

2. Caractérisation de Prolog

(VIDEO-PPT 1.1)

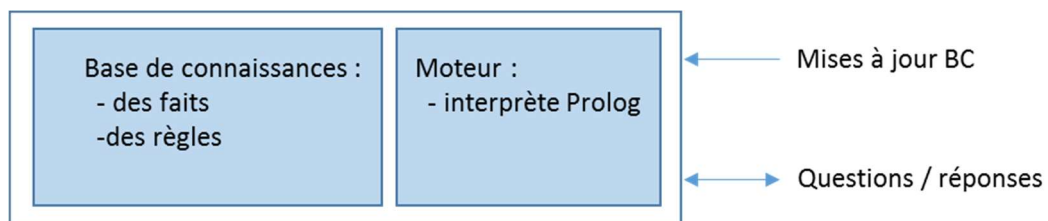
C'est un langage sous-ensemble de la logique (des affirmations vraies, pas de négation), une exécution est une preuve de concept.

⇒ Une rencontre du « 3^{ème} type » :

- pas de : variable désignant un(des) mot(s) mémoire, déclaration, affectation, structures de contrôle (if, while, goto)... rien d'habituel ;
- mais : des relations entre données (prédicats) et des solutions pour un problème à résoudre => non déterminisme.

2.1. Une première caractéristique : pas de contrôle de l'exécution !

Prolog est déclaratif : on exprime le *quoi* (la connaissance, ce qui est vrai) et pas *comment* la machine doit résoudre un problème. Les informations de la base de connaissances (données) sont décrites logiquement sans se préoccuper de la manière dont elles seront utilisées ultérieurement. Le niveau machine (mémoire/variable, séquence de commandes) est masqué. Une exécution est déclenchée par une interrogation. Le moteur (contrôle) applique le principe de raisonnement logique sous-jacent pour trouver toutes les solutions à une question.



En Prolog on exprime 3 choses :

- des faits ;
- des règles ;
- des questions.

Programmer en Prolog c'est :

- mettre à jour la base de connaissances ;
- interroger la base de connaissances.

👉 *Le contrôle de l'exécution ne relève plus du programmeur comme en programmation impérative mais du moteur Prolog.*

Un premier programme :

faits	<code>mot('hello').</code>
	<code>mot('world').</code>
une règle	<code>doubleMot(X,Y) :- mot(X), mot(Y), X\=Y.</code>
une question	<code>doubleMot(X, Y), write(' '), write(X), write(' '), write(Y).</code>

Vous pouvez y remarquer que

- les faits et règles se construisent sur des prédicats en minuscules
- les constantes littérales chaînes de caractères se construisent sur des quotes
- les majuscules (en 1^{er} caractère) sont des variables
- Les items lexicaux de prolog sont le point (termine toute ligne), la virgule (séparatrice de buts et dans les règles le :- séparant à gauche un but (conséquence) vrai si la suite des sous-buts à droite peut être satisfaite

Note : $X \neq Y$ correspond à une contrainte imposant que les variables X et Y doivent avoir des valeurs différentes

En *Gprolog* la base de connaissances est saisie dans un fichier texte avec l'extension *.pl* (*helloWorld.pl*). Celui-ci est ensuite chargé sous l'interpréteur en exécutant la commande :

```
consult('helloWorld.pl').
```

puis peut être rappelé avec la flèche <↑>. La commande *listing.* permet de s'assurer du bon chargement dans la base de connaissances chargée.

Une première interrogation via une pseudo-règle permettant de mémoriser la question :


```
question :- doubleMot(X, Y), write(X), write(' '), write(Y).
```

donnera :

```
|?- question.  
hello world  
  
true ? ;  
world hello  
  
true ? <RC>  
yes
```

Prolog essaie d'effacer le but *question* et propose un premier enchaînement *hello world*. En répondant ; à la poursuite de l'interrogation, nous demandons à l'interpréteur de chercher un second enchaînement *world hello* puis la saisie du *Retour Charriot* arrête la recherche.

Note : en répondant a, l'interpréteur fournit tout de suite toutes les possibilités d'enchaînement.

 *L'exécution d'un programme Prolog à consister à effacer la succession de buts dans la question, ici :*

- 1) *doubleMot(X, Y)*
- 2) *write(' ')*
- 3) *write(X)*
- 4) *write(' ')*
- 5) *write(Y)*

En remplaçant la tête de clause *doubleMot(X, Y)* par sa queue *mot(X) mot(Y)* avec la contrainte X et Y différents, puis en unifiant *mot(X)* puis *mot(Y)* avec les faits, on obtient les deux couples (X,Y) affichés.

☞ *Prolog gère les possibilités multiples d'effacement de but (l'indéterminisme) par l'énumération de toutes les possibilités.*

Il est évidemment possible d'échouer, l'interpréteur n'arrivant pas à établir la « vérité » de la question par effacement de buts :

```
| ?- doubleMot('hello','world').  
yes  
| ?- doubleMot('hello','hello').  
no
```

mini TP : En utilisant le prédicat `avant(X,Y)` fourni, filtrer la question pour n'afficher que le seul succès 'hello world'