

Module

Représentation des connaissances et raisonnements

Master 1 option: Intelligence artificielle

Pr. Okba KAZAR

kazarokba@yahoo.fr

Directeur du Laboratoire d'INFormatique Intelligente (LINFI)

Smart Computer Science Laboratory

Université de Biskra

Introduction à l'Intelligence Artificielle (IA)

- **L'intelligence artificielle**

- Reprenons la phrase de Marvin Minsky, figure célèbre de l'IA des années 50 :

- « Artificial Intelligence is the science of making machines do things that would require intelligence if done by humans".

- **L'IA est la branche de l'informatique qui consiste à concevoir des systèmes intelligents, c'est-à-dire qui soient capables de produire un raisonnement proche de celui de l'être humain.**

- **L'intelligence artificielle**

- **Définition IA**

- **Artificiel.** désigne ce qui n'est pas un produit de la nature.
 - **Intelligence.** "Faculté de connaître, de comprendre et de s'adapter [...]".
Jean-Pol Tassin (Pour la Science, Décembre 1998).
 - Produire une IA consisterait donc à utiliser une technique (informatique) pour concevoir un système capable de raisonnement (résoudre un problème, s'adapter à des situations nouvelles,...).
 - L'IA est une discipline de l'informatique dont le but est que les ordinateurs puissent raisonner comme les humains.
 - Elle s'adresse à une classe de **problèmes exponentiels**.

Historique de l'intelligence artificielle

- ☞ 1943 : McCulloch et Pitts créent le modèle du neurone formel
- ☞ 1948 : Création de la cybernétique (science des systèmes) par Norbert Wiener.
- ☞ 1949 : Hebb établit la première règle d'apprentissage neuronal
- ☞ 1950 Shannon, 1952 Samuel, 1953 Turing : machine pour jouer aux échecs
- ☞ 1956: Dartmouth Workshop - John McCarthy propose le terme « Artificial Intelligence »
- ☞ 1958: MIT – John McCarthy crée le « langage LISP »

Historique de l'intelligence artificielle

- ☞ 1960-1966: Représentation des connaissances et inférences en logique formelle (Méthode de la résolution Robinson)
- ☞ 1965: Zadeh crée la logique floue
- ☞ 1965: Une équipe de Stanford University, sous la direction d'Ed Feigenbaum, réalise le 1er système expert (DENDRAL)
- ☞ Weizenbaum, J., (1966) ELIZA - A computer program for the study of natural language communication between man and machine. Communications of the ACM, 9.1:36-45.
- ☞ General Problem Solver (1969): résolveur de problèmes général
- ☞ 1970: PROLOG

Historique de l'IA

☞ Le renouveau (les premiers systèmes experts) 1969-1979

☞ DENDRAL réalise l'analyse automatique des spectres de masse pour déterminer la structure moléculaire du corps chimique étudié. "Heuristics Dendral : A program for generating explanatory hypotheses in organic chemistry." de B. Buchanan, Sutherland, Feigenbaum, Machine Intelligence, 1969

☞ MYCIN diagnostique les maladies infectieuses du sang et propose un traitement approprié. " Computer based medical consultations : MYCIN" de E. Shortliffe, 1976

☞ L'IA institutionnalisée 1980-aujourd'hui

☞ Une industrie (SE, Systèmes d'apprentissage, interfaces ergonomiques, Data Mining, etc.)

Historique de l'IA

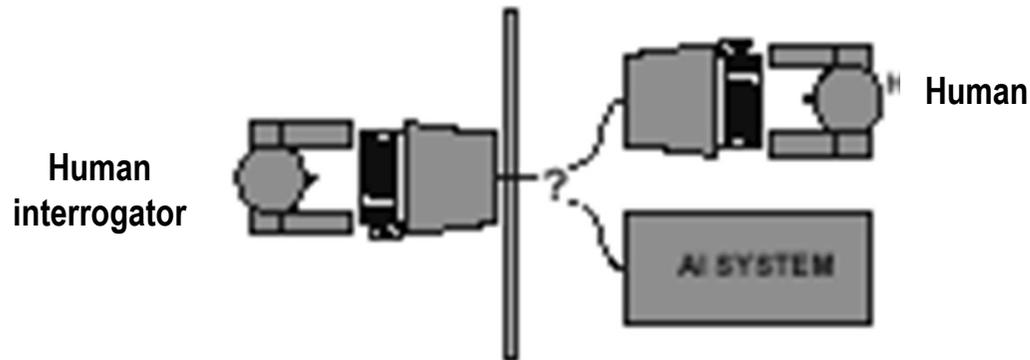
- ☞ Les principaux langages de l'intelligence artificielle
- ☞ Lisp (1960, J. McCarthy)
- ☞ Prolog (1973, A. Colmerauer), Prolog avec contraintes
- ☞ SmallTalk (1972, A. Kay)
- ☞ JAVA (1994) , C++, Scheme, ...

- Notion de Système Intelligent ou
Systèmes à base de connaissance

- Qu'est-ce qu'un Système intelligent ?

- Un programme construit pour :

- modéliser les compétences de résolution de problèmes des humains.
- avoir la même performance que les humains ([Test de Turing](#)).



- **Question:** comment se manifeste l'intelligence?
- On ne connaît pas au préalable le chemin de recherche de la solution;
- On ne connaît pas au préalable ni le temps ni le nombre d'opérations ou d'étapes pour arriver à la solution;
- A chaque étape on choisit la meilleure opération pour atteindre la solution;
- Exemple : jeu d'échec;.....

La machine modèle

- qu'elle comprenne un interlocuteur humain.
- qu'elle soit capable de réaliser les tâches que celui-ci lui demande.
- qu'elle réagisse avec « intelligence ».

- Informatique classique (calcul)

Données + algorithme = résultat

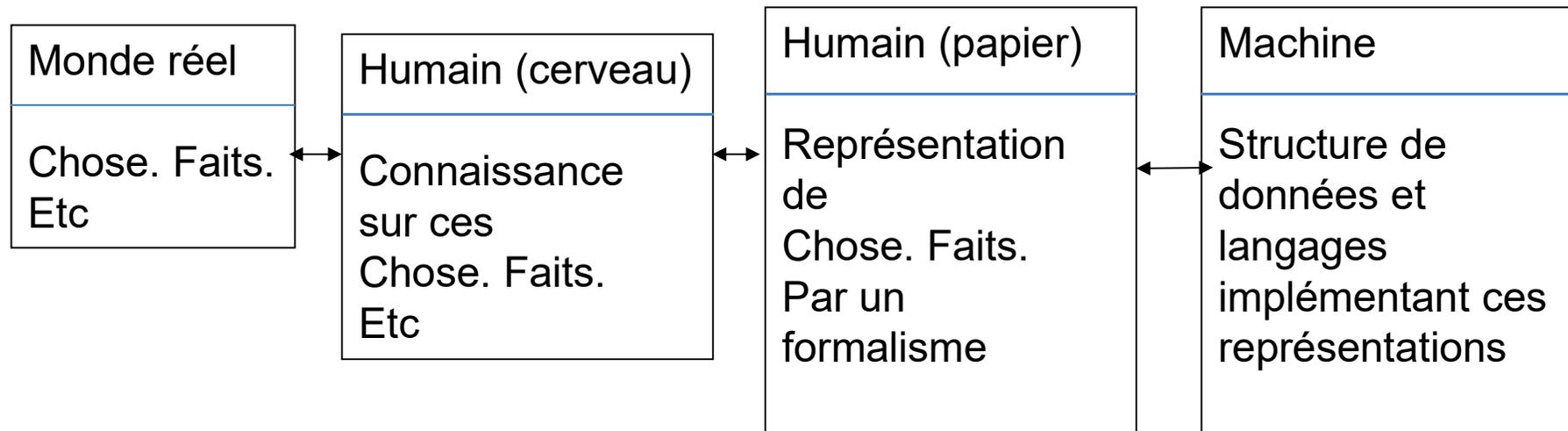
- Informatique intelligente

Connaissances+ raisonnement=resultat

Savoir (expertise) + (heuristique) = résultat

☞ L'un des objectifs majeurs de l'I.A est la résolution de problèmes complexes et dérivant de domaines variés.

☞ Pour résoudre un problème donné, on a souvent besoin de manipuler des connaissances du domaine concerné. Ces connaissances nécessitent des modèles et des **formalismes de représentation** capables de les rendre utilisables sur ordinateur.



- ☞ La représentation des connaissances n'étant pas le but ultime, il faut que les connaissances représentées puissent être manipulées par un **mécanisme de raisonnement** approprié capable de fournir une solution au problème posé. En effet, représentation et raisonnement sont, d'une manière ou d'une autre, très liés.
- ☞ Dans un système à base de connaissances on a donc besoin à la fois :
 - ☞ d'un formalisme pour représenter les connaissances et
 - ☞ d'un mécanisme de raisonnement appelé souvent « moteur d'inférence » qui est capable d'enchaîner des inférences sur les représentations de ces connaissances.

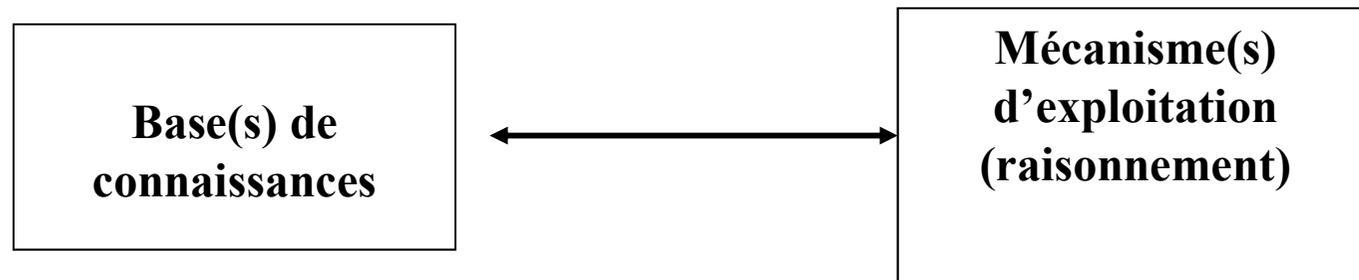


Schéma simplifié d'un système à bases de connaissances

Objectifs, attentes d'un SBC

☞ Inscrire les connaissances en tant que connaissance (pas seulement en tant qu'information) dans un système :

Pour « conserver » des savoirs, des savoir-faire et leur sémantique associée

☞ Disposer d'un « moteur » permettant d'enchaîner des inférences sur ces inscriptions de connaissances :

Pour « exploiter » les savoirs et savoir-faire ainsi « conservés »

Connaissance ? [1]

- ☞ Cognition : faculté de « connaître » -> activités mentales (perception, raisonnement, mémoire, représentation, apprentissage, langage, conscience, émotions, ...)
- ☞ Inférence : production d'une connaissance
 - Avec représentation de la connaissance (déclarative)
 - Sans représentation de la connaissance (incorporée)
- ☞ Raisonnement : enchaînement d'inférences avec un objectif

Connaissance ? [2]

- ☞ Démontrer de la connaissance => démontrer une capacité à mobiliser des informations pour agir ou produire d'autres capacités à agir (connaissances)
- ☞ Connaissance = information + mode d'emploi dans un contexte donné
- ☞ En IA: Connaissance = Information + Sémantique
- ☞ Pas de classement universel des différents types de connaissances

- **Connaissance**

- Donnée → information → connaissance
- La représentation des connaissances est le problème clé en IA.
- Les objets, actions, concepts, situations, relations, etc. sont représentés selon certains formalismes (cerveau vs. mémoire de l'ordinateur).

- **Connaissance**

- Différence entre donnée, information, connaissance :

- Une donnée transporte l'information. Ce sont des signaux non interprétés.

- Exemple. ! ...- - -... C

- L'information est une interprétation de la donnée.

- Exemple. (!, point d'exclamation), (...- - -..., SOS), (C, lettre) ou (C, note)

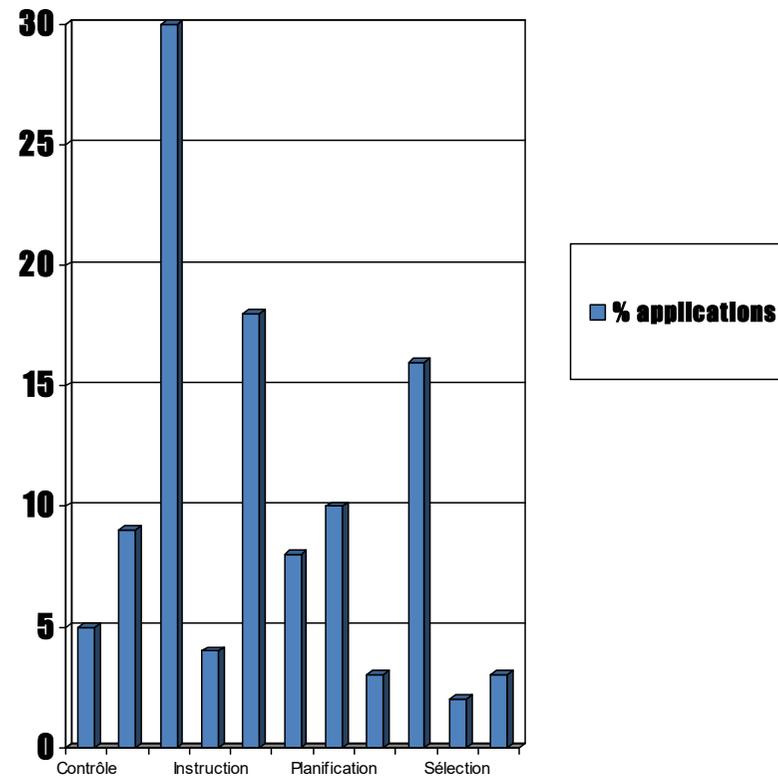
- La connaissance utilise l'information dans le cadre d'actions, dans un but précis. Les actions peuvent être la prise de décisions, la création de nouvelles informations, etc.

- Exemple.

- écrire un “!” pour marquer une exclamation en fin de phrase
 - si le signal ...- - -... reçu alors déclencher l'alerte et envoyer des secours
 - si C apparaît sur une partition alors la référence est la gamme de Do, jouer dans la gamme associée.

- Types de problèmes résolus par les SBC

- Contrôle
- Conception
- Diagnostic
- Instruction
- Interprétation
- Monitoring
- Planification
- Prédiction
- Prescription
- Sélection
- Simulation



- Pourquoi utiliser un SBC ?

- Remplacer un expert

- Automatiser une tâche routinière nécessitant un expert ;
 - Un expert quitte la compagnie ;
 - Besoin d'une expertise dans un environnement hostile.
 - *Exemples*
 - *Drilling advisor (Elf-Aquitaine) ;*
 - *Cooker advisor (Campbell Soup Company) : 95 % de réussite.*

- Assister un expert

- Améliorer la productivité ;
 - Gérer la complexité ;

- Utilisation des SBC

- Agriculture
- Affaires
- Chimie
- Communications
- Informatique
- Éducation
- Électronique
- Ingénierie
- Géologie
- Domaine juridique

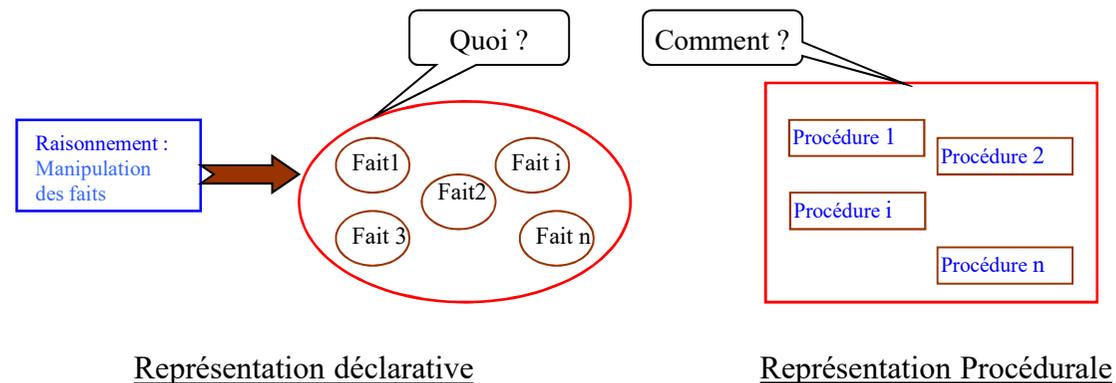
- Manufacture
- Mathématiques
- Médecine
- Météorologie
- Militaire
- Prospection et exploitation minière
- Production d'énergie
- Hydrologie / hydroélectricité
- Espace

1- Représentation des connaissances

I. Représentation des connaissances

Introduction

- ☞ En matière de représentation de connaissances, on distingue –historiquement– deux paradigmes distincts :
- ✓ Le paradigme revendiquant la séparation entre les connaissances et le raisonnement (i.e. la manipulation de ces connaissances) et qui stipule donc une **représentation déclarative** des connaissances;
 - ✓ Le paradigme dans lequel les connaissances et la manière de les manipuler sont inter-mêlées et qui opte donc pour une **représentation procédurale** des connaissances.



I. Représentation des connaissances

Introduction

- ☞ Il y a plusieurs raisons pour préférer la manière déclarative à la manière procédurale :
 - ✓ La représentation déclarative peut être modifiée facilement par ajout ou suppression de nouveaux faits.
 - ✓ Une même collection de faits peut être utilisée de différentes façons moyennant quelques simples modifications et ce, selon le problème à résoudre.
 - ✓ Mieux encore, la représentation déclarative est souvent extensible au-delà de ce qui est explicitement représenté et ce, parce qu'elle est 'liée' (malgré tout) à des processus de raisonnement adéquats qui permettent la dérivation d'une connaissance additionnelle.

I. Représentation des connaissances

Introduction

- ☞ La représentation procédurale a, toutefois, la particularité d'être indispensable à toute implémentation sur machine.
- ☞ En effet, il y a toujours des connaissances difficiles à exprimer de manière déclarative et même celles, de nature déclarative, doivent à un moment ou un autre être manipulées par une procédure opératoire qui représente le mécanisme de raisonnement utilisé
- ☞ Autrement dit, il est nécessaire de spécifier la manière procédurale (le comment) d'utilisation des connaissances déclaratives dans le but de trouver une solution au problème posé.
- ☞ Il en résulte qu'en I.A, on a souvent tendance à marier ces deux types de représentation avec, toutefois, une relative prédominance du déclaratif.

I. Représentation des connaissances

Introduction

- ☞ Au-delà de cette 'fausse' controverse « déclarative vs procédurale », la vraie difficulté réside dans le choix du formalisme à utiliser pour résoudre un problème relevant d'un domaine donné.
- ☞ En effet, il n'existe pas de formalisme "idéal" de représentation de connaissances pour tous les problèmes en I.A.
- ☞ Chaque type de problèmes requiert le choix d'un formalisme qui soit approprié à son domaine de connaissances.
- ☞ A ce niveau, les différents courants de l'IA et les différentes positions concernant la nature du raisonnement intelligent ont leur influence –parfois de manière implicite- sur le choix du formalisme.
- ☞ Par exemple, les logiciens vont toujours préférer les formalismes logiques alors que les 'imitateurs' du comportement humain vont plutôt préférer les réseaux sémantiques ou les frames voire les réseaux de neurones.

- **Représentation des connaissances**
 - C'est le transfert des connaissances d'un expert vers une machine → **Psychologie cognitive**
 - Changement de media
 - Changement de forme de représentation
 - C'est une tâche de modélisation linguistique :
 - puissance expressive
 - applicable pour le raisonnement
 - efficace

- Types de connaissance

Types de connaissances	
Connaissance procédurale	Règles, stratégies, agendas, procédures
Connaissance déclarative	Concepts, objets, faits
Méta-connaissance	Connaissances sur les autres types de connaissance, et comment les exploiter
Connaissance heuristique	Règles
Connaissance structurée	Ensemble de règles : relations inter-concepts, relations entre concept et objet

Les catégories des connaissances

6 catégories des connaissances qui sont :

- ✓ **Connaissances de définition**

Exemple : "Un triangle est un polygone ayant exactement trois cotés ".

- **Connaissances évolutives**

Exemple : " Ali est de 70 cm de taille " (aujourd'hui).

- ✓ **Connaissances incertaines**

Exemple : " Ahmed est né en 465 après J.C ".

- **Connaissances vagues**

Exemple : " les jeunes élèves sont turbulents ". (Jeunes =? turbulents = ?).

- **Connaissances typiques**

Exemple : " habituellement, chaque période d'enseignement est consacrée à une seule matière".

- **Connaissances ambiguës**

Exemple : "Avant le conseil de classe, le professeur savait que trois élèves redoubleraient". Est-ce le nombre global, ou bien 3 cas particuliers connus individuellement ? Est-ce trois exactement, ou au moins trois ?

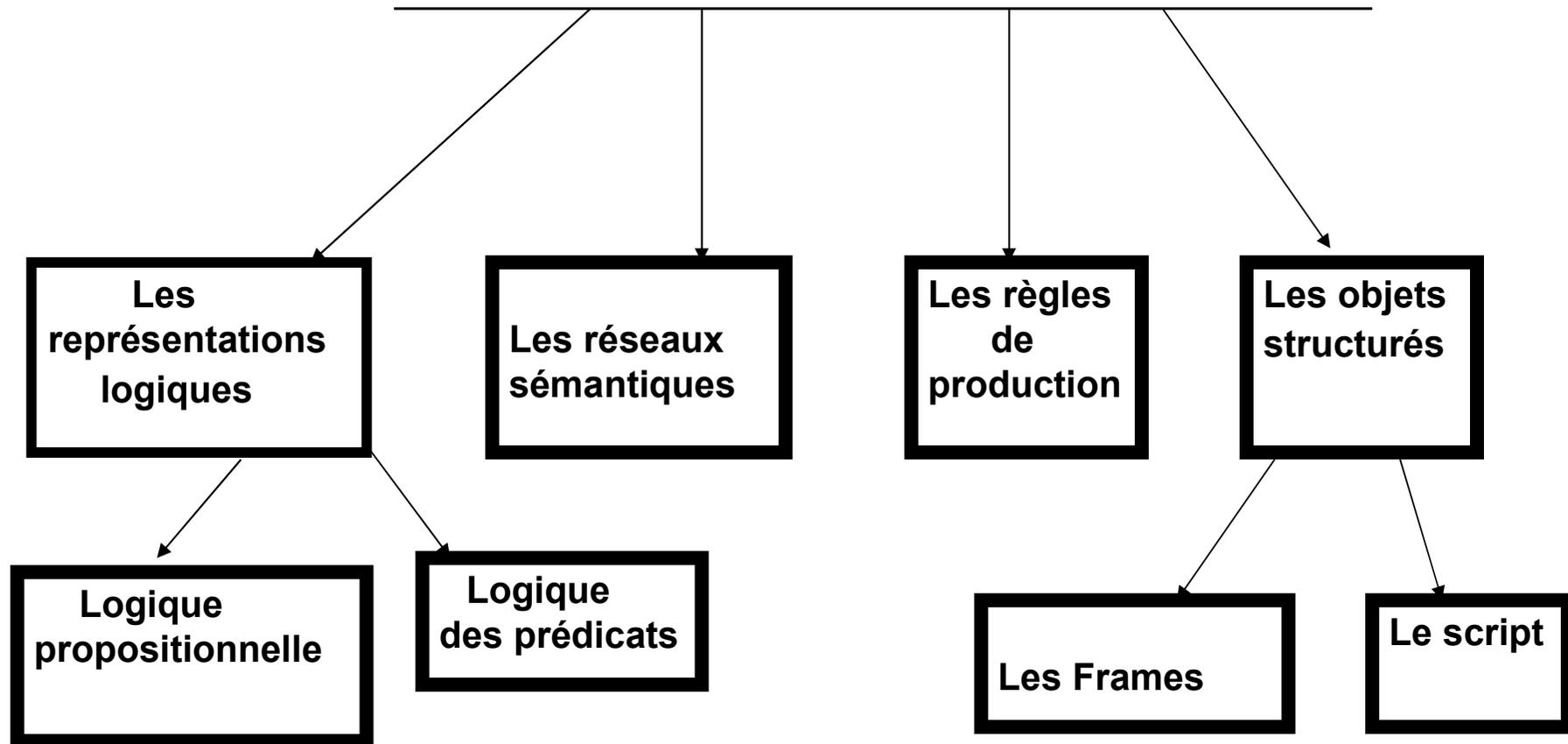
I. Représentation des connaissances

Choix d'un formalisme

Critères de choix d'un formalisme :

- ☞ Voici quelques critères de choix d'un formalisme de représentation qui sont assez objectifs :
- ✓ **Expressibilité** : la capacité à représenter toutes les connaissances pertinentes du domaine.
 - ✓ **Fidélité** : la capacité de 'mapper' facilement les connaissances du domaine avec leur représentation.
 - ✓ **Extensibilité** : la capacité à acquérir de nouvelles connaissances.
 - ✓ **Adéquation** ou 'correction' inférentielle : la capacité à manipuler les connaissances par un mécanisme de raisonnement 'valide'.
 - ✓ **Explication** : la capacité d'expliquer –aux humains- les raisonnements effectués sur les connaissances (plus précisément leur représentation).
 - ✓ **Efficacité calculatoire** : la capacité à guider le processus de raisonnement dans le choix des inférences à appliquer pour une résolution plus rapide.

Modes de représentation des connaissances



I. Représentation des connaissances

Choix d'un formalisme

- ☞ Le choix d'un formalisme pour la représentation des connaissances se fait en général parmi les trois catégories suivantes :
 - ✓ Les formalismes logiques ;
 - ✓ Les réseaux sémantiques ;
 - ✓ Les hiérarchies d'objets structurés.
- ☞ Ces formalismes entrent dans le cadre des représentations déclaratives, mais, ils font appel à des représentations procédurales lorsqu'il s'agit de manipuler les connaissances déclaratives pour trouver une solution à un problème donné.
- ☞ Il arrive aussi qu'une partie des connaissances du domaine soit à caractère particulièrement procédural. Certains formalismes permettent, dans ce cas, de les incorporer dans la représentation déclarative choisie sous forme d'attachement procédural.

I. Représentation des connaissances

Formalismes Logiques

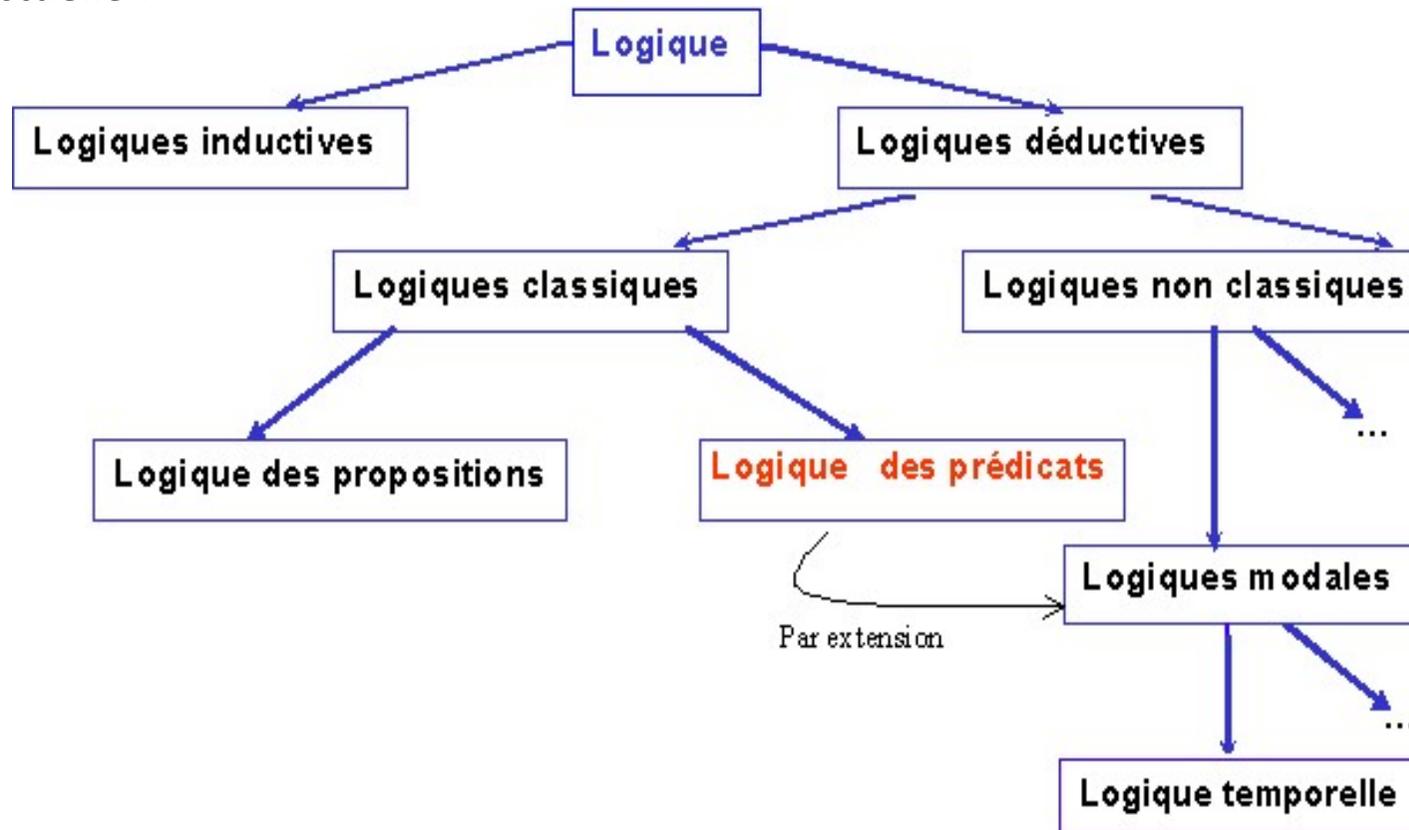
- ☞ Les formalismes logiques et surtout la logique des prédicats jouent un rôle important dans tout ce qui relève de la représentation des connaissances et ceci est dû à plusieurs aspects de ces formalismes :
 - ✓ D'abord, ils sont les premiers –chronologiquement parlant- formalismes utilisés en IA notamment pour les applications de démonstration automatique de théorèmes ;
 - ✓ Certains formalismes peuvent être vus comme une variante syntaxique d'une partie de la logique de prédicats ou l'une de ces extensions ;
 - ✓ Ils sont des formalismes purement syntaxiques dont la sémantique est rigoureusement définie ;
 - ✓ Ils bénéficient d'une base mathématique solide en termes de mécanismes de raisonnement qui procèdent uniquement par manipulation symbolique;
- ☞ On peut résumer ce qui précède en disant qu'ils servent de référence pour les autres formalismes : L'étude d'un formalisme n'est jamais complète sans une comparaison avec la logique des prédicats ou l'une de ses extensions.

I. Représentation des connaissances

Formalismes Logiques

Différentes 'Logiques' :

Le schéma suivant montre les différents types de logiques qui existent à l'heure actuelle :



I. Représentation des connaissances

Formalismes Logiques

Logiques classiques:

☞ La logique des propositions :

☞ La plus simple parmi les formalismes logiques.

☞ Elle souffre d'une limitation au niveau de son expressibilité (pas de quantification, pas de variables, etc.)

☞ Mais bénéficie de propriétés importantes en termes de décidabilité de ses mécanismes de raisonnement.

☞ La logique des prédicats :

☞ La logique des prédicats peut être vue comme une extension de la logique propositionnelle et elles forment ensemble ce qu'on appelle les logiques classiques.

1. Logique propositionnelle (ordre 0)

L'assertion 3 est supérieur à 2 est une proposition logique ayant la valeur vraie. Plusieurs connecteurs logiques sont utilisés pour combiner des propositions élémentaires, tel que : ET, OU, NON, \Rightarrow , \Leftrightarrow . La règle utilisée pour exploiter une base de connaissances en logique propositionnelle est dite « **règles de MODUS PONENS** ».

Elle déclare que :

Si x est vrai et si l'implication $x \Rightarrow y$ est vrai, alors y est vrai.

Cela se traduit formellement par:

$$(x, (x \Rightarrow y)) \Rightarrow y.$$

Exemple

Quand le ciel est bleu, le soleil brille ($x \Rightarrow y$)

Or le ciel est bleu (x)

Donc le soleil brille (y).

2. Logique des prédicats (ordre 1)

Certaines insuffisances exprimées par la logique des propositions telles que les relations entre les objets, la précision de la portée de certaines assertions (généralisation des relations), particularisation des objets sont prises en considération par la logique des prédicats.

Exemple

Assertion : si x est père de y et y est père de z alors x est grand-père de z.

La Formule Bien Formée :

$\forall x, \forall z, \exists y$ [père (y, x) et père (z, y)] \Rightarrow grand-père (z, x)

Une base de connaissances est un ensemble de formules bien formées (FBF) ainsi que des règles sémantiques qui relient ces FBF aux domaines d'application.

Parmi les règles d'inférences utilisées il y a :

- La règle de détachement ou MODUS PONENS donnée par :

Si (P(x) \Rightarrow Q(y)) et (P(x)) alors Q(y)

- La spécialisation universelle ou substitution qui à partir d'une FBF de la forme :

$(\forall x) (F(x))$ et de n'importe quel terme « a » permet d'obtenir la FBF $(F(a))$. En d'autres termes, toutes les occurrences de x dans F sont remplacées par « a »

$(\forall x F(x)) \Rightarrow F(a)$

TOUT être humain est mortel.
OR Socrate est un être humain.

$\forall X$ être-humain(X) \Rightarrow mortel(X)
être-humain(Socrate)
mortel(Socrate)



DONC Socrate est mortel.

I. Représentation des connaissances

Formalismes Logiques

Logique des prédicats :

☞ L'exemple de raisonnement 'classique' qui suit illustre bien le principe de formulation avec la logique des prédicats :

TOUT être humain est mortel.
OR Socrate est un être humain.



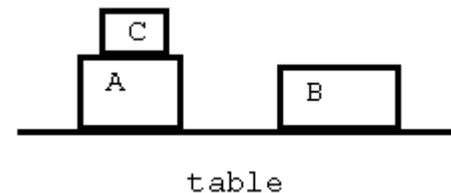
$\forall X$ être-humain(X) \Rightarrow mortel(X)
être-humain(Socrate)
mortel(Socrate)

DONC Socrate est mortel.

- **Logique (Calcul des propositions et prédicats)**

- On peut représenter la situation de la figure de la manière suivante en utilisant les prédicats 'sur', 'surtable', 'libre' :

- $\text{sur}(C,A)$
- $\text{surtable}(A)$
- $\text{surtable}(B)$
- $\text{libre}(C)$
- $\text{libre}(B)$



- Par ailleurs, à l'aide d'opérateurs de la logique du premier ordre, il est possible de définir de nouveaux prédicats: 'ôter', 'empiler' et de donner des équivalences :

- 1) $\text{libre}(x) \Leftrightarrow \neg (y \text{ sur}(y,x))$ (il n'existe pas de y sur x)
- 2) $\text{sur}(y,x) \wedge \text{oter}(y,x) \Rightarrow \text{libre}(x) \wedge \neg \text{sur}(y,x)$
- 3) $\text{libre}(x) \wedge \text{libre}(y) \wedge \text{empiler}(x,y) \Rightarrow \text{sur}(x,y)$

- Il est possible de donner un but à atteindre (par un robot) de la même façon.

- Exemple : $\text{sur}(A,B)$. A partir de cette information et de la règle 3, le système déduit $\text{libre}(A)$, $\text{libre}(B)$ et $\text{empiler}(C,A)$. La règle 2 conduit à l'action $\text{ôter}(C,A)$.

- Exemple de problèmes

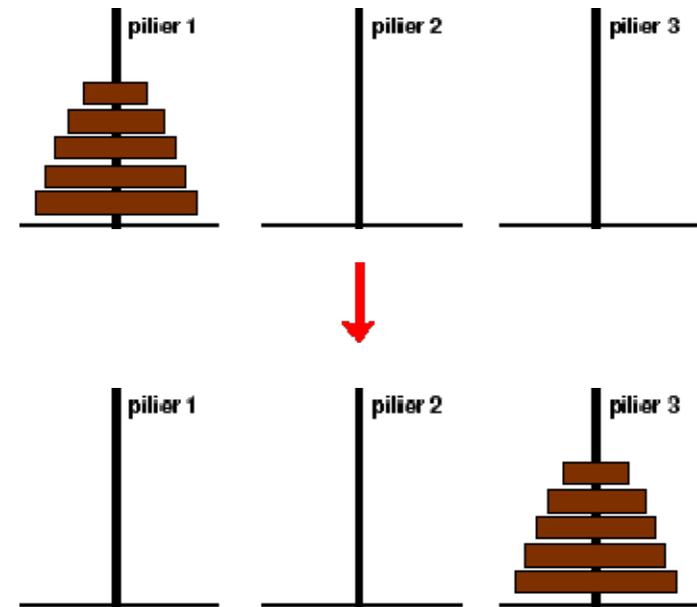
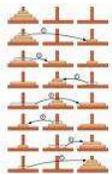
Les tours de Hanoi

- L'univers du problème consiste en un ensemble de trois piliers notés de la gauche vers la droite, p1, p2 et p3, et de deux disques, notés d1 et d2, où le premier est plus petit et posé sur le second.

- Le problème consiste à faire passer les deux disques, d1 et d2, du pilier p1 au pilier p3 en se servant du pilier intermédiaire p2 en respectant la règle ci-dessous :

Le petit disque d1 ne peut jamais être sous la grand disque d2.

- Le problème se généralise à un nombre quelconque de disques.



- Logiques de description

- Les RS mélangeaient plusieurs types de liens. Les LD donnent une place à part entière aux relations entre concepts (relation \rightarrow rôle).

- Exemple

- Humain \equiv Homme \sqcup Femme
 - Humain $\sqsubseteq \exists \text{est_parent}$
 - Femme \equiv Personne \sqcap Féminin
 - Parent \equiv Personne $\sqcap \forall a_enfant. \text{Personne}$
 - Gandmère \equiv Femme $\sqcap \forall a_enfant. \text{Parent}$

I. Représentation des connaissances

Formalismes Logiques

Logiques non classiques:

- ☞ Plusieurs genres de connaissances sont difficilement représentables en logique des prédicats surtout lorsqu'on désire rester le plus fidèle possible à la réalité. Voici des exemples de telles connaissances (exprimées en langage naturel) :
 - ✓ « Il pleut assez souvent en hiver » : comment exprimer la fréquence dans le temps d'un évènement ? et surtout comment s'en servir dans le raisonnement ?
 - ✓ « Moi, je pense que le Couscous est le plus délicieux repas au monde, mais Fatima pense que le Chawarma est le meilleur de tous » : comment exprimer différentes croyances ?
 - ✓ « Ali a vu Hassan en premier, je l'ai vu juste après alors que Samir ne l'a vu que le lendemain » : Comment représenter la chronologie des évènements ?
 - ✓ « Il fait très noir la nuit, sauf en jours de pleine lune » : comment exprimer des exceptions ?
- ☞ Pour augmenter l'expressibilité de la logique des prédicats, d'autres logiques ont vu le jour, comme par exemple : La logique probabiliste, la logique des croyances, la logique temporelle, etc.

I. Représentation des connaissances

Formalismes Logiques

Logiques non monotones:

- ☞ Il existe également d'autres logiques dites **non monotones** qui permettent de représenter les connaissances relatives à des domaines où l'information est à la fois **incomplète** et évolutive.
- ☞ Elles violent le principe de la monotonie selon lequel, si S et S' sont deux ensembles de formules et si S est inclus dans S' , tout théorème démontrable à partir de S est également démontrable à partir de S' .
- ☞ Dans les logiques non-monotones, il se peut qu'une formule qui a été démontrée comme étant 'valide', ne le reste plus lorsque les faits connus comme vrais sont augmentés.
- ☞ Le raisonnement par-défaut est un exemple de raisonnement non-monotonique. Il est utilisé lorsque les connaissances actuelles sont incomplètes et que certaines connaissances par-défaut (sorte d'estimations) peuvent être utilisées pour les compléter jusqu'à preuve de leur invalidité.

I. Représentation des connaissances

Formalismes Logiques

Logiques non monotones:

- ☞ Les systèmes de démonstration automatique basés sur de tels raisonnements non-monotoniques sont bien évidemment plus complexes à manier que les systèmes monotoniques.
- ☞ En effet, il faut à chaque nouveau fait ajouté (respectivement, supprimé) :
 - ✓ Chercher quels faits ont été supposés vrais justement en se basant sur l'absence (respectivement, l'existence) de ce fait nouvellement ajouté et qu'il faut donc supprimer de la base de faits;
 - ✓ Chercher quels faits doivent être supprimés (respectivement, ajoutés) à leur tour puisque leur démonstration repose sur les faits qui viennent d'être supprimés (respectivement, le fait qui vient d'être ajouté).

I. Représentation des connaissances

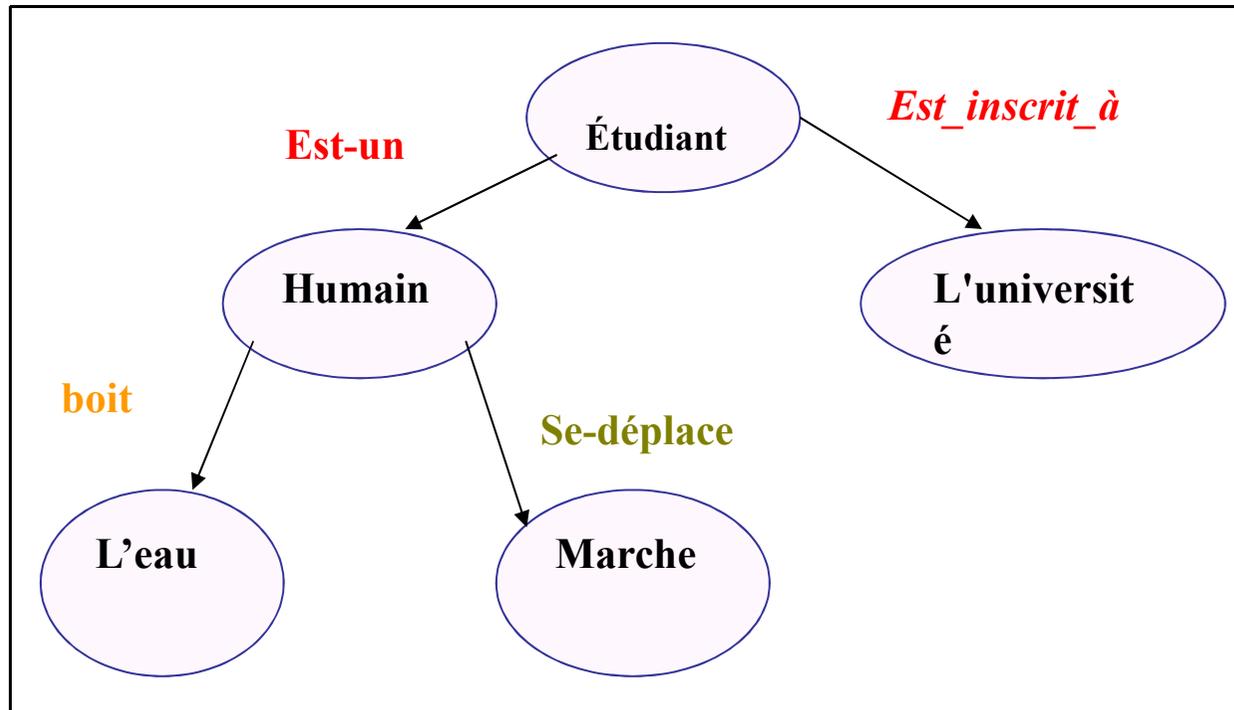
Formalismes Logiques

Quelle Logique ?

- ☞ Comme conclusion, on peut dire que l'utilisation de la logique des prédicats peut être extrêmement utile dans plusieurs applications comme celles de démonstration de théorèmes, surtout avec l'exploitation des techniques heuristiques et autres techniques qui améliorent son efficacité.
- ☞ Toutefois, il existe des genres de connaissances qui sont difficilement 'formulables' en logique de prédicats : la notion de temps, d'incertitude et de croyance n'en sont que des exemples et il en existe bien d'autres.
- ☞ Les logiciens ont trouvé une solution à cette limitation en utilisant d'autres logiques qui sont pour la plupart des extensions de la logique des prédicats.
- ☞ Par ailleurs, on peut penser à représenter et exploiter des connaissances sur les connaissances (méta-connaissances) ou à utiliser des logiques de deuxième ordre pour raisonner sur les prédicats eux-mêmes (en permettant que les quantifications portent sur les prédicats).
- ☞ Il faut, toutefois, garder à l'esprit que lorsqu'on augmente l'expressibilité du formalisme logique utilisé, on perd malheureusement en simplicité et, parfois, en efficacité aussi.

I.3. Paradigmes de Représentation des connaissances

3. Les réseaux sémantiques



Origine

Réflexion sur la mémoire associative humaine, le langage

I. Représentation des connaissances

Réseaux sémantiques

- ☞ L'utilisation des réseaux sémantiques comme formalisme de représentation de connaissances remonte aux travaux du linguiste Quillian (en 1968) sur la mémoire sémantique. En effet, les réseaux sémantiques sont très utilisés dans les travaux sur le traitement et la compréhension des langages naturels.
- ☞ Ce formalisme a comme caractéristique principale le fait que la signification d'un concept dépend du réseau sémantique dans lequel il s'insère et de ses relations avec les autres concepts de ce réseau.
- ☞ Dans ce formalisme, les concepts du domaine à modéliser sont représentées par des nœuds et les relations entre concepts par des arcs étiquetés interconnectant les nœuds associés à ces concepts.
- ☞ Il peut être défini comme étant un formalisme de représentation structurée de connaissances sous forme de graphe sémantique.
- ☞ On rappelle qu'un graphe est une structure mathématique où nœuds et arcs n'ont pas de signification particulière alors que dans un réseau sémantique ils ont une signification spécifique d'où l'utilisation du mot sémantique.

WordNet

Célèbre réseau sémantique

Noeuds : « synsets »

1. *couillon - gogo - naïf - pigeon*

2. *bar - loup - loup de mer - perche de mer*

3. *bar - bistro - brasserie - café - estaminet*

Une entrée = un ensemble de synonymes (**synset**)

Membres d'un synset

- lemmes et non formes fléchies

- mots et non tokens (*loup de mer* : mot composé)

Définitions informelles

Any of various mostly cold-blooded aquatic vertebrates usually having scales and breathing through gills

WordNet

Anglais

Version 3.0 : 120 000 synsets

Miller, 1995 - Fellbaum, 1998

Le réseau sémantique le plus utilisé au monde

Développement à partir de 1985 - Première version 1991

4 sous-réseaux : noms, verbes, adjectifs, adverbes

La granularité de WordNet est beaucoup plus fine, parfois trop

Ex. : 4 sens pour *tribe* "tribu"

WordNet

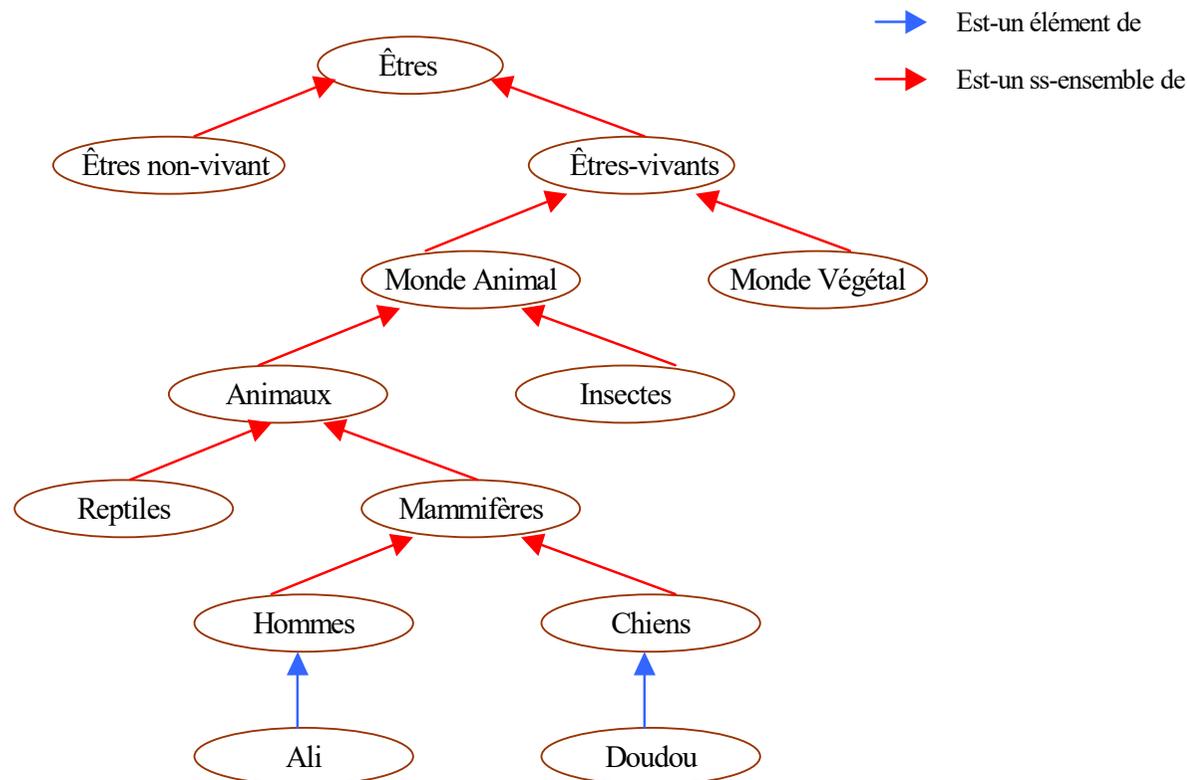
Principales relations entre synsets

sorte de	V/V	<i>exhale/breathe; inhale/breathe</i>
sorte de	N/N	<i>cat/feline</i>
est un	N/N	<i>Eiffel Tower/tower</i>
partie	N/N	<i>France/Europe</i>
membre	N/N	<i>France/European Union</i>
similaire	A/A	<i>dying/moribund</i>

I. Représentation des connaissances

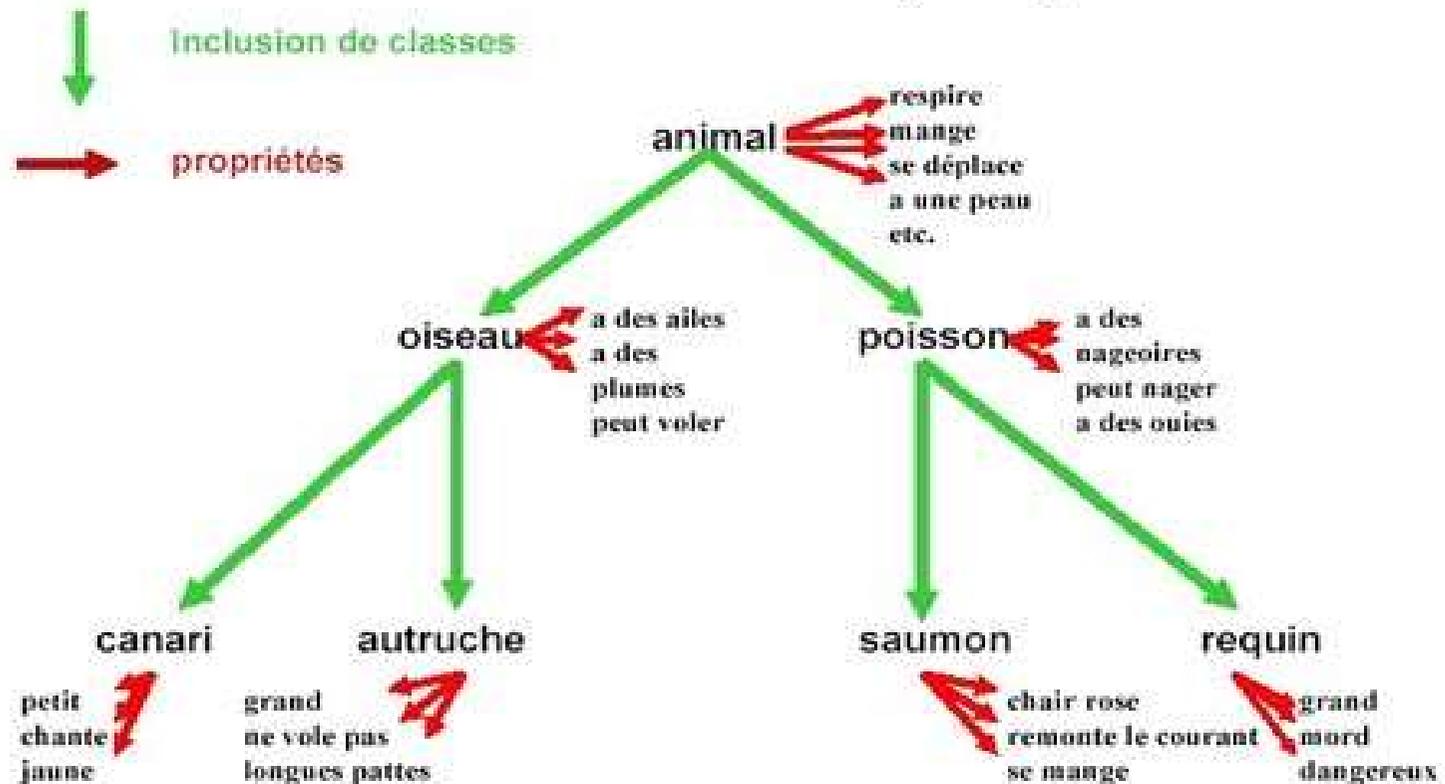
Réseaux sémantiques

☞ Parmi les concepts qu'un réseau sémantique sait très bien modéliser on trouve les concepts fortement hiérarchisés où les relations entre concepts sont de type : est-un sous-ensemble de (arc isa) ou est-un élément de, comme ceci est illustré dans l'exemple suivant :



Réseau sémantique

Collins & Quillian (1969)



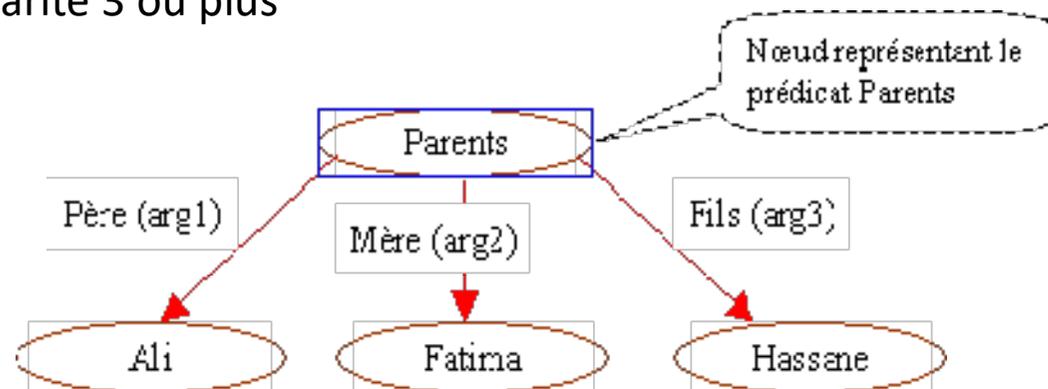
I. Représentation des connaissances

Réseaux sémantiques

Types d'arcs

☞ Les types d'arcs les plus utilisés dans les réseaux sémantiques sont :

- ✓ Est-un- élément
- ✓ Est-un- sous-ensemble
- ✓ Partie de (constituant, attribut)
- ✓ Relations prédicatives d'arité 3 ou plus
- ✓ Relations spatiales
- ✓ Relations temporelles
- ✓ Fonction
- ✓ ...



I. Représentation des connaissances

Réseaux sémantiques

Types d'inférences

- ☞ Le raisonnement dans un réseau sémantique se base essentiellement sur la recherche de chemin le long des arcs afin de trouver de nouvelles associations entre les nœuds (concepts).
- ☞ Il exploite trois grands types d'inférence:
 - ✓ Inférence par héritage
 - ✓ Inférence par composition de relations
 - ✓ Inférence par propagation d'activation

I. Représentation des connaissances

Réseaux sémantiques

Types d'inférences

☞ Inférence par héritage :

- ✓ Elle permet l'héritage de propriétés des nœuds 'pères' vers les nœuds 'fils' dans le cas où ces nœuds sont liés par des arcs 'est-un' (on parle d'une hiérarchie est-un). Par exemple, un chien hérite toutes les propriétés d'un être-vivant.
- ✓ Elle permet donc une réduction des connaissances explicitement représentées (il est inutile de répéter pour chaque nœud les propriétés des nœuds pères).
- ✓ Il faut, toutefois, gérer les exceptions concernant les cas où des propriétés des nœuds pères ne sont pas héritées par certains nœuds fils. Par exemple : Ali est un homme mais il ne doit pas hériter de la propriété d'avoir quatre pattes du concept père Animaux.
- ✓ En effet, les propriétés associées aux nœuds de plus haut niveau doivent être vues comme des propriétés par défaut qui peuvent être annulées par des propriétés plus spécifiques associées à des nœuds de plus bas niveau dans la hiérarchie.
- ✓ D'une manière générale, il faut définir les règles d'héritage afin d'éviter des inférences erronées.

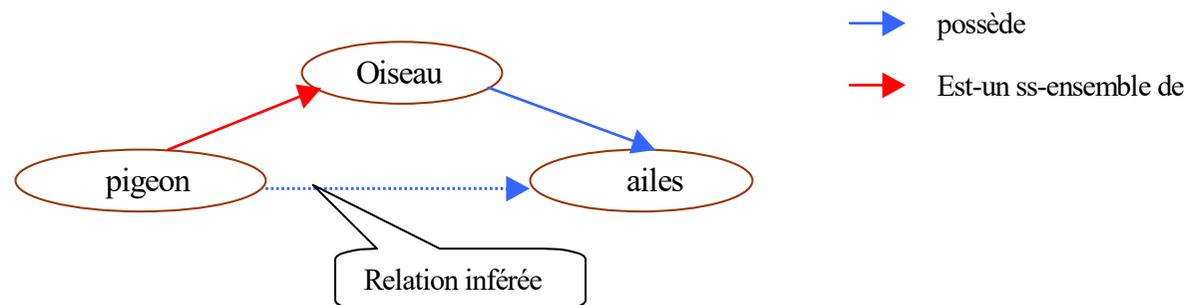
I. Représentation des connaissances

Réseaux sémantiques

Types d'inférences

☞ Inférence par composition de relations :

- ✓ Le cas le plus typique est le cas des relations transitives, où la composition n fois d'une telle relation donne la relation de départ.
- ✓ Par exemple, à partir d'un réseau sémantique où les relations 'doigt *est-une-partie* de Main' et 'Main *est-une-partie* de Corps' sont représentées par des arcs, on peut inférer la relation 'doigt *est-une-partie* de Corps'.
- ✓ De manière plus générale, on peut inférer l'existence d'une relation r (donc d'un arc) entre deux nœuds $n1$ et $n2$ s'il existe un chemin entre $n1$ et $n2$ qui fait intervenir k relations $r1, \dots, rk$ telles que : $r1 \circ r2 \circ \dots \circ rk = r$.



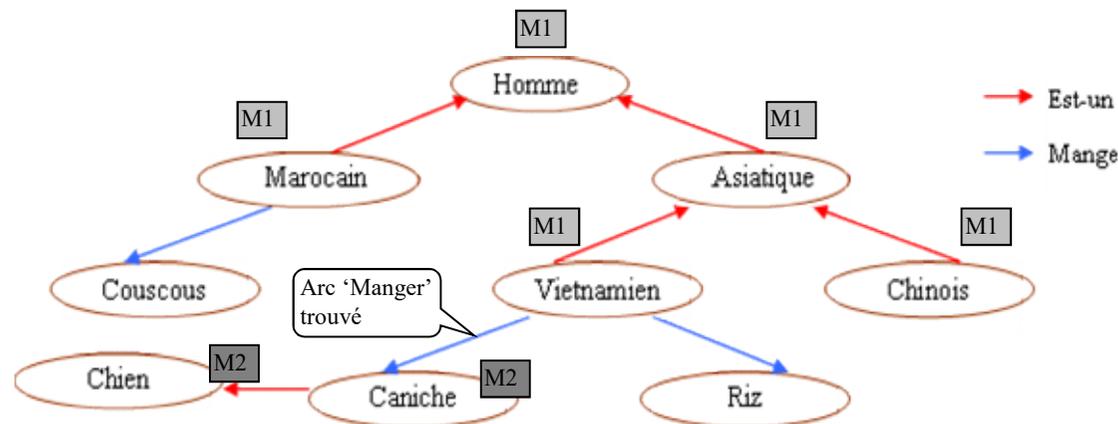
I. Représentation des connaissances

Réseaux sémantiques

Types d'inférences

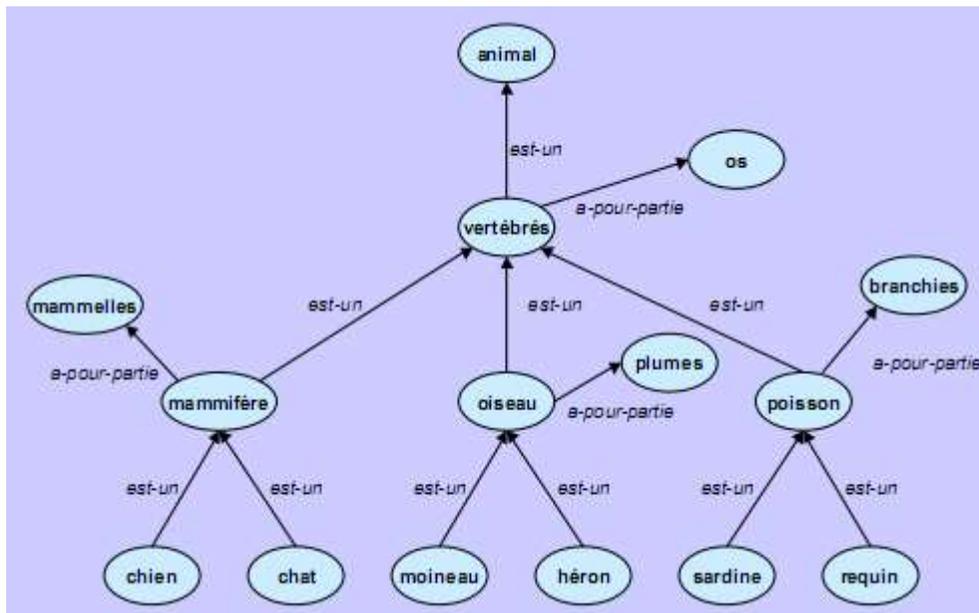
☞ Inférence par 'propagation d'activation' :

- ✓ Cette technique date des premiers travaux de Quillian sur les réseaux sémantiques. Son objectif est d'inférer une relation entre deux nœuds (ou plusieurs), en se basant sur les relations (arcs) existantes entre ces nœuds et d'autres nœuds du réseau.
- ✓ Par exemple, on désire savoir s'il existe des hommes qui mangent des chiens et on a le réseau suivant :



Héritage et inférence

- Les connaissances peuvent être déduites par transitivité dans un arbre hiérarchique



Un moineau est un oiseau qui est lui-même un vertébré

Un oiseau a des plumes -> un moineau a des plumes

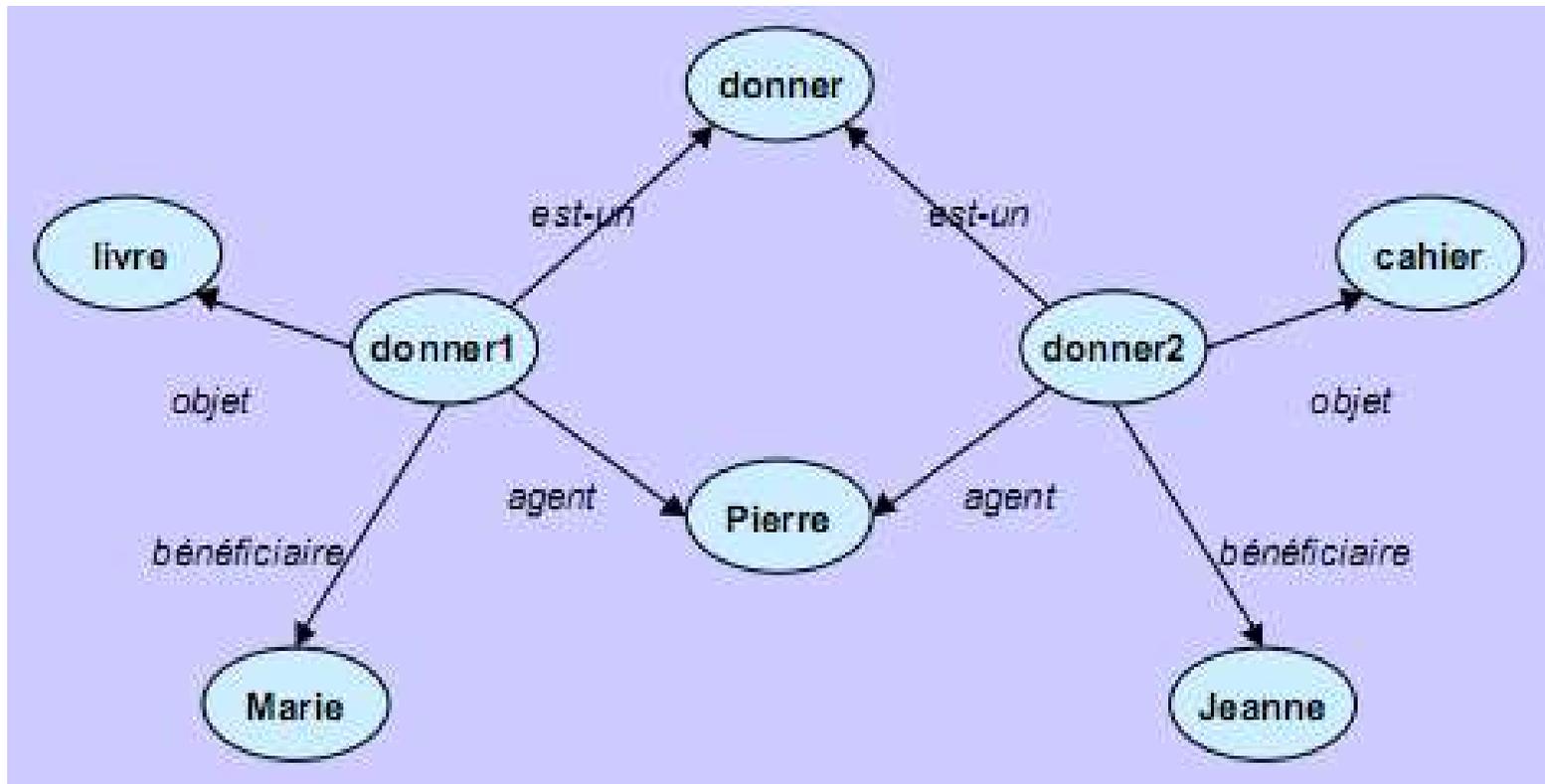
Inférence appelée héritage

Événements particuliers

Instances

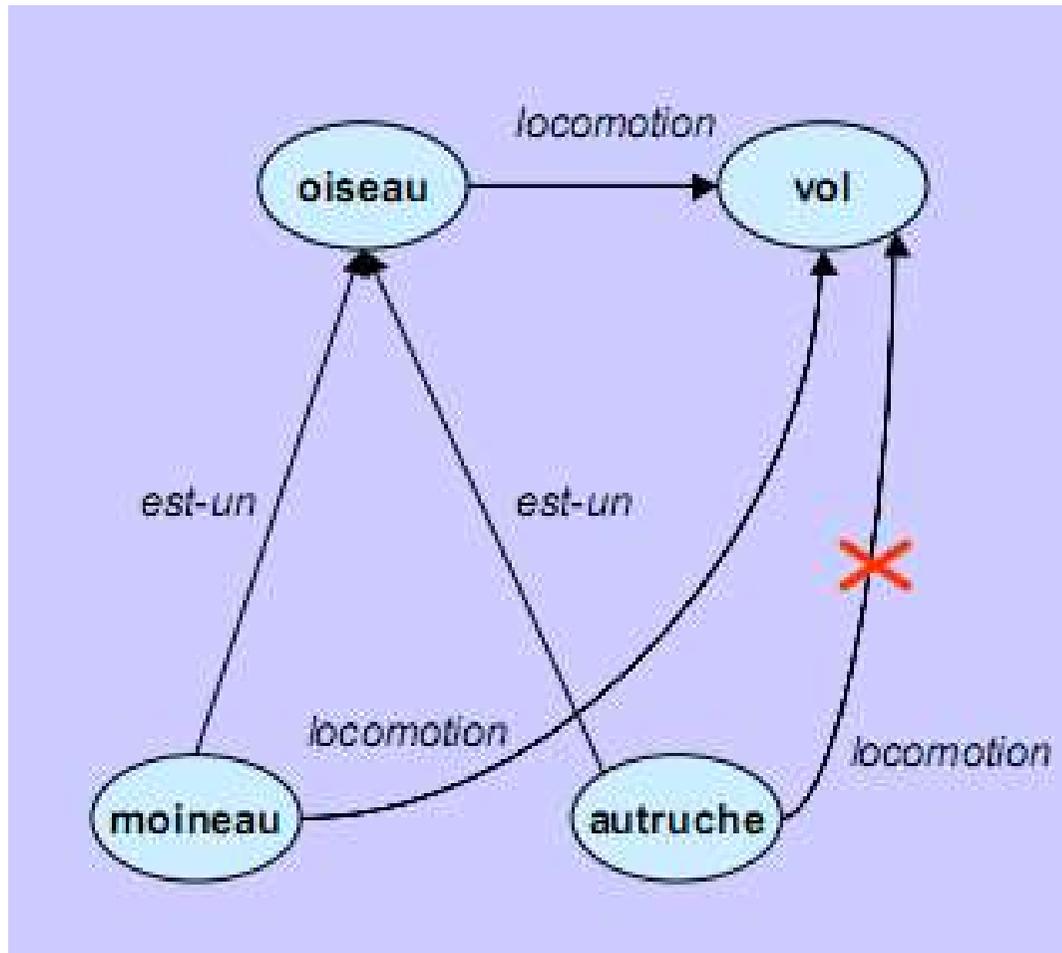
Représentation de la phrase :

Pierre donne un livre à Marie et un cahier à Jeanne



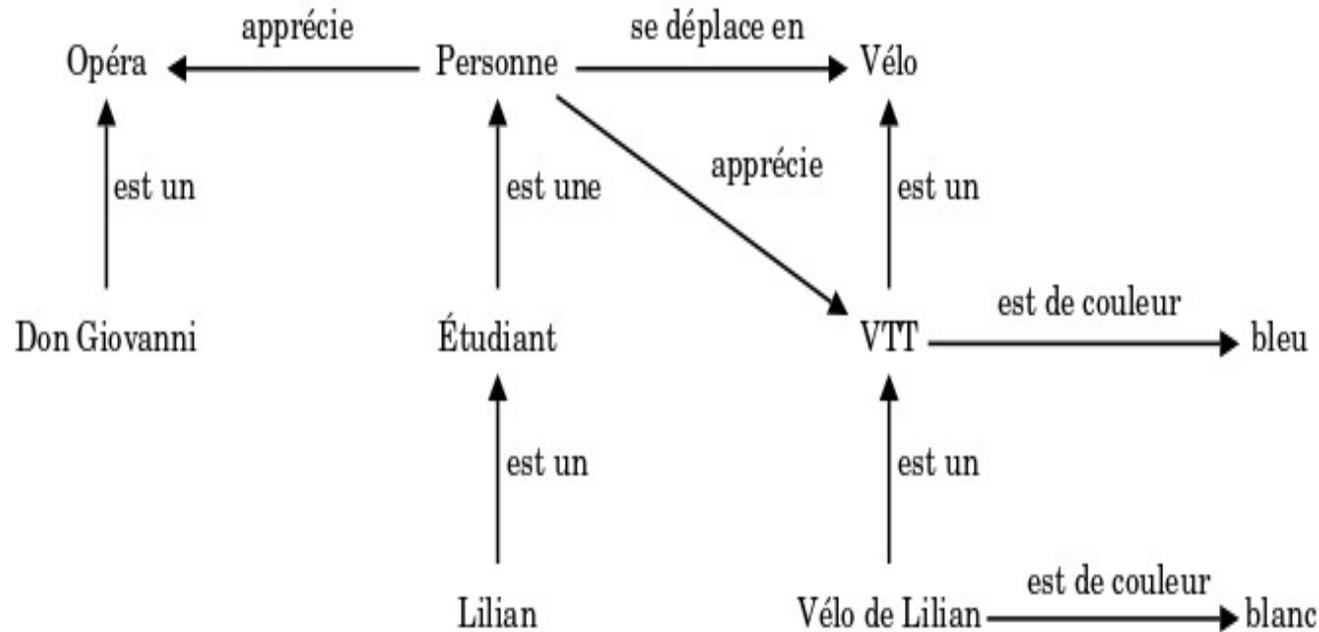
Les exceptions

Le raisonnement par défaut



Tous les
oiseaux volent
par défaut

Les autruches
elles ne volent
pas



- Quels sont les concepts ? Quels sont les objets ?
- Quelle est la couleur du vélo de Lilian, expliquer votre réponse ?
- Peut-on dire que Lilian apprécie l'opéra *Don Giovanni*, expliquer pourquoi ?

Transformer ce réseau en logique de prédicat.

Paradigme 3: Les règles de production

- Connaissance servant à faire le lien entre des informations connues et d'autres informations que l'on peut déduire ou inférer.
 - Exemple.
 - *Si <balle, couleur, rouge> alors j'aime la balle*
 - *Si j'aime la balle alors j'achète la balle*
- Peut exécuter des procédures.
 - *Si délai<30 et âge_étudiant<28 ="oui"
et présent_communication = "oui"
alors réduction_congrés = 50%*

- Règles de production

- Représentent des formes de connaissances variées :

- Relation

- Si batterie morte
alors l'auto ne démarrera pas

- Recommandation

- Si l'auto ne démarre pas
alors prendre un taxi

- Directive

- Si l'auto ne démarre pas & le système d'alimentation en essence est ok
alors vérifier le système électrique

- Stratégie

- Si l'auto ne démarre pas
alors vérifier le système d'alim. en essence puis le système électrique

- Heuristique

- Si l'auto ne démarre pas & l'auto est une Ford de 1962
alors vérifier le radiateur

- Règles de production avec variables
 - Réaliser la même opération sur un ensemble d'objets.
 - Exemple.
 - Si ?x est employé & ?x âge > 65
 - alors ?x peut prendre sa retraite

- **Méta-règles**

- Traduisent une connaissance sur l'utilisation et le contrôle de la connaissance du domaine.
- Disent comment utiliser les autres règles.
- Exemple.

Si auto ne démarre pas & système électrique normal
alors exploiter les règles concernant le système d'alim. en ess.

- **Règles incertaines**

- Traduisent des associations incertaines entre prémisses et conclusions.
- Exemple.

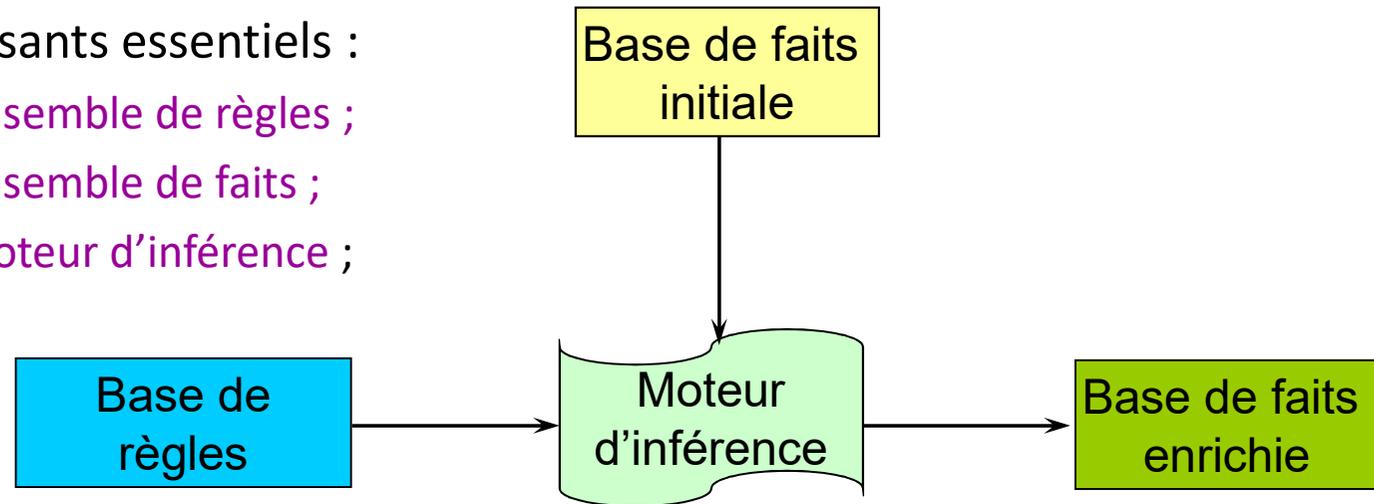
Si inflation élevée
alors taux d'intérêt élevé CF=0.8

Systeme à base de règles

- Raisonement à base de règle

- 3 composants essentiels :

- un ensemble de règles ;
- un ensemble de faits ;
- un moteur d'inférence ;



- Avantages :

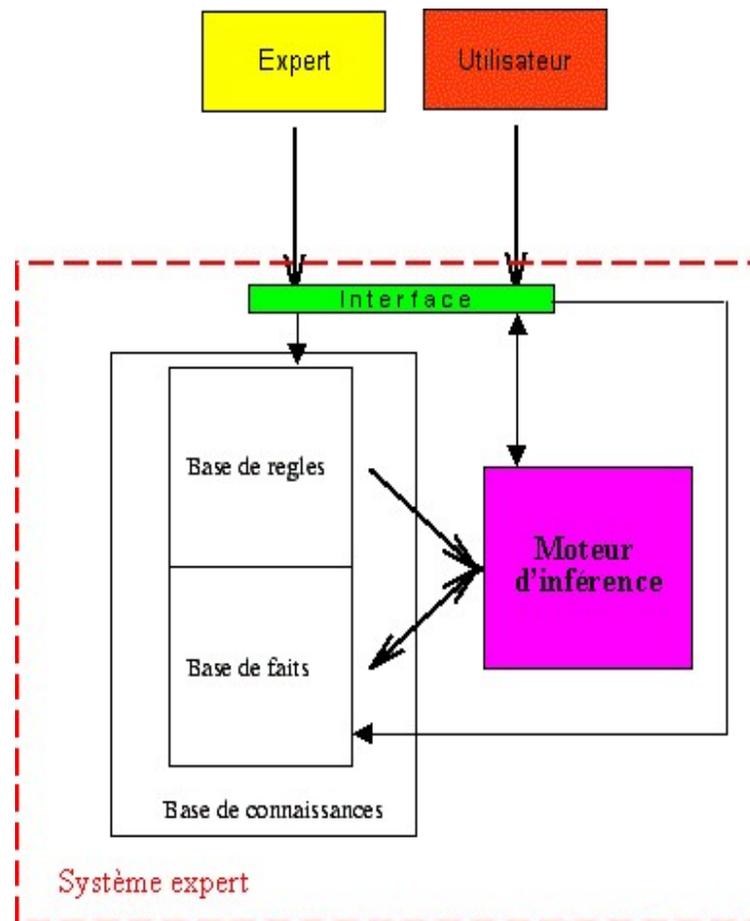
- Facile à comprendre, naturel, modulaire.

- Utilisation :

- Gérer les problèmes de contrôle.

Systeme à base de regles

☞ Tout SBC doit suivre normalement le principe de fonctionnement suivant :



Systeme à base de règles

- Raisononnement à base de règle
 - Principe de fonctionnement :
 - Trouver parmi les règles celles candidates
 - Choisir une de ces règles
 - Exécuter la règle
 - Résolution de conflits
 - Si plusieurs règles satisfaites, choisir laquelle déclenchée
 - Critère d'arrêt
 - Pour déterminer la fin du processus d'inférence (plus aucune règle n'est déclenchée, une solution acceptable a été trouvée, impossibilité de trouver une solution, etc.)

Systeme à base de règles

- Base de règles

- La base de règles rassemble la connaissance et le savoir-faire de l'expert.
- Elle n'évolue donc pas au cours d'une session de travail.
- Une règle se présente sous la forme : **si X alors Y**
 - X est la prémisse
 - C'est une conjonction de conditions, i.e. une suite de comparaison d'attributs et de valeurs à l'aide d'opérateurs.
 - Y est la conclusion
 - La conclusion est une affectation.
- Exemple base de règles
 - si l'âge du patient < 18 et
 - si il a de la fièvre > 39° et
 - si présence d'un germe X
 - alors le patient a peut-être une méningite

Systeme à base de règles

- Base de faits

- La base de faits constitue la mémoire de travail du SBC.
- Elle est variable au cours de l'exécution et vidée lorsque l'exécution se termine.
- Au début de la session, elle contient tout ce qu'on sait à propos du cas examiné avant toute intervention du moteur d'inférence.
- A la fin, elle est complétée par les faits déduits par le moteur ou demandés à l'utilisateur.
- Exemple base de faits
 - Âge est 6
 - Fièvre est 40°
 - Germe X
 - Sexe féminin

Systeme à base de règles

- **Moteurs d'inférence**

- Il existe de nombreux types de moteurs, capables de traiter différentes formes de règles logiques pour déduire de nouveaux faits à partir de la base de connaissance.
- On distingue 3 catégories, basées sur la manière dont les problèmes sont résolus :
 - les moteurs à « *chaînage avant* »
 - les moteurs à « *chaînage arrière* »
 - les moteurs à « *chaînage mixte* »
- Certains moteurs d'inférence peuvent être partiellement pilotés ou contrôlés par des méta-règles qui modifient leur fonctionnement et leurs modalités de raisonnement.

- Mécanismes d'inférence

- Modus Ponens (raisonnement direct)

$((A \rightarrow B) \text{ ET } A) \rightarrow B$

règle fait

Permet de déduire que B est VRAI si la règle de production est vraie ET le fait A est VRAI

Forme la base des systèmes à base de règles

Si ma batterie est à plat

Alors ma voiture ne partira pas

– Modus Tollens (raisonnement indirect)

$((A \rightarrow B) \text{ ET } B) \rightarrow A$
règle action

Permet de déduire que NON A est VRAI si la règle de production est vraie ET l'action NON B est VRAI

Rarement utilisé même si fait appel au sens commun

Sij'ai la tourista

Alors je suis allé en Amérique du Sud

II. Système à base de règles

Moteur d'inférence

Cycle de travail

- ☞ Le moteur d'inférence (appelé parfois moteur de résolution) est le cœur d'un SE.
- ☞ Il peut être indépendant du domaine d'application et il doit être capable d'exploiter la base des connaissances afin de résoudre les problèmes posés par les utilisateurs.
- ☞ En effet, il doit être capable de voir quelles sont les règles applicables au vu de l'état courant de la base des faits et des faits à établir (exprimant les problèmes /questions posés par l'utilisateur).
- ☞ Ce processus de détection des règles applicables fait partie du cycle de travail d'un moteur d'inférence et particulièrement de sa phase d'évaluation.
- ☞ La deuxième phase de ce cycle est la phase d'exécution qui, comme son nom l'indique, concerne l'exécution de la partie 'action' des règles sélectionnées lors de la phase d'évaluation et donc la modification de la base de faits.

II. Système à base de règles

Moteurs d'inférence

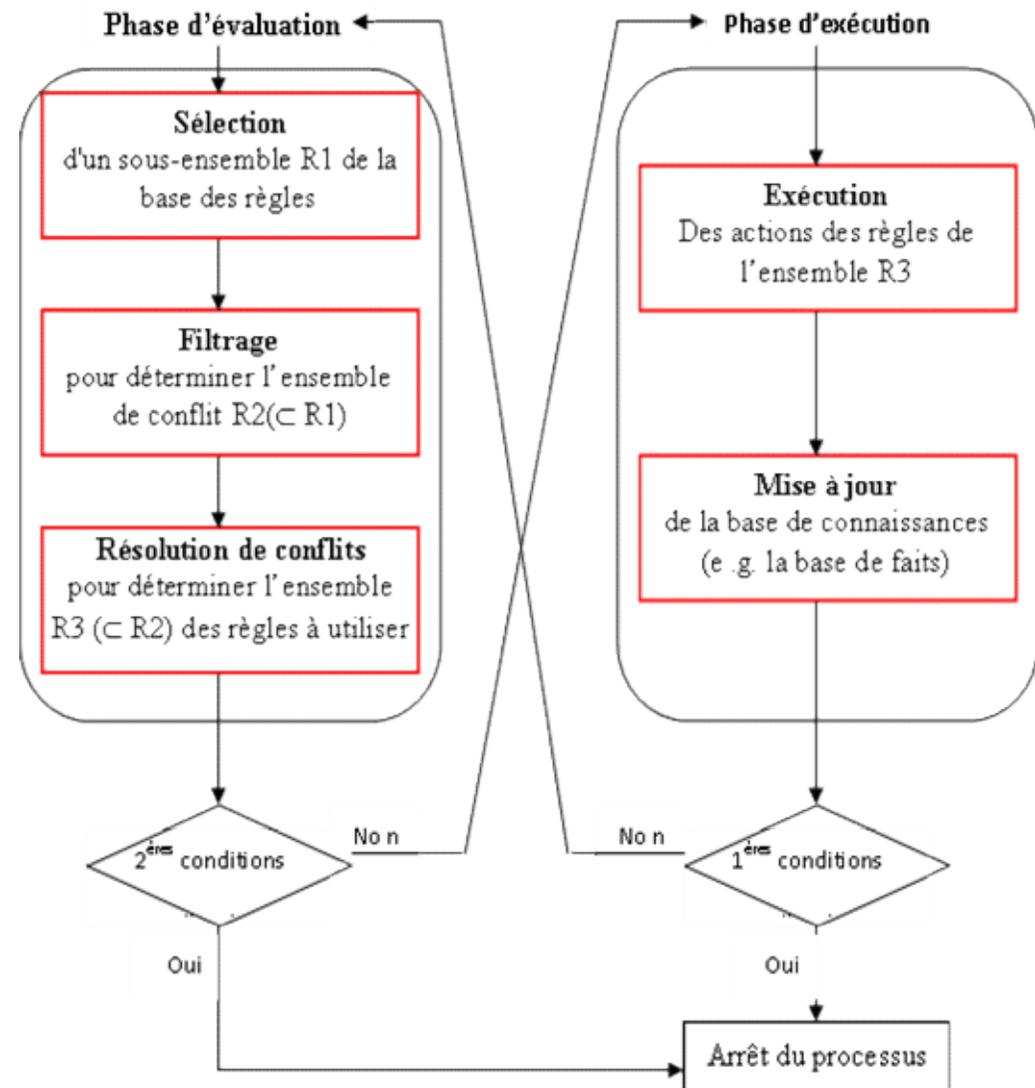
Cycle de travail

- ☞ La phase d'évaluation s'effectue généralement en trois étapes :
 - ✓ La Restriction : cette étape optionnelle concerne une première sélection d'un sous-ensemble de règles dans la base des règles qui –à priori- mérite d'être considéré dans l'étape suivante. Ce choix dépend de l'état courant de la base de faits.
 - ✓ Le filtrage (pattern matching) : Parmi l'ensemble retenu à l'étape précédente, cette étape retient un sous-ensemble, dit de conflit, qui contient seulement les règles effectivement applicables qui sont, généralement, celles dont le côté gauche (côté 'situation' dit aussi 'prémises') est satisfait au vu de l'état actuel de la base de faits.
 - ✓ La résolution de conflits : Parmi les règles retenues suite au filtrage, le moteur décide de la (des) règle(s) qui sera (seront) exécutée(s).
- ☞ La résolution de conflit est souvent indépendante du contexte d'application et de la sémantique des règles, mais, elle est généralement guidée par des soucis d'efficacité (par exemple, choix des règles les moins complexes).
- ☞ Toutefois, des informations heuristiques ou des métarègles peuvent être exploitées pour fixer un ordre de priorité entre ces règles.

Moteurs d'inférence

Cycle de travail

- ☞ Dans la figure ci-contre, il est illustré comment, en général, un moteur d'inférence enchaîne des cycles de travail composés de deux phases.
- ☞ L'arrêt de ce cycle peut survenir dans :
 - ✓ la phase d'évaluation (par exemple, dans le cas d'absence de règles applicables à la situation en cours)
 - ✓ la phase d'exécution (par exemple, dans le cas de l'exécution d'une règle dont la partie 'action' commande l'arrêt du cycle).



Moteurs d'inférence

Caractéristiques

- ☞ L'objectif recherché par un moteur d'inférence lorsqu'il enchaîne des cycles de travail est de trouver une solution au problème (question) posé.
- ☞ Ceci peut être modélisé comme le parcours d'un arbre de recherche dont la racine correspond à l'état initial de la base de faits et les différents arcs correspondent à l'application d'une règle particulière à un état de la base de faits.
- ☞ La manière dont cette recherche est effectuée dépend des caractéristiques du moteur d'inférence utilisé qui peuvent être divisées en quatre catégories :
 - ✓ Formalisme utilisé (généralement basé sur les logiques classiques ou ses extensions) ;
 - ✓ Stratégie de recherche ou de résolution de conflit ;
 - ✓ Type de raisonnement ;
 - ✓ Régime de contrôle ;
 - ✓ Mode de chaînage.

Moteurs d'inférence

Type de raisonnement

- ☞ Un moteur d'inférence peut être basé sur une théorie logique monotone ou non.
- ☞ Lorsqu'un moteur adopte un raisonnement monotone, aucun fait établi (considéré comme vrai) et inséré dans la base de faits ne peut être nié (considéré comme faux) et retiré de la base de faits.
- ☞ Les moteurs d'inférence monotones ne permettent pas de traiter convenablement les problèmes dynamiques dont la nature même fait que les faits changent en permanence.

Moteurs d'inférence

Mode de chaînage

☞ Trois modes de chaînage peuvent être utilisés :

- ✓ le chaînage avant ;
- ✓ le chaînage arrière ;
- ✓ le chaînage mixte.

Modes de chaînage

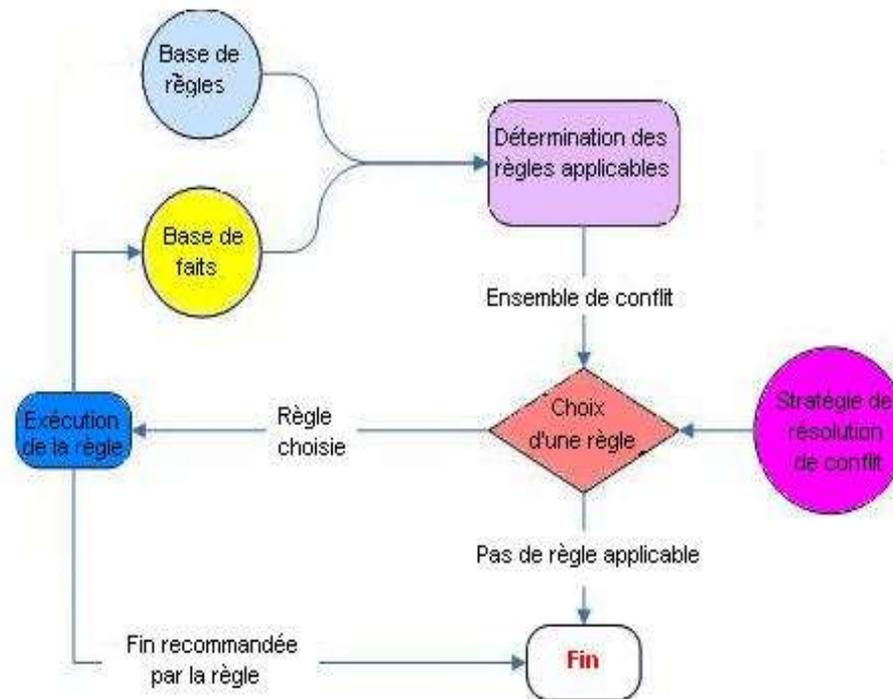
Chaînage avant

- ☞ Dans ce mode (connu aussi comme le mode guidé par les données), le moteur d'inférence procède comme suit :
 - ✓ Il ne sélectionne que les règles dont les conditions (prémises) sont satisfaites au vu de l'état de la base de faits pour arriver au but recherché (un fait qui répond à la question posée).
 - ✓ Il applique ensuite une de ces règles afin d'ajouter d'autres faits à la base.
- ☞ Cet enchaînement est réitéré jusqu'à ce que plus aucun fait nouveau ne puisse être déduit (on parle alors de saturation) ou que le but soit atteint.
- ☞ Le chaînage avant procède généralement par régime irrévocable et monotone et il a l'avantage d'être simple à implémenter tout en permettant de répondre plus rapidement à tout nouveau fait ajouté.
- ☞ Il est, toutefois, sujet au risque de l'explosion combinatoire car il sélectionne toutes les règles même celles sans intérêt pour la résolution du but.

Modes de chaînage

Chaînage avant

☞ Fonctionnement :



- Stratégies de recherche
 1. La recherche en largeur d'abord
 2. La recherche en profondeur d'abord
 3. La recherche en profondeur limitée
 4. La recherche heuristique

Modes de chaînage

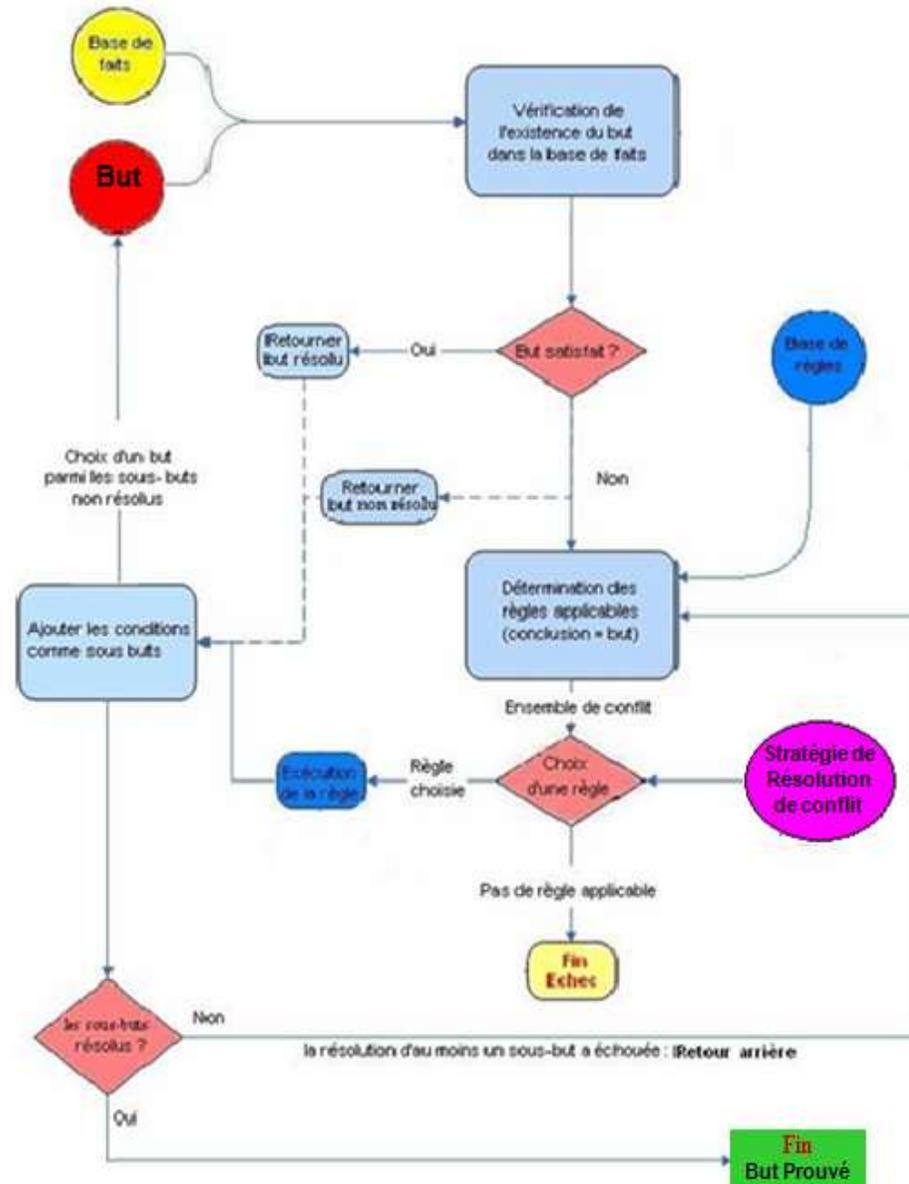
Chaînage arrière

- ☞ Un moteur d'inférence qui suit ce mode de chaînage, part du but et essaie de 'remonter' aux faits pour le prouver (parcours guidé par le but).
- ☞ En effet, le moteur sélectionne les règles dont la partie 'action' (ou conclusion) correspond au but recherché. Les prémisses/conditions (partie 'situation') de ces règles deviennent elles aussi des sous buts à prouver et ainsi de suite.
- ☞ Cet enchaînement s'arrête lorsque tous les sous buts sont prouvés- le but est alors lui aussi prouvé - ou lorsqu'il n'est plus possible de sélectionner des règles.
- ☞ Le chaînage arrière nécessite un régime de contrôle par tentatives. Le moteur d'inférence opère alors un retour arrière (backtracking) pour remettre en cause l'application d'une règle qui débouche sur un échec et pour essayer une règle écartée précédemment.

Modes de chaînage

Chaînage arrière

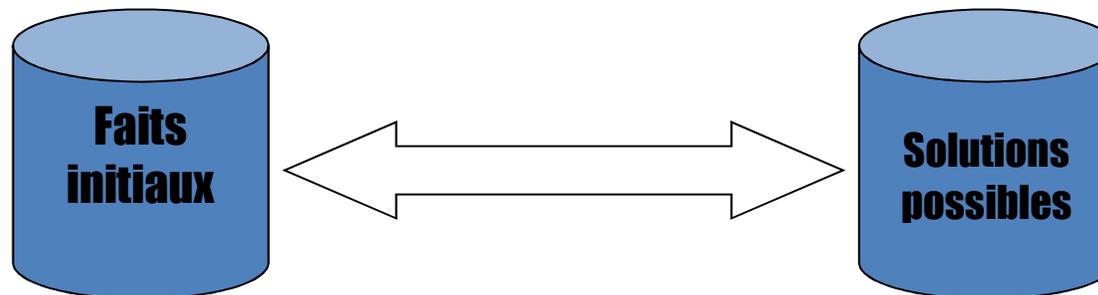
👉 Fonctionnement:



Systeme à base de règles

- **Chaînage mixte**

- Les mêmes règles sont utilisées au chaînage avant et arrière, selon que des faits nouveaux arrivent ou que l'on ait des faits à établir.
- On peut alors aussi bien raisonner à partir des faits que l'on connaît comme prédicats ou comme objectifs.



Modes de chaînage

Chaînage mixte

- ☞ Le chaînage mixte est une combinaison du chaînage avant et du chaînage arrière.
- ☞ Un moteur qui suit ce mode opère généralement comme suit :
 - ✓ Tant que des règles sont applicables (conditions satisfaites), il procède par chaînage avant jusqu'à satisfaction du but ou jusqu'à saturation de la base de faits ;
 - ✓ Si le but n'est toujours pas prouvé, il choisit alors une règle dont la partie conclusion correspond au but et essaie de prouver ses prémisses inconnues par chaînage arrière en veillant à modifier la base de faits à chaque nouveau sous but prouvé;
 - ✓ Si le but n'est toujours pas prouvé, il repart en chaînage avant en tenant compte des nouveaux faits prouvés et éventuellement en demandant de nouveaux faits à l'utilisateur et ainsi de suite jusqu'à ce que le but soit prouvé ou jusqu'à ce qu'aucune règle ne soit applicable.

Systeme à base de connaissance

- Comparaison

	Chaînage avant	Chaînage arrière
Points forts	<ul style="list-style-type: none">✓ Fonctionne bien lorsque le pb se présente "naturellement" avec des faits initiaux.✓ Produit une grande quantité de faits à partir de faits initiaux très nombreux.✓ Adapté à la planification, le contrôle, l'adaptation.	<ul style="list-style-type: none">✓ Fonctionne parfaitement lorsque le pb consiste à prouver une hypothèse✓ Il est focalisé sur le but à prouver et donc des questions pertinentes qui ne déroutent pas l'utilisateur.✓ Contrairement au chaînage avant, il recherche dans la base de connaissance les informations intéressantes pour le problème courant.✓ Adapté au diagnostic et à la prescription.
Points faibles	<ul style="list-style-type: none">○ Souvent, ne perçoit pas certaines évidences.○ Le système peut poser de nombreuses questions, parfois non pertinentes.	<ul style="list-style-type: none">○ Poursuit une ligne de raisonnement, même s'il s'avère qu'il faudrait l'abandonner pour une autre. Les facteurs de croyances et méta-règles peuvent aider à résoudre ce problème.

Exemple

On désire fonctionner le raisonnement déductif et inductif :

Soit la base de fait suivante $BF = \{B,C\}$ et soit la base de règles BR composée de :

1. Si B et D et E Alors F
2. Si G et D Alors A
3. Si C et F Alors A
4. Si B Alors X
5. Si D Alors E
6. Si X et A Alors H
7. Si C Alors D
8. Si X et C Alors A
9. Si X et B Alors D

Questions

- 1- Utilisez le raisonnement déductif et donner le résultat final et les résultats intermédiaires.
- 2- Soit H le fait à établir, donner le fonctionnement en raisonnement inductif.

Systeme à base de connaissance

- Quel chaînage utiliser ?
 - Ce sont les caractéristiques du problème qui vont conditionner le chaînage qu'il est judicieux d'utiliser.
 - Ainsi, lorsque les faits sont peu nombreux ou que le but est inconnu → chaînage avant.
 - Par contre, dans les cas où les buts sont peu nombreux ou précis → chaînage arrière.

I.3. Paradigmes de Représentation des connaissances

4. Le Frame

L'idée générale de cette nouvelle approche est résumée par Minsky : « Lorsqu'on rencontre une situation nouvelle, on sélectionne dans la mémoire une structure réelle qui est un frame (cadre).

FRAME : Chaise

HERITAGE : Meuble

ATTRIBUT:

Style -du -dossier : **doit- être** : droit, rembourré

Nombre -de-pieds : **doit- être** : entier **Par- défaut** : 4

Nombre -de-bras : **doit- être** : 0,1,2

FIN.

I.3. Paradigmes de Représentation des connaissances

5. Le Script

La théorie des scripts (scénarios) a été proposée par SCHANCK et ABELSON comme une spécialisation des frames pour décrire non pas des objets mais des scènes de la vie courante.

- **SCRIPT** " manger- au- restaurant"
- **ELEMENTS:** (restaurant, agent, nourriture, menu, tables, chaises)
- **ROLES:** (clients, serveur, chefs)
- **POINT- DE- VUE:** clients
- **MOMENT :** (heure d'ouverture du restaurant)
- **LIEU :** (emplacement du restaurant)
- **Scénario:**
- **D'abord** : script « Entrer restaurant »
- **Puis** : script « Attirer -l'attention -du -client- de- restaurant »
- **Puis** : script « Prendre- place- à- table »
- **Puis** : script « Passer- commande »
- **Puis** : script « Manger » sauf si (longue attente) alors script « Sortie- en- colère »
- **Puis** : si qualité nourriture >convenable) alors script « Féliciter- le chef »
- **Puis** : script « Payer- l'addition »
- **Enfin** : script « Quitter-restaurant ».

Systeme à base de connaissance

- **Ontologies, qu'est-ce que c'est ?**
 - Selon le dico du net :
 - Organisation hiérarchique de la connaissance sur un ensemble d'objets par leur regroupement en sous-catégories suivant leurs caractéristiques essentielles.
 - Plus généralement :
 - Ensemble des connaissances du domaine d'action d'un agent (pour la réalisation de sa tâche)
 - Ne pas confondre avec :
 - Taxinomie : Classification de concepts
 - Thésaurus :
 - Ensemble de termes organisé (*hiérarchique* ou *thématique* en général)
 - Chaque terme représente un concept du domaine
 - Relations entre les termes (synonymie, hyperonymie...)
 - Exemple. WordNet <http://wordnet.princeton.edu/>

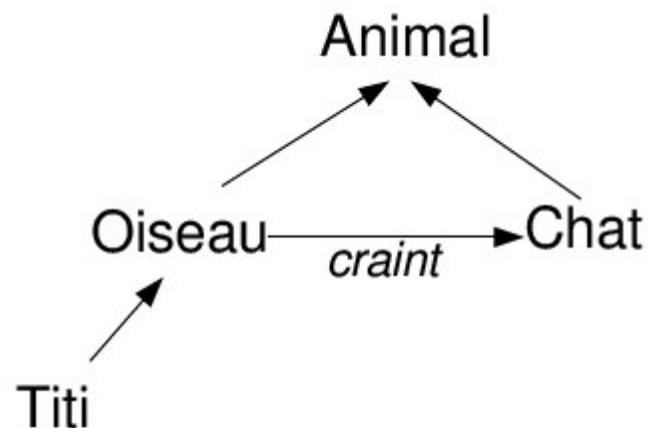
Systeme à base de connaissance

- **Ontologies, qu'est-ce que c'est ?**
 - La définition la plus citée en IA, due à T. Gruber :
“**an ontology is an explicit specification of a conceptualization**”.
 - **conceptualization** pour l'existence d'un modèle abstrait d'un certain aspect du monde, qui se compose d'un ensemble de définitions de **concepts** et de **relations**.
 - **explicit specification** pour la spécification du modèle dans un langage non ambigu, qui peut être compris et manipulé aussi bien par les machines (**agents logiciels**) que par les personnes (**agents humains**).

Systeme à base de connaissance

- Les ontologies

- Ensemble de concepts
- Reliés entre eux
- Regroupement en classe (structure hiérarchique)



protege (http://www.co-ode.org/ontologies/pizza/pizza.owl) : [http://protege.stanford.edu/ontologies/pizza/pizza.owl]

File Edit View Reasoner Tools Refactor Window Help

← → pizza (http://www.co-ode.org/ontologies/pizza/pizza.owl) Search for entity

Data Properties Annotation Properties Individuals OWL Viz DL Query OntoGraf SPARQL Query

Active Ontology Entities Classes Object Properties

Class hierarchy

- DomainConcept
- Country
- Food
 - IceCream
 - Pizza**
 - PizzaBase
 - PizzaTopping
- ValuePartition

OWL Viz Pizza

Asserted model Inferred model

```
graph LR; DomainConcept -- is-a --> Food; Food -- is-a --> Pizza; Pizza -- is-a --> InterestingPizza; Pizza -- is-a --> ThinAndCrispyPizza; Pizza -- is-a --> NonVegetarianPizza; Pizza -- is-a --> RealltalianPizza; Pizza -- is-a --> VegetarianPizzaEquivalent1; Pizza -- is-a --> MeatyPizza; Pizza -- is-a --> Giardiniera; Pizza -- is-a --> Margherita; Pizza -- is-a --> Soho; Pizza -- is-a --> QuattroFormaggi; Pizza -- is-a --> Veneziana; Pizza -- is-a --> Napoletana;
```

To use the reasoner click Reasoner->Start reasoner Show Inferences

Systeme à base de connaissance

- **Les ontologies**

- Ensemble de termes (représentant des concepts)

- Liens entre les concepts :

- Liens hiérarchiques (is-a, part-of...)

- Liens sémantiques (relations)

- Idée :

- La définition d'un terme est l'ensemble de ses relations dans l'ontologie.

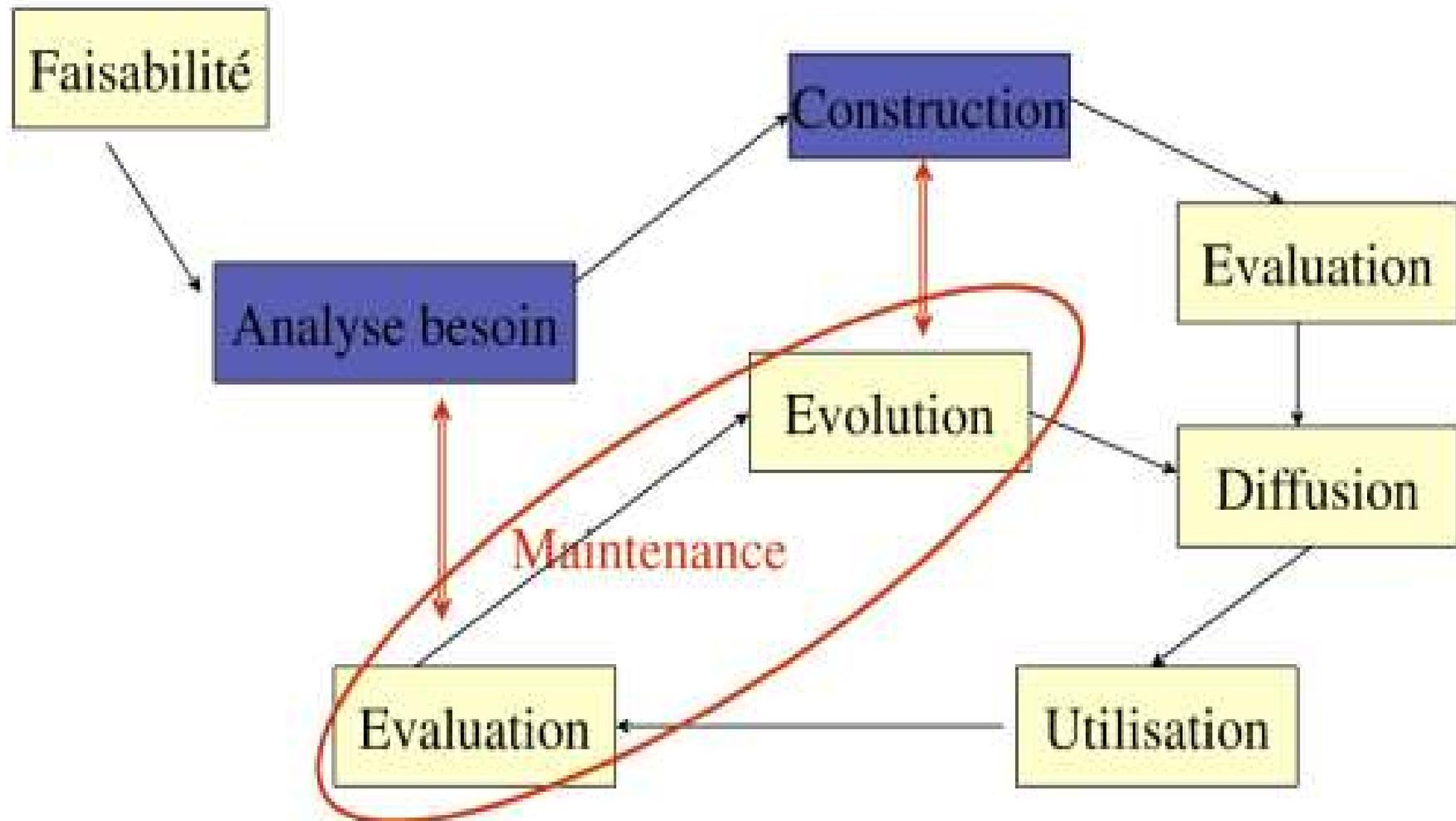
- Différentes formes d'ontologies selon :

- Nature des connaissances à inclure

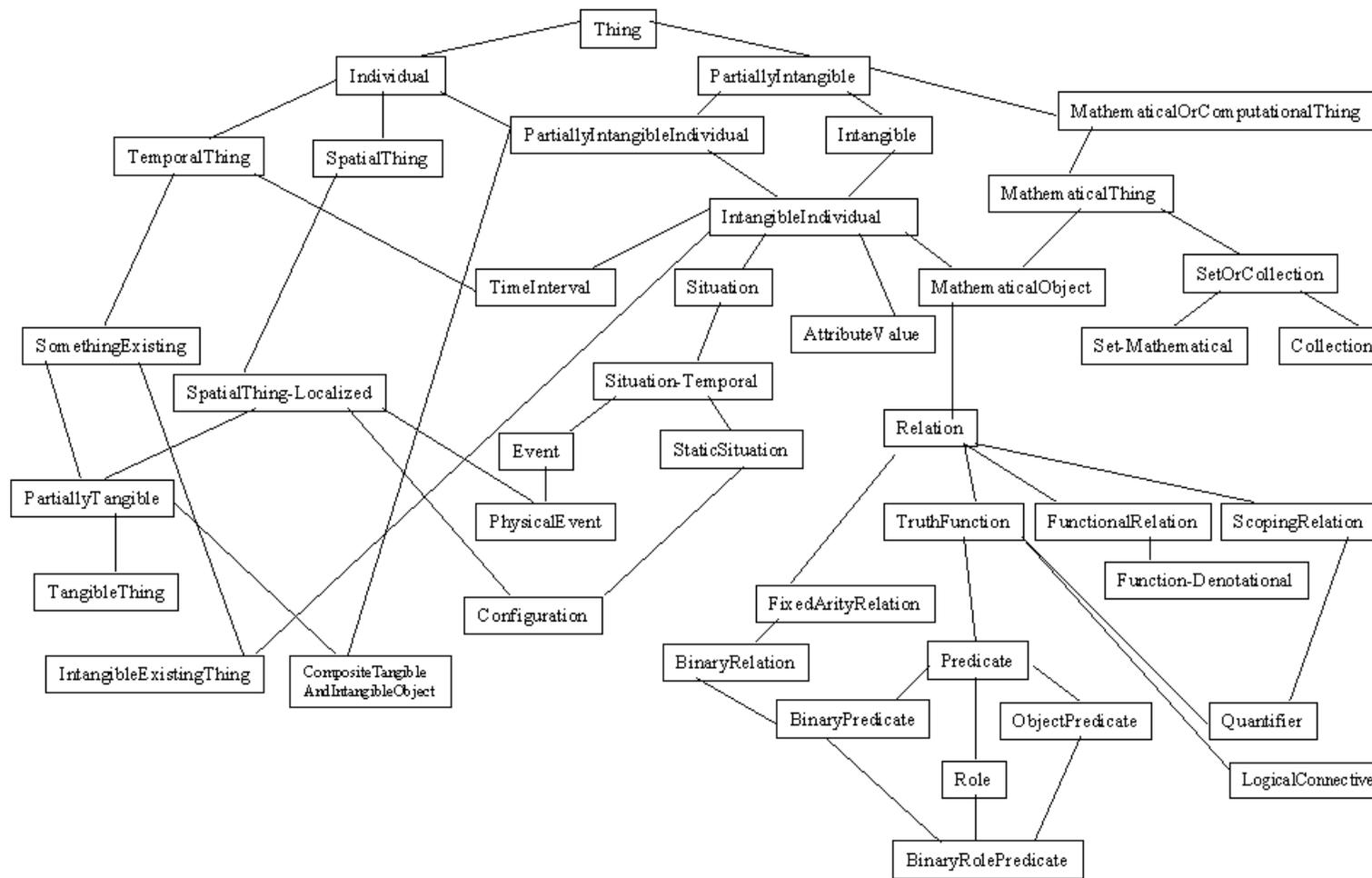
- Niveau de détail (granularité)

- Tâche, domaine, application

Cycle de vie d'une ontologie



Systeme à base de connaissance



Systeme à base de connaissance

- Les ontologies

- L'utilisation effective d'une ontologie nécessite :

- un langage de représentation des ontologies bien défini
 - et des modules associés de raisonnement qui soient efficaces.

Protégé 2000

The screenshot displays the Protégé 2000 ontology editor interface. The main window title is "Ontology1274197235484.owl (http://www.semanticweb.org/ontologies/2010/4/Ontology1274197235484.owl) -...". The menu bar includes File, Edit, Ontologies, Reasoner, Tools, Refactor, Tabs, View, Window, and Help. The toolbar shows navigation and editing icons. The "Active Ontology" tab is selected, and the "Classes" view is active.

The left pane shows a class hierarchy for "SellerPreferences". The hierarchy is as follows:

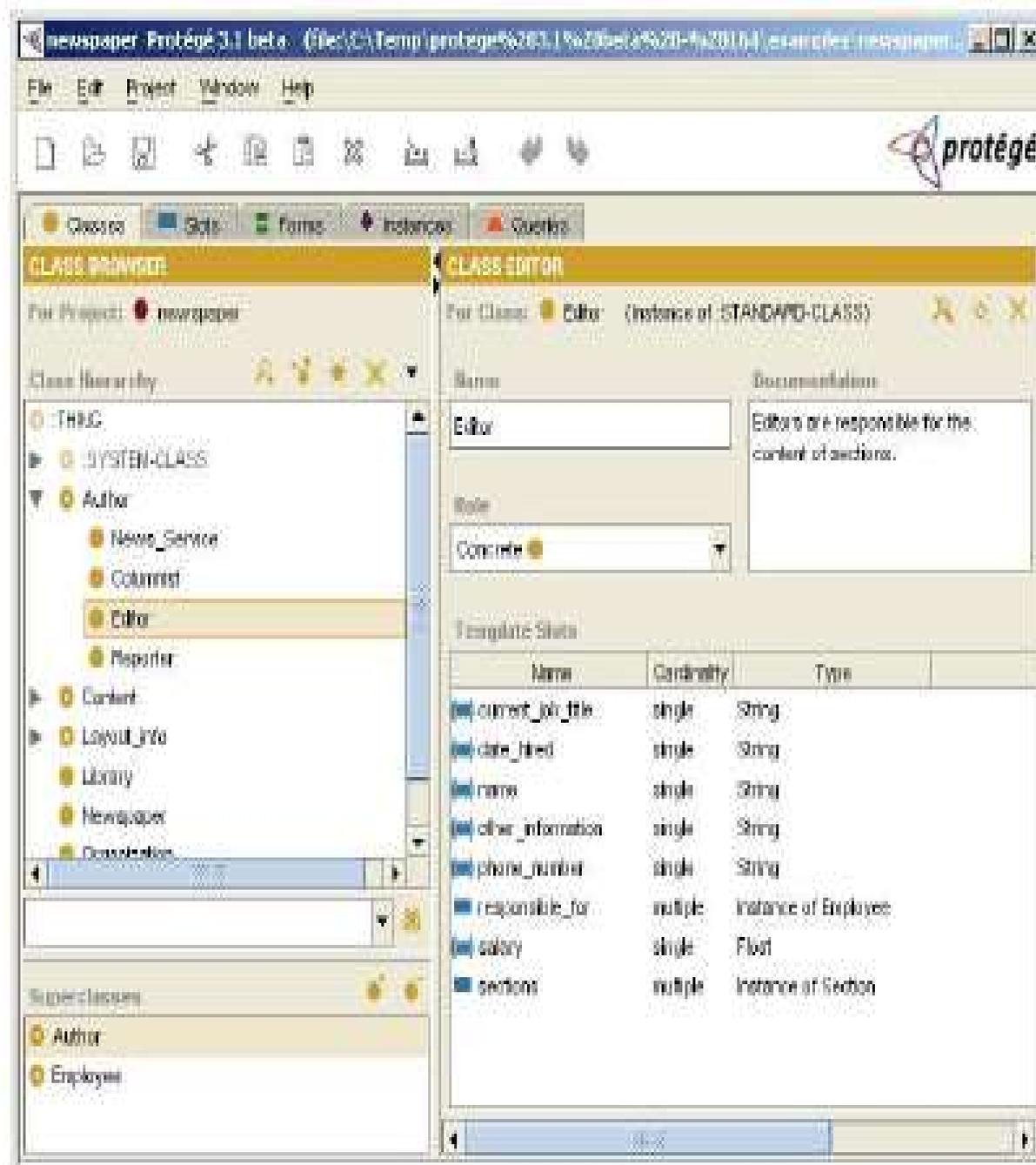
- Thing
 - AirConditioning
 - AlarmSystem
 - BuyerPreference
 - BuyerPreferences
 - CarDetail
 - CarMaker
 - Color
 - Insurance
 - InsurancePlus
 - NavigatorPack
 - Objective
 - SellerPreference
 - SellerPreferences**
 - Vehicle
 - Car
 - LuxuryCar
 - PassengerCar
 - Beta
 - CompactCar
 - MidSizeCar
 - SedanCar
 - SportCar
 - SUV
 - Truck
 - Van

The right pane shows the "Annotations" and "Usage" tabs for the selected class "SellerPreferences". The "Annotations" tab displays the following code:

```
<fuzzyOwl2 fuzzyType="concept">
  <Concept type="weightedSum">
    <Concept type="weighted" value="0.3" base="S1" />
    <Concept type="weighted" value="0.1" base="S2" />
    <Concept type="weighted" value="0.3" base="S3" />
    <Concept type="weighted" value="0.1" base="S4" />
    <Concept type="weighted" value="0.2" base="S5" />
  </Concept>
</fuzzyOwl2>@
```

The "Description" tab for "SellerPreferences" shows sections for Equivalent classes, Superclasses, Inherited anonymous classes, Members, Keys, and Disjoint classes, all of which are currently empty.

At the bottom right of the interface, the status bar indicates "No Reasoner set" and "Show Inferences" is checked.



PROTEGE permet la création, la visualisation et la manipulation d'ontologies dans différents formats.

L'onglet "Classes" permet de définir les concepts de l'ontologie comme une hiérarchie de classes et de leur affecter des propriétés. L'onglet "Slots" permet de spécifier ces propriétés.

Des onglets supplémentaires peuvent être développés par les utilisateurs pour répondre à leurs besoins spécifiques.

✘ Сейчас не удается отобразить рисунок.

Motivation

Base de Données à Base Ontologique

Utilisation accrue des ontologies

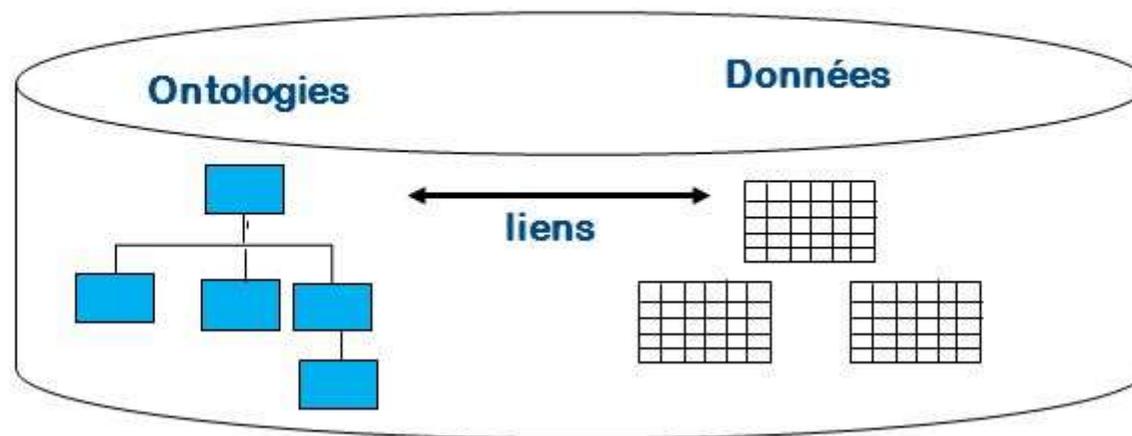
- ❑ Nombreux modèles d'ontologies : RDF-Schema, OWL, PLIB, etc.

Gros volumes de données à base ontologique

- ❑ Méta-données du Web Sémantique
- ❑ Bases de données composants et e-catalogue

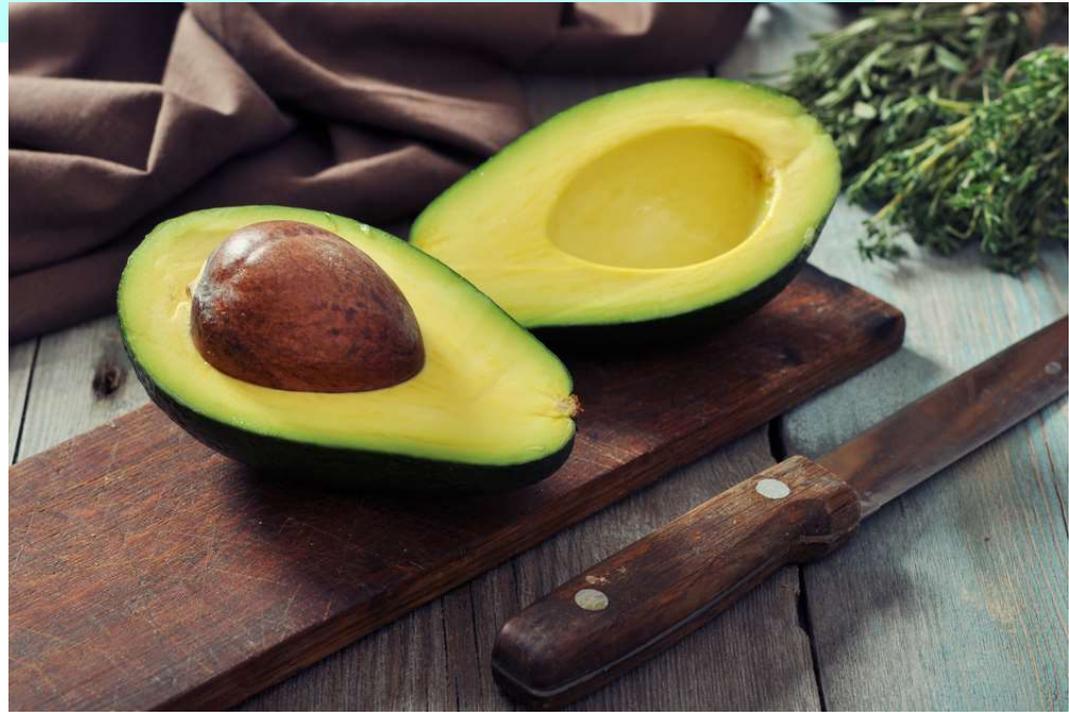
=> Bases de Données à Base Ontologique (BDBO)

- ❑ Ontologies + Données + Liens Ontologies / Données



✘ Сейчас не удается отобразить рисунок.

I. Recherche image par google



I. Représentation des connaissances

Conclusion

Remarques finales

- ☞ La représentation de connaissances est comme tout sujet de recherche relevant de l'IA est l'objet de beaucoup de controverses.
- ☞ Au-delà de ces polémiques, nous devons retenir que chaque type de formalisme a ses points forts et ses points faibles. Il n'existe pas de formalisme 'idéal' pour tout type de problèmes ou pour toute sorte d'applications.
- ☞ Certains problèmes requièrent même plusieurs formalismes pour la représentation de leurs différents aspects.
- ☞ Lors du choix d'un formalisme pour la résolution d'un problème particulier, on doit veiller entre autre à ce que :
 - ☞ toutes les connaissances nécessaires à la résolution du problème puissent être représentées (cf. critère d'expressibilité).
 - ☞ la manipulation des connaissances pour la résolution du problème soit facilitée par le formalisme (cf. critères d'adéquation inférentielle et d'efficacité calculatoire).

I. Représentation des connaissances

Conclusion

Remarques finales

- ☞ Il n'y a pas donc de règle générale pour le choix d'un formalisme, mais l'on sait, toutefois, que :
 - ☞ les formalismes logiques (en particulier, la logique des prédicats) sont plus appropriés lorsque les connaissances du domaine étudié sont uniformes et peu sujettes au contexte de telle sorte qu'elles peuvent être vues comme une 'théorie' du domaine.
 - ☞ les formalismes structurés comme les réseaux sémantiques et les frames peuvent être plus efficaces pour la résolution de certains types de problèmes relevant de domaines où les connaissances sont, par exemple, de nature taxonomique.
- ☞ La tendance actuelle semble privilégier des représentations hybrides comme les logiques de description, les ontologies du Web sémantique qui tentent de marier la rigueur sémantique de la logique des prédicats à l'efficacité inférentielle des systèmes à base de frames ou d'objets.

Fin de la première partie