

Solution TP N°1 : Initiation à Prolog

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

- 1. Présentation de langage :** le nom *Prolog* est l'acronyme de **Program**mation en **log**ique, introduit en 1972 par Alain Colmerauer.
 - Prolog permet d'écrire des programmes sous une forme très proche de la logique des prédicats du premier ordre (plus exactement les clauses de Horn).
 - C'est un langage déclaratif (contrairement aux langages impératifs comme Java, C++, Python, etc.) : écrire un programme revient à décrire un ensemble de connaissances (BC) décomposées sous la forme d'un ensemble de faits regroupé dans une base de faits (BF) et d'un ensemble de règles applicables aux faits, regroupé dans une base de règles (BR).
 - C'est un langage qui raisonne : faire tourner un programme Prolog revient à lui poser des questions auxquelles il va répondre en utilisant les connaissances contenues dans le programme. Pour faire cela, Prolog utilise son « cerveau », un moteur d'inférences en chaînage arrière qui permet d'appliquer les règles aux faits.
 - C'est un langage indéterministe : il permet le traitement de problèmes comportant plusieurs solutions.
 - C'est un langage clair et lisible pour tout utilisateur, et l'écriture d'un programme en est aisée.
- 2. Champs d'application de la programmation logique :** divers domaines d'application sont concernés par la programmation logique tels que :
 - La conception de systèmes experts afin de simuler l'expertise humaine.
 - La conception des SGBDR «Systèmes de Gestion de Bases de Données Relationnelles».
 - Le traitement du langage naturel.
 - L'enseignement assisté par ordinateur.
 - ...
- 3. Installation de Prolog :** plusieurs implémentations de Prolog existent (SWI-Prolog, GNU Prolog, Sicstus Prolog(P),...), aussi bien sous Windows que sous Linux.
 - L'interpréteur Prolog utilisé dans ce module est **SWI-Prolog** qui est gratuit.
 - Le lien de téléchargement de SWI Prolog sous Windows : <http://www.swi-prolog.org/>
 - L'interpréteur en ligne de **SWI-Prolog** : <https://swish.swi-prolog.org>
 - 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 enregistrer.
- 4. Premiers pas en programmation prolog :**
 - Pour lancer SWI-Prolog, cliquez sur le raccourci disponible sur le bureau ou dans le menu Démarrer/ SWI-Prolog.
 - Une fois que vous avez démarré SWI-Prolog, une fenêtre (interpréteur PROLOG) s'ouvre et dont laquelle on peut lancer des requêtes.

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de Relizane

Département Informatique
 Intelligence Artificielle : L3 - SI

Année académique 2021/2022
 Responsable du module : **S. Bella**

- Pour créer un nouveau programme ou fichier, il suffit de cliquer sur *File* ensuite sur *New*.
- Avant l'enregistrement de vos programmes assurez-vous que dans la case type, il y a *Prolog Source* (par exemple *tpl.pl*).
- Après ces deux étapes précédentes, une fenêtre (éditeur de texte) s'ouvre.
- Pour exécuter un programme, allez dans le menu *compile* et choisissez *compile buffer* ou tout simplement tapez *ctrl c* puis *ctrl b*.
- Pour modifier (afficher) un programme, cliquez sur *File* de l'interpréteur ensuite sur *Edit*.
- À chaque modification du programme, n'oubliez pas d'enregistrer et de recharger le fichier dans Prolog.
- Le *prompt* "?-" indique que l'interpréteur attend une requête de votre part. Toute requête doit se terminer par un point "." et un retour charriot. Lorsqu'une requête occupe plusieurs lignes, le prompt devient "|".

5. **Éléments fondamentaux du Prolog** : un programme PROLOG ne contient pas les types de données habituellement utilisés dans d'autres langages, du fait qu'il comporte une base de faits et un ensemble de règles. Dès lors, un programme PROLOG est constitué d'un ensemble de clauses (faits, règles) ; 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.

Élément	Définition	Exemple
Faits	C'est une affirmation inconditionnelle qui décrit des relations entre les données. Soit p est un prédicat alors le fait est de la forme "p(...)."	élève(jamel, 1975, info, 2). élève(sarah, 1974, info, 2). élève(ahmed, 1976, _, 1). masculin(jamel). masculin(ahmed). féminin(sarah). père(farouk, jamel). mère(amel, jamel).
Prédicat	C'est un symbole qui traduit une relation (atome logique). L'arité est le nombre de ses arguments. On identifie un prédicat par son nom et son arité.	élève(jamel, 1975, info, 2). → est une relation d'arité 4 entre les termes jamel, 1975, info et 2 pouvant être interprétée par "Jamel est un élève, né en 1975, étudié l'info en classe 2."
Questions ou requêtes	Une fois le programme chargé, on peut poser des questions sur les faits. Ce sont une suite de relations (atome logique) séparés par des virgules.	?- masculin(jamel). ?- masculin(farid). ?- masculin(X). ?- élève(X, Y, Z, 2)
Termes	Ce sont les objets manipulés par un programme Prolog (les "données" du programme). Les termes atomiques, les variables, les prédicats sont des termes.	
Constantes ou termes atomiques	Nombres (entiers, flottants) : une suite des chiffres, préfixé "-" optionnel. Atomes : une chaîne de caractères des minuscules, de majuscules, des chiffres et des blancs soulignés « _ » qui commence par une minuscule.	23, -492, 0, 23.5, 1.32E-21 ali, num_ali, ali123 'ali_med'

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de Relizane

Département Informatique
Intelligence Artificielle : L3 - SI

Année académique 2021/2022
Responsable du module : **S. Bella**

	Aussi permis : atome entre guillemets « ' » et atome contient les caractères spéciaux (#, \$, &, *,...) MAIS il est recommandé d'utiliser plutôt des atomes simples.	
Variables	Séquences (chaîne de caractères) de majuscules, de minuscules, de chiffres et de blancs soulignés « _ » qui commencent par une majuscule ou par un blanc souligné. Une variable qui commence par « _ » est une variable anonyme (elle représente un objet dont on ne souhaite pas connaître la valeur).	Rel_48, _Relizane, Rel48
Termes composés	Ce sont les objets composés se composent d'un foncteur avec une suite d'arguments. Les arguments peuvent être des atomes, des nombres et des variables. Dans l'exemple, élève est le foncteur principal.	adresse(6, 'Cité Jamel Lafontaine', 'Oran'). élève(amine, 1975, info, 2, adresse(6, 'Cité Jamel Lafontaine', 'Oran')).
Règles	C'est une affirmation conditionnelle qui exprime la façon dont de nouvelles connaissances peuvent être déduites. Soit p, q, r sont des prédicats, alors la règle est du forme "p(...):- q(...),...,r(...)." Si la tête p(...) est vraie si chacun des éléments du corps q(...),...,r(...) est vrai. On appelle ce type de règles des clauses de Horn.	<p> fils(A, B):- père(B, A), masculin(A). fils(A, B):- mère(B, A), masculin(A). parent(X, Y):- père(X, Y). parent(X, Y):- mère(X, Y). grand_parent(X, Y):- parent(X, Z), parent(Z, Y). </p>
Programmes	Programme Prolog est un ensemble de clauses regroupé en paquets.	<p> personne(X) :- homme(X). personne(X) :- femme(X). → ``pour tout X, personne(X) est vrai si homme(X) est vrai ou femme(X) est vrai". </p>

6. Mise en œuvre d'un programme :

Charger de fichiers : les programmes sont dans des fichiers avec le suffixe « .pl » en général.

- Pour charger un programme : `?- consult(nom_du_fichier).` Où `nom_du_fichier` est un atome. Par exemple : `?- consult(tp1).` ou `? consult('tp1.pl').` Ou bien le raccourci avec la commande : `?- [nom_du_fichier].` Ou `?- [nom_du_fichier].` Par exemple : `?- [tp1].` ou `?- ['tp1.pl'].`

- Pour charger de plusieurs fichiers simultanément : `?- [file1, file2].`

Afficher du contenu d'un fichier chargé : `?- listing.`

Afficher une clause particulière, ici père : `?- listing(père).`

Ajouter des commentaires dans un fichier : `% commentaire` ou `/*commentaires*/`

Quitter le Prolog : `?- halt.`

Retourner à l'interpréteur : [CTRL] D.

7. Bonne habitude :

- Écrivez du commentaire avec votre programme (environ une ligne commentaire pour une ligne de commande).
- Choisissez de noms clairs pour les prédicats et des variables, écrivez clairement le sens du prédicat.

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de Relizane

Département Informatique
Intelligence Artificielle : L3 - SI

Année académique 2021/2022
Responsable du module : **S. Bella**

Exercice :

1. Soit les données suivantes :
 - la chèvre est un animal herbivore
 - le loup est un animal cruel
 - un animal cruel est carnivore
 - un animal carnivore mange de la viande
 - un animal herbivore mange de l'herbe
 - un animal carnivore mange des animaux herbivores
 - les carnivores et les herbivores boivent de l'eau
 - un animal consomme ce qu'il boit ou ce qu'il mange
2. Traduire en Prolog les énoncés ci-dessus en faits et en règles puis enregistrer le fichier sous le noms 'exo1'.
3. Recharger votre fichier.
4. Traduire les questions suivantes en Prolog et vérifier les réponses :
 - Est-ce que chèvre est un animal ?
 - Est-ce que chèvre est un animal cruel?
 - Qui sont les animaux ?
 - Qui est l'animal herbivore ?
 - Y a-t-il un animal herbivore qui mange de l'herbe ?
 - Y a-t-il un animal qui boit de l'eau?
 - Y a-t-il un animal qui boit de l'eau et mange?
5. Afficher le contenu de la clause 'consomme'.

Solution :

2. Traduction des énoncés :

/*Faits*/

animal(chèvre).

animal(loup).

herbivore(chèvre).

cruel(loup).

/*Regles*/

carnivore(X) :- cruel(X).

mange(X,viande):- animal(X), carnivore(X).

mange(X,herbe):- animal(X), herbivore(X).

mange(X,Y):- animal(X), animal(Y), carnivore(X),herbivore(Y).

boit(X,eau):- animal(X), (carnivore(X); herbivore(X)).

consomme(X,Y):- mange(X,Y); boit(X,Y).

3. Rechargement : **Compile puis Compile buffer (ou ctrl c puis ctrl b).**

4. Traduction des questions :

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de Relizane

Département Informatique
Intelligence Artificielle : L3 - SI

Année académique 2021/2022
Responsable du module : **S. Bella**

?- animal(chevre).

true.

?- cruel(chevre).

false.

?- animal(X).

X = chevre ;

X = loup.

?- herbivore(X).

X = chevre.

?- mange(X,herbe).

X = chevre ;

false.

?- boit(X,eau).

X = chevre ;

X = loup ;

false.

?- consomme(X,Y).

X = loup,

Y = viande ;

X = chevre,

Y = herbe ;

X = loup,

Y = chevre ;

X = chevre,

Y = eau ;

X = loup,

Y = eau ;

false

6. Contenu d'une clause :

?- listing(consomme).

consomme(A, B) :-

(mange(A, B)

; boit(A, B)

).

true.