Programmation logique
Série de TD n°2

Exercice 1

Ecrire un prédicat PGCD qui calcule le PGCD de deux entiers :
Le PGCD de X et Y est X si X et Y sont égaux,
Sinon c'est le PGCD de X et Y-X si X<Y
Sinon échanger le rôle de X et Y (si Y<X)

Donner l'arbre de résolution Prolog pour la requête PGCD(3,2,X).

Exercice 2

Ecrire un prédicat Fibonnacci calculant un terme U_N de la suite de Fibonnacci définie comme suit :
$U_0 = U_1 = 1$
$U_N = U_{N-1} + U_{N-2}$, $N \geq 2$.

Donner l'arbre de résolution Prolog pour le calcul du terme U_3.

Exercice 3

Ecrire un prédicat factorielle qui calcule la factorielle d'un entier.
Donner l'arbre de raisonnement pour calculer la factorielle de 3.

Exercice 4

Ecrire un prédicat somme qui calcule la somme de N entiers lus de l'input.
Ecrire un prédicat max qui calcule le maximum de N entiers lus de l'input.

Exercice 5

Ecrire un prédicat éléments_pairs(L,LR) donnant la liste LR des entiers pairs d'une liste d'entiers L.
Donner l'arbre de résolution pour la requête éléments_pairs([5,7,0,1], LR)

Exercice 6

Ecrire les prédicats suivants :

- somme: calcule la somme d'une liste de nombres
- longueur: calcule la longueur d'une liste

- maximum: calcule le maximum d'une liste de nombres
- moyenne: calcule la moyenne d'une liste de nombres

Exercice 7

Définir un prédicat inter d'intersection de deux listes d'entiers sans redondance.

1. en utilisant le cut
2. sans le cut mais avec le not

Exercice 8

1. Définir un prédicat début(liste, liste) donnant tous les débuts d'une liste. Par exemple, début([1,5,9], L) donne :
   L=[]
   L=[1]
   L=[1 ,5]
   L=[1,5,9]
2. Utiliser ce prédicat pour définir le prédicat sous_liste(liste, liste) donnant les sous_listes d'une liste. Par exemple, sous_liste([1,5,9], L) donne :
   L=[]
   L=[1]
   L=[5]
   L=[9]
   L=[1,5]
   L=[5,9]
   L=[1,5,9]

Exercice 9 : (Tri par fusion)

Ecrire un programme de tri par fusion en définissant les prédicats suivants :

- séparer(L,L1,L2) qui sépare une liste L en une liste L1 et une liste L2 telles que L1 est composée des éléments de rang impair de L, et L2 est composée des éléments de rang pair de L.

Exemple : séparer([1,12,13,11,2,4,6],L1,L2), donne L1=[1,13,2,6] et L2=[12,11,4]

- fusionner(L1,L2,L) qui pour tout couple de listes ordonnées par ordre croissant L1, L2 donne L, la liste des éléments de L classés par ordre croissant.

Exemple : fusionner([1,3,4,9,18],[2,5,9,10,11],L]) donne L=[1,2,3,4,5,9,9,10,11,18]

- trier(L,LL) qui trie une liste L d'éléments par ordre croissant.

Exemple: trier([1,5,4,2,7,3,5]) donne LL=[1,2,3,4,5,7]