

# IFT-17586 - Intelligence Artificielle I

---

## Séance #1, partie 2

Benoît Potvin

Département d'informatique et de génie logiciel  
Faculté des sciences et de génie  
Université Laval  
7 mai 2003

© Capus, Potvin et Tourigny, 2003

# IFT-17586 - Intelligence Artificielle I

---

1. Introduction aux langages en IA
2. Niveaux d'abstraction
3. Pourquoi utiliser PROLOG
4. Caractéristiques de PROLOG
5. Éléments de syntaxe
6. Exemples
7. Exercices
8. À lire

© Capus, Potvin et Tourigny, 2003

## Introduction aux langages en IA

---

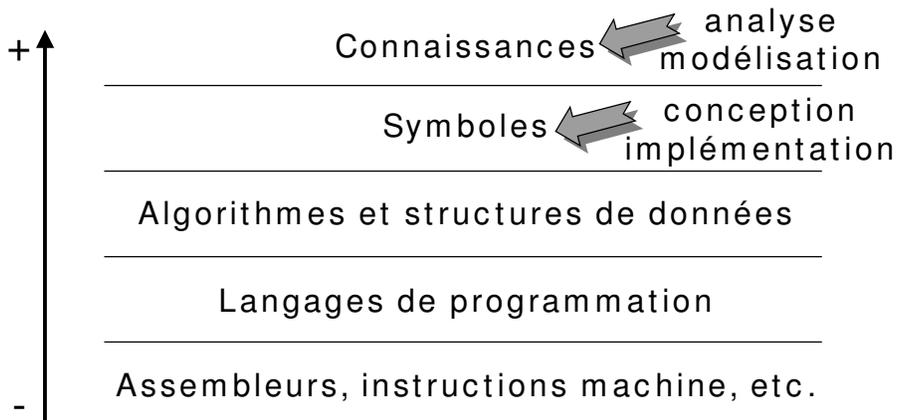
Luger (2002), pages 597-602 :

- Quels sont les langages pour programmer en IA ?
- Comment sélectionner un langage pour programmer en IA ?
- Pourquoi les langages spécifiques à l'IA sont-ils importants ?

© Capus, Potvin et Tourigny, 2003

## Niveaux d'abstraction

---



© Capus, Potvin et Tourigny, 2003

## Pourquoi utiliser Prolog ?

---

- Langage pédagogique
- Prototypage
- Incorporation dans des langages d'actualité comme JAVA  
Developing Intelligent Systems With Java and Prolog (2000), *JavaReport*, Vol. 5, No. 5, pp. 22-32
- Une autre façon de penser les problèmes pour les étudiants en informatique :
  - un paradigme différent de la résolution de problèmes
  - un excitant défi intellectuel

© Capus, Potvin et Tourigny, 2003

## Caractéristiques de Prolog

---

- Développement : début des années 70 pour le traitement du langage naturel
- Exécution de programmes en faisant des inférences à partir de spécifications logiques
- Programmation déclarative (*ce qui est vrai* et *ce qui doit être fait*) par opposition à la programmation procédurale (*comment le faire*)
- Représentation simple des objets et de leurs relations

© Capus, Potvin et Tourigny, 2003

## Éléments de syntaxe

---

- Respect de la norme ISO
- Issue du calcul de prédicats  
(et  $\wedge$ , ) (ou  $\vee$  ; ) (si  $\rightarrow$  :- ) (non  $\neg$  **not** )
- Représentation
  - Relation : prédicat (arité = nombre d'objets)
  - Objet : constante ou variable
- Écriture
  - Prédicats et constantes  $\Rightarrow$  minuscule (ex: toto, 12, a(b), ...)
  - Variables  $\Rightarrow$  majuscule (ex: Toto, A, ...)

© Capus, Potvin et Tourigny, 2003

## Exemples

---

- « *Tout le monde aime Susy* »  
aime(Tous, susy).
- « *Quelqu'un aime à la fois Georges et Susy.* »  
... aime(X, georges) , aime(X, susy).
- « *Georges aime Cathy et Susy.* »  
... aime(georges, cathy) , aime(georges, susy).
- « *Georges aime Cathy ou Susy.* »  
... aime(georges, cathy) ; aime(georges, susy).
- « *Si Georges n'aime pas Cathy alors il aime Susy* »  
aime(georges, susy) :- not(aime(georges, cathy)).

© Capus, Potvin et Tourigny, 2003

## À retenir

---

- En Prolog, la représentation de l'implication est inversée :

Français : « *Si Georges n'aime pas Cathy alors il aime Susy* »

Calcul des  
prédicats :  $\neg \text{aime}(\text{georges}, \text{cathy}) \rightarrow \text{aime}(\text{georges}, \text{susy})$

PROLOG :  `aime(georges, susy) :- not(aime(georges, cathy)).`

© Capus, Potvin et Tourigny, 2003

## Exemple : représentation en PROLOG

---

- Représenter un « monde » : ce que Georges, Cathy et Susy aiment

« <i>Georges aime Cathy.</i> »	<code> aime(georges, cathy).</code>
« <i>Georges aime Susy.</i> »	<code> aime(georges, susy).</code>
« <i>Georges aime le vin.</i> »	<code> aime(georges, vin).</code>
« <i>Susy aime le vin.</i> »	<code> aime(susy, vin).</code>
« <i>Cathy aime le gin.</i> »	<code> aime(cathy, gin).</code>
« <i>Cathy aime Susy.</i> »	<code> aime(cathy, susy).</code>

© Capus, Potvin et Tourigny, 2003

## Exemple : poser une question dans PROLOG

```
aime(georges, cathy).  
aime(georges, susy).  
aime(georges, vin).  
aime(susy, vin).  
aime(cathy, gin).  
aime(cathy, susy).
```

« *Georges aime qui/quoi ?* »

?- aime(georges, X).

X=cathy ;

X=susy ;

X=vin ;

no

« *Georges aime-t-il Cathy ?* »

?- aime(georges, cathy).

yes

« *Georges aime-t-il la bière ?* »

?- aime(georges, bière).

no

« *Cathy aime-t-elle Susy ?* »

?- aime(cathy, susy).

yes

© Capus, Potvin et Tourigny, 2003

## Exercices

- 1) Représenter les connaissances, données en langage naturel, en Prolog en utilisant la relation

**parent(X,Y) : X est le parent de Y.**

« *Pierre et Nathalie sont les parents de Sébastien.  
Marie et Robert sont ses grand-parents maternels.* »

© Capus, Potvin et Tourigny, 2003

## Exercices

---

- 2) À partir des connaissances décrites dans la question (1), traduire les questions suivantes en Prolog. Donner les réponses.

« *Qui sont les parents de Nathalie ?* »

« *Est-ce que Nathalie est la fille de Marie ?* »

« *Est-ce que Robert est le grand-père de Sébastien ?* »

© Capus, Potvin et Tourigny, 2003

## Exercices

---

- 3) Donner les réponses de Prolog aux requêtes suivantes.

?- parent(robert, P), parent(P, sébastien).

?- parent(P, nathalie), parent(M, nathalie).

?- parent(GP, parent(P, sébastien)).

© Capus, Potvin et Tourigny, 2003

## Exercices

---

- 4) Représenter les mêmes connaissances qu'à la question (1), mais en utilisant la relation  $\text{parents}(X, Y, Z)$  : X et Y sont les parents de Z.

© Capus, Potvin et Tourigny, 2003

## Exercices

---

- 5) Donner les réponses de Prolog aux requêtes suivantes

?-  $\text{parents}(P, M, \text{nathalie})$ .

?-  $\text{parents}(\text{Maman}, \text{Papa}, \text{Fils})$ .

?-  $\text{parents}(\text{pierre}, X)$ .

© Capus, Potvin et Tourigny, 2003

## Exercices

---

- 6) Que signifie les requêtes Prolog suivantes ?

?- parents(robert, nathalie, D).

?- parents(robert, marie, nathalie).

© Capus, Potvin et Tourigny, 2003

## À lire

---

- Luger, 2002 :
  - Langages, compréhension et niveaux d'abstraction en IA (pp. 597-600)
  - Survol de PROLOG et LISP (pp. 600-602)
  - Sélection d'un langage d'implémentation (p. 602)
  - Introduction à PROLOG : syntaxe (pp. 603-608)

© Capus, Potvin et Tourigny, 2003