



Fiche TP 1

Initiation à prolog

Objectif

Le but de ce TP est de vous familiariser avec le langage prolog.

Préambule

Prolog est un langage de programmation logique. Ceci veut dire qu'un programme se présente pas sous forme d'une suite d'instructions mais sous forme d'un certain nombre d'**affirmations sur le monde** (des faits et des règles), formant une base de connaissances.

L'utilisation du programme se fera alors en interrogeant cette base de connaissances. Prolog dispose d'un puissant **moteur** lui permettant de déduire certains faits à partir de ceux que contient sa base de connaissance.

Nous allons utiliser SWI-Prolog, une distribution libre et gratuite du langage de programmation Prolog.



SWI Prolog

Vous pouvez télécharger cette distribution sur :

<http://www.swi-prolog.org/Download.html> pour Windows.

<http://www.swi-prolog.org/build/Debian.html> Pour Linux.

L'installation de SWI-Prolog est simple, il suffit de suivre les étapes.

Une fois l'installation achevée, normalement un nouveau dossier sera créé dans le quel vos programmes vont être enregistrés.



Premiers pas en programmation prolog

- Pour lancer SWI-Prolog, cliquer sur le raccourci disponible sur le bureau ou dans le menu Démarrer/ SWI-Prolog.
- Une fois que vous avez démarré SWI-Prolog, vous aurez la fenêtre représentée dans la figure 1. Dans cette fenêtre on peut lancer des requêtes.
- Pour créer un nouveau programme, il suffit de cliquer sur File ensuite sur New et pour modifier un programme cliquer sur File ensuite sur Edit.
- Avant l'enregistrement de vos programmes assurez-vous que dans la casetype, il y a Prolog Source.
- Après ces deux étapes précédentes, vous aurez la fenêtre représentée dans la figure 3.

Remarque : un programme contient la base des connaissances (fait et règles)

- Pour exécuter un programme, il faut aller dans le menu compile et choisir compile buffer ou tout simplement faire **ctrl+c** ou **ctrl+b**.

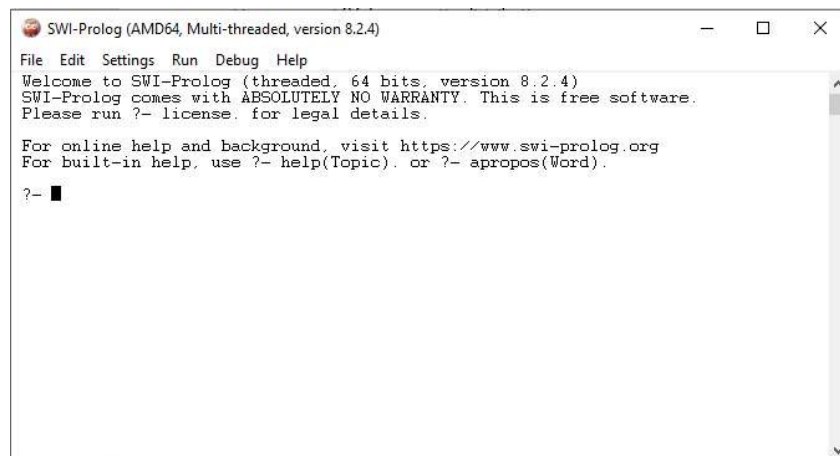


Figure 1

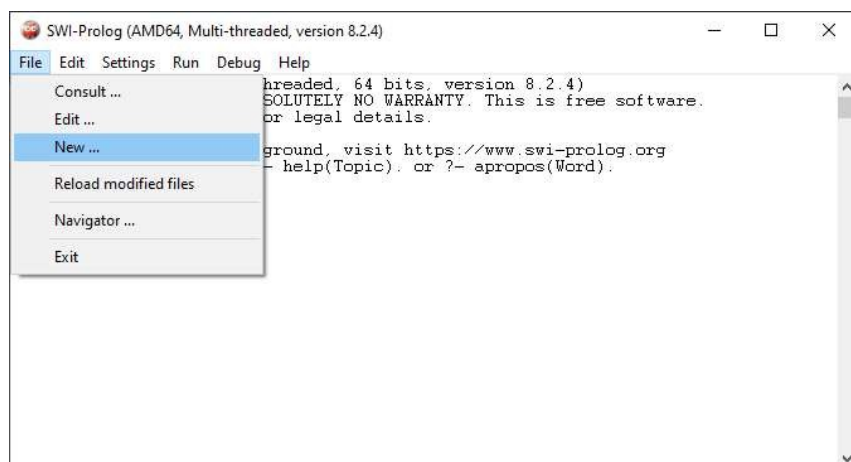


Figure 2

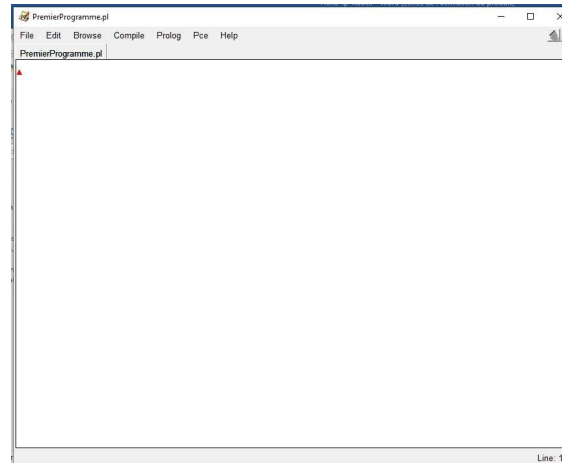


Figure 3

Syntaxe de Prolog

Un programme Prolog est constitué d'un ensemble de clauses ; une clause est une affirmation portant sur des atomes logiques ; un atome logique exprime une relation entre des termes ; les termes sont les objets de l'univers.

A. **Termes** : toute variable ou constante est un terme.

1. **Constantes** : elles commencent par une lettre minuscule et peuvent être une chaîne de caractères ou un nombre (entier ou flottant) (ex. nabil version_2_3 '21 ans' -123 12.56).
2. **Variables** : elles commencent par une lettre majuscule (ex. Var, X, Var_longue_2) ou par « _ » (ex. _objet, _21). Une variable qui commence par « _ » est une variable anonyme (elle représente un objet dont on ne souhaite pas connaître la valeur).

Remarque : On peut avoir un terme composé (appelé aussi foncteur). Un terme composé avec la forme : foncteur(t_1, \dots, t_n). Il commence par une minuscule, et t_1, \dots, t_n sont des termes (variables, constantes ou termes composés). Le nombre d'arguments n est appelé *arité du terme*.

Par exemple, adresse(18,"rue Meghili Mounir",Ville) est un terme composé de foncteur adresse et d'arité 3, dont les deux premiers arguments sont des constantes et le troisième argument est la variable Ville.



B. Prédicats (atomes) : expriment une relation entre les données. Exemple :
mère(farah, amine) fils(amine, farah)

C. Fait : est un prédicat qui termine par un « . ». Il exprime une connaissance acquise (connaissance qui est toujours vraie).

Exemple 1 : étudiant(amine).

Exemple 2 : habite(amine, tlemcen). inscrit(yassine, 'fac des sciences').

- Un fait peut être défini avec des variables. Exemple : égale(X, X). plus_petit(0, X).

D. Règle : c'est une clause de Horn qui exprime la façon dont de nouvelles connaissances peuvent être déduites.

Exemple 1 : personne(X) :- homme(X).

personne(X) :- femme(X).

Exemple 2 : personne(X) :- (homme(X) ; femme(X)).

Ces deux règles ont la même signification qui se lisent "pour tout X, personne(X) est vrai si homme(X) est vrai ou femme(X) est vrai"

Exemple 3 : mortel(X) :- humain(X).

Cette règle se lit : pour tout X, mortel(X) est vrai si humain(X). Exemple 4 :

frangin(X, Y) :- père(Z, X), père(Z, Y).

Cette règle se lit : pour tout X, pour tout Y, frangin(X, Y) est vrai s'il existe un Z tels que Z est le père de X et Z est le père de Y.

E. Requête (question ou but) :

Elle est composée par un ou plusieurs prédicats et qui se termine par un « . ».

?- prédicat_1(...), prédicat_2(...), ...

1. Requête booléenne : qui ne contient pas de variables. Exemples :

?- fils(yassine, med).

?- fils(yassine, med), frère(yassine, sara).

2. Requête existentielle : elle est définie avec une ou plusieurs variables.

Exemple : Si on a dans la base de connaissances



fil(nabil,otman).

soeurs(fatima, nabil).

Requêtes :

?- fil(X, otman).

?- soeurs(X, nabil).

Requêtes utiles

?- listing.

permet d'afficher le contenu de la base de connaissance courante. Ce prédicat peut également prendre en argument le prédicat dont on souhaite connaître la définition courante. Par exemple : listing(femme).

Exercice 1 :

Soit les données suivantes :

mostafa est un homme

ilyes est un homme

samir est un homme

sara est une femme

lylia est une enfant

samir aime tom&jerry

mostafa et ilyes aiment sara

ilyes aime lylia

Q1- traduisez les énoncés ci-dessus en faits avec prolog et enregistrez sous le nom ex1.pl

Q2- lancez SWI-prolog et chargez votre programme à travers le menuFile→Consult

Q3- Testez votre programme. Exemple : femme(sara).

Q4- utilisez la requête : ? -**listing(homme)**. Que fait cette requête ?

Q5- Lancez la requête permettant de retourner toutes les personnes aimées par ilyes.

Q6- En utilisant le variable inconnue (), lancez la requête permettant de tester uniquement si mostafa aime quelqu'un sans retourner ce dernier.

Q7- Ajoutez la règle suivante à la base de connaissances :

X est adorable s'il existe quelqu'un qui aime X, et X est une femme, homme, ou enfant.

Remarque : après chaque modification, rechargez votre programme avec File→Reload modified files



Q8- Lancez une requête qui permet de retourner toutes les personnes adorables sans répétition.

Q9- Ajoutez la règle suivante : X est jaloux de Y si X et Y aime la même femme. Utilisez le symbole (\neq) (différent).

Q10- utilisez : ? - **listing(jaloux)**. Que fait listing dans cette requête?

Q11- affichez les hommes jaloux.

Exercice 2 :

1. Arbre généalogique

A. Déclarations

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).

homme(jean).

homme(paul).

homme(bertrand).

homme(louis).

homme(benoit).

femme(germaine).

femme(christiane).

femme(simone).

femme(marie).

femme(sophie).

pere(albert,jean).

pere(jean,paul).

pere(paul,bertran).

pere(paul,sophie).

pere(jean,simone).

pere(louis,benoit).

mere(germaine,jean).

mere(christiane,simone).

mere(christiane,paul).

mere(simone,benoit).

mere(marie,bertrand).

mere(marie,sophie).



B. Questions

Traduire les questions suivantes en Prolog et vérifier les réponses :

- Q1. Est-ce que Paul est un homme ?
- Q2. Est-ce que Benoit est une femme ?
- Q3. Qui est une femme ?
- Q4. Qui est un homme ?
- Q5. Est-ce que Marie est la mère de Sophie ?
- Q6. Est-ce que Marie est la mère de Benoit ?
- Q7. Qui est la mère de Jean ?
- Q8. Quels sont les enfants de Paul ?
- Q9. Quels sont les hommes qui sont pères ?

C. Définition de prédicats

Définir les prédicats suivants :

- $\text{parent}(X,Y)$: X est un parent de Y, père ou mère ;
- $\text{fils}(X,Y)$: X est le fils de Y ;
- $\text{fille}(X,Y)$: X est la fille de Y ;
- $\text{grand_pere}(X,Y)$: 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.

Exercice 3 :

Adam aime les pommes.

Inès aime les carottes.

Karim aime les oranges.

Les pommes sont des fruits.

Les oranges sont des fruits.

Les carottes sont des légumes.

Ceux qui aiment les fruits sont en bonne santé.

Q1. Formalisez ces faits et règles en PROLOG.

Q2. Quelle est la requête pour savoir qui est en bonne santé ?

Q3. Quelle est la requête pour “Qui aime les pommes?” ?

Q4. Comment savoir les fruits que connaît le programme ?



Exercice 4 :

1- Traduire en Prolog l'énoncé suivant :

- Les étudiants qui ont la grippe doivent prendre un congé de maladie.
- Les étudiants qui ont de la fièvre et qui toussent ont la grippe.
- Ceux qui ont une température supérieure à 38° ont de la fièvre.
- Mohammed tousse et a une température supérieure à 38° .

2- La problématique à résoudre :

- Est-ce que Mohamed doit prendre un congé de maladie ?