

TP2 : Syntaxe du prolog

Consignes :

Il est demandé à l'étudiant de vérifier ces règles syntaxiques sur écran avec prolog et faire les recherches nécessaires sur internet pour compléter (ceci est juste un résumé pour démarrer).

Quelques Règles syntaxiques :

- On représente les commentaires par % à mettre au début de la ligne.
- Toute chaîne de caractères qui commence par une minuscule est une constante. Une chaîne de caractères entre cotes ' ' est une constante même si elle commence par une majuscule.
- Toute chaîne de caractères qui commence par une majuscule est une variable.
- Une variable est dite anonyme si elle commence par _ . Les valeurs affectées à ces variables après unification importent peu. Elles sont utilisées pour des variables dont on ne désire pas connaître les valeurs.
- Un prédicat est une chaîne de caractère commençant par une minuscule. A ne pas confondre avec une constante. Un prédicat a des arguments qui peuvent être des constantes, des variables ou des termes composés.
- Les buts Appelés parfois questions, les buts permettent d'exprimer une interrogation sur l'existence de données vérifiant les prédicats, par exemple : ?-B. Ou plus généralement : ?-B1;;Bm.
- La récursivité est définie par le pas de la récursivité et la condition d'arrêt.
- Il n'y a pas de boucles en prolog, donc la récursivité est le seul moyen d'exprimer la notion d'itération.
- Prolog offre les opérateurs arithmétiques suivants : +,-,*,/, mod.

Le prédicat is permet l'affectation d'une valeur ou d'un résultat d'une opération à une variable, par exemple : ?-X is 5. (exemple : somme(X,Y,S) :- S is X+Y).

- Le prédicat =: permet la comparaison de deux expressions après leur évaluation.
- Les opérateurs <; >;=<;>= et = n = effectuent le calcul avant de comparer les valeurs. Pour utiliser ces opérateurs, les variables doivent être instanciées.
- Prolog permet de calculer toutes les solutions possibles à un ou plusieurs buts. Cependant, à chaque fois qu'il donne une réponse, il perd la précédente. Les prédicats findall, bagof et setof permettent de sauvegarder toutes les solutions dans une liste.
- Le prédicat trace(but) permet de voir tous les appels et leur résultats logiques lors de la résolution d'un but, par exemple, ?-trace.
- Pour afficher sur écran, on utilise write.
- Le prédicat prédéfini fail provoque un échec. Il force ainsi la remontée (le retour en arrière) lors du raisonnement.
- La négation d'un prédicat peut être défini à l'aide du prédicat not

```
not(_p) :- _p, !, fail.  
not(_p).
```

 suivant :
- Les listes en Prolog se notent par des crochets. Par exemple, la liste des nombres entre 0 et 4 sera notée : [0,1,2,3,4]. En général, une liste est notée sous forme [Tête|Queue] où Tête est le premier élément de la liste et Queue est le reste de la liste. Cette notation est récursive puisque Queue est elle-même une liste. Les tableaux en prolog sont des listes.
Exemple : [0,1,2,3] = [0 | [1,2,3]] = [0 | [1 | [2,3]]] et ainsi de suite.
Une liste vide se note comme suit : [].

Applications sur machine

- 1) Les base de connaissances (faits), à mettre dans un nouveau fichier prolog.

pere(farid, djamila).
pere(djamila, nacera).
pere(nacera, farid).
pere(mimi, aridj).
mere(aridj, mahmoud).
mere(mahmoud, mimi).
mere(aridj, mimi).

- Les requêtes à **exécuter sur machine** (des questions ou appelées buts, on va les poser de manière interactive après lecture du fichier)

?- pere(djamila, nacera).
?- mere(aridj, mahmoud).
?- pere(cesar, X).
?- pere(cesar, X), mere(Y,X). (on l'appelle conjonction de clause)

- Les déductions

On définit les clauses suivantes:

masc(X) :- pere(X, Y).
fem(X) :- mere(X, Y).
ancetre(X, X).
ancetre(A, D) :- pere(A, G), ancetre(G, D).
ancetre(A, D) :- mere(A, G), ancetre(G, D).

Exécuter **sur machine** les buts suivants :

?- fem(djamila).
?- fem(nacera).
?- fem(mara).
?- masc(farid).
?- ancetre(aridj,mimi).
?- ancetre(X, mimi).
?- ancetre(X, mimi), ancetre(aridj, X).

- 2) Connaître le plus grand parmi 2 nombres ? ou parmi 3 :

max2(X,Y,X) :- X>=Y.
max2(X,Y,Y) :- X<Y.
max3(X,Y,Z,M) :- max2(X,Y,MM),max2(MM,Z,M).
Que produit max2(X,1,3) ? Pourquoi ? Même question avec max2(1,X,3).

- 3) Afficher N fois 'bonjour' :

ecrit(0).
ecrit(N) :- N>0, write('bonjour'), nl, N1 is N-1, escrit(N1).

- 4) Dire si un nombre est pair :

pair(0).
pair(X) :- X>0, X2 is X-2, pair(X2).

- 5) Appartenance d'un élément à une liste :

?- member(a, [b, c, a]).
Yes
?- member(a, [c, d]).
No

Activités

Activité 1 :

1. Programmer les connaissances suivantes :

Le père de ali est mohamed

Le père de said est mohamed

Si x et Y ont le même père ou la même mère alors ils sont des frères ou des sœurs

La mère de souad est ferooudja

La mère de Wissam est ferroudja

2. Ecrire un programme qui déduit tous les frères et sœurs de cette famille.

Activité 2

Soit le programme Prolog suivant :

homme(ali).

homme(omar).

homme(said).

femme(céline).

femme(lila).

pere(said,ali).

pere(omar,ali).

mere(omar,céline).

mere(said,céline).

parents(X,M,P):-mere(X,M), pere(X,P).

frere(X,Y) :- homme(Y), parents(X,M,P), parents(Y,M,P).

Exécuter le but : **?-frere(omar,said).** et donner tous les résultats possibles.

Activité 3

Etant donnés deux entiers $n1$ et $n2$. Ecrire un programme Prolog qui calcule le quotient entier Q et le reste de la division de $n1$ par $n2$. (Écrire d'abord le programme sur feuille, vérifier et implémenter sur machine)

Activité 4

Quelles sont les réponses de Prolog aux buts suivants (à exécuter sur machine).

?-2=1+1.

?-X=2, X=1+1.

?- X is 3+4.

?-X=5, X is 3+3.

?-P(X,X)=P(a,b).

?-P(X,X)=Q(a,b).

Activité 5

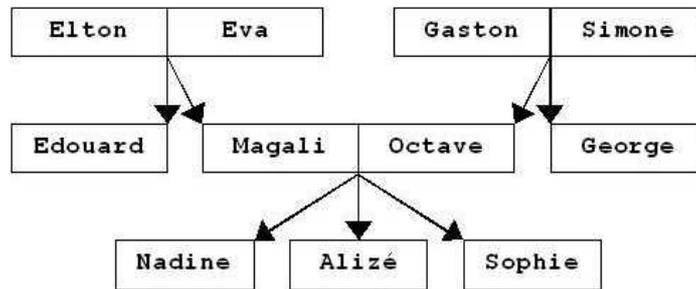
Ecrire un programme Prolog qui affiche les nombres de 1 à N.

Activité 6

Ecrire le programme prolog permettant de calculer la somme de tous les éléments positifs d'une liste d'entiers (positifs et négatifs) ainsi que celle des éléments négatifs.

Activité 6 (perfectionnement) :

Soit l'arbre généalogique suivant :



1. Écrivez une série de *faits* définissant les prédicats homme, femme et parent pour modéliser cet arbre. homme(X) signifiera que X est un homme, femme(X) que X est une femme, et parent(X,Y) que X est un parent (père ou mère) de Y.
2. Rajoutez à votre base des *règles* pour définir les prédicats suivants (pour tous ces prédicats, on conviendra que Préd(X,Y) signifie “X est le/un Préd de Y”). Par exemple, enfant(X,Y) signifiera “X est un enfant de Y”) :

enfant(X,Y)
 fils(X,Y)
 fille(X,Y)
 pere(X,Y)
 mere(X,Y)
 grand_parent(X,Y)
 grand_pere(X,Y)
 grand_mere(X,Y)
 petit_enfant(X,Y)
 petit_fils(X,Y)
 petite_fille(X,Y)
 frere(X,Y)
 soeur(X,Y)
 oncle(X,Y)
 tante(X,Y)