

## 2.4. Approche de l'effacement de buts

(VIDEO-PPT 1.3)

La base de connaissances `familleKennedy2.pl` étant chargée sous l'interpréteur Gprolog, on a vu qu'on pouvait l'interroger.

⇒ Interrogation avec des questions fermées (sans variable) :

➤ John est-il le père de John-john ?

```
|?- pere(john, john_john).  
true ? <RC>  
yes
```

Prolog a trouvé un premier succès, c'est-à-dire une solution au problème. Si on avait demandé à prouver quelque chose de faux, on a vu que la réponse est négative.

```
|?- pere(john, jacqueline).  
no
```

L'interpréteur n'a pas réussi à unifier la question avec un fait, ce qui veut dire « pour autant que je sache, d'après ma base de connaissances, je ne peux pas conclure que `john` est le père de `jacqueline` ».

☞ *En Prolog, tout ce qui ne peut être prouvé dans la base de connaissances est considéré comme faux, c'est l'hypothèse du monde clos <sup>2</sup>.*

⇒ Interrogation avec des questions à inconnues :

➤ De qui John est-il le père ?

```
|?- pere(john, X).  
X=john_john ? ;  
X=caroline  
yes
```

Dans la question il y a une inconnue `x` (la première lettre de l'identificateur est en majuscule), c'est une variable Prolog dont l'existence d'une valeur possible est questionnée. Pour résoudre le problème, Prolog explore le paquet de faits (clauses) `pere` de la base de connaissances et cherche à mettre en correspondance la question avec un fait en instanciant (attribuant une valeur à) `x`.

☞ *Les variables Prolog ont un sens logique, elles sont libres (non instanciées) ou liées (associées définitivement à une valeur pour la question ou la règle dans laquelle elles apparaissent). Dans un langage impératif, une variable correspond à un mot mémoire (état) dans lequel des valeurs différentes sont stockées à des instants différents*

---

<sup>2</sup> Il n'y a pas de négation. Cependant, en section V.2.2 il est montré qu'un prédicat `not` peut-être mis en œuvre en agissant sur le mécanisme d'évaluation de Prolog, (cf. négation par échec).

(affectations). Cette notion de valuation différente au cours du temps n'existe pas en logique, une variable est libre ou liée définitivement. On verra cependant qu'elle peut être liée à un terme composé comprenant d'autres variables, libres lors l'instanciation de la variable les englobant et qui seront liées ultérieurement.

⇒ Interrogation avec des questions composées :

➤ Quelles ont été les épouses de John ?

On conviendra que deux individus ont été époux, s'ils ont eu un enfant en commun. La question peut alors se traduire par : « si John est le père de  $x$ , existe-t-il  $y$ , tel que  $y$  soit la mère de  $x$  ? » .

```
|?- pere(john, X), mere(Y,X).  
X=john_john  
Y=jacqueline ? ;  
  
X=caroline  
Y=jacqueline
```

Dans cette question ouverte, comme dans la question fermée « grand-mère » vue précédemment, il y a 2 buts séparés par une « , » qui correspond à une **conjonction**.

Prolog évalue les buts de **gauche à droite**<sup>3</sup> :

1. recherche d'une solution (instanciation)  $x$  au but `pere(john, X)`
2. **puis**, avec la solution  $x$ , recherche des solutions  $y$  satisfaisant `mere(Y, X)`
3. **retour** sur `pere(john, X)` pour la recherche d'une autre solution  $x$
4. **pour chaque** solution  $x$ , on recherche des solutions  $y$  de `mere(Y, X)`

Dans une conjonction de buts,  $x$  représente le même objet pour tous les buts. Pour une solution  $x$  trouvée ( $x$  est liée / instanciée), Prolog cherche toutes les solutions  $y$  de `mere(Y, X)` puis Prolog revient sur le but `pere(john, X)` pour trouver éventuellement une autre solution  $x$ ... Le retour sur le premier but (backtracking) délie (rend libre)  $x$ .

☞ *Prolog gère l'indéterminisme, ie le fait qu'il y ait deux façons d'effacer le but `pere(john, X)` puis une pour `mere(Y,X)` par l'énumération de toutes les possibilités d'instanciation des couples  $(X,Y)$*

**miniTP** : En utilisant « `familleKennedy2.pl` », trouvez tous les frères de John, observez la succession des succès :

**1) Pourquoi les succès sont-ils doubles pour robert et edward et pas pour rose\_marie ?**

---

<sup>3</sup> C'est un choix d'implémentation pour l'interpréteur car, d'un point de vue logique, il n'y a pas d'ordre entre les buts.

- 2) Assurez-vous que les questions `frere(john,X)` et `frere(X,john)` fournissent les mêmes réponses.
- 3) Vérifiez la correspondance de l'énumération avec la succession des faits et règles dans le fichier
- 4) La variable `X` apparaissant à la question 2) existe aussi dans les règles définissant le prédicat `frere2`, pourrait-on la remplacer par une autre lettre dans un des deux contextes (question ou règles) ?