Université de Montpellier
-
Faculté des Sciences

**Session : 1** Durée de l'épreuve : 3 heures

Date : Janvier 2018

Documents autorisés : aucun

Master Informatique

**Théorie des bases de connaissances (HMIN312)**

**Voir l'annexe pour quelques rappels.**

# Exercice 1. Règles existentielles

On considère les deux bases de connaissances suivantes, composées de faits et de règles existentielles positives.

- $\mathcal{K}_1 = (F_1, \mathcal{R}_1)$, avec :
  $F_1 = \{p(a,b), p(b,a)\}$ et $\mathcal{R}_1 = \{R : p(x,y) \rightarrow \exists z\ p(y,z)\}$.
- $\mathcal{K}_2 = (F_2, \mathcal{R}_2)$, avec :
  $F_2 = \{q(a,b,c)\}$ et $\mathcal{R}_2 = \{R_1 : q(x,y,z) \rightarrow p(y,z); R_2 : p(x,y) \rightarrow \exists z\ q(y,x,z)\}$

**Question 1** Donnez la base de faits saturée obtenue en déroulant l'*oblivious chase* sur chacune des deux bases de connaissances.

**Question 2** Donnez la base de faits saturée obtenue en déroulant le *restricted chase* sur chacune des deux bases de connaissances.

**Question 3** Rappelez ce qu'est un *modèle universel* d'une base de connaissances. Quelle est la relation entre le résultat d'un chase (notamment oblivious ou restricted chase) et un modèle universel de la base de connaissances ?

**Question 4** On rappelle qu'un ensemble de règles $\mathcal{R}$ est un "ensemble à expansion finie" (finite expansion set, *fes*) si pour *toute* base de faits $F$, la base de connaissances $(F, \mathcal{R})$ admet (au moins un) *un modèle universel fini*. Citer des conditions concrètes (c'est-à-dire effectivement vérifiables) qui assurent q'un ensemble de règles est à expansion finie.

**Question 5** Les ensembles de règles $\mathcal{R}_1$ et $\mathcal{R}_2$ sont-ils *fes* ? Justifiez vos réponses.

**Question 5** Toute base de connaissances admet-elle un *modèle* (pas forcément universel) *fini* ? Justifiez votre réponse : si oui, indiquez comment calculer un modèle fini pour une base quelconque, sinon, donnez un contre-exemple.

**Question 6** Montrez que pour toute base de connaissances $\mathcal{K} = (F, \mathcal{R})$ et toute requête conjonctive booléenne $Q$, on a $\mathcal{K} \models Q$ si et seulement si un modèle universel de $\mathcal{K}$ est un modèle de $Q$.

## Exercice 2. Règles existentielles

On considère la base de connaissances $\mathcal{K} = (F, \mathcal{R})$, telle que $F = \{r(a), t(a)\}$ et $\mathcal{R}$ contient les règles suivantes :

$R_1 : t(x_1) \to \exists z_1 p(x_1, z_1)$
$R_2 : p(x_2, y_2) \to s(y_2)$
$R_3 : p(x_3, y_3) \to q(x_3, y_3)$

Soit $Q$ la requête booléenne suivante : $\exists u, v, w (p(u, v) \wedge q(w, v) \wedge s(v) \wedge r(u))$.

**Question 1** Déterminez si $\mathcal{K} \models Q$ en utilisant un mécanisme de chaînage avant (chase).

**Question 2** Déterminez si $\mathcal{K} \models Q$ en utilisant un mécanisme de chaînage arrière (réécriture de requête). Détaillez bien toutes les étapes.

**Question 3** On rappelle qu'un ensemble de règles $\mathcal{R}$ est un "ensemble à unification finie" (finite unification set, *fus*) si pour *toute* requête conjonctive (booléenne) $Q$, l'ensemble des réécritures obtenues par des unificateurs par pièces avec les règles de $\mathcal{R}$ admet une couverture finie (autrement dit, l'ensemble des réécritures les plus générales est fini).

Démontrez que l'ensemble $\mathcal{R}$ ci-dessus est fus.

## Exercice 3. Modélisation en logique

Considérons les phrases suivantes :

1. Les chats et les souris sont des animaux.
2. Rien n'est à la fois un chat et une souris.
3. Tout animal est proie ou prédateur.
4. Tout ce qui chasse est un prédateur.
5. Tout ce qui est chassé est une proie.
6. Toute proie est chassée par quelque chose.
7. Tout chat est plus grand que toute souris.
8. Si quelque chose chasse quelque chose, alors il existe quelque chose qui les chasse tous les deux.

et les langages logiques suivants :

- règles Datalog (et contraintes négatives)
- règles existentielles (et contraintes négatives)
- DL-Lite$_\mathcal{R}$
- $\mathcal{EL}$

Pour chacune des phrases et chacun des langages ci-dessus, décidez si la phrase peut être exprimée dans le langage. Si oui, donnez une formule dans le langage qui décrit la phrase, et si la réponse est non, expliquez brièvement pourquoi.

Vous devez utiliser les prédicats suivants (l'arité des prédicats est indiquée en parenthèses) : $\mathsf{Animal}(1)$, $\mathsf{Proie}(1)$, $\mathsf{Predateur}(1)$, $\mathsf{Chat}(1)$, $\mathsf{Souris}(1)$, $\mathsf{Chasse}(2)$, $\mathsf{PlusGrandQue}(2)$.

## Exercice 4. Calcul des modèles stables

On considère le programme avec RENs (Règles Existentielles avec Négation) $\Pi$ ci-dessous :

```
A(a).
C(X) :- A(X), not B(X).
B(X) :- A(X), not C(X).
B(X) :- C(X), not D(X).
D(X) :- C(X), not B(X).
```

### Question 1

Calculez le grounding $\Pi^G$ du programme $\Pi$.

### Question 2

Calculez le programme réduit associé à l'ensemble d'atomes $\{A(a), C(a)\}$. Cet ensemble est-il un modèle stable du programme $\Pi^G$ ?

### Question 3

Calculez le programme réduit associé à l'ensemble d'atomes $\{A(a), B(a)\}$. Cet ensemble est-il un modèle stable du programme $\Pi^G$ ?

### Question 4

Pouvez-vous répondre, en vous appuyant sur les questions précédentes, mais *sans calcul* à la question : $\{A(a), B(a), C(a)\}$ est-il un modèle stable du programme $\Pi^G$ ?

### Question 5

Utilisez l'algorithme ASPERIX vu en cours pour trouver tous les modèles stables du programme $\Pi$.

### Question 6

Utilisez le graphe de dépendances du programme $\Pi$ pour savoir si ce programme est stratifiable.

### Question 7

Auriez-vous pu répondre à la question précédente (6) sans calculer le graphe de dépendances ?

## Exercice 5. Modèles stables : vrai ou faux ?

Pour chacune des questions suivantes, vous fournirez un contre-exemple si l'affirmation est fausse, et donnerez une démonstration si l'affirmation est vraie. Quand on écrit "un programme contenant les règles $\mathcal{R}$", on considère qu'il peut également contenir d'autres règles.

### Question 1

Soit un programme contenant la REN propositionnelle `b :- a, not b.` Soit $s$ un sommet de l'arbre de recherche d'ASPERIX dont le champ IN contient $a$ mais ne contient pas $b$. Alors aucune feuille ayant $s$ comme ancêtre ne correspond à un modèle stable.

### Question 2

Soit un programme contenant les RENs propositionnelles `c :- a, not b.` et `b :- a, not c.` Alors tout modèle stable contenant `a` contient également `b` ou `c`.

### Question 3

Soit un programme contenant les RENs propositionnelles `c :- a, not b.` et `b :- a, not c.` Alors tout modèle stable contenant `a` contient également `b` ou `c`, mais pas les deux.

## ANNEXE : Rappels

### Chase

Etant donnés une règle $R : B \rightarrow H$ et un homomorphisme $h$ de $B$ dans $F$, l'oblivious chase effectue l'application correspondante si $h$ n'a pas déjà été utilisé pour appliquer $R$ durant le chase, tandis que le restricted chase effectue l'application correspondante si $h$ ne peut être étendu à un homomorphisme de $(B \cup H)$ dans $F$.

### Unificateur par pièces

Il y a plusieurs définitions équivalentes d'unificateur par pièces. En voici une : étant donnée une règle $B \rightarrow H$ et une requête conjonctive booléenne $Q$, un unificateur par pièces $u$ de $Q' \subseteq Q$ et $H' \subseteq H$ est une substitution des variables de $Q' \cup H'$ par des termes de $Q' \cup H'$ telle que :

1. $u(Q') = u(H')$,
2. les variables existentielles de $H'$ ne sont unifiées qu'avec des variables de $Q'$, qui de plus n'apparaissent pas dans $(Q \setminus Q')$. [On dit qu'une variable $v$ est unifiée avec un terme $t$ si $u(v) = u(t)$, ou $u(v) = t$ si $t$ est une constante].

### Logique de description DL-Lite$_{\mathcal{R}}$

En DL-Lite$_{\mathcal{R}}$, une TBox consiste en un ensemble d'inclusions de l'une des formes suivantes :

$$B_1 \sqsubseteq B_2 \quad B_1 \sqsubseteq \neg B_2 \quad S_1 \sqsubseteq S_2 \quad S_1 \sqsubseteq \neg S_2$$

où

$$B \ := A \mid \exists S \qquad\qquad S \ := R \mid R^-$$

avec $A$ un concept atomique et $R$ un rôle atomique.

### Logique de description $\mathcal{EL}$

$\mathcal{EL}$ admet des concepts complexes de la forme suivante :

$$C \ := \top \mid A \mid C_1 \sqcap C_2 \mid \exists R.C$$

où $A$ est un concept atomique et $R$ est un rôle atomique.

Une $\mathcal{EL}$-TBox contient des inclusions de concepts $C_1 \sqsubseteq C_2$ (avec $C_1, C_2$ comme ci-dessus).