

**CENTRE UNIVERSITAIRE DE MILA  
INSTITUT DES SCIENCES ET DE LA TECHNOLOGIE**

**Matière: La programmation logique**

*INTRODUCTION*  
*A*  
*PROLOG*

# Présentation du cours

## I- Définition et historique

## II- Syntaxe et terminologie de Prolog

### II-1 Les termes

#### II-1.1 Les variables

#### II-1.2 Les termes élémentaires

#### II-1.3 Les termes composés

### II-2 Les relations, ou atomes logiques

### II-3 Les clauses

#### II-3.1 Les faits

#### II-3.2 les règles

### II-4 Les buts

## III Les programmes Prolog

### III-1 Définition d'un programme Prolog

### III-2 Exécution de programmes Prolog

### III-3 Signification (sémantique) d'un programme Prolog

## I- Définition et historique

- ✓ **Prolog** est l'un des principaux langages de programmation logique, son nom est un acronyme de **PRO**grammation **LOG**ique. Il a été créé par Alain Colmerauer et Philippe Roussel vers 1972 à Marseille.
- ✓ Prolog est utilisé dans de nombreux programmes d'intelligence artificielle et dans le traitement de la linguistique par ordinateur, il est largement répandu dans les universités, les laboratoires de recherche et dans l'industrie.
- ✓ Sa syntaxe et sa sémantique sont considérées comme très simples et claires.

- ✓ Prolog est basé sur le calcul des prédicats du premier ordre. Les concepts fondamentaux sont l'unification, la récursivité et le retour sur trace (en arrière).
- ✓ Un programme Prolog est constitué d'un ensemble de clauses ; une clause est constituée d'un ou plusieurs atomes logiques ; un atome logique exprime une relation entre des termes ; les termes sont les objets de l'univers.
- ✓ À partir de Prolog, furent développés Gödel, Oz, Visual Prolog, Datalog, et plus récemment CS Prolog ou Actor Prolog pour les ordinateurs de la 5<sup>e</sup> génération.

## II- Syntaxe et terminologie de Prolog

### II-1 Les termes

Les objets manipulés par un programme Prolog (les "données" du programme) sont appelés des termes. On distingue trois sortes de termes:

#### II-1.1 Les variables

- ✓ Les variables représentent des objets inconnus de l'univers. Syntaxiquement, une variable est une chaîne alphanumérique commençant par une majuscule (X3, Prix\_Unitaire, Var\_2 ...).
- ✓ La variable anonyme est notée "\_" et toute variable dont le nom commence par un tiret bas est également une variable anonyme.

- ✓ L'usage des variables en PROLOG permet de transformer des questions fermées en questions ouvertes, c'est-à-dire en questions susceptibles d'avoir plusieurs réponses possibles. Exemples :

**?- patron(X, mohamed).**

**?- patron (ali, X).**

Pour répondre à ces questions, PROLOG doit remplacer la variable X par une des constantes apparaissant précédemment dans le programme, afin que le fait résultant existe dans la base de faits.

- ✓ Les variables peuvent aussi être utilisées pour décrire des faits. Par exemple, **patron(ali, X).** est une transcription de la phrase : «ali est le grand patron».
- ✓ Utilisée dans une conjonction, une variable établit une relation entre les prédicats qui la composent. Exemple: - **patron (ali, X), menuisier(X )**.

De quel(s) menuisier(s) ali est-il le patron ?

- ✓ Les variables anonymes se comportent comme des variables ordinaires dans la recherche des correspondances, mais elles ne prennent jamais de valeurs et n'apparaissent pas dans les réponses.

Exemples :           ?- patron (ali, \_).  
                          yes

La base de fait contient au moins un employé du patron ali.

                          ?- patron( A, \_).  
                          A = ali ;  
                          A = yasser ;  
                          No

Les différentes occurrences du trait de soulignement \_ désignent des variables anonymes différentes.

**II-1.2 Les termes élémentaires:** (ou termes atomiques) représentent les objets simples connus de l'univers. On distingue trois sortes de termes élémentaires:

- ✓ **Les nombres**: entiers ou flottants,
- ✓ **Les identificateurs** (parfois appelés atomes): un identificateur est une chaîne alphanumérique commençant par une minuscule (ex. ali, aX12, ahmed),
- ✓ **Les chaînes de caractères** entre guillemets (ex. "%Mila%", "123").

**II-1.3 Les termes composés**: représentent les objets composés (structurés) de l'univers. Syntaxiquement, un terme composé est de la forme:

**foncteur**( $t_1, \dots, t_n$ )

ou **foncteur** est une chaîne alphanumérique commençant par une minuscule (tête), et  $t_1, \dots, t_n$  sont des termes (paramètres). Le nombre d'arguments  $n$  est appelé **arité**.

Par exemple, **adresse(18,"rue des martyres",Ville)** est un terme composé du foncteur **adresse** d'arité 3, dont les deux premiers arguments sont les termes élémentaires **18** et **"rue des martyres"** et le troisième argument est la variable **Ville**.

## II-2 Les relations, ou atomes logiques

- Un atome logique exprime une relation entre des termes ; cette relation peut être vraie ou fausse. Syntaxiquement, un atome logique est de la forme:

**symbole-de-prédicat( $t_1, \dots, t_n$ )**

ou **symbole-de-prédicat** est une chaîne alphanumérique commençant par une minuscule, et  $t_1, \dots, t_n$  sont des termes. Le nombre d'arguments  $n$  est appelé *arité* de l'atome logique. Par exemple: **pere(al,ahmed)** est une relation d'arité 2 entre les termes élémentaires ali et ahmed pouvant être interprétée par "ali est le père de ahmed".

- De même, **habite(X,adresse(12,"rue r",alger))** est une relation d'arité 2 entre la variable X et le terme composé **adresse(12,"rue r",alger)** pouvant être interprétée par ``une personne inconnue X habite à l'adresse **12,rue r,Alger**".

## II-3 Les clauses

Une clause est une affirmation inconditionnelle (un fait) ou conditionnelle (une règle).

### II-3.1 Les faits :

- Un fait est de la forme: **A.**

A est un atome logique (un prédicat), et signifie que la relation définie par A est vraie. Par exemple, le fait **pere(ali,ahmed).** indique que la relation "ali est le père de ahmed " est vraie.

- Une variable dans un fait est quantifiée universellement. Par exemple, le fait **egal(X,X).** indique que la relation "X est égal à X " est vraie pour toute valeur de x.

## II-3.2 les règles :

➤ Une *règle* est de la forme:  $A_0 :- A_1, \dots, A_n$ .

$A_0, A_1, \dots, A_n$  sont des atomes logiques (des prédicats). Une telle règle signifie que la relation  $A_0$  est vraie si les relations  $A_1$  et ... et  $A_n$  sont vraies.  $A_0$  est appelé **tête de clause** et  $A_1, \dots, A_n$  est appelé **corps de clause**.

➤ Une variable apparaissant dans la tête d'une règle (et éventuellement dans son corps) est quantifiée universellement, exemple :

**père(X,Y) :- parent(X,Y), mâle(X).**

Signifie « si quelqu'un est le parent de quelqu'un d'autre et que c'est un mâle, donc c'est son père ».

➤ Une variable apparaissant dans le corps d'une clause mais pas dans sa tête est quantifiée existentiellement, exemple :

**meme\_pere(X,Y) :- pere(P,X), pere(P,Y).**

se lit: ``pour tout X et pour tout Y, meme\_pere(X,Y) est vrai s'il existe un P tel que pere(P,X) et pere(P,Y) soient vrais".

## **II-4 Les buts**

- Les faits et règles correspondent aux données d'un programme PROLOG, les questions correspondent aux buts.
- Les buts peuvent être élaboré par des prédicats simples. Exemple :

**?- patron( ali, khalil ).**

Ici, on interroge le système pour savoir si ce fait existe dans la base. C'est une question fermée, qui a comme réponse VRAI ou FAUX. D'ou :

**?- patron( ali, khalil ).**

**Yes**

- La conjonction (le ET logique) est notée par une virgule en PROLOG. Une conjonction est vraie si toutes ses composantes, évalués de gauche a droite, sont vraies. Exemple :

**?- patron( ali, khalil ), patron( ali, soheib ).**

Ici, on veut savoir si ali est le patron de khalil et de soheib.

## **III Les programmes Prolog**

### **III-1 Définition d'un programme Prolog**

- ✓ Un programme Prolog est constitué d'une suite de clauses regroupées en paquets. L'ordre dans lequel les paquets sont définis n'est pas significatif.
- ✓ Chaque paquet définit un prédicat et est constitué d'un ensemble de clauses dont l'atome de tête est le même.

- ✓ L'ordre dans lequel les clauses sont définies est significatif.
- ✓ Deux clauses d'un même paquet sont liées par un ou logique. Par exemple, le prédicat **personne** défini par les deux clauses:

**personne(X) :- homme(X).**  
**personne(X) :- femme(X).**

se lit ``pour tout X, **personne(X)** est vrai si **homme(X)** est vrai ou **femme(X)** est vrai''.

## III-2 Exécution de programmes Prolog

- ✓ PROLOG est un langage logique, en théorie vous ne devriez pas vous préoccuper de la façon dont il s'exécute. Cependant il est quelquefois prudent de prendre en compte comment l'algorithme d'inférence agit.

- ✓ "Exécuter" un programme Prolog consiste à poser une question à l'interprète PROLOG.
- ✓ Une question (ou but ou activant) est une suite d'atomes logiques séparés par des virgules.
- ✓ La réponse de Prolog est ``yes'' si la question est une conséquence logique du programme, ou ``no'' si la question n'est pas une conséquence logique du programme.
- ✓ Une question peut comporter des variables, quantifiées existentiellement. La réponse de Prolog est alors l'ensemble des valeurs des variables pour lesquelles la question est une conséquence logique du programme. Par exemple, la question

**?- pere(al,X), pere(X,Y).**

se lit ``est-ce qu'il existe un X et un Y tels que pere(al,X) et pere(X,Y) soient vrais''.

- ✓ La réponse de Prolog est l'ensemble des valeurs de X et Y qui vérifient cette relation. Autrement dit, la réponse de Prolog à cette question devrait être l'ensemble des enfants et petits-enfants d' ali.

### **III-3 Signification (sémantique) d'un programme Prolog**

- ✓ Quand l'interpréteur reçoit une requête, il recherche les règles et les faits dont la partie gauche peut être unifiée avec la requête, et effectue cette unification avec la première règle trouvée.
  
- ✓ Par exemple ayant ce code Prolog :

**frère\_ou\_sœur(X,Y) :- parent(Z,X), parent(Z,Y), X \= Y.**  
**parent(X,Y) :- mère (X,Y).**  
**parent(X,Y) :- père (X,Y).**  
**mère(amina, lina).**  
**père(ali, lina).**  
**père(ali, ahmed).**  
**père(mohamed, ali).**

Il en résulte que la demande suivante est évaluée comme vraie:

**?- frère\_ou\_sœur(lina, ahmed).**  
**oui.**

L'interpréteur arrive à ce résultat en faisant correspondre la règle **frère\_ou\_sœur(X,Y)** en unifiant **X** avec **lina** et **Y** avec **ahmed**. Cela signifie que la demande peut être étendue à **parent(Z,lina)**, **parent(Z,ahmed)**. Faire correspondre cette conjonction en regardant tous les parents possibles de **lina**.

Cependant, **parent(amina,lina)** ne mène pas à une solution viable, parce que si **amina** est substitué pour **Z**, **parent(amina,ahmed)** devra être vrai, mais un tel fait n'est pas

présent. On choisit une autre valeur: **parent(ali, lina)**, qui mène à **frère\_ou\_sœur(lina, ahmed)** vrai.