

Expression Logique et Fonctionnelle ... Évidemment

TP1 : Premiers pas avec DATALOG

Le but des premiers exos de ce TP est de vous familiariser avec l'environnement DATALOG que nous utiliserons au cours de nos prochaines séances, par la création de requêtes et de prédicats simples. Ensuite, vous travaillerez un exercice sur une base de données, et un autre sur des graphes.

Nous utiliserons DES freeware de DATALOG de l'université de Madrid, plus précisément la version 2.4. Lui-même, DES repose sur SWI-PROLOG, freeware PROLOG de l'université d'Amsterdam. Vous pouvez récupérer des distributions (linux, windows ou Max OS X) de DES librement sur le site <http://www.fdi.ucm.es/profesor/fernand/des/> et de SWI-PROLOG sur le site <http://www.swi-prolog.org>.

DES

- Le binaire DES est disponible au M5 dans `/home/enseign/DES/des`.
- Le signal d'invite "DES>" attend que vous saisissiez un *but* (requete) que le moteur DATALOG tentera de satisfaire.
- Puisque vous n'avez pas encore chargé de base, vous ne pouvez pas encore faire grand chose. Une possibilité est de faire une comparaison `42=42`. Observez le résultat. Comparez avec le résultat de `42=2013`.
- Dans l'interprète, vous pouvez changer de répertoire avec `/cd`, faire afficher le répertoire courant avec `/pwd`, et charger une base (nommée *mabase.dl*) avec `/consult mabase.dl`
- Dans l'interprète, vous pouvez obtenir de l'aide en tapant `/help`.
- Pour quitter DES, tapez `/terminate`. Vous vous retrouverez ensuite au niveau de SWI-PROLOG, donc l'interprète vous affiche "?-". Pour quitter SWI-PROLOG, tapez `halt`. (avec le "."!).

Rappel : en DATALOG, tout identificateur commençant par une majuscule est considéré comme une variable, et les prédicats avec paramètres doivent être immédiatement suivis d'une parenthèse ouvrante (il ne doit pas y avoir d'espace entre la fin du nom du prédicat et cette parenthèse).

1 Histoires de jalousie

On considère la base de connaissance suivante :

- *mia* est une femme;
- *jody* est une femme;
- *yolande* est une femme;
- *vincent* aime *mia*;
- *vincent* aime *pierre*;
- *marcellus* aime *mia*;
- *mon chou* aime *lapin*;
- *lapin* aime *mon chou*;
- X est jaloux de Y s'ils aiment tous les deux une même personne.

Question 1 Saisissez les assertions de cette base dans un fichier¹ sous la forme de *faits* DATALOG qui définissent les prédicats suivants : *femme/1*, *aime/2*, *jaloux/2*

Question 2

Chargez votre base. Pour cela vous utiliserez le prédicat `consult` (essayez `help consult`), en lui donnant en paramètre le nom de votre fichier qui doit avoir l'extension `.dl` : "`consult <nom de fichier>`" avec ou sans l'extension. Au chargement, votre fichier est évalué syntaxiquement par l'interprète DATALOG, toute erreur de syntaxe éventuelle est alors annoncée. Vous pouvez également simplement taper `/c <nom de fichier>` Voici différentes possibilités équivalentes pour charger un fichier qui s'appellerait `exemple.dl` dans le répertoire de travail courant (celui indiqué par `/pwd`) :

```
DES> /c exemple
DES> /consult exemple
DES> /c exemple.dl
DES> /consult exemple.dl
```

1. Utilisez l'éditeur de textes que vous souhaitez : emacs, KWrite, Kate, etc. Emacs et Kate vous permettent d'avoir dans une même fenêtre l'éditeur et un shell dans lequel vous pouvez exécuter SWI-PROLOG

Lorsqu'un fichier est chargé (par /consult) la bases chargée précédemment est "oubliée". Donc un "\consult" réinitialise la base de donnée extensionnelle dans le fichier "consulté".

Sachez que Datalog mémorise les résultats déjà calculés. Vous pouvez les afficher à tout moment avec la commande /list_et .

Question 3 Avec la commande

```
DES> /listing
```

vous pouvez afficher le contenu de la base de connaissance courante.

Ce prédicat peut également prendre en argument le prédicat dont on souhaite connaître la définition courante. Par exemple :

```
DES> /listing femme
```

Question 4 Vérifiez (en définissant un *but* DATALOG) que *mia* et *yolande* sont des femmes mais que *lapin* n'est pas une femme.

Question 5 Interrogez DATALOG pour connaître tous les noms de femmes.

Question 6 Interrogez DATALOG pour connaître toutes les femmes que *vincent* aime.

Question 7 Interrogez DATALOG pour connaître tous les hommes que *vincent* aime. Quel problème observez-vous lorsque vous jouez avec la loi logique de commutativité (A et B est équivalent a B et A) ? Pourquoi ? Comment y remédier ?

Question 8 Interrogez DATALOG pour connaître tous ceux dont *vincent* est jaloux.

Question 9 Comment empêcher l'auto-jalousie ?

Lorsqu'on doit utiliser une variable mais qu'on ne désire pas connaître son instantiation on utilise une *variable anonyme*. Celle-ci est représentée par le symbole _ (tiret bas).

Question 10 En utilisant un variable anonyme, interrogez DATALOG pour savoir si *vincent* est jaloux. En déduire la définition du prédicat *unaire* correspondant.

2 Base de données

Une liste d'employés travaillant dans des entreprises d'informatique vous est fournie sous la forme des deux tableaux 1 et 2.

Nom	Age	Sexe	Entré en	à	Echelon
Jean	23	Homme	2000	IBM	1
Jacques	32	Homme	1991	BULL	4
Anne	45	Femme	1993	BULL	3
Fabienne	30	Femme	2002	XEROX	1
Eric	49	Homme	1984	BULL	5
Marc	25	Homme	2000	XEROX	2
Julie	24	Femme	2001	IBM	1
Sonia	35	Femme	1996	IBM	3
Franck	45	Homme	1999	IBM	2
Odette	40	Femme	1994	BULL	4

TABLE 1 – Liste des employés par société.

Salaires	Ech. 1	Ech. 2	Ech. 3	Ech. 4	Ech. 5
IBM	2000	3000	4500	5500	7500
BULL	1600	3000	4500	6000	7500
XEROX	2500	4500	5500	6000	7000

TABLE 2 – Correspondances échelons/salaires.

Question 11

Saisissez dans un fichier les données présentes dans les tableaux 1 et 2, en écrivant un ensemble de faits définissant les deux prédicats :

```

employe(nom:string(varchar),age:number(integer),
        sexe:string(varchar),entree:number(integer),
        entreprise:string(varchar),echelon:number(integer))
salaire(entreprise:string(varchar),echelon:number(integer),
        salaire:number(integer))

```

Contentez-vous de traduire les trois premières lignes du tableau 1, et les deux premières du tableau 2.

Question 12 Quel but permet de lister le nom des employés de chez BULL ? Testez-le à partir des faits que vous avez écrits.

Question 13 Quel but permet d'afficher le salaire de Jacques ?

Question 14 Récupérez le fichier tp1-employees-partiel.dl. Ce fichier contient la totalité des informations contenues dans les deux tableaux sauf les informations concernant les salaires chez XEROX. Complétez donc ces informations.

Question 15 Pour cette question, il n'est pas demandé d'ajouter de nouvelles règles à votre programme. Il s'agit seulement de formuler des requêtes dans l'interprète DATALOG pour répondre aux questions suivantes.

1. Donnez la liste des noms, âges, sexes, années d'entrée et échelons des employés d'IBM.
2. Donnez la liste des noms et âges uniquement des femmes de la base.
3. Donnez la liste des salaires de la société IBM par échelon.

Question 16 Ajoutez à la base de connaissance, la définition du prédicat `jeune_employe(Nom,Entreprise)`. On considère qu'un employé est jeune s'il n'a pas plus de 25 ans.

Utilisez ce prédicat pour répondre aux questions

1. Marc est-il un jeune employé travaillant chez XEROX ?
2. Jacques est-il un jeune employé ?
3. Quels sont les jeunes employés travaillant chez IBM ?

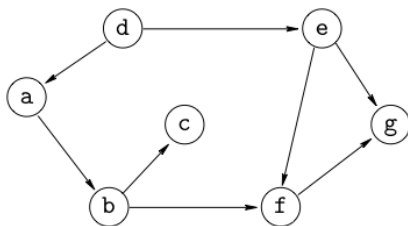
Question 17 Ajoutez des clauses pour le prédicat `cadre(Nom)`. Un employé est un cadre s'il est au moins au 3ème échelon.

Question 18 Définissez le prédicat `gagne_plus_que(Nom1,Nom2)` satisfait lorsque l'employé de nom `Nom1` gagne plus que celui de nom `Nom2`.

Question 19 Donnez, par entreprise, le nombre d'employés.

Question 20 Définissez le prédicat `juniorRiche(Nom)` qui est satisfait pour l'employé de moins de 35 ans qui gagne le plus.

3 Graphes



Question 21 Saisissez le graphe de l'image dans un nouveau fichier *graph.dl*.

Question 22 Utilisez le prédicat `c/2` défini en cours pour afficher toutes les paires de sommets connectés. Rendez votre requête, et son résultat.

Question 23 Ajoutez un ou plusieurs cycles au graphe. Puis, retestez `c/2`.

Question 24 Déclarez un prédicat `impair(X,Y)` qui est vrai s'il existe un chemin de longueur impaire entre les sommets `X` et `Y`. A l'aide d'une requête, affichez les sommets entre lesquels existe un chemin de longueur impaire.