Exercices (HAI933I)

Exercice 1 (Fragment existentiel conjonctif)

On considère les formules fermées suivantes :

- $F_1 = \exists x_1 \exists y_1 \exists z_1\ (p(x_1, y_1) \wedge q(y_1, z_1) \wedge r(z_1))$
- $F_2 = \exists x_2 \exists y_2 \exists z_2 \exists v_2\ (r(x_2) \wedge p(x_2, y_2) \wedge q(y_2, x_2) \wedge q(y_2, z_2) \wedge p(z_2, v_2) \wedge q(v_2, z_2))$
- $F_3 = \exists x_3 \exists y_3\ (p(a, x_3) \wedge q(x_3, y_3))$
- $F_4 = \exists x_4\ (p(a, x_4) \wedge q(x_4, a))$
- $F_5 = \exists x_5 \exists y_5\ (p(y_5, x_5) \wedge q(x_5, y_5))$
- $F_6 = \exists x_6 \exists y_6 \exists z_6\ (s(x_6, y_6, y_6) \wedge s(x_6, y_6, z_6))$

Question 1. Dessinez des graphes traduisant ces formules (vous pouvez considérer F_6 à part).

Question 2. Classifiez ces formules (ou ces graphes) selon la relation *"s'envoie par homomorphisme dans"*.

Question 3. À partir de votre réponse à la question précédente, déterminer les relations de conséquence logique entre ces formules.

Question 4. Mettre chaque formule redondante sous une forme minimale. Autrement dit, en voyant une formule comme un ensemble d'atomes, donner un *core* de cet ensemble d'atomes.

On rappelle qu'un ensemble d'atomes (ou un graphe) est un *core* s'il ne s'envoie pas par homomorphisme dans l'un de ses sous-ensembles stricts (ou sous-graphes stricts). Autrement dit, les seuls homomorphismes d'un core dans lui-même sont des isomorphismes. Et un *core d'un ensemble d'atomes* A est un sous-ensemble de A qui est un core.

Exercice 2 (Règles Datalog)

Question 1. Soit une base de faits F sans variables. Soit $\mathcal{R}$ un ensemble de règles positives Datalog. La saturation de F par $\mathcal{R}$ est-elle nécessairement un *core* ?

Question 2. Même question avec une base de faits F qui peut comporter des variables mais qui est un *core*.

Exercice 3 (Inclusion de requêtes conjonctives)

Etant données deux requêtes conjonctives Q_1 et Q_2, on dit que Q_1 est incluse dans Q_2 (notation $Q_1 \sqsubseteq Q_2$) si, pour toute base de faits F, l'ensemble des réponses à Q_1 dans F est inclus dans l'ensemble des réponses à Q_2 dans F.

1. Reformulez la définition d'inclusion pour des requêtes booléennes en termes de réponse *oui* ou *non*. Quelles sont les relations d'inclusion ($\sqsubseteq$) entre les formules de l'exercice 1 vues comme des requêtes ?

2. Montrer la propriété suivante : étant données des requêtes booléennes Q_1 et Q_2, on a $Q_1 \sqsubseteq Q_2$ si et seulement si il existe un homomorphisme de Q_2 dans Q_1.

3. Définir l'homomorphisme de requêtes conjonctives quelconques (c'est-à-dire avec un nombre quelconque de variables réponses), dans l'idée d'étendre cette propriété ; puis étendre la propriété à des requêtes conjonctives quelconques en utilisant cette notion.

Exercice 4 (Complexité du test d'homomorphisme)

Question 1. On appelle HOM le problème suivant : étant donnés deux ensembles d'atomes Q et F, déterminer s'il existe un homomorphisme de Q dans F. On considère un schéma d'algorithme naïf résolvant HOM:

```
Pour toute application h de variables(Q) dans termes(F)
      Si h(Q) ⊆ F, retourner vrai
Retourner faux
```

Combien y-a-t-il de substitutions h à tester ?

Question 2. Montrer que le problème HOM est NP-complet :

1. Montrer que HOM est dans NP, c'est-à-dire qu'on peut déterminer en temps polynomial si une prétendue solution en est bien une (notion de "certificat polynomial").

2. Montrer qu'il existe une réduction polynomiale d'un problème NP-complet à HOM. Considérer par exemple le problème NP-complet *3-coloration* d'un graphe : étant donné un graphe $G = (V, E)$, existe-t-il une coloration des sommets de G avec 3 couleurs, c'est-à-dire une application c de V dans $\{1, 2, 3\}$ telle que pour tout $xy \in V$, $c(x) \neq c(y)$?

Question 3. En théorie des bases de données, on considère deux mesures de complexité pour les problèmes faisant intervenir des requêtes et des bases de données (ou bases de faits pour nous) : la *complexité combinée*, qui se mesure en considérant la taille toutes les données du problème (c'est la complexité classique), et la *complexité de données (data complexity)*, qui se mesure en ne considérant que la base de données (l'idée étant que la taille de la requête est négligeable par rapport à celle de la base de données).

Considérons notre problème HOM, où Q est une requête et F une base de faits. La complexité combinée se mesure en fonction de Q et F. La complexité de données se mesure en fonction de F uniquement : Q est considérée comme fixe (sa taille est une constante) et ne fait pas partie des données du problème. HOM est-il polynomial en complexité de données ?