

Expression Logique et Fonctionnelle ... Évidemment

TP1 : Premiers contacts avec PROLOG

Le but de cette première séance de TP est de vous familiariser avec l'environnement PROLOG que nous utiliserons au cours de nos prochaines séances, par la création de requêtes et de prédicats simples. Nous utiliserons SWI-PROLOG freeware de PROLOG de l'université d'Amsterdam. Vous pouvez récupérer des distributions (linux, windows ou Max OS X) de SWI-PROLOG librement sur le site www.swi-prolog.org.

SWI-PROLOG

- Pour lancer l'interprète de SWI-PROLOG, tapez **swipl** dans une console. Le signal d'invite “?-” attend que vous saisissiez un *but* que le moteur de résolution PROLOG tentera de satisfaire.
- Pour quitter SWI-PROLOG, tapez **halt.** (avec le “.”!).
- Dans l'interprète, vous pouvez obtenir une aide sur tout prédicat prédéfini à l'aide du prédicat **help**. Par exemple tester “**help(halt).**”.

Rappel : en PROLOG, 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* PROLOG qui définissent les prédicats suivants :

```
femme(?Prenom)
aime(?Prenom1,?Prenom2)
jaloux(?Prenom1,?Prenom2)
```

Question 2 Chargez votre fichier. 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 **.pl** : “**consult(<nom de fichier>).**” sans l'extension (vous pouvez regarder l'aide pour les commande **chdir**, **ls** et **pwd** éventuellement, vous pouvez encadrer les noms de fichiers ou de répertoires d'apostrophes dans ces commandes). Au chargement, votre fichier est évalué syntaxiquement par l'interprète PROLOG, toute erreur de syntaxe éventuelle est alors annoncée. Vous pouvez également simplement taper “[<nom de fichier>].” Voici différentes possibilités équivalentes pour charger un fichier qui s'appellerait **exemple.pl** dans le répertoire de travail courant (celui indiqué par “**pwd.**”) :

```
?- consult(exemple).
?- consult(exemple.pl).
?- [exemple].
?- ['exemple.pl'].
```

Lorsqu'un fichier est chargé (par **consult**) les clauses qui existaient précédemment pour les prédicats définis dans ce fichier sont “oubliées”. Donc un “**consult**” réinitialise la définition des prédicats définis dans le fichier “consulté”.

Question 3 Avec la commande

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

?- 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 :

listing(femme).

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

Question 5 Interrogez PROLOG pour connaître toutes les noms de femmes.

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

Question 7 Interrogez PROLOG pour connaître tous les hommes que *vincent* aime. Quel problème observez-vous? Pourquoi? Comment y remédier?

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

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

Question 9 En utilisant un variable anonyme, interrogez PROLOG pour savoir si *vincent* est jaloux. En déduire la définition du prédicat correspondant.

2 Base de connaissances sur des employés en entreprise

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 10

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, Age, Sexe, Entree, Entreprise, Echelon)
salaire(Entreprise, Echelon, Salaire)
```

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

Question 11 Quel but permet de lister le nom des employés de chez BULL? Testez-le à partir des faits que vous avez écrits.

Question 12 Quel but permet d'afficher le salaire de Jacques?

Question 13 Récupérez le fichier `tp1-employees-partiel.pl`. 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 14 Pour cette question, il n'est pas demandé d'ajouter de nouvelles clauses à votre programme. Il s'agit seulement de formuler des requêtes dans l'interprète PROLOG 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 de toutes les femmes de la base.
3. Donnez la liste des salaires de la société IBM par échelon.
4. Donnez la liste noms et âge des hommes de moins de 25 ans. Vous pourrez utiliser le prédicat prédéfini `=<` qui est satisfait lorsque le membre de gauche est inférieur ou égal au membre droit. Attention, pour être utilisé, il est nécessaire qu'au moment de sa résolution les deux membres soient unifiés à des nombres.
5. Expliquez pourquoi il n'est pas possible d'obtenir la liste des noms uniquement des hommes n'ayant pas plus de 25 ans.

Question 15 Ajoutez à la base de connaissance, des clauses pour définir le 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 16 Ajoutez des clauses pour le prédicat `cadre(?Nom)`. Un employé est un cadre s'il est au moins au 3ème échelon.

Question 17 Ajoutez des clauses pour le prédicat `collegues(?Nom1,?Nom2)`. Deux employés sont collègues s'ils travaillent dans la même entreprise.

Question 18 Ajoutez des clauses pour le prédicat `gagne_plus_que(?Nom1,?Nom2)` satisfait lorsque l'employé de nom `Nom1` gagne plus que celui de nom `Nom2`.