Règles existentielles

1 Modèles

Exercice 1

On considère la base de connaissances $\mathcal{K} = (F, \{R\})$ avec :
$F = person(a)$
$R = person(x) \rightarrow \exists y\ hasParent(x, y) \wedge person(y)$

Question 1 Définir la base de faits saturée F^* et son modèle isomorphe (ou canonique) M^*. Ce modèle est-il fini ?
Remarque : les différentes variantes du chase se comportent toutes de la même façon sur cet exemple.

Question 2 Montrer que $\mathcal{K}$ admet un modèle fini (essayer d'en donner un qui soit minimal au sens du nombre d'atomes dans la base de faits associée) ? Ce modèle est-il universel ? Justifier.

Exercice 2

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 chaque réponse (donner un contre-exemple simple ou les arguments qui prouvent la propriété).

Exercice 3

Soit la base de connaissances $\mathcal{K} = (F, \mathcal{R})$ où F est une base de faits sans variables et $\mathcal{R} = \{R_1, R_2\}$ est un ensemble de règles positives (datalog) :

$F = \{p(a, b), p(b, c)\}$
$R_1 : p(x, y) \rightarrow q(y)$
$R_2 : q(x) \wedge p(x, y) \rightarrow r(y)$

Question 1. Pour chacune des trois interprétations $\mathcal{I}_1$, $\mathcal{I}_2$ et $\mathcal{I}_3$ ci-dessous, dire quelle affirmation est vraie (et justifier votre réponse bien sûr) :

1. Ce n'est pas un modèle de $\mathcal{K}$
2. C'est un modèle de $\mathcal{K}$ mais pas un modèle universel de $\mathcal{K}$
3. C'est un modèle universel de $\mathcal{K}$.

$\mathcal{I}_1 = (D, .^{I_1})$ avec $D = \{a, b, c\}$, $p^{I_1} = \{(a, b), (b, c)\}$, $q^{I_1} = \{b, c\}$, $r^{I_1} = \{b, c\}$
$\mathcal{I}_2 = (D, .^{I_2})$ avec $D = \{a, b, c\}$, $p^{I_2} = \{(a, b), (b, c)\}$, $q^{I_2} = \{b, c\}$, $r^{I_2} = \{c\}$
$\mathcal{I}_3 = (D, .^{I_3})$ avec $D = \{a, b, c\}$, $p^{I_3} = \{(a, b), (b, c)\}$, $q^{I_3} = \{b\}$, $r^{I_3} = \{c\}$

Question 2. On rappelle qu'un modèle universel M de $\mathcal{K}$ assure la propriété suivante (P) : pour toute requête conjonctive booléenne Q, si M est un modèle de Q alors $\mathcal{K} \models Q$. Montrez que cette propriété n'est pas vraie si on considère un modèle de $\mathcal{K}$ qui n'est pas universel. Pour cela, vous prendrez la base de connaissances $\mathcal{K}$ de cet exercice, vous choisirez un modèle M de $\mathcal{K}$ non universel (éventuellement pris dans la question précédente si cela convient) et vous construirez un contre-exemple à (P) pour $\mathcal{K}$ et M.

Exercice 4. Règles existentielles et contraintes négatives

Définition préalable. On ajoute un prédicat (0-aire) particulier, noté $\bot$, dont la valeur est *faux* dans toute interprétation. Une *contrainte négative* est une règle dont le corps est une conjonction d'atomes et dont la tête est réduite au prédicat $\bot$.

Exemples typiques parmi d'autres :
$\forall x\ (grand(x) \wedge petit(x)) \rightarrow \bot$; *concept disjoints*
$\forall x \forall y\ (supStrict(x, y) \wedge supStrict(y, x)) \rightarrow \bot$; *antisymétrie de relations*

On considère maintenant des bases de connaissances de la forme $\mathcal{K} = (F, \mathcal{R}, \mathcal{C})$ où F est une base de faits, $\mathcal{R}$ un ensemble de règles existentielles (positives) et $\mathcal{C}$ un ensemble de contraintes négatives.

Question 1 Soit une base sans règles existentielles, c'est-à-dire de la forme $\mathcal{K} = (F, \emptyset, \mathcal{C})$. Cette base est-elle forcément satisfiable ? Donner un contre-exemple ou justifier votre affirmation.

Question 2 Donner une requête permettant de tester si une base de connaissances de la forme $\mathcal{K} = (F, \mathcal{R}, \mathcal{C})$ est insatisfiable (c'est-à-dire telle que la réponse à la requête est oui sur $\mathcal{K}$ si et seulement si $\mathcal{K}$ est insatisfiable).

Question 3 Sachant que le problème de réponse à une requête conjonctive booléenne sur une base de connaissances de la forme $\mathcal{K} = (F, \mathcal{R}, \emptyset)$ n'est pas décidable, montrer que le problème de test de la satisfiabilité d'une base de connaissances de la forme $\mathcal{K} = (F, \mathcal{R}, \mathcal{C})$ ne l'est pas non plus.

2 Chase

Exercice 5

On considère un mécanisme de chaînage avant simple (connu sous le nom de *oblivious chase*), qui procède *en largeur.* On désignera par F_i la base de faits obtenue après une étape de largeur (donc i ne correspond pas à la longueur de la dérivation courante mais bien au nombre d'étapes de largeur effectuées).

Question 1 Soit la base de connaissances $K = (F, \mathcal{R})$, avec $F = \{q(a)\}$ où a est une constante, et $\mathcal{R} = \{R_1, R_2, R_3\}$, où :
$R_1 = q(x) \rightarrow \exists z\ p(x, z)$
$R_2 = p(x, y) \rightarrow q(y)$
$R_3 = q(x) \wedge q(y) \rightarrow p(x, y)$

1. Définir F_1, F_2 et F_3 obtenues par le chaînage avant simple en indiquant quelles applications de règles sont effectuées.
2. Le mécanisme de chaînage avant simple s'arrête-t-il ici ?

Question 2 On se pose maintenant les mêmes questions avec un mécanisme de chaînage avant qui diffère du chaînage avant simple en n'effectuant que des applications de règles "localement non redondantes" : à une étape $i > 0$, l'application d'une règle $B \rightarrow H$ sur F_{i-1} selon un homomorphisme h n'est effectuée que si h ne peut pas s'étendre à un homomorphisme de H dans F_{i-1} (autrement dit il n'existe pas d'homomorphisme h' de H dans F_{i-1} tel que pour toute variable frontière x, $h('x) = h(x)$). Ce mécanisme est connu sous le nom de chaînage restreint ("restricted chase").

Exercice 6

On considère la base de connaissances $\mathcal{K} = (F, \mathcal{R} = \{R_1, R_2\})$ ci-dessous :
$\mathcal{F} = \{p(a, b)\}$
$R_1 : p(x, y) \rightarrow \exists z\ p(x, z) \wedge q(z, y)$
$R_2 : q(x, y) \rightarrow q(y, x)$

On suppose que les algorithmes de chase procèdent en largeur (à chaque étape, tous les homomorphismes des corps de règles dans la base de fait courante sont considérées, puis les applications de règles correspondant à ces homomorphismes sont effectuées séquentiellement si elles sont acceptables selon le critère du chase considéré).

Rappels. On rappelle qu'étant donnés une règle $R : B \rightarrow H$ et un homomorphisme h de B dans F, l'application correspondante est effectuée si : R n'a pas déjà été appliquée avec h (*oblivious chase*) ; R n'a pas déjà été appliquée avec un homomorphisme qui envoie la frontière de R de la même façon que h (*semi-oblivious chase*) ; h ne peut pas être étendu à un homomorphisme de $B \cup H$ dans F (*restricted chase*). Quant au *core chase*, il applique la règle comme le ferait le restricted chase, puis calcule le core de la base de faits obtenue.

Question 1. L'oblivious chase s'arrête-t-il ? Quelle base de faits (éventuellement infinie) est (ou serait) construite par ce chase ?

Question 2. Mêmes questions pour le semi-oblivious chase.

Question 3. On suppose qu'à chaque étape de largeur, le restricted chase considère R_1 avant R_2. S'arrête-t-il ? Quelle base de faits (éventuellement infinie) est (ou serait) construite par ce chase ?

Question 4. On suppose qu'à chaque étape de largeur, le restricted chase considère R_2 avant R_1. S'arrête-t-il ? Quelle base de faits (éventuellement infinie) est (ou serait) construite par ce chase ?

Question 5. Que produit le core chase ?

3 Questions de décidabilité et complexité

Exercice 7

Le core chase s'arrête si et seulement si la base de connaissances a un modèle universel fini. C'est donc la variante de chase qui s'arrête "le plus souvent". Cependant, il est considéré comme plus coûteux que le restricted chase, et pour cette raison il n'est pas utilisé en pratique. Pourtant les deux variantes de chase reposent sur un 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 à tester ?

Question 2. Expliquer pourquoi le core chase est considéré comme plus coûteux que le restricted chase en vous appuyant sur votre réponse à la question précédente.

Question 3. 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 4. 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 toutes les données du problème, 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. Quelle est la complexité de données de HOM ?

Exercice 8

Le problème consistant à déterminer si une requête conjonctive est conséquence d'une base de connaissances avec des règles existentielles n'est pas décidable. Cependant, différentes conditions sur les ensembles de règles assurent l'arrêt du chase (ou d'une certaine variante de chase) sur n'importe quelle base de faits, et donc la décidabilité du problème. Nous étudions ici une condition classique en théorie des bases de données, appelée *weak-acyclicity*.

Voir article à RW 2014, définitions 16,17 et 18 (pages 19 à 21) : positions d'un prédicat, graphe des positions, weak-acyclicity.

Question 1. L'ensemble de règles ci-dessous est-il weakly-acyclique :
$R_1 : p(x) \rightarrow \exists y \; r(x, y) \wedge q(y)$
$R_2 : r(x, y) \rightarrow p(x)$

Question 2. Même question avec :
$R_1 : p(x) \rightarrow \exists y \exists z \; r(x, y) \wedge r(y, z) \wedge r(z, x)$
$R_2 : r(x, y) \wedge r(y, x) \rightarrow p(x)$

Question 3. Un ensemble de règles datalog est-il nécessairement weakly-acyclique ?

Question 3. La condition de weak-acyclicity assure l'arrêt du semi-oblivious chase. Montrer qu'elle n'assure pas l'arrêt de l'oblivious chase.

Question 4. Montrer que la weak-acyclicity n'est pas une condition nécessaire d'arrêt de l'oblivious chase (mais seulement une condition suffisante).