

Prolog : TP7

jeudi 27 novembre 2014

Maud LERAY
Corentin NICOLE
groupe 1.2

Questions

```
/**
TP 7 Base de Donnes Dductives (BDD) - Prolog

@author Corentin Nicole
@author Maud Leray
@version Annee scolaire 2014/2015
*/

/*
=====
=====
Dfinition des prdicats
=====
*/
% =====
% SECTION 1 : Base de donnees
% =====

assemblage(voiture, porte, 4).
assemblage(voiture, roue, 4).
assemblage(voiture, moteur, 1).
assemblage(roue, jante, 1).
assemblage(porte, tole, 1).
assemblage(porte, vitre, 1).
assemblage(roue, pneu, 1).
assemblage(moteur, piston, 4).
assemblage(moteur, soupape, 16).

piece(p1, tole, lyon).
piece(p2, jante, lyon).
piece(p3, jante, marseille).
piece(p4, pneu, clermontFerrand).
piece(p5, piston, toulouse).
piece(p6, soupape, lille).
piece(p7, vitre, nancy).
piece(p8, tole, marseille).
piece(p9, vitre, marseille).

demandeFournisseur(dupont, lyon).
demandeFournisseur(michel, clermontFerrand).
demandeFournisseur(durand, lille).
demandeFournisseur(dupond, lille).
demandeFournisseur(martin, rennes).
demandeFournisseur(smith, paris).
demandeFournisseur(brown, marseille).

fournisseurReference(f1, dupont, lyon).
fournisseurReference(f2, durand, lille).
```

```
fournisseurReference(f3, martin, rennes).
fournisseurReference(f4, michel, clermontFerrand).
fournisseurReference(f5, smith, paris).
fournisseurReference(f6, brown, marseille).

livraison(f1, p1, 300).
livraison(f2, p2, 200).
livraison(f3, p3, 200).
livraison(f4, p4, 400).
livraison(f6, p5, 500).
livraison(f6, p6, 1000).
livraison(f6, p7, 300).
livraison(f1, p2, 300).
livraison(f4, p2, 300).
livraison(f4, p1, 300).

% =====
% SECTION 2 : Operations relationnelles
% =====

% ==== Q 2.1 Selection ====
piece_a_lion(NumPiece,Piece) :-
    piece(NumPiece,Piece,lyon).

% ==== Q 2.2 Projection ====
projection(NomPiece,LieuFab) :-
    piece(_,NomPiece,LieuFab).

% ==== Q 2.3 Union, intersection et difference ensembliste ====
union_fourn(Nom,Ville) :-
    demandeFournisseur(Nom,Ville).

union_fourn(Nom,Ville) :-
    fournisseurReference(_,Nom,Ville),
    not(demandeFournisseur(Nom,Ville)).

inter_fourn(Nom,Ville) :-
    fournisseurReference(_,Nom,Ville),
    demandeFournisseur(Nom,Ville).

diffens_fourn(Nom,Ville) :-
    demandeFournisseur(Nom,Ville),
    not(fournisseurReference(_,Nom,Ville)).

% ==== Q 2.4 Produit cartesien ====
prod_fournliv(NumFournR,Nom,Ville,NumFournL,Piece,Qte) :-
    fournisseurReference(NumFournR,Nom,Ville),
    livraison(NumFournL,Piece,Qte).

% ==== Q 2.5 Jointure ====
join_fournliv(NumFourn,Nom,Ville,Piece,Qte) :-
    fournisseurReference(NumFourn,Nom,Ville),
```

```
    livraison (NumFourn, Piece, Qte) .

join_fournliv_plus (NumFourn, Nom, Ville, Piece, Qte) :-
    fournisseurReference (NumFourn, Nom, Ville) ,
    livraison (NumFourn, Piece, Qte) ,
    Qte > 350 .

% ==== Q 2.6 Division ====

division_bis ([], _) .

division_bis ([T|Q], Fourn) :-
    livraison (Fourn, T, _) ,
    division_bis (Q, Fourn) .

division (Fourn) :-
    findall (NumP, piece_a_lion (NumP, _) , R) ,
    division_bis (R, Fourn) .

% ==== Q 2.7 Calculs de totaux ====

qte_livraison (Fourn, Qte) :-
    livraison (Fourn, _, Qte) .

total_livraison_add ([], 0) .

total_livraison_add ([T|Q], TotalSum) :-
    total_livraison_add (Q, Total) ,
    TotalSum is Total + T .

total_livraison (Fourn, Total) :-
    findall (Qte, qte_livraison (Fourn, Qte) , ListeQte) ,
    total_livraison_add (ListeQte, Total) .

% =====
% SECTION 3 : Au del de l'algre relationnelle
% =====

% ==== Q 3.1 composant_to_make ====

piece_to_make ([], []) .

piece_to_make ([T|Q], ListeAllToMakeRes) :-
    composant_to_make (T, ListeAllToMake1) ,
    piece_to_make (Q, ListeAllToMake2) ,
    append (ListeAllToMake1, ListeAllToMake2, ListeAllToMakeRes) .

composant_to_make (Composant, ListeAllToMake) :-
    findall (CompToMake, assemblage (Composant, CompToMake, _) , ListeCompToMake) ,
    piece_to_make (ListeCompToMake, ListeAllToMakePiece) ,
    append (ListeCompToMake, ListeAllToMakePiece, ListeAllToMake) .
```

```
% ==== Q 3.2 Total_piece_pour_composant ====

make_total([],0).
make_total([composant_qte(_,Qte)|Q],TotalRes):-
    make_total(Q,Total),
    TotalRes is Total + Qte.

total_piece_liste_2([],0).

total_piece_liste_2([composant_qte(Composant,Qte)|Q],TotalRes) :-
    total_piece_pour_composant(Composant,Total1),
    total_piece_liste_2(Q,Total2),
    TotalRes is Total1*Qte + Total2.

total_piece_liste_1([],1).

total_piece_liste_1([composant_qte(Composant,Qte)|Q],TotalRes) :-
    total_piece_pour_composant(Composant,Total1),
    total_piece_liste_2(Q,Total2),
    TotalRes is Total1*Qte + Total2.

total_piece_pour_composant(Composant,Total) :-
    findall(composant_qte(CompToMake,Qte),
        assemblage(Composant,CompToMake,Qte),
        ListeCompToMake),
    total_piece_liste_1(ListeCompToMake,Total).

/*
=====
=====
Tests
=====

% ===== Q 2.1 Selection =====
[eclipse 5]: piece_a_lion(N,P).

N = p1
P = tole
Yes (0.00s cpu, solution 1, maybe more) ? ;

N = p2
P = jante
Yes (0.00s cpu, solution 2)
% =====

% ===== Q 2.2 Projection =====
[eclipse 7]: projection(N,L).

N = tole
```

```
L = lyon
Yes (0.00s cpu, solution 1, maybe more) ? ;

N = jante
L = lyon
Yes (0.00s cpu, solution 2, maybe more) ? ;

N = jante
L = marseille
Yes (0.00s cpu, solution 3, maybe more) ? ;
% =====

% ==== Q 2.3 Union, intersection et difference ensembliste ====
[eclipse 12]: inter_fourn(N,V).

N = dupont
V = lyon
Yes (0.00s cpu, solution 1, maybe more) ? ;

N = durand
V = lille
Yes (0.00s cpu, solution 2, maybe more) ? ;

N = martin
V = rennes
Yes (0.00s cpu, solution 3, maybe more) ? ;

N = michel
V = clermontFerrand
Yes (0.00s cpu, solution 4, maybe more) ? ;

N = smith
V = paris
Yes (0.00s cpu, solution 5, maybe more) ? ;

N = brown
V = marseille

[eclipse 20]: diffens_fourn(Nom,Ville).

Nom = dupond
Ville = lille
% =====

% ===== Q 2.4 Produit cartesien =====
[eclipse 22]: prod_fournliv(Nfr,Nom,Ville,Nl,Piece,Qte).

Nfr = f1
Nom = dupont
Ville = lyon
Nl = f1
Piece = p1
Qte = 300
```

```
Yes (0.00s cpu, solution 1, maybe more) ? ;

Nfr = f1
Nom = dupont
Ville = lyon
Nl = f2
Piece = p2
Qte = 200
Yes (0.00s cpu, solution 2, maybe more) ? ;

Nfr = f1
Nom = dupont
Ville = lyon
Nl = f3
Piece = p3
Qte = 200
Yes (0.00s cpu, solution 3, maybe more) ? ;

Nfr = f1
Nom = dupont
Ville = lyon
Nl = f4
Piece = p4
Qte = 400
Yes (0.00s cpu, solution 4, maybe more) ? ;

to continue...
% =====

% ===== Q 2.5 Jointure =====
[eclipse 26]: join_fournliv(NumF,Nom,Ville,Piece,Qte).

NumF = f1
Nom = dupont
Ville = lyon
Piece = p1
Qte = 300
Yes (0.00s cpu, solution 1, maybe more) ? ;

NumF = f1
Nom = dupont
Ville = lyon
Piece = p2
Qte = 300
Yes (0.00s cpu, solution 2, maybe more) ? ;

NumF = f2
Nom = durand
Ville = lille
Piece = p2
Qte = 200
Yes (0.00s cpu, solution 3, maybe more) ?
```

```
[eclipse 28]: join_fournliv_plus(NumF,Nom,Ville,Piece,Qte).

NumF = f4
Nom = michel
Ville = clermontFerrand
Piece = p4
Qte = 400
Yes (0.00s cpu, solution 1, maybe more) ? ;

NumF = f6
Nom = brown
Ville = marseille
Piece = p5
Qte = 500
Yes (0.00s cpu, solution 2, maybe more) ? ;

NumF = f6
Nom = brown
Ville = marseille
Piece = p6
Qte = 1000
Yes (0.00s cpu, solution 3, maybe more) ? ;
% =====

% ===== Q 2.6 Division =====

| ?- division(R).

R = f1 ? ;

R = f4 ? ;

% =====

% ==== Q 2.7 Calculs de totaux ====

| ?- total_livraison(f1,Total).

Total = 600

yes
| ?- total_livraison(f6,Total).

Total = 1800

yes
=====

% ==== Q 3.1 composant_to_make ====

| ?- composant_to_make(voiture,ListeDesTrucPourFaireMaCaisse).
```



```
ListeDesTrucPourFaireMaCaisse = [porte,roue,moteur,toile,vitre,jante,pneu,piston,soupape]  
=====
```

**/*

Listing 1: tp7.pl