

TP N°1 IA : Prise en main du langage PROLOG

1 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 et installer SWI-Prolog.

2 Programmer en PROLOG

Programmer en Prolog signifie :

- Spécifier des faits (vérifiés) sur les objets et leurs relations ;
- Définir des règles sur les objets et leurs relations ;
- L'utilisateur pose ensuite une question sur les objets et leurs relations et le programme donne sa réponse.

3 Notions fondamentales

3.1 Les faits

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.

- D'abord, le nom de la relation ou prédicat (débutant par une minuscule),
- Le/Les arguments (exemple « jean ») séparés par une virgule.
- Un fait se termine toujours par un point «.».

Les espaces ne jouent pas de rôle et le commentaire peut aussi être placé entre /* ... */.

Les faits sont sous la forme : p ou p(...), avec p un prédicat :

```
pere(jean,paul). /* traduit "Jean est le pere de Paul"*/
```

Un programme Prolog est au moins constitué d'un ou plusieurs faits car c'est grâce aux faits que Prolog va pouvoir rechercher des preuves pour répondre aux requêtes de l'utilisateur. Ce sont en quelque sorte les hypothèses de travail.

3.2 Règles

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 qui dit : 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)).
```

3.3 Questions

La question est-elle prouvable à partir des faits et règles ?

Les questions peuvent contenir des variables ou non. Elles sont sous la forme : : s(...), ..., t(...). Exemple :

```
pere(jean,X), mere(annie,X).
```

Vous remarquerez que X est en majuscule car la majuscule signifie que c'est une variable.

Si une question contient une ou plusieurs variables Prolog cherche toutes les valeurs des variables qui rendent la question prouvable.

4 Mes premiers pas avec 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 ...).

Remarque : 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 prenom ?)*/
```

Qu'affiche le programme ?

```
?- prenom(vincent).
```

Qu'affiche le programme ?

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

La variable X va prendre pour valeur chaque nom d'animal.

Remarque : le point virgule ; sert à passer au résultat suivant de la question.

Quel est le résultat ?

```
?- prenom(X). /* Quels sont tous les prenom s ? */
```

```
?- 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. Quel est le résultat ?

Remarque : le point virgule ; sert à passer au résultat suivant de la question.

Autres commandes utiles •?- `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.

5 Exercices

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

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

Exercice 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 ?
- Qui sont les enfants de Paul ?
- Qui sont les hommes qui sont pères ?

Exercice 3 Définir les prédicats suivants :

- `parent(X,Y)` : X est un parent de Y, père ou mere ;
- `fils(X,Y)` : X est le fils de Y ;
- `filles(X,Y)` : X est la fille de Y ;
- `grand_pere(X,Y)` : X est le grand-père de Y ;
- `grand_mere(X,Y)` : X est la grand-mère de Y ;
- `frere(X,Y)` : X est le frère de Y ;
- `soeur(X,Y)` : X est la soeur de Y.