Licence Informatique, semestre 5 2019-20
Eléments de logique pour l'informatique (Info 315) 21 mai 2021

Examen - 8 janvier 2020

L'examen dure 2 heures. L'énoncé est composé de 4 pages. Toutes les réponses devront être clairement justifiées. Le seul document autorisé est une page A4 manuscripte recto-verso. Le tableau résumant les règles du système G est donné à la fin du sujet.
Inscrivez votre nom sur chaque copie et numérotez-les. Cacheter toutes les copies. Recopier le numéro d'anonymat sur les intercalaires et sur le QCM.

Exercice 1 *QCM (8 points)*

Le **numéro d'anonymat** de la copie principale (pas le numéro d'étudiant) doit être reporté sur l'énoncé du QCM. Utiliser un style bleu ou noir pour cocher les cases. **N'oubliez pas de rendre le QCM avec vos copies.**

Exercice 2 *Résolution (4 points)*

Soit la formule $P \stackrel{\text{def}}{=} (\exists x, \forall y, R(x,y)) \Rightarrow (\forall y, \exists x, R(x,y))$

1. Skolemiser la formule $\neg P$.

   Correction :

   $\neg((\exists x, \forall y, R(x,y)) \Rightarrow (\forall y, \exists x, R(x,y))) \equiv (\exists x, \forall y, R(x,y)) \wedge (\exists y, \forall x, \neg R(x,y)))$ *La skolemisation introduit deux constantes a et b. On obient la formule* $(\forall y, R(a,y)) \wedge (\forall x, \neg R(x,b))$

2. Donner le domaine de Herbrand et la base de Herbrand associés à la formule skolemisée précédente.

   Correction : *La domaine de Herbrand est formé de deux éléments* $\{a,b\}$ *et la base de Herbrand a 4 formules atomiques* $\{R(a,a), R(a,b), R(b,a), R(b,b)\}$

3. En utilisant la méthode de résolution, montrer que P est valide.

   Correction : *A partir de la forme de skolem précédente, on trouve deux clauses* $R(a,y)$ *et* $\neg R(x,b)$ *qu'il suffit de combiner avec la substitution* $\{x \leftarrow a; y \leftarrow b\}$ *pour déduire la clause vide.*

4. Donner une interprétation dans laquelle la réciproque $(\forall y, \exists x, R(x,y)) \Rightarrow (\exists x, \forall y, R(x,y))$ est fausse.

   Correction : *On prend une interprétation avec deux constantes a et b. On prend* $R(a,a)$ *et* $R(b,b)$ *vrais. On a bien* $(\forall y, \exists x, R(x,y))$ *vrai dans cette interprétation mais* $(\exists x, \forall y, R(x,y))$ *est faux.*

Exercice 3 *Filtrage sur les termes (4 points).*

On considère un ensemble de termes construits sur une signature $\mathcal{F}$ qui comporte une constante `a`, un symbole de fonction unaire `f` et un symbole de fonction binaire `g`. On notera $\mathcal{T}$ l'ensemble des termes avec variables construits sur la signature $\mathcal{F}$.

1. Définir à l'aide d'équations récursives une fonction `vars` qui calcule pour un terme de $\mathcal{T}$ l'ensemble des variables qui apparaissent dans ce terme.

   Correction :

   $$\begin{array}{rlrl} \texttt{vars}(x) & = \{x\} \text{ si } x \text{ est une variable} & \texttt{vars}(\texttt{a}) & = \emptyset \\ \texttt{vars}(\texttt{f}(t)) & = \texttt{vars}(t) & \texttt{vars}(\texttt{g}(t,u)) & = \texttt{vars}(t) \cup \texttt{vars}(u) \end{array}$$

2. On appelle *motif* un terme de $\mathcal{T}$ dans lequel chaque variable apparaît au plus une fois.

   (a) Donner un exemple de terme qui n'est pas un motif.

   Correction : *avec x une variable,* $\texttt{vars}(\texttt{g}(x,x))$ *n'est pas un motif.*

(b) Définir à l'aide d'équations récursives une fonction `motif` qui étant donné un terme t de $\mathcal{T}$, renvoie un booléen qui est vrai exactement lorsque t est un motif.

Correction :

$$\begin{aligned} \texttt{motif}(x) &= \textit{vrai si } x \textit{ est une variable} \\ \texttt{motif}(\texttt{a}) &= \textit{vrai} \\ \texttt{motif}(\texttt{f}(t)) &= \texttt{motif}(t) \\ \texttt{motif}(\texttt{g}(t,u)) &= \texttt{motif}(t) \textit{ et } \texttt{motif}(u) \textit{ et } \texttt{vars}(t) \cap \texttt{vars}(u) = \emptyset \end{aligned}$$

La fonction décrite ci-dessus est mathématiquement correcte mais correspond à un algorithme peu efficace. Pour implanter une telle opération, il est judicieux d'introduire une fonction auxiliaire `motife` *qui prend en argument une liste de variable l et un terme t qui vérifie que t est un motif dont les variables sont m et n'apparaissent pas dans l'ensemble l. La fonction renvoie soit l'information* `PasUnMotif`*, soit l'information* $\texttt{Motif}(l \cup m)$*.*

$$\begin{aligned} \texttt{motife}(l,x) &= \textit{si } x \in l \textit{ alors } \texttt{PasUnMotif} \textit{ sinon } \texttt{Motif}(l \cup \{x\}) \\ &\quad \textit{(lorsque } x \textit{ est une variable)} \\ \texttt{motife}(l,\texttt{a}) &= \texttt{Motif}(l) \\ \texttt{motife}(l,\texttt{f}(t)) &= \texttt{motife}(l,t) \\ \texttt{motife}(l,\texttt{g}(t,u)) &= \textit{si } \texttt{motife}(l,t) = \texttt{PasUnMotif} \textit{ alors } \texttt{PasUnMotif} \\ &\quad \textit{sinon si } \texttt{motife}(l,t) = \texttt{Motif}(m) \textit{ alors } \texttt{motife}(m,u) \end{aligned}$$

$$\texttt{motif}(t) = \texttt{motife}(\emptyset,t) \neq \texttt{PasUnMotif}$$

3. Soient deux termes p et t de $\mathcal{T}$, tels que p est un motif, on dit que t est une instance de p s'il existe une substitution telle que $\sigma(p) = t$.

(a) Définir à l'aide d'équations récursives une fonction `match` qui étant donnés deux termes p et t de $\mathcal{T}$, tels que p est un motif, cherche s'il existe une substitution telle que $\sigma(p) = t$. La fonction `match` renvoie la substitution si elle existe et échoue sinon.

Correction : *La fonction se définit de manière récursive sur le premier argument (le motif)*

$$\begin{aligned} \texttt{match}(x,t) &= \{x \leftarrow t\} \\ \texttt{match}(\texttt{a},t) &= \textit{si } t = \texttt{a} \textit{ alors } \emptyset \textit{ sinon } \texttt{echec} \\ \texttt{match}(\texttt{f}(p),t) &= \textit{si } t = \texttt{f}(u) \textit{ alors } \texttt{match}(p,u) \textit{ sinon } \texttt{echec} \\ \texttt{match}(\texttt{g}(p,q),t) &= \textit{si } t = \texttt{g}(u,v) \textit{ alors } \texttt{match}(p,u) \cup \texttt{match}(q,v) \textit{ sinon } \texttt{echec} \end{aligned}$$

(b) Quel est le résultat de la fonction `match` lorsque $p = g(x,y)$ et $t = g(f(x),a)$?

Correction : *le résultat est la substitution $\{x \leftarrow f(x); y \leftarrow a\}$. Dans la cas du filtrage du motif p avec le terme t, il faut une substitution σ telle que $p\sigma = t$ (contrairement à l'unification pour laquelle la substitution s'applique aux deux termes. Le remplacement de x par $f(x)$ ne pose donc pas de problème.*

(c) En quoi le fait que p est un motif simplifie-t-il la procédure par rapport au cas général ?

Correction :

Dans le cas d'un motif $\texttt{g}(p,q)$ *les variables de p sont distinctes des variables de q. Les deux substitutions solutions de* $\texttt{match}(p,u)$ *et* $\texttt{match}(q,v)$ *ne portent donc pas sur les mêmes variables, il sufit de les réunir. Dans le cas contraire il faudrait vérifier que les termes substitués dans les deux parties sont égaux.*

Par exemple le $\texttt{g}(x,x)$ *ne filtre pas le terme* $\texttt{g}(\texttt{a},y)$

Exercice 4 *Modélisation cryptographique (5 points).*

Les protocoles cryptographiques sont utilisés pour garantir des transactions sur des réseaux qui sont non sûrs avec des acteurs qui peuvent tricher. De nombreux protocoles utilisés en pratique ont des failles, même si on suppose que les clés de cryptage sont sûres. La logique est utilisée pour modéliser ces protocoles et en garantir certaines propriétés. Nous donnons ici un exemple très simplifié.

— Les *acteurs* sont les entités qui souhaitent échanger.

— Les communications se font par échange de *messages*, tout ce qui est envoyé est vu par tout le monde, les messages peuvent également être transformés avant d'être renvoyés.

— Notre modélisation logique va utiliser les symboles de prédicats suivants qui sont tous unaires :
  — $\texttt{acteur}(a)$ est vrai si a est un acteur
  — $X(m)$ est vrai s'il y a diffusion possible du message m
— Un symbole de fonction binaire p permet de mettre bout à bout deux messages pour en faire un nouveau.
— La théorie suppose que les identifiants des acteurs peuvent circuler comme des messages et que l'on peut former de nouveaux messages en les mettant bout à bout ou bien en les séparant. On a donc les deux propriétés suivantes que l'on supposera vraies dans la suite :

$\mathrm{P}_1 : \forall z, \texttt{acteur}(z) \Rightarrow X(z)$

$\mathrm{P}_2 : \forall m_1\, m_2, X(p(m_1, m_2)) \Leftrightarrow (X(m_1) \wedge X(m_2))$.

Le protocole qui nous intéresse concerne une télévision avec un décodeur dans lequel est insérée une carte à puce. La carte et le décodeur sont liés à un utilisateur x. La transaction se passe entre le décodeur et la carte à puce de x. Le décodeur vérifie auprès de la carte à puce que l'abonnement de x est à jour. Pour cela il envoie un message qui est composé de son nom et d'une donnée s. La carte à puce vérifie que l'utilisateur a bien payé et renvoie le message s associé à son propre nom, le décodeur en recevant ce message en déduit que l'abonnement est réglé.

Le protocole se modélise en introduisant deux symboles de fonction unaires pour représenter des acteurs à savoir le décodeur $D(x)$ et la carte à puce $C(x)$ associés à l'utilisateur x, un symbole de constante $\texttt{s}$ pour représenter le message échangé ainsi qu'un symbole de prédicat unaire $\texttt{paye}(x)$ qui est vérifié si l'utilisateur x est à jour de son abonnement.

On introduit des nouveaux axiomes dans la théorie, les deux premiers spécifient que la carte à puce et le décodeur sont des acteurs et le suivant modélise la transaction souhaités à savoir la carte répond au message envoyé par le décodeur après vérification du paiement :

$\mathrm{P}_3 : \forall x, \texttt{acteur}(C(x))$

$\mathrm{P}_4 : \forall x, \texttt{acteur}(D(x))$

$\mathrm{P}_5 : \forall x, X(p(D(x), s)) \wedge \texttt{paye}(x) \Rightarrow X(p(C(x), s))$

Questions

1. On note $\mathcal{T}h_0$ la théorie dans laquelle les 5 propriétés énoncées ci-dessus (P_1 jusqu'à P_5) sont vraies.
   (a) Montrer par la méthode de résolution que $\mathcal{T}h_0 \models \forall x, X(p(D(x), s)) \Rightarrow X(p(C(x), s))$.

   Correction : *Mise en forme clausale :*

   P_1 *: 1 clause :* $C_1 \stackrel{\text{def}}{=} \neg \texttt{acteur}(z) \vee X(z)$

   P_2 *: 3 clauses :* $C_2 \stackrel{\text{def}}{=} \neg X(p(m_1, m_2)) \vee X(m_1)$, $C_3 \stackrel{\text{def}}{=} \neg X(p(m_1, m_2)) \vee X(m_2)$ *et* $C_4 \stackrel{\text{def}}{=} \neg X(m_1) \vee \neg X(m_2) \vee X(p(m_1, m_2))$.

   P_3 *: 1 clause :* $C_5 \stackrel{\text{def}}{=} \texttt{acteur}(C(x))$

   P_4 *: 1 clause :* $C_6 \stackrel{\text{def}}{=} \texttt{acteur}(D(x))$

   P_5 *: 1 clause :* $C_7 \stackrel{\text{def}}{=} \neg X(p(D(x), s)) \vee \neg \texttt{paye}(x) \vee X(p(C(x), s))$

   P_6 : $\neg \forall x, X(p(D(x), s)) \Rightarrow X(p(C(x), s)) \equiv \exists x, X(p(D(x), s)) \wedge \neg X(p(C(x), s))$, *la skolemisation introduit une constante a et on obtient deux clauses* $C_8 \stackrel{\text{def}}{=} X(p(D(a), s))$ *et* $C_9 \stackrel{\text{def}}{=} \neg X(p(C(a), s))$

   Résolution :
   — C_9 *avec* C_4 *et* $\{m_1 \leftarrow C(a); m_2 \leftarrow s\}$ *donne* $C_{10} \stackrel{\text{def}}{=} \neg X(s) \vee \neg X(C(a))$.
   — C_8 *avec* C_3 *et* $\{m_1 \leftarrow D(a); m_2 \leftarrow s\}$ *donne* $C_{11} \stackrel{\text{def}}{=} X(s)$.
   — C_{10} *avec* C_{11} *donne* $C_{12} \stackrel{\text{def}}{=} \neg X(C(a))$.
   — C_1 *avec* C_5 *et* $\{z \leftarrow C(x)\}$ *donne* $C_{13} \stackrel{\text{def}}{=} X(C(x))$.
   — C_{12} *avec* C_{13} *et* $\{x \leftarrow a\}$ *donne la clause vide.*

(b) L'axiome P_5 est-il utile pour montrer ce résultat ? pouvait-on l'éliminer a priori ?

Correction : *La clause C_7 correspondant à l'axiome P_5 n'est pas utilisée. Comme aucune une autre clause ne mentionne le prédicat* `paye`*, cette clause ne peut pas mener à la clause vide et donc aurait pu être éliminée a priori.*

(c) Est-il possible pour le décodeur de recevoir le message $X(p(C(x), s))$ alors que x n'a pas payé son abonnement ?

Correction : *oui, le message $X(p(C(x), s))$ peut toujours être forgé à partir de la demande initiale $X(p(D(x), s))$ du décodeur sans que la carte ne vérifie le paiement.*

2. Pour pallier au problème précédent, le protocole est modifié pour utiliser deux messages différents s_1 et s_2 qui ne sont a priori connus que de la carte et du décodeur. Le décodeur envoie le message s_1 et la carte lui renvoie le message s_2. On modifie donc l'axiome P_5 qui devient : $\forall x, X(p(D(x), s_1)) \wedge \mathtt{paye}(x) \Rightarrow X(p(C(x), s_2))$. On obtient une nouvelle théorie $\mathcal{T}h_1$.

(a) Proposer une formule T qui dit que si un des utilisateurs a payé alors tous les utilisateurs peuvent tricher (à savoir faire aboutir la transaction entre leur décodeur et leur carte sans avoir payé).

Correction : $T \stackrel{\text{def}}{=} (\exists x, \mathtt{paye}(x)) \Rightarrow \forall y, (X(p(D(y), s_1)) \Rightarrow X(p(C(y), s_2))$

(b) Donner de manière informelle les étapes d'échanges de messages qui permettent à un utilisateur de faire croire qu'il a payé en indiquant à chaque fois quelle propriété est utilisée.

Correction : *Soit a l'utilisateur qui a payé et b celui qui veut tricher. L'utilisateur b récupère le message de son décodeur $p(D(b), s_1)$ et le transforme en un message du décodeur de a $p(D(a), s_1)$. Le message est envoyé à la carte de a qui vérifie qu'il a payé et renvoie le message $p(C(a), s_2)$. Ce message est à nouveau intercepté et transformé en un message $p(C(b), s_2)$ qui fait croire au décodeur de b que celui-ci a bien payé son abonnement.*