Série de Travaux Pratiques n°1 : Création de règles et de requêtes en Prolog

Exercice n°1 : Soit une partie de la base de connaissances animalière qui décrit les caractéristiques des animaux, au travers du prédicat animal qui donne : le nom d'un animal, sa classe, son régime alimentaire, sa principale caractéristique, son lieu d'habitation, son poids moyen (en kg) et son âge moyen (en années).

```
animal(tigre, mammifère, carnivore, rayures, savane, 100, 12).
animal(zèbre, mammifère, herbivore, rayures, savane, 250, 30).
animal(aigle, oiseau, carnivore, large, hauteurs, 5, 25).
animal(lion, mammifère, carnivore, crinière, savane, 150, 12).
animal(serpent, reptile, carnivore, long, désert, 5, 3).
animal(moineau, oiseau, granivore, petit, champs, 0.04, 13).
animal(lézard, reptile, insectivore, petit, désert, 0.03, 5).
```

Donner les requêtes qui permettent d'avoir :

1. Tous les mammifères
2. Tous les mammifères qui sont des insectivores.
3. Tous les oiseaux qui ne sont pas carnivores.
4. Tous les mammifères qui ont des rayures
5. Les animaux dont le poids est compris entre 1 et 10 Kg.
6. Les mammifères qui sont des herbivore et dont l'âge est inférieur à celui du zèbre.
7. Les mammifères et les reptiles qui vivent dans les mêmes lieus.
8. Les couples d'oiseaux qui ont les mêmes caractéristiques.

Exercice n°2. Soit la base de connaissances en Prolog suivantes :

```
masculin(amine).                    feminin(samia).
masculin(nabil).                    feminin(louisa).
masculin(badr).                     feminin(amira).
masculin(jamel).   % «paquet» de clauses   feminin(nadia).

enfant(badr,samia).                 enfant(nadia,badr).
enfant(badr,amina).                 enfant(nabil,louisa).
enfant(louisa,amine).               enfant(jamel,nadia).
enfant(amira,badr).

pere(X,Y) :- enfant(Y,X), masculin(X).
```

1. Quelle est la requête Prolog qui permet d'avoir le père de nabil (sans utiliser le prédicat père)
2. fils(X,Y) qui exprime que X est un fils de Y
3. fille(X, Y) qui exprime que X est une fille de Y
4. frere-ou-sœur(X, Y) qui exprime que X est frère ou sœur de Y.

Il est à noter qu'un individu n'est pas son propre frère ou sa propre sœur.

5. Définissez les prédicats suivants : mere/2, grand-pere/2, frere/2, tante/2, cousin/2

Série de travaux pratiques n°2 : Prolog et les bases de données

On considère la base de données « fournisseurs–pièces–projets ». Elle est composée de 4 prédicats faits :

– Le prédicat des fournisseurs affecte à chaque fournisseur un numéro d'identification, un nom, une priorité et une ville.

```
fournisseur (f1,  omar ,  20,  annaba).
fournisseur (f2 ,  ali ,  10 ,  alger).
fournisseur (f3 ,  nabil,  30 ,  alger).
fournisseur (f4 ,  rami,  20 ,  annaba).
fournisseur (f5 ,  said,  30 ,  oran).
```

– Le prédicat des pièces détachées précise la référence, le nom, la couleur, le poids et le lieu de stockage de chaque pièce.

```
piece(p1,  ecrou ,  rouge ,  12,  annaba).
piece(p2 ,  boulon , vert ,  17 ,  alger).
piece(p3 ,  vis ,  bleu ,  17 ,  setif).
piece(p4 ,  vis ,  rouge ,  14 ,  annaba).
```

– Le prédicat des projets indique la référence, la nature et le lieu d'assemblage de chaque projet.

```
projet(pj1,  disque,  alger).  projet(pj2,  scanner, setif).
projet(pj3,  lecteur,  oran).  projet(pj4,  console,  annaba).
projet(pj5,  capteur,  annaba).
```

– La table des livraisons contient le numéro d'identification d'un fournisseur, la référence de la pièce à livrer, la quantité livrée ainsi que la référence du projet auquel cette pièce est affectée.

```
livraison(f1,  p1,  pj1,  200).   livraison(f1,  p1,  pj4,  700).
livraison(f2 ,  p3,  pj1,  400).   livraison(f2 ,  p3,  pj2,  200).
livraison(f2,  p3,  pj3,  200).   livraison(f2,  p3,  pj4,  500).
livraison(f3,  p3,  pj1,  200).   livraison(f3,  p4,  pj2,  500).
livraison(f4,  p2,  pj3,  300).   livraison(f4,  p4,  pj5,  300).
livraison(f5,  p1,  pj4,  100).   livraison(f5,  p2,  pj4,  500).
livraison(f5,  p3,  pj1,  100).   livraison(f5,  p4,  pj3,  200).
```

1. Créer le prédicat « pjOran(Reference,Nature) » donnant les informations concernant les projets en cours à oran.
2. Créer le prédicat « fpDistincts(Fournisseur,Piece) » faisant apparaître les numéros d'identification des fournisseurs et des pièces pour tous les fournisseurs et toutes les pièces non situés dans la même ville.
3. Créer le prédicat « pjF1P1(Projet,Ville) » faisant apparaître tous les projets alimentés par le fournisseur f1 ou utilisant la pièce p1.
4. Trouver tous les détails des projets de annaba.
5. Trouver les références des fournisseurs du projet pj1 .
6. Quelles sont les livraisons dont la quantité est comprise entre 300 et 750 ?
7. Trouver tous les triplets (fournisseur,piece,projet) tels que le fournisseur, la pièce et le projet soient situés dans la même ville.
8. Trouver les références des pièces provenant d'un fournisseur de annaba.
9. Trouver les références des pièces provenant d'un fournisseur de annaba, et destinées à un projet de annaba.

Série de Travaux Pratiques n°3 : Prolog : Arithmétique, Contrôle, Négation et la Récursivité

Partie 1 : Arithmétique, Contrôle, Négation

Q1. Ecrire le prédicat qui calcule la distance Euclidienne entre deux points A et B.

Q2. Ecrire le programme Prolog qui calcule la valeur absolue d'un nombre.

Q3. Ecrire le prédicat qui donne le maximum de 2 nombres.

Q4. Ecrire le programme prolog qui étant donné 3 nombre calcule le maximum des deux premiers nombres puis donne le minimum entre ce nombre obtenu et le troisième nombre.

Q5. Ecrire le prédicats pair(X), qui permet de vérifier si un nombre naturel est pair.

Partie 2 : Récursivité

Q6. Ecrire le programme prolog qui implémente les fonctions suivantes :

a. factoriel

b. La suite de fibonnacci définie de façon récurrente par :

fib(0) = fib(1)=1

fib(N+2)= fib(N+1) + fib(N), $N \geq 0$

c. PGDC propriétés du PGCD **D** de **X** et **Y**

- si **X** et **Y** sont égaux, **D** vaut **X**
- si **X** < **Y** alors D est le PGCD de **X** et de **Y** - **X**
- si **Y** < **X** alors échanger le rôle de **X** et **Y**

Q7. Définir un prédicat calculant le nième terme de la suite : $U_0 = 2$, $U_n = 2U_{n-1}+3$

Série de Travaux Pratiques n°4 : Les Listes

Q1. Ecrire en Prolog le prédicat element, de deux manières : avec et sans le cut (!). element est le prédicat qui permet de savoir si X est un élément de la liste L. Tester la différence entre les deux définitions sur un exemple.

Q2. Définir un prédicat **ajoute1(L,L1)** où L est une liste de nombres, et L1 une liste identique où tous les nombres sont augmentés de 1.

Q3. Définir le prédicat « suivants(X, Y, L) » où étant donné une liste L, le prédicat renvoie le suivant d'un élément X, avec X et Y se suivent immédiatement dans la liste L.

```
?- suivants(a,b,[a,b,c]).
true ;
?- suivants(a,X,[a,b,c]).
X = b ;
?- suivants(X,b,[a,b,c]).
X = a ;
?- suivants(X,Y,[a,b,c]).
X = a, Y = b ;
X = b, Y = c ;
```

Q4. Ecrire le prédicat qui :

a. Donne l'élément maximum d'une liste d'entiers.

b. Calcule le nombre N d'occurrences de l'élément X dans la liste L (occ(L,X,N)).

Exemple ?occ([z,a,r,a,t],a,N).

N=2

c. Supprimer des doublons consécutifs dans une liste L pour obtenir une liste L1.

Remarque: L'ordre des éléments doit être respecté.

Exemple: ?- compresser([a,a,a,a,b,c,c,a,a,d,e,e,e,e],L1).

L1 = [a,b,c,a,d,e].

d. permet de partager une liste en deux parties. On appellera ce prédicat split(L,N,L1,L2) où L est la liste de départ. N est le nombre des éléments dans la première liste L1 et L2 est la seconde liste.

e. inverse les éléments d'une liste.

Exemple: ?- renverse([c, b, a, d, b],L1).

L1 = [b, d, a, b, c].

Q5. Définir le prédicat : **partition(X, L, LinfX, LsupX)** qui étant donné un nombre X et une liste L, partitionne cette liste en deux listes : LinfX est la liste composée des éléments de L qui sont inférieurs à X, et LsupX est la liste composée des éléments de L qui sont supérieurs ou égaux à X.

Exemple : ?- partition(4,[3,8,4,1,6,5,2],LinfX,LsupX).

LinfX = [3,1,2]
LsupX = [8,4,6,5]
yes