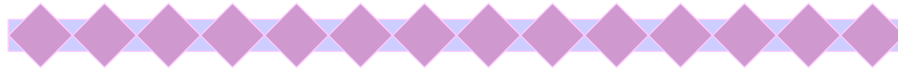


IFT-17586

Intelligence artificielle I



Récapitulation

© Tourigny, Potvin et Capus (2003)

1

Objectifs de ce cours



- **Reconnaître** des problèmes/applications de l'IA
- **Expliquer/choisir/utiliser** des techniques de recherche de l'IA
- **Expliquer/choisir/utiliser** des modes de représentation des connaissances
- **Utiliser** un langage de programmation déclaratif : PROLOG
- **Concevoir** des systèmes d'IA en suivant une approche déclarative :
 - jeux, système à base de connaissances, système de traitement automatique du langage naturel, ...

© Tourigny, Potvin et Capus (2003)

2

Problèmes/applications de l'IA

◆ 4 approches en IA

	humaine	rationnelle
processus	Modélisation cognitive Penser comme un humain	Lois de la pensée Penser rationnellement
résultat	Test de Turing Agir comme un humain	Agent rationnel Agir rationnellement

© Tourigny, Potvin et Capus (2003)

3

Recherche dans un espace d'états

◆ Espace d'états

◆ Recherche dans un espace d'états :

- Dans quelle direction ?
 - À partir des données : chaînage avant
 - À partir d'un but : chaînage arrière
 - Combinaison buts + données
- Dans quel ordre ?
 - En profondeur d'abord
 - En largeur d'abord
 - En profondeur d'abord avec approfondissement itératif

© Tourigny, Potvin et Capus (2003)

4

Recherche dans un espace d'états

- ◆ Heuristique
- ◆ Recherche heuristique
 - Hill-Climbing le plus escarpé
 - Meilleur d'abord
 - Minimax
 - Alpha-Beta

© Tourigny, Potvin et Capus (2003)

5

Recherche dans un espace d'états

- ◆ Production
- ◆ Système de production
 - Mémoire de travail
 - Mémoire à long terme
 - Cycle de contrôle

© Tourigny, Potvin et Capus (2003)

6

Programmation Prolog

◆ Éléments de syntaxe

- Opérateurs
 - A et B : A , B
 - A ou B : A ; B
 - si A alors B : B :- A
 - non A : **not** A
- Prédicats et constantes ⇒ **minuscule**
 - ex: toto, 12, a(b), ...
- Variables ⇒ **majuscule**
 - ex: Toto, A, ...

© Tourigny, Potvin et Capus (2003)

7

Programmation Prolog

◆ Composantes d'un programme : clauses

- Faits (ou affirmations)
 - a(1).
 - b(2).
- Règles (d'inférence ou de déductions)
 - a(X):-b(X).
- Questions (résolution de problèmes)
 - ?-a(2).
 - yes

© Tourigny, Potvin et Capus (2003)

8

Programmation Prolog

- ◆ 3 caractéristiques de base : unification, appel et retour arrière

$| \text{?- } a(b(c), X, [d(Z), e, f, g]) = Z(C, Z(d), [Y | W]).$
 $X = a(d) , Z = a , C = b(c) , Y = d(a) , W = [e,f,g]$

$| \text{?- } a(X) = a(b(X)).$

Erreur

Programmation Prolog

- ◆ 3 caractéristiques de base : unification, appel et retour arrière

$\text{reduire}(A * B + A * C, A * (B + C)).$

$| \text{?- } \text{reduire}(X, a * (b + c)).$

$X = a * b + a * c$

$| \text{?- } \text{reduire}(a * b + a * c, X).$

$X = a * (b + c)$

$| \text{?- } \text{reduire}(2 + 3 * b, X).$

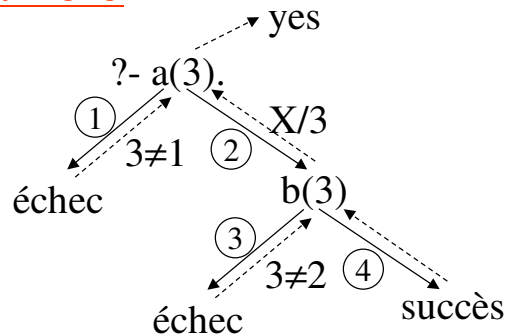
no

Programmation Prolog

- ◆ 3 caractéristiques de base : unification, appel et retour arrière

```

① a(1).
② a(X):- b(X).
③ b(2).
④ b(3).
    
```



© Tourigny, Potvin et Capus (2003)

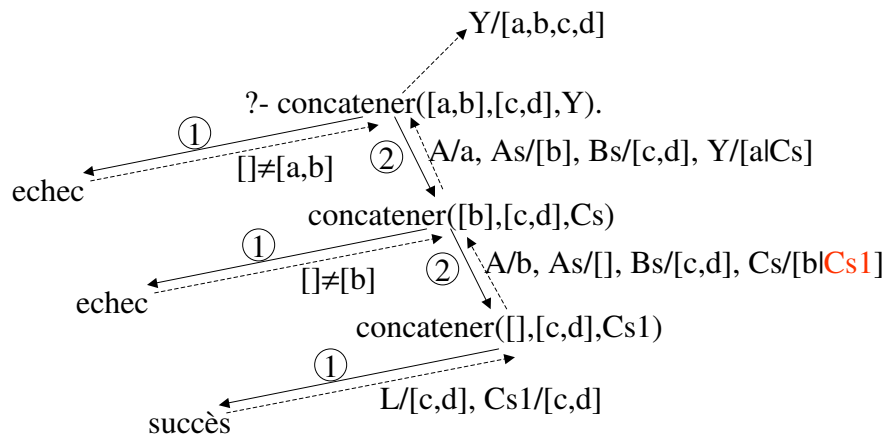
11

Programmation Prolog

Construction d'un arbre de résolution

```

① concatener([],L,L).
② concatener([A|As], Bs,[A|Cs]):-concatener(As,Bs,Cs).
    
```

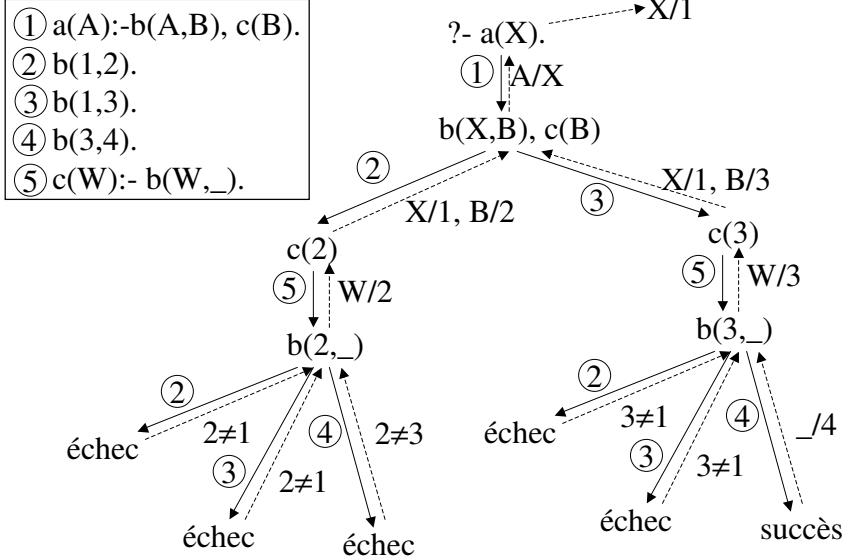


© Tourigny, Potvin et Capus (2003)

12

Programmation Prolog

Construction d'un arbre de résolution



© Tourigny, Potvin et Capus (2003)

13

Programmation Prolog

- ◆ Règles récursives
- ◆ Assertions
- ◆ Listes et récursivité
- ◆ Variable anonyme
- ◆ Opérateurs
- ◆ Prédicats ensemblistes
 - bagof/3
 - setof/3
 - findall/3

© Tourigny, Potvin et Capus (2003)

14