



ÉCOLE CENTRALE LYON

ELECTIF
A&R
RAPPORT

Algorithmes et raisonnement

Élève :
Flavio SESTU

Enseignant :
Emmanuel DELLANDREA

16 février 2022

Table des matières

1	Introduction	2
2	Arbre généalogique	2
2.1	Prédicat epoux(Mari, Femme)	2
2.2	Prédicat enfant/2	2
2.3	Prédicat homme/1	2
2.4	Prédicat femme/1	3
2.5	Prédicats pere/2, mere/2, parent/2, fils/2, fille/2	3
2.6	Relation Question VI	4
2.7	Pedro est son propre grand-père	4
3	Livres	5
3.1	Nom d'un auteur dont le prénom est Victor	5
3.2	Livre valant 40 euros	6
3.3	Titre d'un livre de Fredric Dard	6
3.4	Prix du livre "le silence"	6
3.5	Livres d'Umberto Ecco	6
3.6	Même titre, auteurs différents	6
3.7	Livres avec prix supérieur à 200	7
3.8	Règle écrivain/prix : prédicat livre _{EP} /2	7
3.9	Livres pouvant être achetés avec 400 euros	7
4	Conclusion	7

1 Introduction

L'objectif de ce rapport est de rendre compte de la prise en main de Prolog, langage de programmation logique.

Je présenterai donc succinctement :

- Un premier travail sur un arbre généalogique, permettant de prendre en main le langage,
- Un second sur les livres, permettant de travailler sur les "termes" Prolog.

2 Arbre généalogique

2.1 Prédicat `epoux(Mari, Femme)`

Ayant déjà un prédicat `epouse(Femme, Mari)`, le prédicat `epoux(Mari, Femme)` peut être déduit de la façon suivante :

```
1 epoux(Mari, Femme) :- epouse(Femme, Mari).
```

Ainsi, chaque couple de variables `Mari` et `Femme` tels que `epouse(Femme, Mari)` est vrai impliquera que `epoux(Mari, Femme)` est vrai.

2.2 Prédicat `enfant/2`

D'après l'énoncé :

`E` est enfant d'un père `P` si `enfant(E,P)` est donné.

`E` est enfant d'une mère `M` si `enfant(E,M)` est donné.

De plus,

`E` est enfant de `M` si `M` est épouse de `P`.

`E` est enfant de `P` si `P` est l'époux de `M`.

Ainsi, j'ai complété à la main (comme demandé) la liste des enfants de sorte à ce que si `enfant(E,P)` et `epouse(M,P)` existent, j'ajoute `enfant(E,M)` et de même pour l'enfant d'une mère ayant un époux.

2.3 Prédicat `homme/1`

Ajout des hommes : suivre l'arbre généalogique en déduisant des prénoms. Rien de bien compliqué pour le moment, ça servira surtout pour la question suivante.

2.4 Prédicat femme/1

Une femme va être toute valeur n'étant pas un homme mais qui est définie (est déjà apparue dans le programme). En effet, si une valeur est précédemment définie, on sait que si c'est un homme on aurait l'information du fait de la question précédente, cependant si la valeur n'a pas été définie on ne peut pas dire que c'est une femme sans avoir d'information sur le fait que ce soit un homme ou non.

La règle est décrite comme ceci :

```
1  femme(X) :- nonvar(X), \+ homme(X).
```

En effet, toute valeur qui est une valeur précédemment définie (`nonvar(X)`) et qui n'est pas un homme (`\+ homme(X)`, avec `\+` la négation) sera ainsi une femme.

2.5 Prédicats pere/2, mere/2, parent/2, fils/2, fille/2

On peut définir le prédicat `pere/2` tel qu'un enfant a un homme pour père ou qu'un enfant a une mère qui a un époux / qui est une épouse.

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

Note : le ';' fonctionne comme un OU, tandis que les virgules font office de ET.

De même, on peut définir la mère telle qu'un enfant l'a pour mère et que c'est une femme, ou tel qu'un enfant a un père qui est un époux / qui a une épouse.

```
1  mere(Mam,Enf) :- enfant(Enf,Mam), femme(Mam) ; epouse(Mam,Dad), enfant(Enf,Dad).
```

De ces deux prédicats, on peut déduire le prédicat `parent/2`, tel qu'un parent est père ou mère :

```
1  parent(Par, Enf) :- pere(Par, Enf) ; mere(Par, Enf).
```

Ensuite, à partir du prédicat `enfant/2`, on peut déduire les prédicats `fils/1` ou `fille/1` à l'aide des prédicats `homme/1` et `femme/1`.

```
1  fils(Enf,Par) :- enfant(Enf,Par), homme(Enf).  
2  fille(Enf,Par) :- enfant(Enf,Par), femme(Enf).
```

Tous les prédicats précédents (pour les 5 questions) ont été testés et ont le comportement escompté.

2.6 Relation Question VI

La question 6 demande de trouver quelles personnes peuvent satisfaire la question suivante :

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

Pour cela, il est pratique de remonter la question à l'envers : on cherche la mère de X, puis le fils de cette mère, puis la soeur de ce fils, ... et on peut pour cela créer la règle suivante :

```
1 rel(P1,P2) :- mere(Mam,P1), fils(F1,Mam), soeur(So1, F1), pere(Pere, So1),  
mere(Mam2, Pere), epoux(Mari,Mam2), fils(F2,Mari), frere(P2, F2).
```

En ayant précédemment défini frere/2 et soeur/2 tels que :

```
1 % Fr1 est le fr re de Fr2 (qui est h ou f)  
2 frere(Fr1,Fr2) :- parent(P,Fr1), parent(P,Fr2), homme(Fr1), Fr1 \= Fr2.  
3  
4 % So1 est la soeur de So2 (qui est h ou f)  
5 soeur(So1,So2) :- parent(P,So1), parent(P,So2), femme(So1), So1 \= So2.
```

Ainsi, de tête, on trouve que jacques, pierre et alex peuvent tous les trois correspondre pour loic en entrée. Pour alex, on ne trouve personne car jean n'a pas de frère.

On trouve bien le résultat escompté avec les requêtes rel(loic, P2). (jacques, pierre et alex apparaissent) et rel(alex, P2). (on obtient false.). Cependant, pour la seconde requête, on obtient possiblement false car on ne connaît pas de mère de jean.

La question était aussi simplifiable (ex : la soeur du fils de la mère de loic, c'est la soeur de loic).

2.7 Pedro est son propre grand-père

Enoncé : 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 !

On peut donc ajouter les prédicats suivants, que l'on déduit du texte :

```
1 homme(pedro).
2 homme(yannick).
3 epouse(brigitte,pedro).
4 epouse(elisabeth,yannick).
5 enfant(yves,brigitte).
6 enfant(elisabeth,brigitte).
7 enfant(luc,elisabeth).
8 enfant(yves,pedro).
9 enfant(pedro,yannick).
10 enfant(luc,yannick).
```

Ainsi que le prédicat `gd_pere/2` :

```
1 gd_pere(GdP,Enf) :- parent(P,Enf), parent(GdP,P), homme(GdP).
```

Ainsi, la requête `gd_pere(pedro,pedro)` renvoie bien `true`. comme attendu.

3 Livres

On définit les faits suivants :

```
1 livre(auteur(fredric,dard),titre('l"espion'),prix(45)).
2 livre(auteur(victor,hugo),titre('les mis rables'),prix(300)).
3 livre(auteur(eugene,delonay),titre('pendule de foucault'),prix(125)).
4 livre(auteur(victor,hugo),titre('fantasia chez les ploucs'),prix(200)).
5 livre(auteur(eugene,chang),titre('le silence'),prix(180)).
6 livre(auteur(umberto,ecco),titre('le nom de la rose'),prix(185)).
7 livre(auteur(umberto,ecco),titre('pendule de foucault'),prix(195)).
```

3.1 Nom d'un auteur dont le prénom est Victor

On utilise la requête suivante :

```
1 livre(auteur(victor,Nom),_,_).
```

Cette requête retourne comme attendu deux fois `Nom = hugo`, car il y a deux livres dont l'auteur a pour prénom victor, et pour les deux l'auteur est Victor Hugo.

Dans la requête, `victor` (en minuscules) nous permet de contraindre le premier terme d'auteur à être victor (le prénom recherché), `Nom` avec une majuscule est une variable et nous permet de connaître les noms correspondants, et les `"_"` sont des variables anonymes qui nous permettent de laisser passer n'importe quelle information en deuxième et troisième terme de livre.

3.2 Livre valant 40 euros

De même qu'avec la requête précédente, on va simplement imposer un prix de 40 euros, en cherchant le nom du livre. (Il n'y a aucun livre correspondant comme on aurait pu le savoir en regardant les faits, on obtient donc false.).

```
1 livre(_,Titre,prix(40)).
```

3.3 Titre d'un livre de Fredric Dard

```
1 livre(auteur(fredric,dard),Titre,_).
```

Ici, le prix ne nous importe pas, on impose l'auteur et on cherche le titre. On obtient bien Titre = titre('l'espion'). comme attendu. On remarquera qu'on n'obtient pas simplement la chaîne de caractères correspondant au titre, pour cela on aurait dû utiliser la requête suivante :

```
1 livre(auteur(fredric,dard),titre(Titre),_).
```

3.4 Prix du livre "le silence"

```
1 livre(_,titre('le silence'),prix(P)).
```

On obtient comme attendu $P = 180$.

3.5 Livres d'Umberto Ecco

```
1 livre(auteur(umberto,ecco),titre(Titre),_).
2
3 /* On obtient :
4 Titre = 'le nom de la rose' ;
5 Titre = 'pendule de foucault'. */
```

Le résultat est celui attendu.

3.6 Même titre, auteurs différents

Pour tester X différent de Y, on utilise $X \neq Y$. Ainsi, on cherche 2 livres avec la même variable pour le titre (même titre, forcément), deux variables différentes pour les auteurs, et la condition $A1 \neq A2$ permet d'éviter d'obtenir comme réponse le même livre en double.

```
1 livre(A1,Titre,_), livre(A2,Titre,_), A1 \= A2.
```

On trouve bien les deux livres appelés "Pendule de Foucault" d'Umberto Ecco et d'Eugene Delonay.

3.7 Livres avec prix supérieur à 200

```
1 livre(_,titre(Titre),prix(P)), P>=200.
```

Cette requête permet d'extraire le prix pour imposer une condition dessus, et le titre pour savoir quels livres correspondent. En pratique, sachant qu'il y a des livres avec le même titre mais de différents auteurs, il faudrait aussi afficher l'auteur.

Il y a deux livres correspondants : Les Misérables et Fantasia chez les ploucs.

3.8 Règle écrivain/prix : prédicat $\text{livre}_{EP}/2$

```
1 livre_E_P(E,P) :- livre(E,titre(T),P), print(T).
```

`print(T)` permet d'afficher le titre du livre trouvé.

En testant avec la requête `livre_E_P(auteur(victor, hugo),prix(300))`., on trouve bien Les Misérables.

3.9 Livres pouvant être achetés avec 400 euros

```
1 livre(A,T,prix(P)), P<=400.
```

La requête ci-dessus permet de connaître chaque livre qui peuvent être achetés avec 400 euros, conformément à la question posée. Ainsi, Prolog retourne les 7 livres énoncés dans les faits.

4 Conclusion

Ce travail mené sur la programmation logique a permis de bien prendre en main les premiers principes de Prolog. Le prochain travail portera notamment sur les listes.

Repo Github personnel