IAO2 MEDIAN-2016

POUR S'ECHAUFFER

Calculer la composée $\sigma = \sigma_2 \circ \sigma_1$, avec :

$\sigma_1 = \{ < x \,.\, f(f(x)) > ; < y \,.\, g(f(x),y) > ; < z \,.\, g(x,f(y)) > \}$

$\sigma_2 = \{ < x \,.\, f(A) > ; < y \,.\, g(y,A) > ; < z \,.\, B > \}$

Cette question est très simple : il suffit, partant de x puis de y, d'appliquer σ_1 puis σ_2.

Donc : $x \xrightarrow{\sigma_1} f(f(x)) \xrightarrow{\sigma_2} f(f(f(A)))$

$y \longrightarrow g(f(x),y) \longrightarrow g(f(f(A)),g(y,A))$

$z \longrightarrow g(x),f(y)) \longrightarrow g(f(A),f(g(y,A)))$

Finalement, le résultat est :

$\sigma = \sigma_2 \circ \sigma_1 = \{ < x \,.\, f(f(f(A)) > ; < y \,.\, g(f(f(A)),g(y,A)) > ; < z \,.\, g(f(A),f(g(y,A))) > \}$

N-VALIDITE

Soit n un entier strictement positif. Une formule du calcul des prédicats (ω) est n-valide si et seulement si (ω) est vraie dans toute interprétation dont le domaine compte n éléments. Soit enfin (ω) : $[\forall x \;\; P(x,x)] \;\Rightarrow\; [\forall x \; \forall y \;\; P(x,y)]$.

1) Montrer par un bref raisonnement que la formule suivante est 1-valide.
2) Montrer par un contre-exemple qu'elle n'est pas 2-valide.

Question 1 :

Cette question est a priori simple, mais elle peut déstabiliser…. Soit D un domaine quelconque à un élément, que nous noterons a. Donc, D = {a}. Pour montrer $p \Rightarrow q$, on suppose p vraie et on en déduit que q est vraie. On suppose donc que, pour tout élément x du domaine D, x est en relation avec x par P. Donc, ici, P(a,a) est vraie. Dès lors, $\forall x \; \forall y \;\; P(x,y)$ est vraie car x et y ne peuvent prendre que la valeur a. En conséquence, si I est un modèle à un élément de $\forall x \; P(x,x)$, alors I est un modèle de $\forall x \; \forall y \;\; P(x,y)$. Donc ($\omega$) est 1-valide.

Question 2 :

C'est encore plus simple, mais sait-on jamais … ?

Soit D = {a ; b} un domaine quelconque à deux éléments. Remarque : on aurait tout aussi bien pu prendre D = {0 ; 1} ou autre. Soit par exemple I = < D, ψ > l'interprétation suivante :

$\psi : D \longrightarrow D$, avec : $\psi(a) = A$ et $\psi(b) = B$.

$\psi(P) : \;\; D^2 \longrightarrow \{V ; F\}$

$(A,A) \longrightarrow V$

$(B,B) \longrightarrow V$

$(A,B) \longrightarrow F$

$(B,A) \longrightarrow V$

On a $val_I(\forall x \; P(x,x)) = V$ et $val_I(\forall x \; \forall y \;\; P(x,y)) = F$. Donc, $val_I(\omega) = F$, et (ω) n'est pas 2-valide.

POLITIQUE

Soit la phrase : « Les hommes politiques qui ne résolvent aucun problème ne gagnent pas d'élection ».

1) Traduire cette phrase en Calcul des Prédicats. On utilisera les prédicats :
   POL(x) : "x est un homme politique".
   PB(x) : "x est un problème".
   RES(x,y) : "x résout y".
   GAG(x,y) : "x gagne y »
   ELE(x) : "x est une élection ».
   Deux conseils :
   - procéder par étape (fortement recommandé)
   - et faire extrêmement attention au parenthésage, faute de quoi la question suivante pourrait s'avérer très problématique….
2) Mettre la formule obtenue sous forme clausale.

Question 1 :

Le principe désormais bien connu est de découper la phrase en niveaux bien distincts.

Niveau 1 : « les hommes politiques qui vérifient la propriété P vérifient la propriété Q ».

$\forall x\, ((\, pol(x) \wedge P(x)) \Rightarrow Q(x))$

Niveau 2 pour P(x) : « x ne résout aucun problème ».

$\forall z\, (pb(z) \Rightarrow \neg res(x,z))$

Niveau 2 pour Q(x) : « x ne gagne aucune élection ».

$\forall y\, (ele(y) \Rightarrow \neg gag(x,y))$

En définitive, la phrase logique est :

$\forall x\, ((pol(x) \wedge \forall z\, (pb(z) \Rightarrow \neg res(x,z))) \Rightarrow \forall y\, (ele(y) \Rightarrow \neg gag(x,y)))$

Il faut donc comprendre que la sous-formule $\forall z\, (pb(z) \Rightarrow \neg res(x,z))$ appartient bien à la partie antécédente de l'implication de niveau supérieur, circonscrite par les parenthèses rouges.

Question 2 :

Il n'y a pas de difficultés particulières ici, c'est du "classique".

Etape 1, élimination des $\Rightarrow$:

$\forall x\, (\neg(pol(x) \wedge \forall z\, (\neg pb(z) \vee \neg res(x,z))) \vee \forall y\, (\neg ele(y) \vee \neg gag(x,y)))$

Etape 2, propagation du $\neg$:

$\forall x\, ((\neg pol(x) \vee \exists z\, (pb(z) \wedge res(x,z))) \vee \forall y\, (\neg ele(y) \vee \neg gag(x,y)))$

Etape 4, skolémisation (de type II) :

$\forall x\, ((\neg pol(x) \vee (pb(f(x)) \wedge res(x,f(x)))) \vee \forall y\, (\neg ele(y) \vee \neg gag(x,y)))$

Etape 5, suppression des quantifications universelles :

$(\neg pol(x) \vee (pb(f(x)) \wedge res(x,f(x)))) \vee (\neg ele(y) \vee \neg gag(x,y))$

Ou encore :

$\neg pol(x) \vee \neg ele(y) \vee \neg gag(x,y) \vee (pb(f(x)) \wedge res(x,f(x)))$

Etape 6, mise sous forme conjonctive normale :

$(\neg pol(x) \vee \neg ele(y) \vee \neg gag(x,y) \vee pb(f(x))) \wedge (\neg pol(x) \vee \neg ele(y) \vee \neg gag(x,y) \vee res(x,f(x)))$

Etape 7, séparation des clauses, avec renommage des variables :

$\{\neg \text{pol}(x) \vee \neg \text{ele}(y) \vee \neg \text{gag}(x,y)) \vee \text{pb}(f(x))\}$

$\{\neg \text{pol}(u) \vee \neg \text{ele}(v) \vee \neg \text{gag}(u,v)) \vee \text{res}(u,f(u))\}$

TOLERANCE

1) Mettre sous la forme d'expressions du calcul des prédicats L1 les énoncés ci-dessous. On utilisera à cet effet les prédicats suivants (ou plutôt leurs abréviations soulignées) :

<u>RES</u>TAURANT (x)	*x va au restaurant*
<u>ACC</u>OMPAGNER (x,y)	*x est accompagné de y*
<u>CAR</u>NIVORE (x)	*x est carnivore*
<u>VEG</u>ETARIEN (x)	*x est un végétarien*
<u>BOU</u>CHER (x)	*x est un boucher*

H1 "Tous ceux qui vont au restaurant sont carnivores ou sont accompagnés par un boucher."

H2 "Certains végétariens vont au restaurant, et ne sont alors accompagnés que de végétariens."

H3 "Aucun végétarien n'est carnivore."

G : "certains végétariens sont bouchers."

2) On veut montrer, à l'aide d'une démonstration par réfutation, que, étant donnés H1, H2 et H3, alors il en résulte G. Il est demandé de mettre le problème sous forme clausale. On donnera le détail des calculs (sur la feuille de gauche). :

3) Effectuer la démonstration et en tirer les conséquences.

<u>Question 1 :</u>

H1 : le $\vee$ inclusif est ici correct, en revanche, le "ou bien" serait inapproprié. Le parenthésage de la partie conséquente est hautement souhaitable.

$\forall x\ (\text{res}(x) \Rightarrow (\text{car}(x) \vee \exists y\ (\text{acc}(x,y) \wedge \text{bou}(y))))$

H2 : le "alors" sert à souligner que ceux qui ne sont accompagnés que de végétariens sont bien les "certains végétariens" dont il est question juste auparavant. De plus, la phrase ne dit pas que ceux-là sont nécessairement accompagnés, mais que s'ils le sont, c'est forcément par un végétarien. Cela dit, si vous avez ajouté le fait qu'ils étaient accompagnés, ceci sera sans incidence, sinon l'ajout d'une clause qui s'avèrera inutile.

$\exists x\ (\text{veg}(x) \wedge \text{res}(x) \wedge \forall y\ (\text{acc}(x,y) \Rightarrow \text{veg}(y)))$

H3 : sans problème.

$\forall x\ (\text{veg}(x) \Rightarrow \neg \text{car}(x))$

G : $\exists y\ (\text{veg}(x) \wedge \text{bou}(y))$

<u>Question 2 :</u>

H1 : détails, sans commentaires.

$\forall x\ (\neg \text{res}(x) \vee (\text{car}(x) \vee \exists y\ (\text{acc}(x,y) \wedge \text{bou}(y))))$

$\forall x\ (\neg\, res(x) \vee (car(x) \vee (acc(x,f(x)) \wedge bou(f(x)))))$

$\neg\, res(x) \vee car(x) \vee (acc(x,f(x)) \wedge bou(f(x)))$

$(\neg\, res(x) \vee car(x) \vee acc(x,f(x))) \wedge (\neg\, res(x) \vee car(x) \vee bou(f(x)))$

C1 : $\{\neg\, res(x) \vee car(x) \vee acc(x,f(x))\}$

C2 : $\{\neg\, res(y) \vee car(y) \vee bou(f(y))\}$

H2 : idem

$\exists x\ (veg(x) \wedge res(x) \wedge \forall y\ (acc(x,y) \Rightarrow veg(y))))$

$\exists x\ (veg(x) \wedge res(x) \wedge \forall y\ (\neg\, acc(x,y) \vee veg(y))))$

$veg(A) \wedge res(A) \wedge \forall y\ (\neg\, acc(A,y) \vee veg(y))))$

$veg(A) \wedge res(A) \wedge (\neg\, acc(A,y) \vee veg(y))$

C3 : $\{veg(A)\}$

C4 : $\{res(A)\}$

C5 : $\{\neg\, acc(A,z) \vee veg(z)\}$

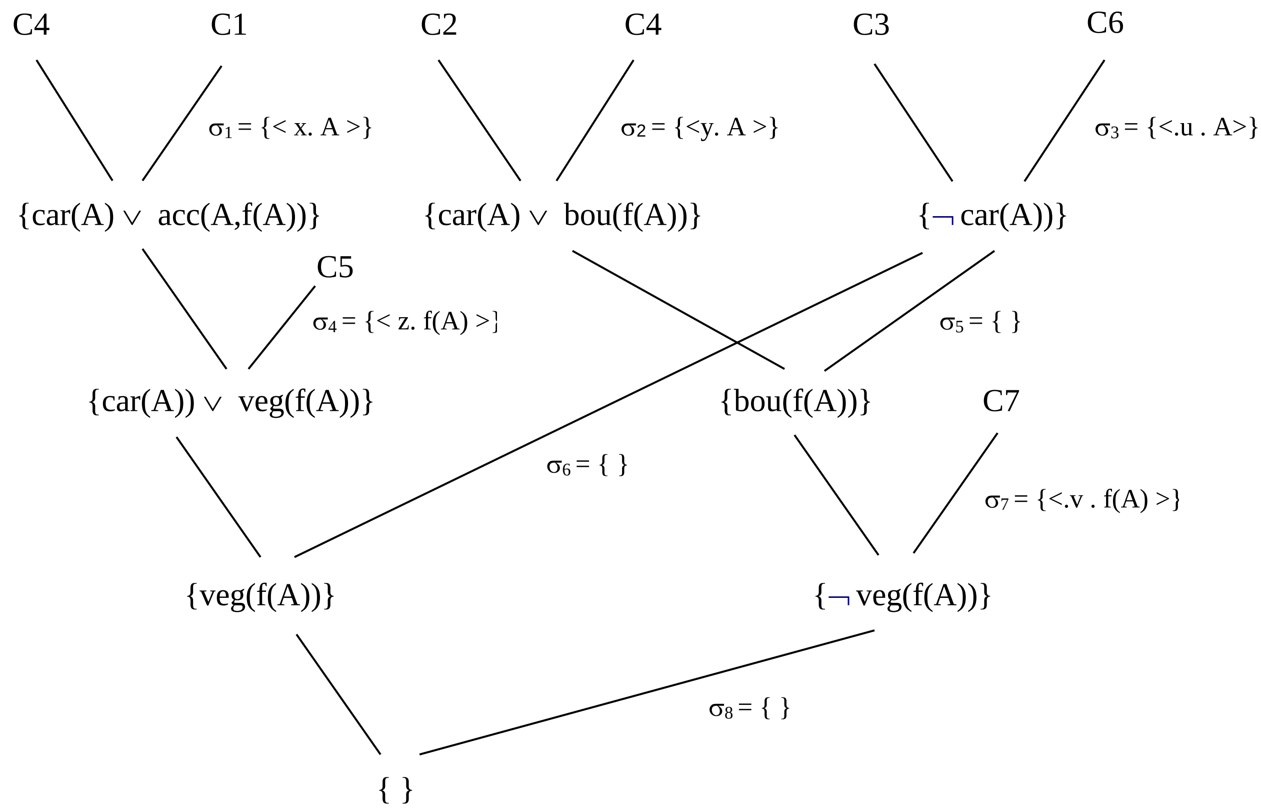
H3 : C6 : $\{\neg\, veg(u) \vee \neg\, car(u)\}$

$\neg$ G : pour la réfutation bien sûr

C7 : $\{\neg\, veg(v) \vee \neg\, bou(v)\}$

Question 3 :

Nous allons essayer de privilégier les constantes.



Il va de soi que ce graphe n'est qu'un exemple parmi une multitude ! On peut notamment faire intervenir C7 beaucoup plus tôt. Quoi qu'il en soit, on peut tirer les conséquences suivantes :

- Théorique : G est la conséquence logique de {H1 ; H2 ; H3}, autrement dit, quand les hypothèses sont tenues pour vraies, alors « il existe bien au moins un végétarien qui est boucher ». Et après tout pourquoi pas ? (sic)
- Pratique : on a v ——> f(A) par la substitution σ_7. Donc, ce végétarien est un boucher en tant qu'accompagnateur d'un végétarien !

UNE BELLE INCONNUE

On considère les formules suivantes :

R1 : $\forall e\ APPEND(NIL,e,e)$

R2 : $\forall x\ \forall y\ \forall z\ \forall t\ [\ APPEND(x,y,z) \Rightarrow APPEND(cons(t,x),y,cons(t,z))\ ]$

R3 : $\underline{INC}(NIL,NIL)$

R4 : $\forall u\ \forall v\ \forall r\ \forall s\ \mathbf{(}\ (\ \underline{INC}(u,v) \wedge\ APPEND(v,cons(r,N\ \ \Rightarrow\ \ \underline{INC}(cons(r,u),s)\ \mathbf{)}$

1) Appliquer la stratégie Prolog à la question : $\exists\omega\ \ INC\ (cons(1,cons(2,NIL)),\omega)$, où *INC* est l'abréviation *de INCONNUE.* On pratiquera le renommage "à la Prolog".

2) Par quel terme plus explicite pourrait-on remplacer le prédicat INC ? Donner une signification de la règle R4.

3) Question de cours : en quoi la formulation de APPEND est-elle remarquable ? Quels problèmes permet-elle de résoudre ?

Question 1 :

Ecrivons les formules sous la forme de clauses. C'est très simple puisque ce sont déjà des clauses de Horn…laissées en implication.

C1 : $\{APPEND(NIL,e,e)\}$

C2 : $\{\neg APPEND(x,y,z) \vee APPEND(cons(t,x),y,cons(t,z))\}$

C3 : $\{INC(NIL,NIL)$

C4 : $\{\neg INC(u,v)\ \vee\ \neg APPEND(v,cons(r,NIL),s)\ \vee\ INC(cons(r,u),s)\}$

$\neg\omega$: $\{\neg INC(cons(1,cons(2,NIL)),\omega)\}$

Dès lors, on obtient :

C4 $\qquad \neg\omega$

$\sigma_1 = \{< r.\ 1 > ; < u\ .\ cons(2,NIL) > ; < s\ .\ \omega > \}$

C4 $\qquad \{\neg INC(cons(2,NIL),v)\ \vee\ \neg APPEND(v,cons(1,NIL),\ \omega)\}$

$\sigma_2 = \{< r_1.\ 2 > ; < u_1\ .\ NIL) > ; < s_1\ .\ v > \}$

$\{\neg INC(NIL,v_1)\ \vee\ \neg APPEND(v_1,cons(2,NIL),v)\ \vee\ \neg APPEND(v,cons(1,NIL),\ \omega)\}$

C3

$\sigma_3 = \{< v_1.\ NIL > \}$

$\{\neg APPEND(NIL,cons(2,NIL),v)\ \vee\ \neg APPEND(v,cons(1,NIL),\ \omega)\}$

C1

$\sigma_4 = \{< e.\ cons(2,NIL) > ; < v\ .\ cons(2,NIL) > \}$

$\{\neg APPEND(cons(2,NIL),cons(1,NIL),\ \omega)\}$

C2

$\sigma_5 = \{< t.\ 2 > ; < x\ .\ NIL > ;$
$< y\ .\ cons(1,NIL) > ; < \omega\ .\ Cons(2,z) > \ \}$

$\{\neg APPEND(NIL,cons(1,NIL),z\}$

C1

$\sigma_6 = \{< e_1.\ Cons(1,NIL) > ; < z\ .\ cons(1,NIL) > \}$

$\{\ \}$

On remarque qu'à quelques reprises il a été nécessaire de renommer les variables des règles avant de réemployer celles-ci, selon les principes de renommage spécifiques à Prolog. En revanche, selon ces mêmes principes, il n'a jamais été nécessaire de renommer les variables des résolvants.

Remarque : dès l'instant où il ne restait plus que des APPEND à éliminer, on savait d'ores et déjà ce que serait le résultat final….

Résultat :

- Théorique : ω est la conséquence logique de {R1 ; R2 ; R3 ; R4}.
- Pratique, à mettre en exergue en Prolog :

$$\omega \xrightarrow{\sigma_1} \mathrm{cons}(2,z) \xrightarrow{\sigma_2} \mathrm{cons}(2,\mathrm{cons}(1,\mathrm{NIL}))$$

Et donc : < ω . cons(2,cons(1,NIL)) >

Question 2 :

Un exemple ne fait pas loi, évidemment. Cependant, il peut déclencher une intuition, non ? A savoir que ce programme aurait pour effet de créer la liste inverse de la liste d'entrée.
Une lecture attentive des règles R3 et surtout R4 pourrait confirmer cette intuition :
R4 : « si v est la liste inverse de u et si la concaténée de (r . nil) et de v est égale à la liste s, alors la concaténée de (r . nil) et de u est l'inverse de la liste s, ce qui semble frappé au coin du bon sens !
Donc, le prédicat INC pourrait être interprété en INV.

Question 3 :

Vu en TD …., mais assimilable à du cours, voici quelques exemples. Pour APPEND(u,v,w) :

- Si on connaît les listes u et v, alors renvoi de w concaténée de u et v.
- Si on connaît les listes u et w, alors renvoi de la liste v qui concaténée à u va donner w.
- Si on connait les listes v et w, alors renvoi de la liste u qui concaténée à v va donner w.
- Si connaît la liste w, alors renvoi de tous les couples (u,v) de listes qui concaténées vont donner w.

Au-delà, ce qui rend un tel prédicat remarquable, c'est d'une part la concision de sa formulation, et plus fort encore le fait que tous les exemples d'application supra ne résultent pas de modifications du programme initial, mais bien différemment de questions posées différentes, sans que celui-ci soit en quoi que ce soit changé !