



ÉCOLE CENTRALE LYON

UE PRO
ALGORITHME ET RAISONNEMENT
RAPPORT

Rapport de TD1

Élèves :

Mattéo ROUSSEAU

Enseignant :

Emmanuel DELLANDREA

7 mars 2024

Table des matières

1	Travail 1 - Famille	2
1.1	Base de donnée	2
1.2	Question 1	2
1.3	Question 2	2
1.4	Question 3	3
1.5	Question 4	3
1.6	Question 5	4
1.7	Question 6	5
1.8	Question 7	5
2	Travail 2 - Livre	7
2.1	Base de donnée	7
2.2	Question 1	7
2.3	Question 2	8
2.4	Question 3	8
2.5	Question 4	8
2.6	Question 5	9
2.7	Question 6	9
2.8	Question 7	9
2.9	Question 8	10
2.10	Question 9	10
3	Conclusion	11

1 Travail 1 - Famille

1.1 Base de donnée

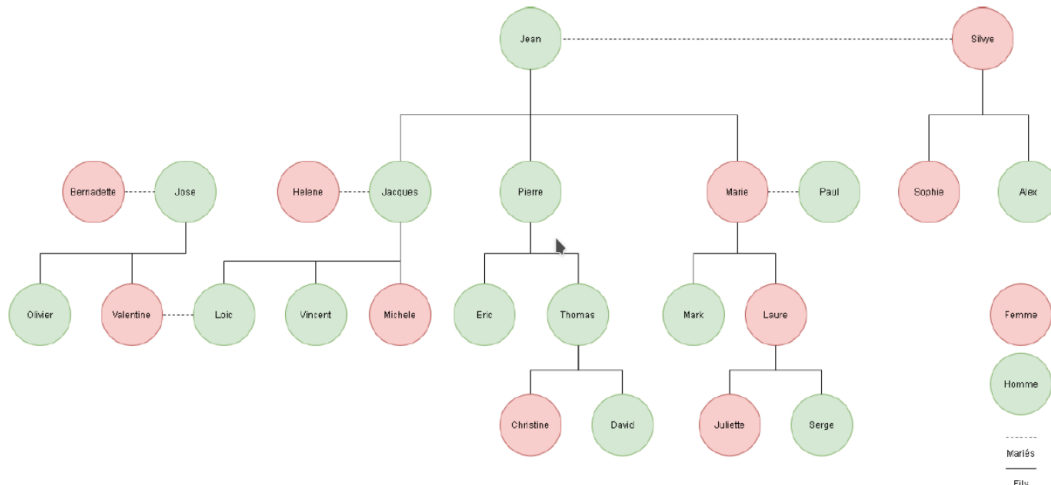


FIGURE 1 – Arbre généalogique considéré lors du BE

On considère cet arbre généalogique comme base de donnée pour ce premier exercice. Les trois relations codées sous la forme de prédicats sont les suivantes :

- La relation $epouse(Femme, Mari)$.
- La relation $enfant(Enfant, Père)$.
- La relation $enfant(Enfant, Mère)$.

On commence donc par écrire chaque relation présente sur l'arbre généalogique dans un fichier 'famille.pl'.

1.2 Question 1

Écrire le prédicat $epoux(Mari, Femme)$ que vous déduirez de la relation $epouse(Femme, Mari)$.

Un homme est l'époux d'une femme si la femme est l'épouse de l'homme. On obtient donc :

$$epoux(Homme, Femme) : \neg epouse(Femme, Homme).$$

1.3 Question 2

Compléter (à la main) la base de données et compléter le prédicat $enfant/2$ (enfant à 2 paramètres) de sorte que :

- E est enfant de M si M est épouse de P .
- E est enfant de P si P est l'époux de M .

La base de donnée telle qu'elle est donnée dans l'énoncé ne prend en compte le lien de parenté (prédicat enfant/2) qu'avec l'un des deux parents. Pour cette question, on rajoute donc tous les liens de parenté qui découlent du mariage de deux parents :

```

6
7  enfant(jacques, jean).
8  enfant(jacques, sylvie).
9  enfant(pierre, jean).
10 enfant(pierre, sylvie).
11 enfant(marie, jean).
12 enfant(marie, sylvie).
13

```

FIGURE 2 – Exemple de prédicats traduisants des relations de parenté une fois le mariage prit en compte

1.4 Question 3

Ajouter le prédicat homme/1 pour chaque individu (masculin de couleur verte dans le graphe) à cette base :

On ajoute les prédicats dans notre fichier 'famille.pl' :

```

48
49 % Q3
50 homme(jean).
51 homme(jose).
52 homme(jacques).
53 homme(pierre).
54 homme(paul).
55 homme(alex).
56 homme(olivier).
57 homme(vincent).
58 homme(eric).
59 homme(loic).
60 homme(thomas).
61 homme(mark).
62 homme(david).
63 homme(serge).
64

```

FIGURE 3 – Prédicats des hommes de l'arbre généalogique

1.5 Question 4

Ajouter la règle pour les femmes de l'arbre généalogique :

1.6 Question 5

Écrire les prédicats basiques : $pere(Dad, Enf)$, $mere(Mam, Enf)$, $parent(Par, Enf)$, $fil(Enf, Par)$, $fille(Enf, Par)$:

Voici les prédicats obtenus pour décrire les relations demandées :

```
69
70 % Q5
71 pere(Dad, Enf) :-
72     (enfant(Enf, Dad),
73     homme(Dad));
74     (enfant(Enf, Mom),
75     epouse(Mom, Dad)).
76
77 mere(Mom, Enf) :-
78     (enfant(Enf, Mom),
79     femme(Mom));
80     (enfant(Enf, Mom),
81     epoux(_, Mom)).
82
83 parent(Par, Enf) :-
84     pere(Par, Enf);
85     mere(Par, Enf).
86
87 fils(Enf, Par) :-
88     enfant(Enf, Par),
89     homme(Enf).
90
91 frere(Enf1, Enf2) :-
92     parent(Par, Enf1),
93     parent(Par, Enf2),
94     Enf1 \= Enf2,
95     homme(Enf1).
96
```

FIGURE 4 – Prédicats des relation pere, mere, parent, fils et fille

1.7 Question 6

Qui est "le frère du fils du mari de la mère du père de la soeur du fils de la mère de loic" ?

Voici les prédicats obtenus pour décrire les relations demandées :

```
?- mere(X, loic), fils(Y, X), soeur(Z, Y), pere(A, Z), mere(B, A), epoux(C, B), fils(D, C), frere(E, D).
X = helene,
Y = vincent,
Z = michele,
A = D, D = jacques,
B = sylvie,
C = jean,
E = pierre ;
X = helene,
Y = vincent,
Z = michele,
A = D, D = jacques,
B = sylvie,
C = jean,
E = pierre ;
X = helene,
Y = vincent,
Z = michele,
A = D, D = jacques,
B = sylvie,
C = alex,
E = alex ;
X = helene,
Y = vincent,
Z = michele,
A = D, D = jacques,
B = sylvie,
C = alex,
E = alex ;
```

FIGURE 5 – Requête et résultat de la question 6 avec "loic".

On remarque que la requête mène à plusieurs résultats possibles pour la variable E qui sont $E = pierre$ et $E = alex$. Les autres résultats affichés mènent tous aux deux nom évoqués juste avant.

En remplaçant "loic" par "alex" dans la requête, on obtient le résultat suivant (car dans notre arbre "jean" n'a pas de père) :

```
?- mere(X, alex), fils(Y, X), soeur(Z, Y), pere(A, Z), mere(B, A), epoux(C, B), fils(D, C), frere(E, D).
false.
```

FIGURE 6 – Requête et résultat de la question 6 avec "alex".

1.8 Question 7

Un jeune homme (pedro) se marie avec une dame nommée "brigitte". Ensemble, ils ont un enfant (un garçon nommé "yves"). Quelque temps après, le père de pedro ("yannick") rencontre "elisabeth" la fille de "brigitte". Ils se marient également et ont un fils "luc". En admettant qu'un beau-fils est comme un fils et un beau père comme un père, pouvez-vous démontrer que "pedro" est son propre grand-père !

Voici l'implémentation dans *famille.pl* des nouvelles données :

```
% Q7
homme(pedro).
epouse(brigitte, pedro).

homme(yves).
enfant(yves, brigitte).
enfant(yves, pedro).

homme(yannick).
enfant(pedro, yannick).

enfant(elisabeth, brigitte).

epouse(elisabeth, yannick).
enfant(luc, elisabeth).
enfant(luc, yannick).
```

FIGURE 7 – Implémentation des nouvelles données.

On modifie ensuite le prédicat *pere/2* qui ne tient pas compte des liens du mariage dans les liens de parenté. Voici les modifications effectuées :

```
pere(Dad, Enf) :-
    (enfant(Enf, Dad), homme(Dad));
    (enfant(Enf, Mom), epouse(Mom, Dad));
    (epoux(Enf, Epouse), pere(Dad, Epouse)).
```

FIGURE 8 – Modification du prédicat *pere/2*.

On effectue ensuite la requête dans la console prolog qui nous renvoie le (ou les) grand-père de "pedro" :

```
?- pere(A, pedro), pere(B, A).
A = yannick,
B = pedro ;
false.
```

FIGURE 9 – Requête et résultat de la recherche du grand père de "pedro".

On obtient bien que "pedro" est son propre grand-père !

2 Travail 2 - Livre

2.1 Base de donnée

Dans cette partie on considère une base de donnée autour du prédicat livre/3 qui est constitué des prédicats auteur/2, titre/1 et prix/1.

L'arbre formé à partir de ces relations est le suivant :

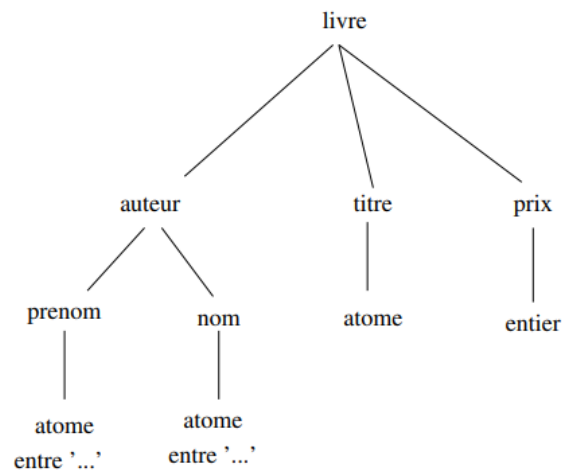


FIGURE 10 – Définition du prédicat livre/3.

2.2 Question 1

Quel est le nom d'un auteur de livre dont le prénom est victor ?

On cherche ici le nom d'un auteur de livre dont le prénom est victor. La réponse souhaitée est :

```

?- livre(auteur(victor, Nom), _, _).
Nom = hugo ;
Nom = hugo.

```

FIGURE 11 – Réponse à la question 1.

2.3 Question 2

Existe-t-il un livre valant 40 Euros ?

On cherche ici un livre dont le prix vaut 40 euros. On procède de la même manière qu'à la question précédente mais pour le prix :

```
?- livre(X, Y, prix(40)).  
false.
```

FIGURE 12 – Réponse à la question 2.

Il n'y a donc pas de livre dont le prix est de 40 euros.

2.4 Question 3

Quel est le titre d'un livre de fredric dard ?

On cherche ici un livre dont l'auteur est fredric dard. On procède de la même manière qu'à la question précédente mais pour le titre :

```
?- livre(auteur(fredric, dard), Titre, _).  
Titre = titre('l"espion').
```

FIGURE 13 – Réponse à la question 3.

Le titre est 'l"espion'.

2.5 Question 4

Quel est le prix du livre "le silence" ?

On cherche ici le prix du livre "le silence". On procède de la même manière qu'à la question précédente mais pour le prix avec une condition sur le titre :

```
?- livre(_, titre('le silence'), Prix).  
Prix = prix(180).
```

FIGURE 14 – Réponse à la question 4.

Le prix est de 180.

2.6 Question 5

Quels sont les livres d'umberto ecco ?

On cherche ici les livres de umberto ecco. On procède de la même manière qu'à la question précédente mais pour les livres entiers avec une condition sur l'auteur :

```
?- livre(auteur(umberto, ecco), Titre, _).
Titre = titre('le nom de la rose') ;
Titre = titre('pendule de foucault').
```

FIGURE 15 – Réponse à la question 5.

Les livres sont 'le nom de la rose' et 'pendule de foucault'.

2.7 Question 6

Existe-t-il des livres du même titre écrits par des écrivains différents ?

On cherche ici les livres ayant le même titre mais un auteur différent :

```
?- livre(Auteur1, Titre, _), livre(Auteur2, Titre, _), Auteur1 \= Auteur2.
Auteur1 = auteur(eugene, delonay),
Titre = titre('pendule de foucault'),
Auteur2 = auteur(umberto, ecco) ;
Auteur1 = auteur(umberto, ecco),
Titre = titre('pendule de foucault'),
Auteur2 = auteur(eugene, delonay) ■
```

FIGURE 16 – Réponse à la question 6.

Les livres sont 'le nom de la rose' et 'pendule de foucault'.

2.8 Question 7

Y a-t-il des livres dans la base avec un prix supérieur à 200 ?

On cherche ici les livres ayant un prix supérieur à 200 :

```
?- livre(Auteur, Titre, prix(Prix)), Prix >= 200.
Auteur = auteur(victor, hugo),
Titre = titre('les misérables'),
Prix = 300 ;
Auteur = auteur(victor, hugo),
Titre = titre('fantasia chez les ploucs'),
Prix = 200 ;
false.
```

FIGURE 17 – Réponse à la question 7.

Les livres sont 'les misérables' et 'fantasia chez les ploucs'.

2.9 Question 8

Écrire une règle permettant de connaître les livres écrits par un écrivain donné E dont le prix est égal à P .

On écrit cette règle dans le fichier livre.pl :

```
auteur_prix_fixes(E, P, Livres) :-
    findall((E, Titre), livre(E, Titre, P), Livres).
```

FIGURE 18 – Ecriture de la règle dans le fichier prolog.

On a utilisé la règle findall/3() de prolog qui permet de trouver tous les éléments qui respectent une conditions et les attribuent dans une liste.

On peut maintenant réaliser le test de cette règle et on obtient :

```
?- auteur_prix_fixes(auteur(victor, hugo), prix(200), X).
X = [(auteur(victor, hugo), titre('fantasia chez les ploucs'))].
```

FIGURE 19 – Test de la règle.

2.10 Question 9

Quels sont les livres que l'on peut acheter avec 400 euros ?

On écrit cette règle dans le fichier livre.pl sous la forme *achats_possibles/4* :

Dans un premier temps, on fait une vérification du prédicat livre/3 pour obtenir les informations d'un livre. Ensuite, on vérifie que le budget est supérieur au prix du livre et si c'est bien le cas, on met à jour le budget restant.

On fait ensuite un appel récursif de la règle avec le budget mis à jour et de nouvelles variables de stockages des données d'achat.

On fini ensuite par combiner les données du premier achat avec les données d'achat de l'appel récursif en des listes qui sont les variables du prédicats *achats_possibles/4*.

```
achats_possibles(Budget, Livres_achetes, Prix_des_livres_achetes, Total_cout) :-
    livre(_, titre(Titre), prix(Prix)),
    Prix <= Budget,
    Budget_restant #= Budget - Prix,
    achats_possibles(Budget_restant, Autres_livres_achetables, Prix_des_autres_livres_achetables, Cout_des_autres_achats),
    Livres_achetes = [Titre | Autres_livres_achetables],
    Prix_des_livres_achetes = [Prix | Prix_des_autres_livres_achetables],
    Total_cout #= Cout_des_autres_achats + Prix.

achats_possibles(Budget, [], [], 0).
```

FIGURE 20 – Ecriture de la règle dans le fichier prolog.

On peut maintenant réaliser le test de cette règle et on obtient :

```
?- achats_possibles(400, Livres_achetes, Prix_des_livres_achetes, Total_cout).
Livres_achetes = ['l'espion', 'l'espion', 'l'espion', 'l'espion', 'l'espion', 'l'espion', 'l'espion', 'l'espion'],
Prix_des_livres_achetes = [45, 45, 45, 45, 45, 45, 45, 45],
Total_cout = 360 ;
Livres_achetes = ['l'espion', 'l'espion', 'l'espion', 'l'espion', 'l'espion', 'l'espion', 'l'espion', 'l'espion'],
Prix_des_livres_achetes = [45, 45, 45, 45, 45, 45, 45, 45],
Total_cout = 315 ;
Livres_achetes = ['l'espion', 'l'espion', 'l'espion', 'l'espion', 'l'espion', 'l'espion', 'l'espion', 'pendule de foucault'],
Prix_des_livres_achetes = [45, 45, 45, 45, 45, 45, 125],
Total_cout = 395 ;
Livres_achetes = ['l'espion', 'l'espion', 'l'espion', 'l'espion', 'l'espion', 'l'espion', 'l'espion'],
Prix_des_livres_achetes = [45, 45, 45, 45, 45, 45, 45],
Total_cout = 270 ;
Livres_achetes = ['l'espion', 'l'espion', 'l'espion', 'l'espion', 'l'espion', 'l'espion', 'pendule de foucault', 'l'espion'],
Prix_des_livres_achetes = [45, 45, 45, 45, 45, 125, 45],
Total_cout = 395 ;
```

FIGURE 21 – Test de la règle.

3 Conclusion

Nous avons pu au travers de ce BE prendre en main prolog et ses particularités comme l'appel récursif au sein des règles pour le traitement des listes.