



ÉCOLE CENTRALE LYON

ALGORITHME ET RAISONNEMENT
BE1
RAPPORT

Base de données sur les familles

Élèves :
Hugo PUYBAREAU

Enseignant :
Emmanuel DELLANDREA

4 mars 2024

Table des matières

1	Introduction	2
2	Question 1	2
3	Question 2	3
4	Question 3	4
5	Question 4	5
6	Question 5	5
7	Question 6	6
8	Question 7	7

1 Introduction

L'objectif de ce premier devoir maison est de construire une base de données sur une famille sous PROLOG. Il est également demandé de rendre compte des différents liens entre les membres de la famille.

La famille recrée est représentée sur l'arbre suivant :

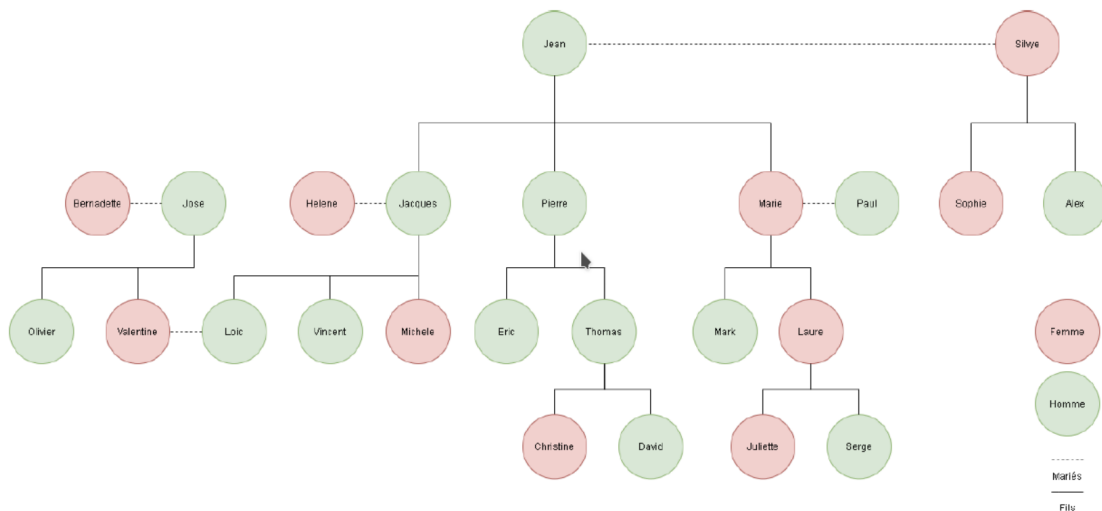


FIGURE 1 – Arbre généalogique de la famille

2 Question 1

Il est demandé d'écrire le prédicat *epoux*(*Mari*, *Femme*) que vous déduirez de la relation *epouse*(*Femme*, *Mari*).

J'ai rédigé cela de la manière suivante :

```
epoux(H,F) :- epouse(F,H) .
```

FIGURE 2 – Code pour la question 1

On effectue le test suivant et on voit que le code renvoie bien toutes les mariages importés.

```
?- epoux(H, F).
H = jean,
F = sylvie ;
H = paul,
F = marie ;
H = jacques,
F = helene ;
H = jose,
F = bernadette ;
H = loic,
F = velentine.
```

FIGURE 3 – Test pour la question 1

3 Question 2

Il est demandé de 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.

J'ai complété la base de données de cette façon :

```
enfant(jacques, sylvie).
enfant(marie, sylvie).
enfant(pierre, sylvie).

enfant(alex, jean).
enfant(sophie, jean).

enfant(laure, paul).
enfant(mark, paul).

enfant(loic, helene).
%enfant(helene, helene).??????
enfant(michele, helene).
enfant(vincent, helene).

enfant(olivier, bernadette).
enfant(valentine, bernadette).
```

FIGURE 4 – Code pour compléter la base de données dans la question 2

J'ai complété le prédicat de la manière suivante :

```
enfant(E,M) :- enfant(E,P), epouse(M,P).
enfant(E,P) :- enfant(E,M), epoux(P,M).
```

FIGURE 5 – Code pour le prédicat

On test et on voit que cela marche puisque *Silvy*, par exemple, a bien pour enfant *Jacques*, *Pierre*, *Marie* et *Jean* a bien pour enfant *Sophie* et *Alex*.

```
E = alex,
M = jean ;
E = sophie,
M = jean ;
E = laure,
M = paul ;
E = mark,
M = paul ;
E = mark,
M = marie ;
E = laure,
M = marie ;
E = serge,
M = laure ;
E = juliette,
M = laure ;
E = jacques,
M = sylvie ;
E = marie,
M = sylvie ;
E = pierre,
M = sylvie
```

FIGURE 6 – Test pour le prédicat

4 Question 3

Il nous est demandé de compléter la base de données en listant tous les hommes de la famille.

```
%-----
% hommes (homme)
% -----
homme(jean).
homme(loic).
homme(paul).
homme(pierre).
homme(jacques).
homme(mark).
homme(thomas).
homme(vincent).
homme(eric).
homme(david).
homme(serge).
homme(jose).
homme(alex).
homme(olivier).
```

FIGURE 7 – Code pour la question 3

On teste avec la requête suivante et on a ce résultat :

```
?- homme(X)
|
X = jean ;
X = loic ;
X = paul ;
X = pierre ;
X = jacques ;
X = mark ;
X = thomas ;
X = vincent ;
X = eric ;
X = david ;
X = serge ;
X = jose ;
X = alex ;
X = olivier.
```

FIGURE 8 – Test de la requête

5 Question 4

Il nous est demandé d'ajouter un test pour les femmes. Pour ça on utilise une négation :

```
femme(X) :- nonvar(X), not(homme(X)).
```

FIGURE 9 – Code pour la question 4

On teste :

```
?- femme(sylvie).
true.

?- femme(olivier).
false.
```

FIGURE 10 – Test de la question 4

Ce que l'on obtient est logique. Cependant, on ne peut pas écrire 'femme(X).' et obtenir une liste de toutes les femmes car le code implanté est un test et qu'il ne permet pas d'initialiser des femmes comme on l'a fait pour les hommes.

6 Question 5

On code les prédicats de la manière suivante :

On test pour 'mere', le fonctionnement est identique pour les autres prédicats :

```
%enfant(E,M) :- enfant(E,P), epouse(M,P).
% enfant(E,P) :- enfant(E,M), epoux(P,M). %On met ça en note parce que
% ça fait des boucles infinies sur les prédicats de la question 5.

pere(Dad, Enf) :- enfant(Enf, Dad), homme(Dad).
mere(Mam, Enf) :- enfant(Enf, Mam), femme(Mam).
parent(Par, Enf) :- pere(Par, Enf); mere(Par, Enf).
fils(Enf, Par) :- enfant(Enf, Par), homme(Enf).
fille(Enf, Par) :- enfant(Enf, Par), femme(Enf).▲
```

FIGURE 11 – Code des prédicats

```
?- mere(Mam, Enf).
Mam = sylvie,
Enf = alex ;
Mam = sylvie,
Enf = sophie ;
Mam = marie,
Enf = mark ;
Mam = marie,
Enf = laure ;
Mam = laure,
Enf = serge ;
Mam = laure,
Enf = juliette ;
Mam = sylvie,
Enf = jacques ;
Mam = sylvie,
Enf = marie ;
Mam = sylvie,
Enf = pierre ;
Mam = helene,
Enf = loic ;
Mam = helene,
Enf = michele ;
Mam = helene,
Enf = vincent ;
Mam = bernadette,
Enf = olivier ;
Mam = bernadette,
Enf = valentine ;
Mam = brigitte,
Enf = yves ;
Mam = elisabeth,
Enf = luc ;
Mam = elisabeth,
Enf = pedro ;
Mam = brigitte,
Enf = pedro.
```

FIGURE 12 – Test de la question 5

7 Question 6

Il faut trouver qui est "le frère du fils du mari de la mère du père de la soeur du fils de la mère de loic". Je rajoute donc le code des prédicats 'frere' et 'soeur'.

J'utilise ensuite la requête suivante :

On trouve *Alex*, *Jacques* et *Pierre* ce qui paraît logique si on cherche la réponse à la main dans l'arbre.

Pour Alex, j'utilise la même requête et je trouve *Jacques* et *Pierre*. Je m'attendais plutôt à trouver 'False' ou une erreur car *Jean* n'a pas de mère.

Je ne comprends pas pourquoi je n'obtiens pas une erreur.

```
frere(F,Enf):- %Un frère c'est si les deux parents sont les mêmes sinon c'est demi
frère
    homme(F),
    dif(F,Enf),
    pere(Dad,Enf),
    mere(Mam,Enf),
    pere(Dad,F),
    mere(Mam,F).

soeur(S,Enf) :- %Idem
    femme(S),
    dif(S,Enf),
    pere(Dad,Enf),
    mere(Mam,Enf),
    pere(Dad,S),
    mere(Mam,S).
```

FIGURE 13 – Code pour la question 6

```
?- frere(Loic, Fils), epoux(Mari, Mere), pere(Pere, Soeur), mere(MereSoeur, Soeur), fils(X, MereSoeur).
Loic = loic,
Fils = vincent,
Mari = Pere, Pere = jean,
Mere = MereSoeur, MereSoeur = sylvie,
Soeur = marie,
X = alex ;
Loic = loic,
Fils = vincent,
Mari = Pere, Pere = jean,
Mere = MereSoeur, MereSoeur = sylvie,
Soeur = marie,
X = jacques ;
Loic = loic,
Fils = vincent,
Mari = Pere, Pere = jean,
Mere = MereSoeur, MereSoeur = sylvie,
Soeur = marie,
X = pierre ;
```

FIGURE 14 – Requête pour la question 6

8 Question 7

On ajoute dans l'arbre de famille les relations décrites en comptant que les beau-fils sont comme des fils, etc. On trouve avec '*pere(pedro, Pere), pere(Pere, GrandPere).*' que '*GrandPere = pedro*'.