Université de Montpellier
-
Faculté des Sciences

Session : 1	Durée de l'épreuve : 3 heures
Date : Janvier 2016	
	Documents autorisés : tous
Master Informatique	Matériel utilisé : aucun
Théorie des bases de connaissances (HMIN312)	

Exercice 1. Règles existentielles

On considère des bases de connaissances composées de faits (conjonctions d'atomes fermées existentiellement) et de règles existentielles positives (ne contenant pas $\perp$).

Question 1 De telles bases de connaissances sont-elles toujours satisfiables ?

Question 2 De telles bases de connaissances admettent-elles toujours au moins un modèle fini ?

Question 3 De telles bases de connaissances admettent-elles toujours au moins un modèle universel ?

Question 4 De telles bases de connaissances admettent-elles toujours au moins un modèle universel fini ?

Justifier chacune de vos réponses (donner un contre-exemple simple ou les arguments qui prouvent la propriété).

Exercice 2. Règles existentielles linéaires et contraintes négatives

Définitions préalables.
- Une règle *linéaire* est une règle existentielle positive dont le corps est réduit à un seul atome. Exemple : $\forall x \forall y (p(x,y) \rightarrow \exists z (p(y,z) \wedge q(z)))$
- Une *contrainte négative* est une règle dont le corps est une conjonction d'atomes et dont la tête est réduite au symbole $\perp$. Exemple : $\forall x \forall y \forall z ((p(x,y) \wedge r(x,y,z)) \rightarrow \perp$

Question 1 Une base de connaissances composée de faits et de règles linéaires a-t-elle toujours un modèle universel fini ? Justifier votre réponse.

Question 2 On rappelle qu'un ensemble de règles $\mathcal{R}$ est dit f.u.s. si toute requête conjonctive booléenne q peut-être réécrite avec $\mathcal{R}$ en un ensemble fini $\mathcal{Q}$ de requêtes conjonctives booléennes tel que : pour pour toute base de connaissances $\mathcal{K} = (F, \mathcal{R})$, où F est une base de faits, on a $\mathcal{K} \models q$ ssi il existe $q' \in \mathcal{Q}$ pour laquelle $q' \geq F$ (il existe un homomorphisme de q' dans F).

Tout ensemble de règles linéaires est-il f.u.s. ? Justifier votre réponse.

Question 3 On considère maintenant des bases de connaissances de la forme $\mathcal{K} = (F, \mathcal{R})$, où $\mathcal{R}$ est composée de règles linéaires et de contraintes négatives. On s'intéresse à deux problèmes :
- satisfiabilité : étant donnée $\mathcal{K} = (F, \mathcal{R})$, $\mathcal{K}$ est-elle satisfiable ?
- réponse à une requête conjonctive : étant données $\mathcal{K} = (F, \mathcal{R})$ et une requête conjonctive q, calculer toutes les réponses à q dans $\mathcal{K}$.

Proposer des mécanismes finis permettant de résoudre ces deux problèmes.

Question 4 Citer une logique de description pour laquelle toute inclusion se traduit en une règle linéaire ou en une contrainte négative.

Exercice 3. Logiques de description expressives, tableau

Question 1 La règle de tableau simplifiée $\sqsubseteq^{at}$ ne peut être utilisée que pour des inclusions dont le côté gauche est un concept atomique. Pour chacune des inclusions suivantes, trouvez une inclusion équivalente telle que le côté gauche est un concept atomique. Par exemple, l'inclusion $\neg B \sqsubseteq \neg A$ est équivalente à l'inclusion $A \sqsubseteq B$, dont le côté gauche est le concept atomique A. Justifiez brièvement que les inclusions proposées sont équivalentes aux inclusions originales.

- $D \sqcap E \sqsubseteq \bot$
- $\forall R.B \sqsubseteq \neg A$
- $A \sqcap \forall R.A \sqsubseteq B$
- $\exists R.F \sqsubseteq \neg B \sqcup D$

Question 2 Soit $\mathcal{T}$ la TBox suivante :

$$\mathcal{T} = \{D \sqsubseteq F, \neg E \sqsubseteq \forall S.E, F \sqsubseteq A, A \sqsubseteq \exists S.G, B \sqcap E \sqsubseteq \bot, D \sqsubseteq \forall S.B\}$$

Pour chacune des inclusions suivantes, déroulez l'algorithme de tableau pour décider si l'inclusion est impliquée par $\mathcal{T}$ ou pas. Si l'inclusion n'est pas impliquée, vous devez utiliser le résultat de l'algorithme pour construire un modèle de $\mathcal{T}$ dans lequel l'inclusion n'est pas satisfaite. Indiquez clairement les règles de tableau utilisées. Si vous le souhaitez, vous pouvez utiliser des optimisations présentées dans le cours.

- $D \sqsubseteq E$
- $B \sqsubseteq A$

Exercice 4. Logique de description $\mathcal{EL}$

Question 1 Supposons que nous ayons la $\mathcal{EL}$-TBox $\mathcal{T}$ qui comporte les inclusions suivantes : (notez que dans la 3e inclusion, il s'agit du concept spécial 'top')

$$A \sqsubseteq \exists R.A \quad A \sqcap E \sqsubseteq B \quad \exists R.\top \sqsubseteq E \quad B \sqsubseteq \exists S.D \quad D \sqsubseteq \exists V.E$$

et que nous ayons l'ABox $\mathcal{A} = \{A(a)\}$.

(a) Construisez le modèle canonique compact pour la base de connaissances $(\mathcal{T}, \mathcal{A})$. Vous pouvez le représenter sous forme graphique.

b) Pour chacune des inclusions et assertions suivantes, indiquez si elle est impliquée par $(\mathcal{T}, \mathcal{A})$. Justifiez vos réponses en vous référant au modèle canonique compact.

$$A \sqsubseteq B \qquad B \sqsubseteq A \qquad B(a) \qquad D(a)$$

Question 2 Dans cette question, nous considérons la possibilité d'utiliser le modèle canonique compact pour répondre à des requêtes conjonctives. Appelons **QueryCM** l'algorithme qui prend en entrée une base de connaissance $\mathcal{K}$ (décrite en $\mathcal{EL}$) et une requête conjonctive booléenne Q, et qui retourne OUI s'il existe un homomorphisme de la requête dans le modèle canonique compact de $\mathcal{K}$ et sinon retourne NON.

(a) Pour chacun des énoncés suivants, décidez s'il est vrai ou faux. Justifiez vos réponses.
   - Si **QueryCM** retourne OUI, alors $\mathcal{K} \models Q$.
   - Si $\mathcal{K} \models Q$, alors **QueryCM** retourne OUI.

(b) Maintenant restreignons-nous à des *requêtes de chemins dirigés*, dont les atomes de rôles créent un chemin dirigé $R_1(t_0, t_1), R_2(t_1, t_2), \ldots, R_n(t_{n-1}, t_n)$. Par exemple, la requête $\exists x, y R(x, y) \wedge A(y) \wedge S(y, a)$ est de cette forme, tandis que $\exists x, y, z R(x, y) \wedge S(z, y)$ ne l'est pas. Les énoncés précédents sont-il vrais dans ce cas restreint ? Justifier.

Exercice 5. Modèles stables : Questions de cours

Dans tout cet exercice, on considère le programme Π composé des règles existentielles avec négation (REN) suivantes :

$$\begin{array}{ll} (R_0) & p(a) \\ (R_1) & p(X), \mathrm{not}\, q(X) \to r(X, Y) \\ (R_2) & p(X), \mathrm{not}\, r(X, Y) \to q(X) \\ (R_3) & r(X, Y), \mathrm{not}\, u(X) \to u(X) \end{array}$$

Question 1 Le programme Π est-il stratifiable ? Vous justifierez votre réponse par un dessin.

Question 2 Calculez le "grounding" Π_G du programme Π (attention, il ne s'agit pas du "grounding" de sa skolemization, mais bien du "grounding" de Π).

Question 3 On considère $E_1 = \{p(a), r(a, a)\}$ et $E_2 = \{p(a), q(a)\}$. En utilisant la définition par point fixe (et pas l'algorithme vu en cours), justifiez si, oui ou non, E_1 et E_2 sont des modèles stables de Π_G.

Question 4 Calculez une skolemization Π^S du programme Π.

Question 5 Quel est le domaine de Herbrand utilisé pour calculer le "grounding" Π^S_G du programme Π^S ?

Question 6 En utilisant le domaine de Herbrand de la question 5, calculer le "grounding" de la règle R_1.

Question 7 En utilisant l'algorithme vu en cours, trouvez tous les modèles stables de Π^S (définis comme les modèles stables de Π^S_G).

Question 8 Nous avons vu en cours que si un programme Π était stratifiable, alors il avait exactement un seul modèle stable. La réciproque est-elle vraie ? Justifier votre réponse.

Exercice 6. Modèles stables : Optimisations

Considérons tout d'abord le programme Π_1 composé des règles existentielles avec négation (REN) suivantes :

$$\begin{array}{ll} (R_0) & p(a) \\ (R_1) & p(X), \text{not}\, q(X,Y) \to r(X, f_1(X)) \\ (R_2) & r(X,Y), \text{not}\, p(b) \to s(Y, f_2(Y)) \\ (R_3) & s(X,Y), \text{not}\, p(b) \to r(Y, f_3(Y)) \end{array}$$

Question 9 Après application de la règle R_1, un des deux sommets de l'arbre généré par l'algorithme vu en cours contient les champs suivants :

IN	OUT	MBT
$p(a)$	$\emptyset$	$q(a, Y_0)$

Expliquez pourquoi, sans développer le sous-arbre issu de ce sommet, on peut affirmer qu'il ne mène à aucun modèle stable.

Question 10 Rappelons qu'un sommet (IN, OUT, MBT) satisfait MBT lorsque pour toute disjonction $D = C_1 \vee \ldots \vee C_k$ dans MBT, il existe un C_i dans D qui se projette par homomorphisme dans IN.

Donnez une propriété permettant d'affirmer qu'aucun successeur d'un sommet ne peut satisfaire MBT. Cette propriété devra être suffisamment forte pour qu'un algorithme qui l'implémente puisse couper la recherche sur le sommet de la question 9.

Considérons maintenant le programme Π_2 composé des règles existentielles avec négation (REN) suivantes :

$$\begin{array}{ll} (R_0) & p(a) \\ (R_1) & p(X), \text{not}\, q(X,Y) \to r(X, f_1(X)) \\ (R_2) & p(X), \text{not}\, r(X,Y) \to s(X, f_2(X)) \\ (R_3) & u(X), \text{not}\, q(X,Y) \to v(X, f_3(X)) \\ (R_4) & v(X,Y), \text{not}\, r(X,Y) \to q(X,Y) \end{array}$$

Question 11 Après application de la règle R_1, un des deux sommets de l'arbre généré par l'algorithme vu en cours contient les champs suivants :

IN	OUT	MBT
$p(a)$	$\emptyset$	$q(a, Y_0)$

Expliquez pourquoi, sans développer le sous-arbre issu de ce sommet, on peut affirmer qu'il ne mène à aucun modèle stable.

Question 12 Donnez une propriété permettant d'affirmer qu'aucun successeur d'un sommet ne peut satisfaire MBT. Cette propriété devra être suffisamment forte pour qu'un algorithme qui l'implémente puisse couper la recherche sur le sommet de la question 11.