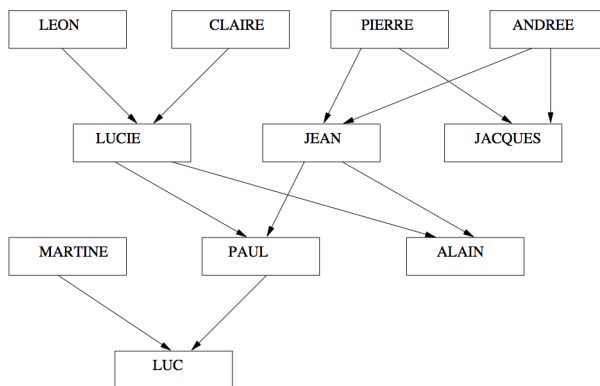


Feuille de T. P. 1 : Premiers pas avec PROLOG

Exercice 1 : Soit l'arbre généalogique suivant :



- 1) Quelles sont les constantes? les prédicats unaires ? binaires?
- 2) Donner la représentation des liens de parenté avec des prédicats instanciés.
- 3) Ecrire le programme PROLOG représentant l'arbre généalogique ci-dessus. (Attention les constantes commencent par une lettre minuscule et les variables par une lettre majuscule.)
 Une règle s'écrit : $A : - B_1, B_2, \dots, B_n.$ et un fait s'écrit : $A.$
 Les commentaires : $/ * \dots * /$
- 4) Ecrire les règles PROLOG qui expriment :

```

enfant(X,Y)
fils(X,Y)           fille(X,Y)
frère-ou-soeur(X,Y)
frère(X,Y)         soeur(X,Y)
grand-parent(X,Y)
grand-père(X,Y)    grand-mère(X,Y)
  
```

X différent de Y s'écrit: $X \neq Y$

- 5) Ecrire l'arbre d'effacement des buts pour :

- mere(claire,lucie), mere(claire,X), mere(X,Y)
- fille(lucie,claire), fille(X,claire), fille(X,Y)

- 6) Comment exprimer le fait qu'une personne est grand-parent sans pour autant spécifier ses petits enfants ?
- 7) Tester le programme PROLOG avec Gnu Prolog.
 - Gnu Prolog est installé sur votre compte sous Linux.
Pour une utilisation sur votre machine personnelle, URL :
http://www.gprolog.org/
 - Documentation, URL : *http://www.gprolog.org/#manual*

Pour utiliser Gnu Prolog :

- création (avec un éditeur quelconque) d'un fichier d'extension .pl qui contient le programme.
- compilation (2 manières possibles) :
 - `gplc nom_prog.pl`
 - `gplc -o nom_prog_exe nom_prog.pl`

Dans le premier cas, l'exécutable a le même nom que le programme source, dans le deuxième cas, il peut avoir un nom différent.

- execution
 - `./ nom_prog_exe`

*affichage d'une nouvelle invite |?-
il est alors possible d'interroger le programme*
- quelques commandes : **a** : abort, **c** : continue, **d** : debug, **h** : help, **b** : break, **e** : exit, **t** : trace.

Exercice 2 : Soit la carte d'un restaurant qui comporte pour les hors-d'oeuvres : cèleri, pâté, melon; pour les viandes : rôti, steak; pour les poissons : sole; rouget; pour les desserts : flan, fruit.

- 1) Ecrire le programme PROLOG représentant la carte du restaurant.
- 2) Un repas est constitué d'un hors-d'oeuvre, d'un plat et d'un dessert, par ailleurs, un plat peut être soit de la viande, soit du poisson. Traduire en PROLOG :
 - les règles `repas(X,Y,Z)` et `plat(X)`.
 - Quels sont les repas qui comportent du cèleri en hors-d'oeuvre ?
 - Quels sont les repas qui ne comportent pas de fruit en dessert ?

- 3) Traduire en langage naturel :
 - repas(X,Y,Z), !.
 - repas(X,Y,Z), poisson(Y),!.
 - repas(X,Y,Z), !,poisson(Y).
- 4) Tester le programme avec Gnu Prolog
- 5) Reprendre le programme et ajouter une boisson qui peut être de l'eau, du vin ou de la bière.
- 6) Ecrire la règle repas-regime(X,Y,Z) qui exprime que la valeur calorique du repas est inférieure à 2000 calories. Pour cela, on attribue une valeur calorique à chacun des mets.
- 7) Tester le programme avec Gnu Prolog
- 8) Ecrire l'arbre déffacement pour
 - plat(Y) , Y \ = roti .
 - plat(Y) , Y\ = roti, !.
 - plat(Y), ! , Y \ = roti.

(facultatif) Exercice 3 : Une agence de voyage propose à ses clients des séjours d'une ou deux semaines à Rome, Tunis, ou Londres. Le catalogue de l'agence contient pour chaque destination, le prix du voyage (indépendant de la durée) et le prix d'une semaine de séjour, qui varie selon la destination et le confort choisi : hôtel, chambre chez l'habitant ou camping.

- 1) Ecrire un programme qui décrit le catalogue (les prix seront choisis arbitrairement)
- 2) Ecrire la règle voyage(V,D,H,S) qui s'interprète par : le voyage dans la ville V, pendant D semaines, avec l'hébergement H coûte S euros.
- 3) Ecrire la règle voyage-eco(V,D,H,S) qui exprime que le coût de ce voyage est inférieur à 2000 euros.
- 4) Tester le programme avec Gnu Prolog