

Exercice 1.

Calculez, si il existe, un unificateur le plus général de la paire d'atomes (A1;A2). En cas d'échec, indiquez pourquoi les atomes ne sont pas unifiables.

p, q, r sont des symboles atomiques, x; y; z; ; u ; v sont des variables, f; g; h sont des symboles fonctionnels, a; b sont des constantes :

A1 = q(f(f(a, x), b), f(z, f(c, v))) ; **A2** = q(f(y, z), y)

N.B : Dérouler l'algorithme pour la paire 1., Pour les autres donner l'upg ou justifier l'échec

Exercice 2.

utiliser le raisonnement en logique (cours) pour le problème suivant :

1. Tous les élèves de 2ième année suivent les cours de logique formelle et de Prolog.
2. Colmérauer est un créateur de Prolog.
3. Tout élève qui étudie une matière souhaite rencontrer le créateur de cette matière.
4. Les élèves de 2ième année souhaitent rencontrer Colmerauer.

a) Modélisez en logique du premier ordre l'énoncé ci-dessus en utilisant les prédicats :

- _ élève(x,y) : x est élève en yième année.
- _ étudie(x,y) : x étudie la matière y.
- _ créateur(x,y) : x est un créateur de y.
- _ rencontre(x,y) : x souhaite rencontrer y.

Bien entendu, vous complétez cette modélisation, si c'est nécessaire, avec des constantes, des variables et des fonctions.

b) Prouvez que la dernière affirmation est une conséquence logique de l'ensemble des autres affirmations.

Exercice 3 Prolog

3-1 : Soit le programme Prolog suivant définissant le prédicat pred.

R1: pred([], []).

R2: pred([X|L1], [X|L2]) :- X mod 2 =:= 0, !, pred(L1, L2).

R3: pred([X|L1], L2) :- pred(L1, L2).

a) **Construisez l'arbre d'effacement du but** **pred([5, 6, 3, 7, 4], L)**. On rappelle que mod est la fonction modulo, et que Expr1 =:= Expr2 s'efface si les expressions arithmétiques Expr1 et Expr2 sont évaluées à la même valeur.

b) **Expliquez exactement ce que fait pred.**

3-2 : On rappelle que le nombre C_n^k **de sous-ensembles à k éléments d'un ensemble de cardinal n est donné par la formule :**

$$C_n^k = \frac{n!}{k!(n-k)!}$$

a) Que vaut C_n^0 ? Montrer que : $C_n^{k+1} = \frac{(n-k)}{k+1} C_n^k$

b) Ecrire un programme **Prolog** permettant de calculer C_n^k . On l'appellera par c(K,N,R) où K et N seront fournis au programme tandis que R contiendra le résultat retourné. Le prédicat **integer(X)** vérifie si X est entier ou pas.

c) Ecrire une variante du même programme, qui dès le calcul achevé affiche à l'écran le résultat (par exemple à l'appel de c(2,3,R)) : le resultat est égal à 3