

Solution série de TP 1: Création de règles et de requêtes en Prolog

Exercice 01:

1. Tous les mammifères.
 - ?- animal(X, mammifere, _, _, _, _).
2. Tous les mammifères qui sont des insectivores.
 - ?- animal(X, mammifere, insectivore, _, _, _, _).
3. Tous les oiseaux qui ne sont pas carnivores.
 - ?- animal(X, oiseau, Regime, _, _, _, _), Regime \== "carnivore ".
4. Tous les mammifères qui ont des rayures.
 - ?- animal(X, mammifere, _, rayures, _, _, _).
5. Les animaux dont le poids est compris entre 1 et 10 Kg.
 - ?- animal(X, _, _, _, Poid, _), Poid >= 1, Poid <= 10.
6. Les mammifères qui sont des herbivore et dont l'âge est inférieur à celui du zèbre.
 - ?- animal(X, mammifere, herbivore, _, _, _, Y), animal(zebre, _, _, _, _, Z), Z < Y.
7. Les mammifères et les reptiles qui vivent dans les mêmes lieux.
 - ?- animal(X, mammifere, _, _, L, _, _), animal(Y, reptile, _, _, L, _, _), X \= Y.
8. Les couples d'oiseaux qui ont les mêmes caractéristiques.
 - ?- animal(X, oiseau, _, C, _, _, _), animal(Y, oiseau, _, C, _, _, _), X \= Y.

Exercice 02 :

- 1) Quelle est la requête Prolog qui permet d'avoir le père de nabil (sans utiliser le prédicat père).
 - ?- enfant(nabil, X), masculin(X).
- 2) fils (X,Y) qui exprime que X est un fils de Y.
 - fils(X,Y) :- enfant(X,Y), masculin(X).
- 3) fille(X, Y) qui exprime que X est une fille de Y.
 - fille(X,Y) :- enfant(X,Y), feminin(X).
- 4) frere-ou-sœur(X, Y) qui exprime que X est frère ou sœur de Y.
 - frere-ou-soeur(X,Y) :- enfant(X,Z), enfant(Y,Z), X \= Y.
- 5) Définissez les prédicats suivants : mere/2, grand-pere/2, frere/2, tante/2, cousin/2 .
 - mere(X,Y) :- enfant(Y,X), feminin(X).
 - grand-pere(X,Y) :- pere(X,Z), enfant(Y,Z).
 - frere(X,Y) :- enfant(X,Z), enfant(Y,Z), masculin(X), X \= Y.
 - Tante(X,Y) :- feminin(X), enfant(X,K), enfant(Z,K), enfant(Y,Z).
 - Cousin(X,Y) :- enfant(X,Z1), enfant(Y,Z2), frere-ou-soeur(Z1,Z2).

Solution série de TP 2 : Prolog et les bases de données

1. Créer le prédicat « *pjOran(Reference,Nature)* » donnant les informations concernant les projets en cours à oran.
 - *pjOran(Reference, Nature) : - projet(Reference, Nature, oran).*
2. Créer le prédicat « *fpDistincts(Fournisseur,Piece)* » faisant apparaître les numéros d'identification des fournisseurs et des pièces pour tous les fournisseurs et toutes les pièces non situés dans la même ville.
 - *fpDistincts(Fournisseur, Piece) : - fournisseur(Fournisseur, _, _, VilleFournisseur), piece(Piece, _, _, _, VillePiece), VilleFournisseur \= VillePiece.*
3. Créer le prédicat « *pjF1P1(Projet,Ville)* » faisant apparaître tous les projets alimentés par le fournisseur *f1* ou utilisant la pièce *p1*.
 - *pjF1P1(Projet, Ville) : - (livraison(f1, _, Projet, _) ; livraison(_, P1, Projet, _)), projet(Projet, _, Ville).*
4. Trouver tous les détails des projets de annaba.
 - *?- Projet (Reference, Nature, annaba).*
5. Trouver les références des fournisseurs du projet *pj1* .
 - *?- livraison(Fournisseur, _, pj1, _).*
6. Quelles sont les livraisons dont la quantité est comprise entre 300 et 750 ?
 - *?- livraison(Fournisseur, Piece, Projet, Quantite), Quantite >= QuantiteMin, Quantite <= QuantiteMax.*
7. Trouver tous les triplets (fournisseur,piece,projet) tels que le fournisseur, la pièce et le projet soient situés dans la même ville.
 - *tripletsMemeVille(Fournisseur, Piece, Projet) : - livraison(Fournisseur, Piece, Projet, _), fournisseur(Fournisseur, _, _, Ville), piece(Piece, _, _, _, VillePiece), projet(Projet, _, VilleProjet), Ville = VillePiece, Ville = VilleProjet.*
8. Trouver les références des pièces provenant d'un fournisseur de annaba.
 - *piecesFournisseurAnnaba(Piece) : - livraison(Fournisseur, Piece, _, _), fournisseur(Fournisseur, _, _, annaba).*
9. Trouver les références des pièces provenant d'un fournisseur de annaba, et destinées à un projet de annaba.
 - *piecesFournisseurEtProjetAnnaba(Piece) : - livraison(Fournisseur, Piece, Projet, _), fournisseur(Fournisseur, _, _, annaba), projet(Projet, _, annaba).*