Rappels cours précédent *(VIDEO-PPT 2.1)*

- Base de connaissances = faits + règles = paquets de clauses
- Règle ≡ `tete :- queue.`
  - `tete` = 1 but (conclusion de la règle)
  - `queue` = conjonction de sous-buts (hypothèse de la règle)
- Hypothèse du monde fermé : si une information n'est pas déclarée dans la base de connaissances, elle est considérée comme fausse.
- Variable Prolog : une variable Prolog a un sens logique, c'est une inconnue qui est soit libre, soit liée, elle ne peut être instanciée (liée) qu'une seule fois.
- Portée d'une variable : toute la règle, rien que la règle

2.5. Catégories syntaxiques

Cette section ne présente que quelques éléments de syntaxe. Pour plus de précisions, cf. le manuel utilisateur de Gprolog.

i) Clauses

La base de connaissances définit des faits (sans variable) et des règles (avec variables a priori) :

- `pere(john, john_john).`
- `grandPere(X, Y) :- pere(X, Z) , pere(Z, Y).`

Sous l'interpréteur, les questions ont la forme :

- `mere(X, caroline).`

ii) Termes

Les objets manipulés par Prolog sont des termes, on distingue : les atomes, les variables, les nombres, les termes composés.

- **Atomes :**
  - **Constante symbolique** : suite de caractères, nombres, tirets bas, commençant par une ***lettre minuscule*** :

    ```
    daniel
    x007
    ```

  - **Chaine de caractères** :

    ```
    'toto'
    ```

- **Variables**

- Suite de caractères, nombres, tirets bas, commençant par une ***lettre majuscule***

```
Daniel
X007
X
```

- **Nombres**
  - Des **entiers** : `1`
  - Des **flottants** : `3.14`

- **Termes composés**

Un terme composé est une donnée représentant un arbre décrit sous une forme : `foncteur(arg1, …, argn)`. La racine (appelée foncteur) est une constante symbolique. Les arguments sont des termes.

➢ Fait : `possede(jean, voiture(electrique)).`

Ce fait caractérise (partiellement) une relation `possede` portant sur deux termes `jean` et `voiture(electrique)`. Attention, ce dernier terme **n'est pas un prédicat**[4] (qu'il faudrait évaluer) mais une **donnée structurée** représentant un arbre à deux arguments (qui n'est pas évalué).

➢ Questions :

```
|?- possede(jean, X).
X = voiture(electrique)

|?- possede(jean, voiture(X))).
X = electrique
```

Les arguments d'un terme composé peuvent à leur tour être des termes composés comme ici :

➢ Fait :

```
possede(jean, livre(lesRobots, auteur(isaac, asimov))).
```

➢ Question :

```
|?- possede(jean, livre(X, Y)).
X= lesRobots, Y = auteur(isaac, asimov)
```

[4] Prolog est fondé sur la logique du 1er ordre : un prédicat ne peut pas avoir un prédicat en paramètre

Si un terme composé n'est pas un prédicat, il peut cependant le devenir. Ainsi, il est possible de compléter la base de connaissances dans « `familleKennedy3.pl` » avec la règle :

```
complete_pere_par_mere :-   epoux(X,Y), mere(Y,Z),
                            non(pere(X,Z)),
                            assertz(pere(X,Z)).
```

qui utilise l'arbre `pere(X,Z)` avec les liaisons de variable `{X=joseph, Z=rose_marie)` pour ajouter un fait supplémentaire augmentant le prédicat `pere` avec l'usage du méta-prédicat `assertz`.

Remarques :

- `non` est un méta-prédicat défini (`not` absent de *Gprolog*), nous reviendrons ultérieurement sur sa mise en œuvre
- pour permettre l'ajout ou le retrait de faits en *Gprolog* avec, il faut que le prédicat ait été rendu « dynamique »

☞ *le préfixe « méta » est utilisé ici dans son acception informatique classique pour laquelle un « méta-X » est « un.e X qui traite des X ». par exemple, en programmation objet, une « méta-classe » est « une classe sur une classe ».*

- **Listes (en extension)**

Une liste s'écrit sur des termes. Elle peut s'énumérer en extension :

`[]`	liste vide
`[pierre]`	liste à 1 élément (constante)
`[X, pierre]`	liste à 2 éléments (une variable, une constante)
`[pierre, X, paul]`	liste à 3 éléments (constante , variable, constante)
`[X1,…,Xn]`	liste à *n* éléments

ou par emboîtement développé à droite d'arbres binaires (présenté plus loin en récursivité).

miniTP : En utilisant « arbresComposes.pl », cherchez si Jean (ou tout autre personne) :

1) Aime un « type » d'animal `X` ?
2) Aime les livres écrits par un auteur (désigné par son seul nom de famille) ?