

Expression Logique et Fonctionnelle ... Évidemment

TP2 : Un puzzle logique et des prédicats récursifs en DATALOG

1 Princesses et tigres (2/4)

Pour résoudre cet exercice, nous vous invitons à utiliser la correction des princesses (1/4) disponible [ici](#).

Le premier prisonnier à réussi à s'en sortir. Le roi, mécontent (il était un peu sadique), modifie les affiches puis en fait venir un deuxième. Voici ce que le nouveau prisonnier pouvait lire :

- 1 -

Il y a une princesse dans
cette cellule et un tigre
dans l'autre

- 2 -

Il y a une princesse dans
une cellule et il y a un tigre
dans une cellule

Seulement le roi a aussi modifié les règles du jeu : maintenant, *une des deux affiches ment, et l'autre dit la vérité!* Répondez aux questions suivantes :

Question 1 Discutez avec vos voisins, essayez de trouver une solution sans formaliser. Ne passez pas plus de 5 minutes là-dessus.

Question 2 Traduisez les deux affiches en formules logiques.

Question 3 Écrivez une formule logique qui inclut aussi bien le contenu des affiches, que les règles du jeu pour ce jour.

Question 4 Implémentez un programme qui résout le puzzle en Datalog, en suivant la solution du premier puzzle.

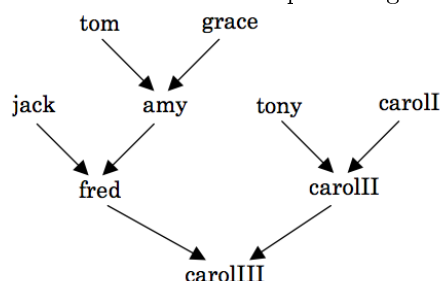
2 Arbre généalogique

Dans cet exercice, vous travaillez avec la généalogie de la famille vue en TD, que nous rappelons :

```
% la EDB
pere(tom,amy).
pere(jack,fred).
mere(grace,amy).
mere(amy,fred).
% le programme (les regles)
parent(X,Y) :- mere(X,Y).
parent(X,Y) :- pere(X,Y).

ancetre(X,Y) :- parent(X,Y).
ancetre(X,Y) :- parent(X,Z), ancetre(Z,Y).
```

Le code ci-haut ne donne qu'un fragment de la EDB. Voici l'arbre complet :



Question 5 Dans un fichier *famille.dl*, saisissez la EDB complète, donc toutes les personnes qui apparaissent dans l'image, pour les prédicats *pere/2* et *mere/2*. Pour vous faciliter le travail, vous pouvez remplacer les chiffres romains par des chiffres arabes. Saisissez également le prédicat *ancetre/2*. Puis, testez ce dernier prédicat, pour vous assurer de le comprendre.

Question 6 Calculez le plus petit point fixe du programme, à l'aide de DATALOG. Rendez les requêtes qui vous permettent de faire cela, ainsi que leur résultats, dans votre compte-rendu.

Question 7 Cette question est à rendre sur papier, au début de votre prochaine séance de TD. Construisez l'arbre de preuve pour le fait que *Grace* est l'ancêtre de *CarolIII*.

Le but du reste de cet exercice est de programmer un nouveau prédicat récursif *mg/2*, à lire, *même génération*. Cette tâche se décompose en deux questions :

Question 8 L'idée du *cas de base* est simple : chaque personne est de la même génération qu'elle-même. Pourtant, puisqu'il n'est pas défini ce qu'est une personne, vous devrez penser à ce point. Lorsque votre définition du cas de base sera correcte, elle devrait produire :

```
DES> mg(X,Y)

{
  mg(amy,amy) ,
  mg(carol1 , carol1 ) ,
  mg(carol2 , carol2 ) ,
  mg(carol3 , carol3 ) ,
  mg(fred , fred ) ,
  mg(grace , grace ) ,
  mg(jack , jack ) ,
  mg(tom,tom) ,
  mg(tony , tony)
}
Info: 9 tuples computed .

DES>
```

Question 9 Pour le cas récursif, sachez qu'il faut remonter de la racine de l'arbre aux feuilles. Avec la bonne définition, vous obtiendrez l'affichage suivant pour la requête *mg(X,Y)* :

```
DES> mg(Y,M)

{
  mg(amy,amy) ,
  mg(amy, carol1 ) ,
  mg(amy, jack ) ,
  mg(amy, tony ) ,
  mg(carol1 , amy) ,
  mg(carol1 , carol1 ) ,
  mg(carol1 , jack ) ,
  mg(carol1 , tony ) ,
  mg(carol2 , carol2 ) ,
  mg(carol2 , fred ) ,
  mg(carol3 , carol3 ) ,
  mg(fred , carol2 ) ,
  mg(fred , fred ) ,
  mg(grace , grace ) ,
  mg(grace , tom) ,
  mg(jack , amy) ,
  mg(jack , carol1 ) ,
  mg(jack , jack ) ,
  mg(jack , tony ) ,
  mg(tom, grace ) ,
  mg(tom,tom) ,
  mg(tony , amy) ,
  mg(tony , carol1 ) ,
  mg(tony , jack ) ,
  mg(tony , tony)
}
Info: 25 tuples computed .
```

3 Oooops... déjà à la fin ?

Si vous avez terminé toutes les questions précédentes au cours de la séance de TP, prévenez votre enseignant, puis abordez la dernière question du TD2.

Question 10 Faites l'exercice de négation (exo 2) du TD2.