$. Donner une dérivation dans le système G de $A, C \vdash D$
   (les règles du système G, dont celles pour le connecteur $\Leftrightarrow$ sont données à la fin du sujet et on pourra utiliser la règle *mp* introduite à la question précédente).
5. Mettre les formules B et D en forme clausale.
6. Montrer **en utilisant la résolution** que la formule $\forall x, F(e, x, x)$ est conséquence logique des formules B et D.
7. Peut-on en déduire que $\forall x, F(e, x, x)$ est conséquence logique des formules A, B et C ? Justifiez votre réponse.

Exercice 9 On se donne une signature avec une constante a, un symbole de fonction binaire f et deux symboles de prédicat P d'arité un et R d'arité deux.

1. Pour les formules suivantes, dire si elles sont bien formées syntaxiquement (c'est-à-dire qu'elles respectent les contraintes d'arité de la signature). Si oui, donner la forme arborescente en indiquant pour chaque variable si elle est libre ou liée. Si non expliquer pourquoi.
   (a) $(\forall x, P(f(x, a))) \Rightarrow \exists y, R(x, y)$
   (b) $(R(f(x, y), a)) \vee (\neg(\forall x, P(y)))$
   (c) $\neg(\neg R(f(f(a, b))) \Rightarrow \forall x, P(x))$
   (d) $(\forall x, R(x, f(x))) \Rightarrow \exists x, P(f(f(y, a), x))$
   (e) $(\forall x, R(f(x, y), P(x))) \wedge \exists y, R(y, x)$
2. On veut définir une fonction qui vérifie qu'une formule (bien formée syntaxiquement sur la signature donnée) est close (ne contient pas de variables libres). Pour cela on introduit deux fonctions `varterm` qui étant donnés un ensemble de variables V et un terme t renvoie `vrai` si toutes les variables du terme t appartiennent à l'ensemble V et faux sinon et une fonction `varform` qui étant donnés un ensemble de variables V et une formule A renvoie `vrai` si toutes les variables *libres* de la formule A appartiennent à l'ensemble V.
   (a) Donner les équations récursives qui définissent les fonctions `varterm` et `varform` (on pourra utiliser librement les fonctions de base sur les ensembles comme le test d'appartenance $x \in V$, l'ensemble singleton $\{x\}$ ou l'opération d'union $A \cup B$).
   (b) En déduire une fonction `clos` qui étend donnée une formule F renvoie `vrai` si cette formule est close.