

## TP1 : Prise en main du langage Prolog

### 1. Introduction

*Prolog* est un langage de programmation déclaratif permettant de résoudre des problèmes logiques. Le nom *Prolog* est l'acronyme de *PRO*grammation *LOG*ique. Le principe de la programmation logique est de décrire l'énoncé d'un problème par un ensemble d'expressions et de liens logiques au lieu de définir pas à pas la succession d'instructions que doit exécuter l'ordinateur pour résoudre la problématique.

*Prolog* est utilisé principalement en Intelligence Artificielle. En *Prolog*, il suffit de décrire ce que l'on sait sur le domaine étudié, (en Intelligence Artificielle, on appelle cela une base de connaissances), puis on pose une question à propos de ce domaine et *Prolog* va nous répondre, sans que l'on ait à lui dire comment construire sa réponse.

Un programme *Prolog* est composé d'une suite de faits et de relations entre ces faits exprimés par des règles selon la syntaxe suivante :

### 2. Installation de Prolog

Plusieurs implémentations de *Prolog* existent, aussi bien sous Windows que sous Linux. Parmi elles, il y a **SWI-Prolog** qui est un interpréteur *Prolog* gratuit et disponible, il s'installe aisément sur tout système d'exploitation et qui peut aussi s'utiliser en ligne et en partageant ses programmes ici : <https://swish.swi-prolog.org>

Téléchargez la dernière version stable selon la version de votre Windows sur le site [swi-prolog.org](https://swi-prolog.org) et installer SWI-Prolog sous Windows

### 3. Notions fondamentales

Contrairement aux langages de programmation impératifs (comme le Pascal, le langage C, le Java, Perl, Ruby, Python, etc.) on ne trouvera pas de boucles For, de test If, de "fonctions", etc. dans *Prolog*.

Un programme *Prolog* est construit à partir d'une base de connaissance qui décrit le problème à résoudre.

- La base de connaissances est composée d'un ensemble de clauses.
- Les clauses représentent des faits et/ou des règles.
- L'utilisateur pose ensuite une question et le programme donne sa réponse.
- Généralement, on place toutes les déclarations de faits au début du programme même si ce n'est pas obligatoire.
- Un fait se termine toujours par un point «.». Les faits sont des données élémentaires que l'on considère vraies. Ce sont des formules atomiques constituées d'un prédicat suivi ou pas d'une liste ordonnée d'arguments mis entre parenthèses qui sont les objets auxquels s'appliquent le prédicat.
- Un programme *Prolog* est au moins constitué d'un ou plusieurs faits car c'est à partir d'eux que *Prolog* va pouvoir rechercher des preuves pour répondre aux requêtes de l'utilisateur. Ce sont en quelque sorte les hypothèses de travail.

Quelques exemples de faits :

mange.

herbivore(chevre).

mange(loup, chevre).

Les règles énoncent la dépendance d'un prédicat par rapport à d'autres prédicats, elles concernent des catégories d'objets / faits :

Faits : manger(pomme).  
Afaim

Règle : Jean mange une pomme s'il a faim s'écrit : manger(pomme) :-Afaim.

Le «si» s'écrit « :-» en Prolog et correspond à l'implication  $\Rightarrow$

Il peut y avoir plusieurs conditions derrière le «:-», séparées par des virgules ou des points-virgules.

- La virgule correspond à un ET logique (conjonction).
- Le point-virgule correspond à un OU logique (disjonction)

Quelques exemples de règles :

mange(loup, chevre).

cruel(loup) :-mange(loup, chevre).

carnivore(loup) :-mange(loup, chevre), animal(chevre).

La relation telle que si on est le père du père ou de la mère de quelqu'un alors on est son grand-père se traduit par:

grandpere(xavier,yves) :-pere(xavier,joe), pere(joe,yves).

grandpere(xavier,yves) :-pere(xavier,joe), mere(joe,yves).

ou encore par : grandpere(xavier,yves) :-pere(xavier,joe), (pere(joe,yves) ; mere(joe,yves)).

#### 4. Premier programme en Prolog

En utilisant le bloc note de Windows, créer un nouveau fichier nommé *essai.pl* (pl représente l'extension du programme).

On y mettra un ensemble de faits et n'oubliez pas de sauvegarder votre fichier :

animal(chien).

animal(chat).

prenom(paul).

prenom(pierre).

prenom(jean).

Maintenant, double cliquez sur l'interpréteur Prolog (installé) et cliquez sur *File/Consult* pour ouvrir votre fichier *essai.pl* :

- Par défaut Prolog attend l'extension .pl pour les fichiers source en Prolog (ce qui peut parfois prêter à confusion avec les fichiers en Perl ...).
- L'extension .pl n'est pas indiquée dans l'instruction *consult*.
- En Prolog chaque ligne se termine par un point.

Nous n'allons pas maintenant "exécuter" le programme, mais nous allons poser à l'interpréteur Prolog un ensemble de questions pour lesquelles l'interpréteur consultera les faits et les règles inscrits dans le programme et répondra en conséquence.

PROLOG fonctionne comme une base de données : on charge le programme et ensuite on l'interroge dans la fenêtre de terminal où on a lancé prolog (commande *swipl* sur les serveurs de la FDS), celle où se trouve le prompteur « ?- »

?- prenom(jean). (*jean est un prénom ?*)

L'interpréteur répond par *true* : jean est bien un prénom dans la base des faits inscrits dans votre programme.

?- prenom(vincent). L'interpréteur nous répond *false* : vincent n'est pas un prénom dans la base des faits inscrits dans le programme.

?- animal(X). Quels sont tous les animaux ?

La variable X va prendre pour valeur chaque nom d'animal. Pour passer d'une valeur à une autre il faut appuyer sur la touche point-virgule. Une fois que la liste est terminée, Prolog la fini par un point. Le résultat est alors :

X = chien;  
X = chat.

Quels sont tous les prenom ? (?- prenom(X). À essayer, même que pour la précédente)

?- animal(\_). existe-t-il des animaux ?

Cette fois l'expression animal(\_) sera vraie chaque fois que Prolog rencontre un animal (peut importe lequel) dans la base des faits du programme. Comme Prolog rencontre 2 animaux, la réponse à notre question est :

true ;  
true.

?- listing. La commande listing permet d'afficher le code source de la base de faits du programme courant

?- halt. La commande halt permet de sortir de l'interpréteur Prolog.

Interruption brutale : « CTRL C » puis « a » pour abort.

### Quelques exercices

**1/ Définissez la famille ci-dessous dans l'éditeur. Attention à ne pas mettre de majuscules aux prénoms ni aux prédicats. N'oubliez pas les points.**

homme(albert).	femme(simone).	pere(jean,simone).
homme(jean).	femme(marie).	pere(louis,benoit).
homme(paul).	femme(sophie).	mere(germaine,jean).
homme(bertrand).		mere(christiane,simone).
homme(louis).	pere(albert,jean).	mere(christiane,paul).
homme(benoit).	pere(jean,paul).	mere(simone,benoit).
femme(germaine).	pere(paul,bertrand).	mere(marie,bertrand).
femme(christiane).	pere(paul,sophie).	mere(marie,sophie).

**2/ Traduire les questions suivantes en Prolog et vérifier les réponses :**

Est-ce que Paul est un homme ?

Est-ce que Benoit est une femme ?

Qui est une femme ?

Qui est un homme ?

Est-ce que Marie est la mère de Sophie ? de Benoit ?

Qui est la mère de Jean ?

Quels sont les enfants de Paul ?

Quels sont les hommes qui sont pères ?

**3/ Définir les prédicats suivants :**

- parent(X,Y) : X est un parent de Y, père ou mère ;
- fils(X,Y) : X est le fils de Y ;
- fille(X,Y) : X est la fille de Y ;
- grand\_pere(X,Y) : X est le grand-père de Y ;

- $\text{grand\_mere}(X,Y)$  : X est la grand-mère de Y ;
- $\text{frere}(X,Y)$  : X est le frère de Y ;
- $\text{soeur}(X,Y)$  : X est la soeur de Y.

**4/ Définir les prédicats suivants :**

Traduire en Prolog l'énoncé suivant :

Marie aime le vin

Pierre est un voleur

Pierre aime tous ceux qui aiment le vin

Si quelqu'un est un voleur et aime quelque chose alors il le vole

Qui vole quoi?