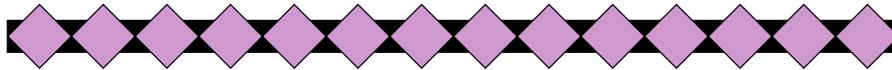


IFT-17586

Intelligence artificielle I



Révision

Benoît Potvin

Département d'informatique et de génie logiciel
Faculté des sciences et de génie, Université Laval
Été 2003

Objectifs



- ◆ Nommer, expliquer et utiliser les modes importants de représentation des connaissances de l'IA
- ◆ Choisir les modes de représentation les plus appropriés dans un contexte donné
- ◆ Résoudre des problèmes en utilisant des techniques de résolution de l'IA, notamment pour les jeux, la planification, le traitement des langues naturelles et les systèmes à base de connaissances
- ◆ Concevoir et implémenter des solutions en utilisant une approche déclarative

Examen final

- ◆ Date : mercredi 6 août 2003
- ◆ Heure : 18h30-21h30
- ◆ Local : PLT-2751
- ◆ Aucune documentation permise

Examen final

- ◆ Plan
 - 50% Partie théorique
 - TALN
 - SBC / Planification
 - RC
 - AA, etc.
 - 50% Partie PROLOG
 - Méta-programmation / SBC / Planification
 - TALN

Revision

◆ Partie 1 : Théorie

- Représentation des Connaissances
- Raisonnements
- Systèmes à Base de Connaissances
- Traitement Automatique du Langage Naturel
- Planification
- Apprentissage Automatique

Revision

◆ Partie 2 : Conception / implémentation

- Autres particularités de PROLOG
- Implémentation des modes de représentation des connaissances
- Méta-programmation
- Conception de SBC
- Implémentation du TALN
- Conception de planificateurs

Revision

◆ Partie 3 : Laboratoires

- Les liens de parenté
- Qui a volé les biscuits ?
- Le singe et les bananes
- À la recherche d'un téléphone
- Méta-programmation
- Application TALN
- Planification

Revision

◆ Travaux pratiques

- TP2 : représentation des connaissances
- TP3 : Conception d'un SBC
- TP4 : TALN

Prédicats particuliers

◆ ! : stopper la récursion

– exemple :

a(1).	a(2).	a(3).	?- c(X).	?- e(X).
			X = 2 ;	no
b(2).	b(3).	b(4).	X = 3	
c(A) :- a(A), b(A).			?- d(X).	?- f(X).
d(A) :- !, a(A), b(A).			X = 2 ;	X = 2
e(A) :- a(A), !, b(A).			X = 3	
f(A) :- a(A), b(A), !.				

Logique des propositions

◆ Représentation en Lprop

– ex : $p \rightarrow q$

◆ Raisonnement en Lprop

– ex : $p \rightarrow q, p \vdash q$ (Modus Ponens)

Logique des prédicats

◆ Représentation en Lpred

–ex: $\forall X : p(X) \rightarrow q(X)$

◆ Raisonnement en Lpred

–ex: $\forall X : p(X) \rightarrow q(X), p(a) \vdash q(a)$

Logique clause

◆ Représentation en LClaus

–ex: $\{ \neg p(X) \vee q(X), p(a) \}$

◆ Raisonnement en LClaus

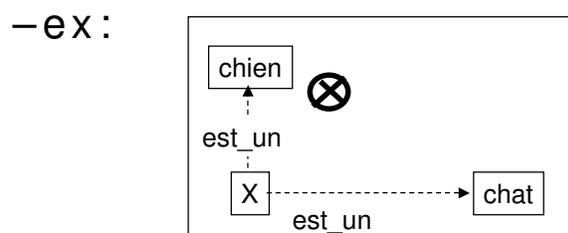
–ex: $\neg p(X) \vee q(X)$ $p(a)$
 X/a \swarrow \searrow
 \downarrow
 $q(a)$ (principe de résolution)

Réseaux sémantiques

◆ Représentation en RSém

-ex: gentil (curled arrow) milou → est_un chien

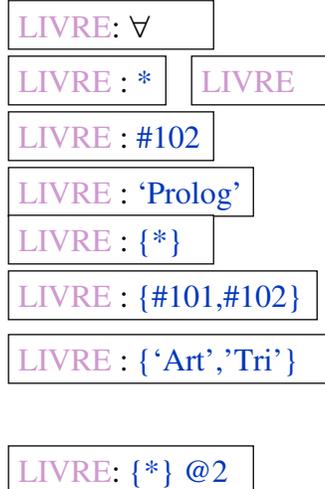
◆ Raisonnement en RSém



Graphes conceptuels

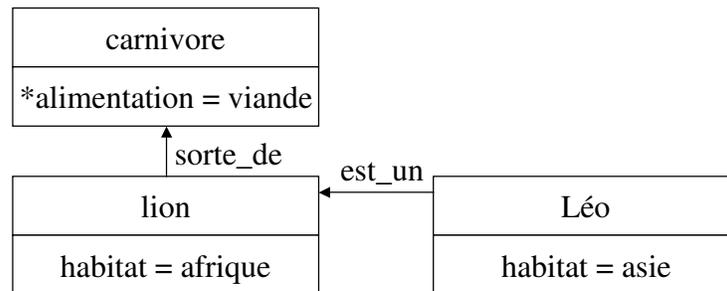


- Marie achète tous les livres
- Marie achète un livre
- Marie achète le livre (particulier)
- Marie achète un livre intitulé "Prolog".
- Marie achète des livres
- Marie achète les livres (particulier)
- Marie achète un livre intitulé 'Art' et un autre intitulé 'Tri'
- cardinalité d'un ensemble: deux livres
Marie achète 2 livres



Schémas

- sous-réseau sémantique identifié par un noeud accompagné de ses attributs et valeurs
- représente une classe ou un objet et ses attributs



© Capus, Potvin et Tourigny, 2003

15

Scripts

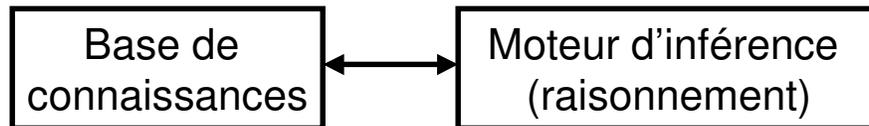
- Déf.: Représentation structurée décrivant une suite stéréotypée d'événements (suite d'événements) dans un contexte particulier
 - Ex.: Script du restaurant, script d'une journée d'un étudiant, script d'une journée d'un professeur
- Utilité des scripts
 - Fournir de des connaissances sur le contexte, Aider à la compréhension de textes
- Composantes d'un script:
 - conditions d'entrée, résultat, objets spécifiques (entités impliquées), rôles de chaque entité, scènes (ensemble des suites d'événements)

© Capus, Potvin et Tourigny, 2003

16

Le raisonnement en IA

- ◆ Séparation entre connaissance et raisonnement



Systemes à base de connaissances

- ◆ Architecture d'un SE (luger2002, p.249)

Systemes à base de connaissances

- ◆ **Modèle conceptuel**
(luger,2002,p255)

Conception d'un SBC en Prolog

- ◆ Base de faits \Rightarrow un prédicat fait/1
fait(F) où F est un fait du domaine
- ◆ Base de règles \Rightarrow ajout d'opérateurs
si Condition **alors** Conclusion où :
 - Condition = Fait
 - Condition = Condition1 **et** Condition2
 - Condition = Condition1 **ou** Condition2
- ◆ Moteur d'inférence \Rightarrow 2 prédicats
 - ch_arrière/1 : prouver qu'un but est vrai
 - ch_avant/0 : déduire tous les faits possibles

Méta-programmation

S'il existe 2 clauses C1 et C2 telles que
P est une sous-expression disjonctive de C1
et
 $\neg P$ est une sous-expression disjonctive de C2

alors

- enlever P dans C1 (la nouvelle clause est CA)
- enlever $\neg P$ dans C2 (la nouvelle clause est CB)
- ajouter dans la base de clauses la clause (CA \vee CB)

\Rightarrow Enlever P dans C sachant que P est une sous-expression disjonctive de C (on obtient CA)

GRAMMAIRE

V_T : {symboles Terminaux} non dérivables

V_N : {symboles Non terminaux dérivables
(catégories syntaxiques) }

R : {RÈGLES DE RÉÉCRITURE}

axiome S

$\langle \text{phrase} \rangle \rightarrow \langle \text{syntagme_nominal} \rangle, \langle \text{syntagme_verbal} \rangle$

$\langle \text{syntagme_nominal} \rangle \rightarrow \langle \text{nom} \rangle$

$\langle \text{syntagme_verbal} \rangle \rightarrow \langle \text{verbe} \rangle, \langle \text{syntagme_nominal} \rangle$

$\langle \text{nom} \rangle \rightarrow \text{'Tarzan'} | \text{'Jane'}$

GRAMMAIRE ATTRIBUÉE

$\langle \text{syntagme_verbal(SV)} \rangle \rightarrow \langle \text{verbe(V)} \rangle ,$
 $\langle \text{syntagme_nominal(SN)} \rangle$
 $\{ \text{SV} = \text{gv}(\text{V}, \text{SN}) \}$

- ATTRIBUTS : paramètres

SV V SN

- ACTIONS ou procédures

« *Tarzan embrasse Jane* »

TALN en PROLOG

◆ Grammaire #9

- Pour spécifier cette condition, utiliser les { }

p(ACTION(AGNT, OBJ)) --> sn(AGNT), sv(ACTION, OBJ), { AGNT \= OBJ }.

sn(AGENT) --> a, nc(AGENT).

sv(ACTION,OBJET) --> v(ACTION), sn(OBJET).

a --> [le].

a --> [un].

nc(chien) --> [chien].

nc(homme) --> [homme].

v(mordre) --> [mord].

l ?- p(A,P,[]).

A = mordre(chien,homme) , P = [le,chien,mord,le,homme] ;

...

l ?- p(A,[le,chien,mord,un,homme],[]).

A = mordre(chien,homme)

l ?- p(A,[le,chien,mord,un,chien],[]).

no

l ?- p(mordre(chien, chien),P,[]).

no

Planification

◆ Opérateurs

Liste **P** :
Éléments qui **doivent être vrais** pour
que l'opérateur puisse s'appliquer

opérateur

Liste **A** :
éléments à **Ajouter** à
l'état suite à l'application
de l'opérateur

Liste **S** :
éléments à **Supprimer** de
l'état suite à l'application
de l'opérateur

La liste **S** spécifie tout ce qui est nécessaire
pour satisfaire les axiomes de schémas

Planification

◆ Description des opérateurs :

Opérateur → **P** = liste des préconditions

→ **A** = liste à ajouter

→ **S** = liste à supprimer

– Formulation possible en PROLOG :

possible(Opérateur, P).

ajoute(Opérateur, A).

supprime(Opérateur, S).

Apprentissage automatique



- ◆ 3 approches principales :
 - SYMBOLIQUE : algorithmes d'induction
 - CONNEXIONNISTE : réseaux de neurones
 - SOCIALE ET ÉMERGENTE : algorithmes génétiques