Prolog

Programe

- nomprgm(var1, var2)
- nomprgm(0). ← initialisé

Question

homme(X) ?

Prédicat

Règle = Prédicat

Fait : homme(Socrate). → S ∈ homme

Règle : mortel(S) :- homme(S). → S ∈ mortel SI S ∈ homme.

Exemples

chemin(X,Y) :- arc(X,Y).
chemin(X,Z) :- arc(X,Y), chemin(Y,Z).

⇒ arc(a,b).
arc(c,b).

Sauf si accolades dans la règle → Paradoxe / Inversion

chemin(a,c) ? → false.
arc(b,c).
chemin(a,c) ? → true

parent(X,Y) :- enfant(Y,X)
parent(père, fils) ? → true

Symboles

= ⇒ unification (question) / test
is ⇒ affectation (fonction)
<, > ⇒ questions / tests
rl ⇒ saut ligne (fonction)
write(X) ⇒ écrit (fonction) ⇒ égalité classique

Listes

[] : vide
[T|R] : élément tête / reste
[_|R] : liste à partir de i+1

Créer un programe en Prolog :

Definitely - prgm(0,0). Toujours Vrai.

Mise en place règles - prgm(X,Y) :- regle1, regle2, regle3, fct(...), regle n.
ou opération / fonction

Fonctions utilisables :

findall(X,Y,L) : Trouve tout variable X tel que Y et les stocke dans L. (L = liste)

between(x,y,i) for(x à y incrémenté de i)

Le Cut : Instruction ! , "coupe-choix" ; Empêche le retour arrière.

⇒ Si plusieurs solutions existaient sauf la solution courante est préservée.
Si le prédicat précédent est vrai, on continue, sinon on passe à la définition suivante.