

T.P. Prolog n°1

1) Familiarisation avec l'environnement Prolog.

La rédaction d'un programme Prolog exige d'avoir deux fenêtres ouvertes en même temps :

- L'interpréteur SWI-Prolog dans lequel vous lancez les requêtes et chargez la base de connaissance.
- La fenêtre d'édition de SWI-prolog contenant la base de connaissance en cours de rédaction.

a) Commencez par copier le fichier **astronomie.pl** dans votre répertoire H:\Prolog que vous aurez créé auparavant (**.pl** est l'extension correspondant aux programmes Prolog).

b) Ouvrez-le, l'interpréteur SWI-Prolog devrait se lancer. Editez la base de connaissance avec File / Edit... N'oubliez pas de sauver le fichier (Ctrl+S) après chaque modification.

c) Chargez le programme Prolog en mémoire à l'aide du menu File/Consult... ou File/Reload... ou par Compile/Compile buffer dans la fenêtre d'édition.

Prolog charge alors le programme dans sa mémoire. S'il y a des erreurs de syntaxe, Prolog vous les indique avec le numéro de la ligne erronée. Retournez dans la fenêtre du programme pour corriger les erreurs. Lorsque vous retournerez dans la fenêtre Prolog, vous pourrez retrouver les commandes précédentes à l'aide des touches fléchées.

Lorsque le programme est syntaxiquement correct, vous pouvez taper par exemple les requêtes suivantes :

```
astre(mars).  
astre(X).  
satellite(X,jupiter).  
satellite(P,soleil).  
satellite(X,Y).
```

Pour avoir toutes les réponses possibles, taper un ; ou un **espace** après chaque réponse obtenue.

d) Pour mieux comprendre ce que fait Prolog (et par la suite, déboguer votre programme), vous pouvez faire précéder une requête du prédicat **trace**. Par exemple :

```
trace, satellite(X,uranus).
```

Prolog fait alors du pas à pas. Vous passez au pas suivant en tapant Entrée. Quatre types d'informations sont affichés :

Call : c'est l'appel du prédicat

Exit : Retour avec succès

Fail : Retour avec échec

Redo : Retour en arrière, Prolog essaie une autre solution possible.

Si vous ne souhaitez pas faire du pas à pas sur un appel particulier de prédicat (parce que vous savez qu'il fonctionne bien), tapez la touche 's' (skip) au lieu de Entrée.

e) ajouter dans le fichier **astronomie.pl** le prédicat **planete/1** qui définit une planète comme étant un astre qui est le satellite d'une étoile. Pour que Prolog prenne en compte le prédicat **planete**, sélectionnez le dans l'éditeur avec la souris, puis dans le menu faites **Compile/Consult selection**. Testez la règle en posant des questions concernant ce nouveau prédicat.

f) ajouter dans le fichier **astronomie.pl** le prédicat **satellite_planete/1** qui définit un satellite de planète. Testez la règle en utilisant la trace et observez les recherches de solutions effectuées par Prolog :

```
trace, satellite_planete(S).
```

Pour quitter le mode trace, tapez le prédicat : **nodebug**.

Swi-Prolog fournit également une trace graphique, entrez la requête :

```
gtrace, satellite_planete(S).
```

2) Allons au restaurant

Récupérez le fichier **menu.pl** ; ce fichier contient des faits qui représentent la carte d'un restaurant.

- a) Par une simple requête, demandez la liste des hors d'œuvre disponibles.
- b) Définissez le prédicat **plat/1** (règle) qui dit qu'un plat est à base de viande ou de poisson.
- c) Définissez le prédicat **repas/3** qui dit qu'un repas est constitué d'un hors d'œuvre, d'un plat et d'un dessert.
- d) Prenez en compte dans Prolog les règles **plat** et **repas**, puis demandez la liste des repas possibles.
- e) Comment demander la liste des repas comprenant du poisson ?
- f) Considérons maintenant les valeurs caloriques des différents aliments. Ajoutez les faits suivants à votre programme :

```
calories(artichauts, 102).  
calories(crevettes, 230).  
calories(oeufs, 220).  
calories(grillade_de_boeuf, 612).  
calories(poulet, 242).  
calories(loup, 256).  
calories(sole, 146).  
calories(glace, 195).  
calories(tarte, 278).  
calories(fraises, 99).
```

- g) Demandez l'affichage de la valeur calorique des hors d'œuvre.
- h) Définissez le prédicat **valeur_calorique/4** qui calcule la valeur calorique d'un repas.
- i) Définissez le prédicat **repas_equilibré/3** (valeur calorique inférieure à 700) et demandez la liste des repas équilibrés à base de viande.
- j) Demandez quels sont les repas comportant des crevettes.
- k) Demandez quels sont les repas ne comportant pas de fraises.
- l) Complétez le programme de façon qu'un repas comporte une boisson à choisir parmi le vin, l'eau minérale et la bière.

3) La famille...

Récupérez le fichier **famille.pl** ; ce fichier contient des prédicats **parent/2**, **homme/1** et **femme/1**. Ecrire et les prédicats **pere/2**, **mere/2**, **grandPere/2**, **grandMere/2**, **frere/2**, **soeur/2**, **oncle/2** et **tante/2**, **cousin/2**, **cousine/2**, **demi_frere/2**, **demie_soeur/2**, **arriere_grand_parent/2**.

4) Faire le programme et tester l'exemple suivant vu en TD

1 - Un club d'aviron décide d'utiliser Prolog pour composer les équipes lors de compétitions d'aviron.

- a) Représenter par des faits l'âge des athlètes (repérés par leur nom) et leur place préférée (rameur ou barreur).
- b) Ecrire ensuite le prédicat **equipe/4** qui donne les quatre noms d'une équipe possible : 3 rameurs et un barreur.
- c) Ecrire le prédicat **equipe2/5** qui comporte en plus **l'âge moyen** de l'équipe.
- d) Ecrire le prédicat **equipe3/5** qui comporte en plus **l'âge moyen maximal** de l'équipe.

```
?- equipe3(B,R1,R2,R3,25).  
B=dupont, R1=martin, R2=durand, R3=lefevre
```