



Université d'Ottawa • University of Ottawa

**Faculté de Génie – EECS**

**CSI2520 : PARADIGMES DE PROGRAMMATION**

**Hiver 2014 – Solution Tutorat 1**

**Exercice 1**

Soit la base de faits Prolog suivante :

```
born(charles, lisa, philip).  
born(anne, lisa, philip).  
born(bob, lisa, philip).  
born(edward, lisa, philip).  
born(diana, mary, richard).  
born(paul, diana, charles).  
born(john, diana, charles).
```

Quel sera le résultat de la requête suivante :

?- born(S, lisa, Y), born(G, M, S)

Donner toutes les solutions qui seront obtenues, dans l'ordre où elles seront trouvées.

Solution :

```
S = charles,  
Y = philip,  
G = paul,  
M = diana ;  
S = charles,  
Y = philip,  
G = john,  
M = diana ;  
false.
```

## **Exercice 2**

Soit la base de faits suivantes :

parent( marie, robert ).  
parent( paul, robert ).  
parent( paul, emma ).  
parent( robert, vickie ).  
parent( robert, anne ).  
parent( robert, gail ).  
parent( robert, stan ).  
parent( stan, jack ).  
female(marie).  
female(emma).  
female(gail).  
female(anne).  
female(vickie).

Et le prédicat suivant :

sister(X, Y) :- parent(Z, X), parent(Z, Y), female(X).

Donner, dans l'ordre, toutes les solutions qui seront produites par le prédicat suivant :

?- sister(X,gail).

**Solution :**

**X = vickie ;**

**X = anne ;**

**X = gail ;**

**false.**

### **Exercice 3**

Soit les prédicats suivants créés afin de signifier que deux personnes qui combattent le même ennemi sont des alliés :

```
combat(paul,pierre).  
combat(jean,simon).  
combat(jean,pierre).  
alliés(X,Y) :- combat(X,Z),combat(Y,Z).
```

Lister dans l'ordre toutes les solutions qui seront trouvées par la requête suivante :

```
?- alliés(X,Y).
```

Solution :

```
X = Y, Y = paul ;  
X = paul,  
Y = jean ;  
X = Y, Y = jean ;  
X = jean,  
Y = paul ;  
X = Y, Y = jean.
```

### **Exercice 4**

Définir une base de faits des personnes qui se connaissent mutuellement :

```
connaitre (robert,stan).
```

```
connaitre(robert,stan).  
connaitre(robert,hugo).  
connaitre(sara,hugo).  
connaitre(christian,hugo).  
connaitre(robert,sara).  
connaitre(mustapha,robert).  
connaitre(stan,christian).  
connaitre(robert,sara).  
connaitre(hugo,mustapha).  
connaitre(sara,mustapha).  
connaitre(sebastian,robert).  
connaitre(sebastian,hugo).  
connaitrem(X,Y):-connaitre(X,Y);connaitre(Y,X).
```

Ecrire les requêtes Prolog qui répondent aux questions suivantes:

Est ce que je connais cette personne?

```
?- connaitrem(robert,sebastian).  
true ;  
false.
```

```
?- connaitrem(robert,hugo).  
true ;  
false.
```

```
?- connaitrem(robert,alexandre).  
false.
```

Combien de personnes me connaissent?

```
?- aggregate_all(count, connaitrem(robert,X), Count).
```

Qui peut m'introduire à cette personne? (Quelqu'un que je connais qui connaît la personne)

```
?- connaitrem(sara,X),connaitrem(stan,X).  
X = robert ;  
X = robert ;  
false.
```

### **Exercice 5**

Soit le programme suivant:

```
p(X) :- a(X).  
p(X) :- b(X),c(X),d(X).  
p(X) :- f(X).
```

```
a(1). b(1). c(1). b(2). c(2). d(2). f(3). b(4). c(4). d(4).
```

a) Lister l'ensemble des solutions qui seront obtenus à partir de la requête :

```
?- p(X).
```

**?- p(X).**

**X = 1 ;**

**X = 2 ;**

**X = 4 ;**

**X = 3.**

### **Exercice 6**

Considérez la base de faits suivantes:

trajet('Windsor', 'Toronto',bus).

trajet('Windsor', 'Burlington',bus).

trajet('Toronto', 'NorthBay',bus).

trajet('Toronto', 'Ottawa',bus).

trajet('Toronto', 'Kingston',bateau).

trajet('Kingston', 'Ottawa',bateau).

Donnez les requêtes suivantes:

1. y a t il un chemin entre Windsor et Ottawa?

**?- trajet('Windsor','Ottawa',X).**

**false.**

2. Peut-on visiter la ville du Québec?

**?- trajet(X,'Quebec',Y).**

**false.**

3. Imprimer les villes qui sont dans le trajet de Toronto à Ottawa.

**?-trajet('Toronto',X,Y),trajet(X,'Ottawa',Z).**

**X = 'Kingston',**

**Y = Z, Z = bateau.**

4. Peut on avoir un chemin agréable entre Burlington et Ottawa? (un chemin est dit agréable lorsqu'on alterne les trajets en bus et en bateau!).

**agréable(X,Y):-trajet(X,Z,T),trajet(Z,Y,W),T\=W.**

**?- agréable('Toronto','Ottawa').**

**false.**

**trajet('Kingston', 'Ottawa',bus).**

**?- agréable('Toronto','Ottawa').**

**true.**