

EXAMEN, BASES DE DONNÉES AVANCÉES, JANVIER 2014

Les documents sont autorisés.

Les exercices sont indépendants.

Le barème est sur plus de 20 mais il suffit d'avoir 20 points pour avoir 20.

Exercice 1 : Écriture de Requêtes - 10 Points

On considère le schéma de base de données **AERIEN** contenant les trois relations $\{\text{COMPAGNIES}, \text{VOLS}, \text{APPAREILS}\}$ telles que :

— $\text{sort}(\text{COMPAGNIES}) = \{\text{NomC}, \text{Pays}, \text{Personnel}\}$

— $\text{sort}(\text{VOLS}) = \{\text{NomA}, \text{Origine}, \text{Destination}\}$

— $\text{sort}(\text{APPAREILS}) = \{\text{NomA}, \text{TypeA}, \text{NomC}\}$

On commence par considérer une instance donnée du schéma **AERIEN** ci-dessous :

COMPAGNIES			APPAREILS		
NomC	Pays	Personnel	NomA	TypeA	NomC
Air Talence	France	100 000	AT515	A380	Air Talence
Data Airlines	USA	60 000	AT308	A320	Air Talence
Go Pessac	France	5 000	AT125	747	Air Talence
India Connect	Inde	40 000	GL823	A380	Go London
Air Bordeaux	France	25 000	GL320	A320	Go London
Bob Airlines	USA	75 000	DA100	A320	Data Airlines
Go London	Angleterre	23 000	AB238	A320	Air Bordeaux
			BA715	747	Bob Airlines
			GP963	737	Go Pessac

VOLS		
NomA	Origine	Destination
AT515	Talence	New-York
AT515	New-York	Paris
AT515	Paris	Talence
AT125	Talence	Tokyo
AT125	Tokyo	Talence
GL823	Londres	Paris
GL823	Paris	Londres
BA715	Los Angeles	Chicago
BA715	Chicago	Los Angeles
BA265	Chicago	Boston
BA265	Boston	Chicago

Dans les question suivantes, quand on demande d'écrire une requête en algèbre, on laisse le choix entre les variantes avec ou sans attributs.

1. Pour les requêtes suivantes du calcul relationnel, donnez les résultats renvoyés et donnez une requête équivalente en algèbre :
 - $\{x \mid \exists y \text{ COMPAGNIES}(x, \text{France}, y)\}$.
 - $\{x \mid \forall x_1 \forall x_2 \text{ APPAREILS}(x_1, \text{A380}, x) \Rightarrow \neg \text{VOLS}(x_1, x_2, \text{Paris})\}$.
2. Pour les requêtes suivantes de l'algèbre relationnelle, donnez les résultats renvoyés et donnez une requête équivalente en calcul :
 - $\pi_{\text{TypeA}}(\sigma_{\text{NomC}=\text{AirTalance}}(\text{COMPAGNIES} \bowtie \text{APPAREILS}))$.
 - $\pi_{\text{NomC}}(\pi_{\text{NomC,TypeA}}(\text{APPAREILS}) \div R)$. Avec R la relation suivante :

$$R$$

TypeA
A320

3. On considère la liste de requêtes suivantes :
 - Retourner les pays qui hébergent une compagnie qui dessert New-York.
 - Retourner les compagnies qui possèdent au moins deux A380.
 - Retourner les villes qui sont desservies par des compagnies ayant plus de 40000 employés, dans lesquelles ne se posent aucun A380 et qui ne sont pas joignables depuis New Delhi.
 - Retourner les pays dont toutes les compagnies ne possèdent pas de 747 ou ont moins de 10000 employés.

On demande d'écrire chaque requête en calcul **et** en algèbre.

4. Écrire les deux requêtes suivantes en Datalog :
 - Lister les compagnies qui permettent de faire le trajet Talence-Tokyo.
 - Lister les compagnies qui permettent de faire le trajet Talence-Tokyo en utilisant toujours le même type d'avion.

Exercice 2 : Datalog Bien Fondé - 7 Points

On considère le jeu suivant. Le plateau de jeu est un arbre de rang non-borné dont les feuilles peuvent être de deux types : *blanches* ou *noires* (voir Figure 1). Il y a deux joueurs **J1** et **J2** qui partagent un unique jeton qui à chaque tour de jeu est posé sur un nœud interne de l'arbre.

Les joueurs jouent à tour de rôle et un tour est joué de la façon suivante : le joueur courant choisit un enfant du nœud sur lequel se trouve le jeton et déplace le jeton sur celui-ci. C'est ensuite à l'autre joueur de jouer. Initialement, le jeton se trouve sur la racine de l'arbre et **J1** joue en premier. Un joueur gagne si il pose le jeton sur une feuille blanche, il perd si il pose le jeton sur une feuille noire.

1. L'un des deux joueurs a t'il une stratégie gagnante pour l'arbre de la Figure 1 ? Si oui, la donner.

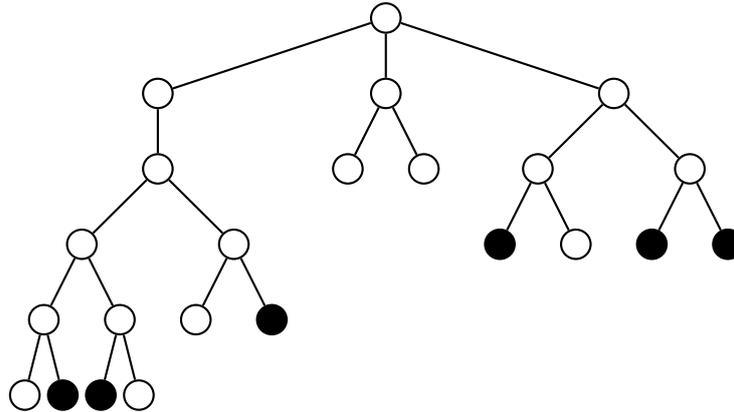


FIGURE 1 – Plateau de jeu

2. Montrez que pour tout arbre, l'un des deux joueurs a toujours une stratégie gagnante.
3. On représente les arbres de jeu comme instance du schéma de base de données contenant les deux relations suivantes :
 - ENFANT binaire qui représente la relation du même nom entre les nœuds de l'arbre.
 - NOIR unaire qui contient les feuilles noires.
 Donnez un programme *datalog*[¬] qui, évalué selon la sémantique bien fondée, calcule l'ensemble des nœuds pour lesquels un joueur qui commence à jouer depuis ce nœud a une stratégie gagnante.
4. Évaluez votre programme sur l'exemple de la Figure 1.
5. A votre avis, est-il possible d'écrire un programme équivalent pour la sémantique stratifiée de *datalog*[¬]. Si la réponse est oui, donnez ce programme. Si la réponse est non expliquez quel est le problème (on ne demande pas ici de preuve formelle mais simplement votre intuition).

Exercice 3 : Datalog et Nombres Premiers - 7 Points

On généralise de manière simple la sémantique de *datalog* (classique) aux instances infinies. Soit P un programme *datalog* et \mathbf{I} une instance (potentiellement infinie) sur $edb(P)$, on note

$$P(\mathbf{I}) = \bigcup_{j>0} T_P^j(\mathbf{I}) \quad \text{où } T_P \text{ est l'opérateur de conséquence immédiate}$$

A noter que la définition n'est ici plus effective (calculable) dès que \mathbf{I} est infinie. De plus, par définition, la généralisation s'étend également à *datalog*[¬] stratifié.

On considère maintenant un schéma de base de données contenant une unique relation binaire *succ*. On voit l'ensemble \mathbb{N} des entiers naturels comme une instance (infinie) de

ce schéma. C'est-à-dire qu'on a $dom(\mathbb{N}) = \{0, 1, 2, 3, \dots\}$ et on interprète $succ$ comme la relation successeur (i.e. $succ(0, 1)$, $succ(1, 2)$, $succ(2, 3)$, ...).

1. Écrire un programme $datalog^{\neg}$ stratifiable $premier$ tel que $premier(\mathbb{N})$ retourne une relation unaire contenant tous les nombres premiers.
2. Écrire un programme $datalog^{\neg}$ stratifiable P tel que $P(\mathbb{N})$ retourne une relation d'arité 0 (i.e. *vrai* ou *faux*) et qui retourne *vrai* si et seulement si le dernier théorème de Fermat est vrai (il n'existe aucun $n > 2$ tel que l'équation $a^n + b^n = c^n$ a une solution dans les entiers positifs).