

cours 1

Introduction

L'IA Symbolique :

- **Intègre l'expertise humaine dans un algorithme.**
- **Restitue des connaissances et raisonnement logique.**
- **Est lisible et compréhensible par l'humain.**
- **Aide à la décision.**
- **Permet l'explicabilité.**
- **Représente des connaissances.**

Elle utilise des systèmes experts, qui sont une autre famille d'IA.

Base de connaissances

Une base de connaissances est composée de :

- **Faits**
- **Règles**

Elle utilise aussi un moteur d'inférence.

Exemple :

- **Tous les hommes sont mortels.**
- **Socrate est un homme.**

La requête "Est-ce que Socrate est mortel ?" serait résolue en appliquant les règles logiques de la base de connaissance et en manipulant les faits.

Programmation logique

Programmation déclarative

La programmation logique est un paradigme de programmation très différent de la programmation procédurale. C'est une programmation déclarative. Le programme décrit les connaissances et le problème, mais pas comment résoudre le problème. Le moteur d'inférence associé au langage de programmation résout le problème.

Prolog

Le premier interpréteur Prolog a été créé en 1972 à la faculté des sciences de Luminy à Marseille par Alain Colmerauer et Philippe Roussel. Prolog s'appuie sur les principes de la logique du premier ordre et est principalement utilisé en intelligence artificielle. Sa vocation initiale est le traitement du langage naturel.

Programme Prolog :

```
mortel(X) :- homme(X).  
homme('Socrate').
```

Requête Prolog :

```
mortel('Socrate').
```

Réponse Prolog :

```
true
```

Requête Prolog :

```
mortel(X).
```

Réponse Prolog :

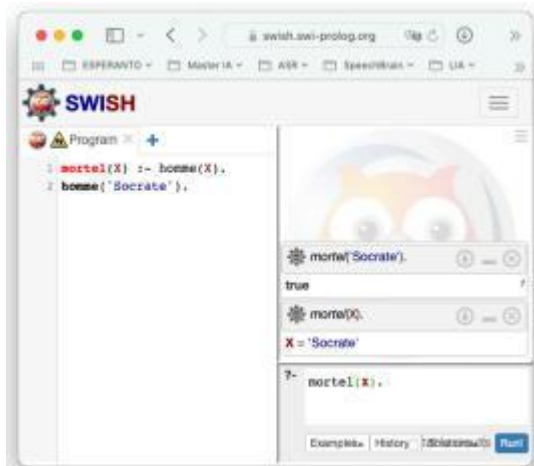
```
X='Socrate'
```

Ressources pédagogiques

Il existe plusieurs ressources pédagogiques pour apprendre Prolog, notamment :

- L'interpréteur SWI Prolog (logiciel libre disponible sous Linux, MacOS et Windows).
- Le site "Learn Prolog Now!" (<https://lpn.swi-prolog.org/lpnpage.php?pageid=online>).
- L'interpréteur en ligne SWIsh (<https://swish.swi-prolog.org/>).

Le visuel suivant présente l'interpréteur en ligne SWISH:



Prolog déduit logiquement de nouveaux faits sur la situation que nous avons décrite et rend ses déductions comme réponses. Son fonctionnement de base est de décrire la situation considérée et de poser une question.

Fonctionnement de base

- **Penser de manière déclarative, pas de manière procédurale.**
- **Utile pour un grand nombre d'applications en intelligence artificielle.**
- **Nécessite un état d'esprit différent.**
- **Langage de haut niveau.**
- **Pas aussi efficace que le C.**
- **Pertinent pour du prototypage rapide.**

Faits, règles, et clauses

Prolog est basé sur trois constructions de base :

- **Les faits**
- **Les règles**
- **Les requêtes**

Les règles et les faits sont des clauses.

Programme Prolog :

```
mortel(X) :- humain(X).  
humain(X) :- homme(X).  
humain(X) :- femme(X).  
homme('Socrate').  
femme('Aspasie').
```

Dans l'exemple au dessus, il y a :

- **2 faits**
- **3 règles**
- **5 clauses**

Les faits

Les faits sont des données élémentaires considérées comme vraies. Un fait est une affirmation inconditionnelle.

Ce sont des formules atomiques constituées d'un prédicat, suivi ou pas d'une liste ordonnée d'arguments : `nom_du_predicat(arg1, arg2, ..., argn)`.

Exemples :

```
homme(socrate).  
femme(aspasie).  
satellite(terre, soleil).  
satellite(lune, terre).  
satellite(europe, jupiter).  
satellite(ganymede, jupiter).
```

Les règles

Une règle comporte une tête, suivi d'un corps. Entre les deux se trouvent les deux caractères ':-' (syntaxe Édimbourg). Une règle est une affirmation conditionnelle. La règle signifie que la tête est vraie si le corps est vrai.

Exemple :

```
mortel(X) :- humain(X).  
humain(X) :- homme(X).  
humain(X) :- femme(X).
```

Dans l'exemple au dessus, il y a :

- 3 règles
- 3 clauses
- 3 prédicats

Le ';' dans le corps d'une règle représente une disjonction (OU logique). La conjonction (ET logique) s'écrit avec une virgule.

```
humain(X) :- homme(X) ; femme(X).  
pere(X,Y) :- homme(X), parent(X,Y).
```

Syntaxe et terminologie

Les termes

Les objets manipulés par Prolog sont appelés des termes.

Il y a 3 types de termes :

- **Les variables**
- **Les termes élémentaires**
- **Les termes composés**

Les variables

Une variable Prolog représente le même objet tout au long de sa durée de vie et ne peut pas changer de valeur (similaire à une variable mathématique).

Une variable s'écrit sous la forme d'une chaîne alpha-numérique commençant par une majuscule ou par un souligné. La variable anonyme s'écrit "_". C'est un objet dont on ne souhaite pas connaître la valeur.

Exemples :

- **X**
- **Var_longue_2**
- **_toto**
- **_21**

Les termes élémentaires

Les termes élémentaires sont les objets simples connus de l'univers.

Il y a trois types de termes élémentaires :

- Les identificateurs (ou atomes). Un identificateur est une chaîne alpha-numérique commençant par une minuscule : **toto**, **aX12**, **jean_Paul_2**
- Les chaînes de caractères entre simples quotes : **'Hello World!'**, **'Socrate'**
- Les nombres : entiers ou flottants

Les termes composés

Les termes composés représentent les objets structurés de l'univers. Un terme composé s'écrit sous la forme : **foncteur(t1, ..., tn)**

Le nombre n d'arguments d'un terme complexe est appelé arité.

En Prolog, il est possible de définir deux prédicats avec le même foncteur mais avec une arité différente. Prolog les traitera comme deux prédicats différents. Dans la doc. Prolog l'arité d'un prédicat est généralement indiquée par le suffixe "/" suivi d'un nombre représentant l'arité.

Exemple :

```
cool(yolanda).  
ecoute2LaMusique(mia).  
ecoute2LaMusique(yolanda):- cool(yolanda).  
joueAirGuitar(mia):- ecoute2LaMusique(mia).  
joueAirGuitar(yolanda):- ecoute2LaMusique(yolanda).
```

Cette base de connaissances définit :

- cool/1
- ecoute2LaMusique/1
- joueAirGuitar/1

Exercice

Exprimer en Prolog les propositions suivantes et identifier les objets, les faits, les règles :

- la chèvre est un animal herbivore
- le loup est un animal cruel
- toute chose cruelle est carnivore
- un animal carnivore mange de la viande et 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
- Question : y a-t-il un animal cruel et que consomme-t-il ?

Une correction possible :

animal(chèvre).

herbivore(chèvre).

animal(loup).

cruel(loup).

carnivore(X) :- cruel(X).

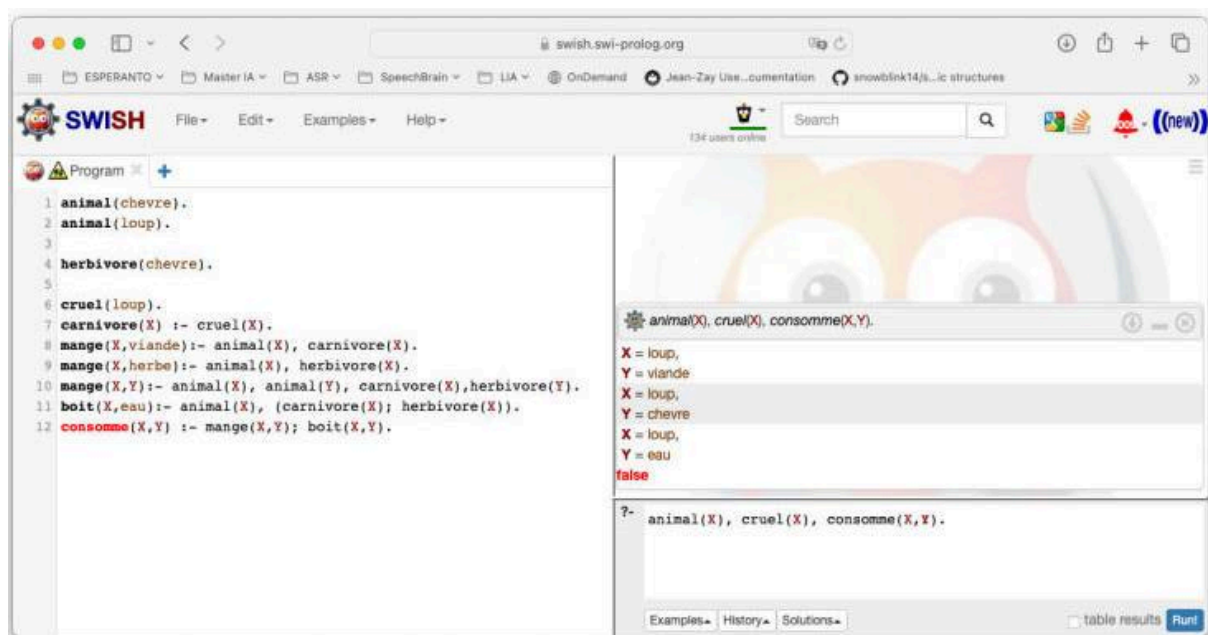
mange(X,viande) :- animal(X), carnivore(X).

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

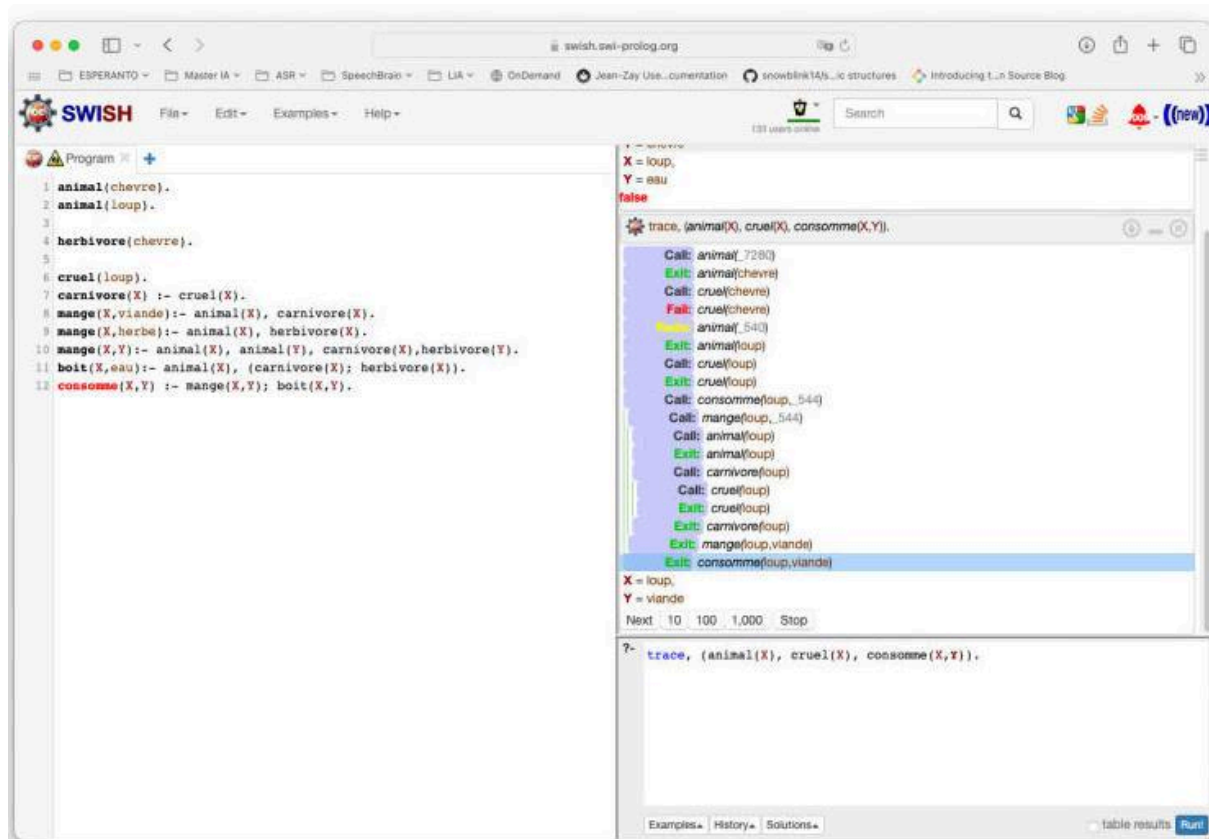
**mange(X,Y) :- animal(X), animal(Y), carnivore(X),
herbivore(Y).**

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

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



Dans l'image suivante, nous pouvons voir le résultat d'une requête Prolog, qui affiche un traçage de l'exécution de la requête :



Références :

SWI-Prolog. (s. d.). <https://www.swi-prolog.org/>

Learn ProLog Now ! (s. d.).

<https://lpn.swi-prolog.org/lpnpag.php?pageid=top>

Harrison, A. (1995). The Integration of Multiple and Diverse Knowledge Representation Paradigms using a Blackboard Architecture.

<https://doi.org/10.21954/ou.ro.0000fb85>

Getting Started [SDA SE Wiki].

https://sewiki.iai.uni-bonn.de/research/jtransformer/tutorial_old/stepbystep

PROLOG – Programmation Logique

© Patrick Blackburn, Johan Bos & Kristina Striegnitz

Traduction : Yannick Estève

Cours dispensé à CERI – Avignon Université