Complexité des langages de requête

Le but de ce TD est de se familiariser avec les preuves de complexité des langages de requête.

Rendu

Le rendu de ce TD doit être fait sur le moodle, avant 23:59 le 17 mars en fournissant un fichier PDF contenant une version électronique, des scans ou des photographies *lisibles* de vos réponses au TD. Des pénalités seront comptées en cas de rendu tardif (0,5 point par heure de retard). Les exercices sont par difficulté croissante, il n'est pas nécessaire de finir tous les exercices de la fiche de TD pour avoir une bonne note.

1 Unions de requêtes conjonctives

On considère le langage $\mathcal{L}$ des requêtes composées d'unions de requêtes conjonctives.

Exercice 1. Quelle est la complexité en données du langage $\mathcal{L}$?
C'est à dire, étant donné une requête $\mathcal{Q} \in \mathcal{L}$ fixée, quelle est la complexité du problème $\{(x, D) \mid x \in \mathcal{Q}(D)\}$?

Exercice 2. Quelle est la complexité combinée du langage $\mathcal{L}$?
C'est à dire, quelle est la complexité du problème $\{(x, D, \mathcal{Q}) \mid x \in \mathcal{Q}(D)\}$?

2 α-acyclicité

Exercice 3. Montrer qu'une requête conjonctive α-cyclique peut devenir α-acyclique quand un atome est ajouté à la requête.

Exercice 4. Donner une caractérisation plus simples des requêtes conjonctives α-acycliques sur les schémas relationnels dont toutes les relations sont d'arité ≤ 2. Prouver l'équivalence de cette caractérisation avec la définition des requêtes α-acycliques.

3 FO+μ^+ capture la parité en présence d'un ordre total

On étend dans cet exercice la preuve de capture de parité du nombre d'éléments d'un ordre total par FO+μ^+ à un cadre un peu plus général.

On considère un schéma relationnel arbitraire contenant une relation n-aire R. On considère la requête Q : « R contient un nombre pair de tuples. »

Exercice 5. Montrer (sans utiliser le théorème de Fagin !) que Q peut s'exprimer en logique du second ordre existentielle.

Exercice 6. On suppose un ordre total sur les constantes du domaine, et que la base de données contient une relation S qui, pour toute base de données, contient la relation de *successeur immédiat* pour cet ordre total. Montrer que Q peut s'exprimer en FO+μ^+.

4 Datalog

Exercice 7. Prouver que l'évaluation de Datalog est PTIME en complexité en données.

Dans la complexité en données, le schéma des règles est fixé et donc l'arité des règles est fixé. Le nombre de constantes, lui, est borné par la taille de l'entrée.

Exercice 8. Prouver que l'évaluation de Datalog est EXPTIME-complète en complexité combinée, en montrant qu'on peut simuler l'exécution d'une machine de Turing déterministe en $O(n^k)$ pour une certaine constante k par un programme Datalog.

Suggestion : vous pouvez partir du problème $\{(M, x, T) \quad : \quad M(x)$ *s'arrête en moins de* T *étapes*$\}$ *qui est EXPTIME-complet et utiliser les règles Datalog suivantes (*t *et* p *représentent des nombres en binaire inférieurs ou égaux à* T*) :*

— $next(t, t')$ vrai quand $t' = t + 1$;
— $strictlyLess(t, t')$ vrai quand $t < t'$
— $notEqual(t, t')$ vrai quand $t \neq t'$
— $state(t, q)$ vrai quand la machine est dans l'état q au temps t ;
— $position(t, p)$ vrai quand la tête de la machine est à la position p au temps t ;
— $value(p, t, c)$ vrai quand la bande contient c au temps t à la position p.