

Ch2 : Représentation de la connaissance

Prof. Konan Marcellin BROU

marcellin.brou@inphb.ci

2019-2020

Sommaire

- ❑ **Introduction**
- ❑ **Techniques pour la RC**
- ❑ **Approche logique**
- ❑ **Règles de production**
- ❑ **Logique des prédicats**
- ❑ **Les schémas**
- ❑ **Bibliographie**

I. Introduction

□ Objectifs :

- Comprendre les différents types de formalisme de représentation de la connaissance.
- Savoir quand utiliser un type de représentation de la connaissance

I. Introduction

□ 1.1. Le but de la RC

- **Modéliser un domaine particulier d'application de sorte que la représentation ou le modèle obtenu soit manipulable par une machine.**
- **Deux types de connaissances dans une BC :**
 - **Connaissances factuelles (statiques) : concepts mis en jeu dans l'expertise ;**
 - **Connaissances dynamiques : règles de comportement des connaissances factuelles.**

I. Introduction

□ 1.2. Qu'est-ce que la connaissance ?

- **Compétence** qui permet de résoudre des problèmes.
- **Données brutes**
 - Collection d'éléments de valeur brute ou de faits servant à calculer, raisonner et mesurer ;
 - Peuvent être collectées, stockées ou traitées;
 - Ne possèdent pas de contexte ni de sens.
 - **Exemples :**
 - **!**
 - **...- - -...**
 - **C**

■ **Information** = donnée + sens attaché à la donnée :

- Proviennent de l'organisation des données, mettent en valeur les relations entre les différents éléments de ces données ;
- Fournissent un contexte et un sens aux données.
- **Exemples :**
 - **!** : point d'exclamation
 - **...- - -...** : SOS (Save Our Soals)
 - **C** : lettre ou note

I. Introduction

■ **Connaissance**

- **Ce que l'on a appris par l'étude ou la pratique.**
- **Connaissance = information + mode d'emploi pour entreprendre une action.**
- **Viennent de la compréhension de l'information dans son contexte ;**
- **Utile au processus de décision.**

□ **Exemples :**

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

I. Introduction

□ 1.3. Qu'est-ce que la représentation ?

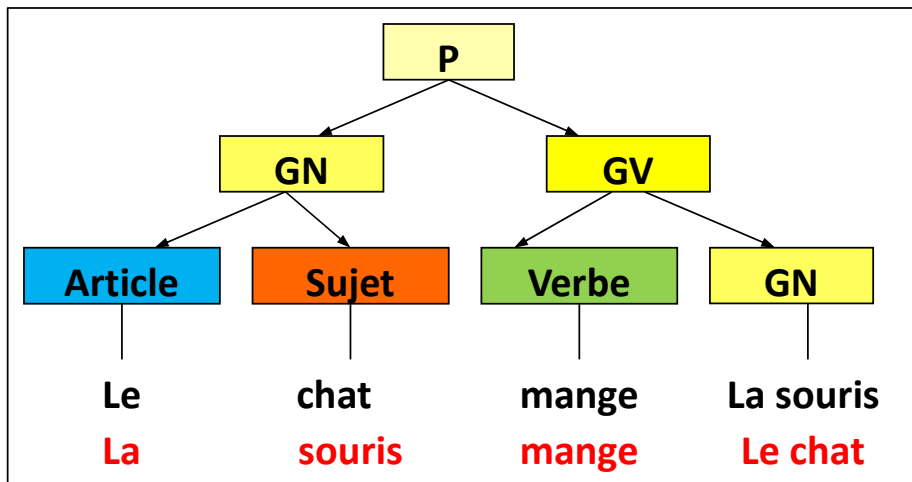
- Ensemble de conventions syntaxiques et sémantiques rendant possible la description d'objets.
- **Syntaxe** : symboles qui peuvent être utilisés, associés à la façon dont ils peuvent être assemblés.

- **Sémantique** : comment le sens se trouve intégré aux symboles et dans les arrangements de symbole qui sont autorisés par la syntaxe.

- L'ordinateur manipule des objets syntaxiques, sans sens.
- La sémantique associe un sens aux objets syntaxiques.

I. Introduction

■ Exemple :



- Ces deux phrases sont syntaxiquement correctes mais la 2^{ème} est sémantiquement incorrecte.

- RC est le point clé dans tout problème d'IA.
 - Trouver une représentation lisible par l'expert et exploitable par un programme d'IA (Moteur d'Inférences).
- RC consiste à trouver une correspondance entre un monde extérieur et un système symbolique.

I. Introduction

□ 1.4. Connaissance symbolique

■ Connaissances numériques :

- Manipulées habituellement par les ordinateurs.

■ Connaissances symboliques :

- Utilisent des symboles
- Symboles stockées dans des fichiers texte.

■ Exemple :

- Représenter la phrase : **"Toto est allé à Tipatipa"**
- Représentation par une chaîne de caractères
- Représentation par une structure de données

■ Représentation par une chaîne de caractères

- Chaîne stockée dans un fichier de texte.
- Questions :
 - **« Qui est allé à Tipatipa ? ».**
 - **« Où est parti Toto ? »**
- Difficile de répondre à ces questions
 - **RC retenue n'intègre pas des informations sur l'action "aller", ni aucune compréhension de la chaîne de caractères.**

I. Introduction

- **Représentation par une structure de données**

- **Inclue des éléments de signification de la phrase**

Attribut	Valeur
Action	aller
Agent	Toto
Source	?
Destination	Tipatipa
Temps	passé
Moyen	?

- **Représentation structurée plus appropriée que la représentation par chaîne de caractères.**

- **Plusieurs formalismes de RC :**

- **Logique**
- **Frames ou schémas : utilisés par les chercheurs en psychologie dans le cadre de la perception et la vision des objets ;**
- **Réseaux sémantiques : utilisés par les linguistes pour représenter la sémantique des phrases**
- ...

I. Introduction

□ 1.5. Typologie de la connaissance

■ On distingue plusieurs types de connaissance :

□ **Définition** : toujours vraie

- Ex. « Un quadrilatère est un polygone ayant exactement 4 côtés ».

□ **Evulsive/Atemporelle** : peut être modifiée/constante

- Ex. « Toto est élève de 2^oC4 au Lycée Tranquille ».

□ **Incertaine/Certaine** : pas avec certitude/sûre

- Ex. « Toto est né vers 1978 ».

□ **Vague ou Floue** : évaluation difficile

- Ex. « Les jeunes élèves sont turbulents ».

□ **Typique ou Universelle** : plausible mais peut être contredite (habituellement)

- Ex. « Habituellement les baoulés aiment boire le vin ».

□ **Ambigüe** : plusieurs significations

- Ex. « Fatou coupe ses oignons »
(Bulbe comestible ou cor du pied)

I. Introduction

□ 1.6. Caractéristiques d'un bon système de RC

■ Adéquation représentationnelle

- Représenter tous les genres de connaissances nécessaires au domaine.

■ Adéquation inférentielle

- Inférer de nouvelles connaissances à partir des anciennes.

■ Efficacité inférentielle

- Incorporer un supplément d'informations dans les structures des connaissances pour aider les mécanismes d'inférences.

■ Efficacité acquisitionnelle

- Acquérir de nouvelles connaissances car les connaissances de l'expert évoluent.

■ Extensibilité

- Prise en compte de nouvelles connaissances.

■ Simplicité

- Permettre aux non informaticiens puissent transmettre leur savoir au système.

I. Introduction

- **Connaissance explicite**
 - Recherche des erreurs et justification
 - Essayer d'avoir toutes les connaissances pour justifier le raisonnement.

II. Techniques pour la RC

□ **2.1. Connaissances déclaratives et procédurales**

■ **Principe de base d'un formalisme de RC**

- **Distinction explicite entre ces deux sortes de connaissances.**

■ **Programmation classique**

- **Programmation procédurale de type Pascal.**
- **Ces deux catégories de connaissances sont imbriquées.**

■ **Exemple : modélisation du domaine médical.**

□ **Connaissances sur les maladies représentées selon deux approches :**

- **Représentation procédurale**
- **Représentation déclarative**

II. Techniques pour la RC

□ 2.2. Représentation procédurale

- Connaissance représentée comme une collection de procédures qui indiquent **comment** utiliser la connaissance
- Exemple :

```
PROCEDURE maladieA
DEBUT
  vérifier symptôme 1
  ...
  vérifier symptôme m
FIN
```

■ Deux connaissances imbriquées dans la procédure :

- Connaissances factuelle sur la maladie
 - "maladieA" concerne les symptômes 1, etc.
- Connaissance dynamique : manière de diagnostiquer la maladieA
 - vérifier symptôme1 avant symptôme2, etc.

II. Techniques pour la RC

□ 2.3. Représentation déclarative

- Représente ce que l'on sait (**quoi**) dans une collection statique de faits et de règles d'inférences.
- Séparation des deux sortes de connaissance :
 - Connaissances sur la maladie et connaissances sur la manière de faire un diagnostic.

- 1^{ère} étape : définir les connaissances du domaine d'application (informations descriptives)

```
C1 : SI symptôme 1 ET .... ET symptôme m
      ALORS maladie A
...
Cn :      .....
```

- 2^{ème} étape : définir les procédures d'exploitation

```
ALGORITHME VerifierButX
DEBUT
1. chercher l'ensemble" des Ci tel que
   symptômei1 ET ... symptômeim → X
2. choisir" un Cj approprié
3. vérifier" les nouveaux buts (symptômejk) de
   Cj
FIN
```


II. Techniques pour la RC

□ Exemple de BC

■ Base de connaissance

C1 : $A \rightarrow E$
C2 : $B \rightarrow D$
C3 : $H \rightarrow A$
C4 : $E, G \rightarrow C$
C5 : $E, K \rightarrow B$
C6 : $D, E, K \rightarrow C$
C7 : $G, K, F \rightarrow A$

■ Base de faits initiale : **{H, K}**

■ But à prouver : **C**

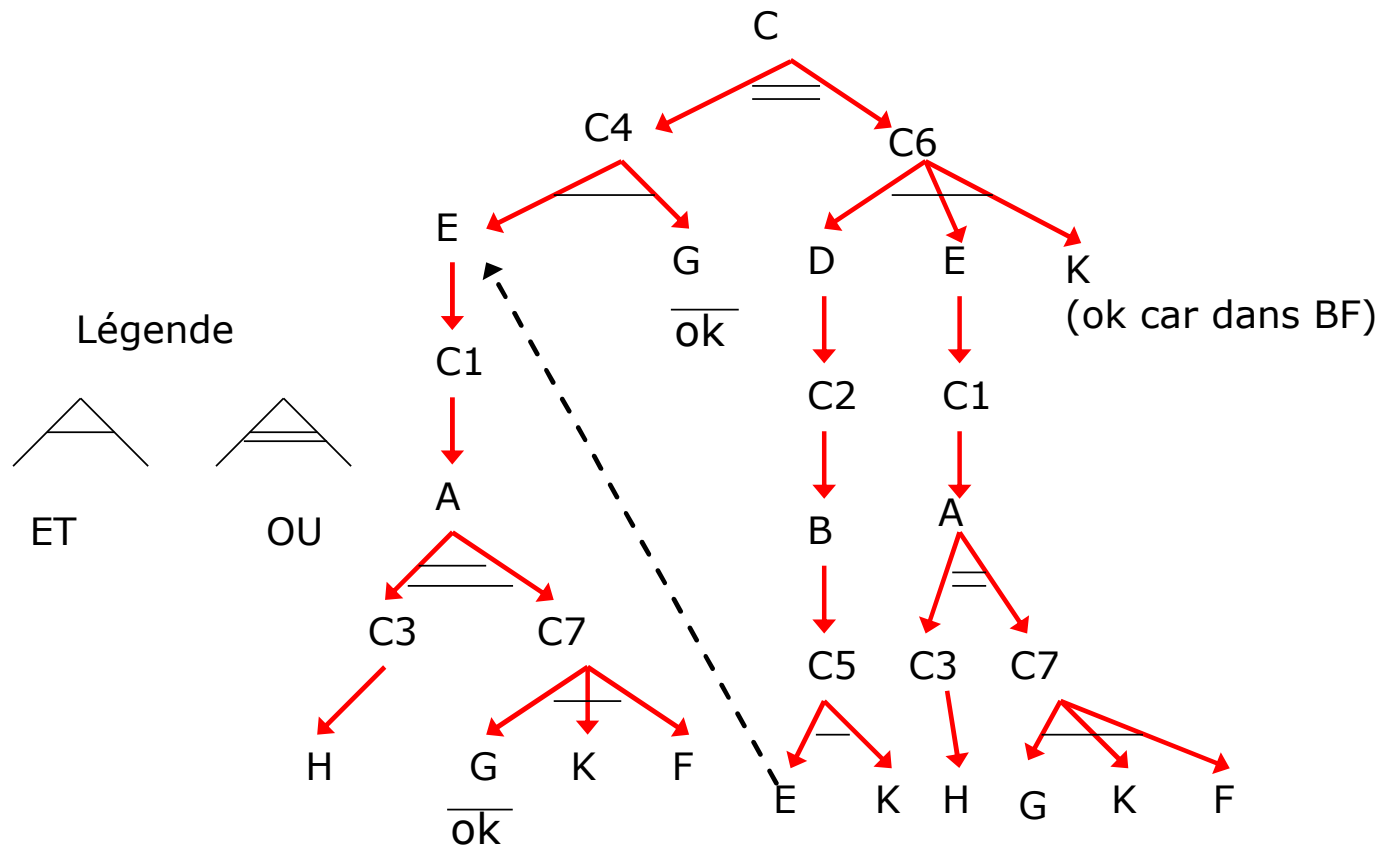
■ Stratégie d'arrêt du moteur :

- Arrêt si le but recherché est prouvé
- Arrêt s'il n'y a plus rien à déduire

■ Stratégie de choix :

1. Règle qui donne le but si elle est sélectionnée
2. Une règle non encore utilisée

II. Techniques pour la RC



C1 : $A \rightarrow E$
C2 : $B \rightarrow D$
C3 : $H \rightarrow A$
C4 : $E, G \rightarrow C$
C5 : $E, K \rightarrow B$
C6 : $D, E, K \rightarrow C$
C7 : $G, K, F \rightarrow A$

BF initiale : H, K
But à prouver : C

II. Techniques pour la RC

□ 2.4. Comparaison des deux approches

■ Avantages représentation procédurale :

□ Facilité de représentation :

- Des connaissances sur la manière de faire les choses ;
- Des connaissances qui ne rentrent pas dans de nombreux schémas déclaratifs simples ;
- Des connaissances heuristiques sur la manière de faire les choses efficacement.

□ Rectitude du raisonnement :

- Heuristiques spécifiques pour induire un raisonnement naturel.

□ Facilité de codage

□ Facilité de compréhension du processus lui-même

■ Avantages représentation déclarative :

□ Modularité :

- pouvoir mettre les connaissances dans n'importe quel ordre.

II. Techniques pour la RC

□ **Gain de place :**

- Connaissance stockée une seule fois quel que soit le nombre de façon différentes dont on peut l'utiliser.

□ **Mises à jour des connaissances sans remise en cause de la structure globale du système :**

- Ajoute ou modification des informations sans modifier les procédures d'interprétation qui sont indépendantes du domaine.

□ **Explication du raisonnement :**

- Dans l'approche procédurale, l'explication d'un résultat d'exécution de procédures est quasi impossible.
- Dans l'approche déclarative, les modules d'interprétation utilisent les connaissances du domaine d'application pour produire une trace utilisée pour la production des explications.

II. Techniques pour la RC

□ 2.5. Formalisme de RC

■ RC sujet de recherche pluridisciplinaire :

- **Sciences cognitives** : étudient l'intelligence humaine
- **Psychologie** : étudient le comportement de l'homme
- **Linguistique** : étudie les langues
- **Mathématiques**
- **Informatique** : BD, systèmes experts, graphes conceptuels, programmation logique, IA en général,...

■ Trois principaux types d'approche de RC :

□ Approche logique :

- Utiliser la logique mathématique comme outil de RC.

□ Approche sémantique :

- Utilisée par les linguistes pour représenter la sémantique des phrases.

□ Approche hybride entre la sémantique et la logique :

- Utilise la notion de "frame" ou schéma
- Mise en évidence par les chercheurs en psychologie.

III. Approche logique

□ 3.1. Principe

- **Utiliser la logique mathématique comme outil pour la RC.**
- **Formalisme logique adapté :**
 - à la résolution de problème ;
 - au calcul formel ;
 - à la démonstration de théorèmes ;
 - à l'interrogation de bases de données.
- **Chefs de file de cette approche**
 - **Concepteurs et utilisateurs du langage PROLOG.**

III. Approche logique

- **3.2. Représentation de la connaissance**
 - **Connaissances du domaine construites à partir des prédicats de base assemblés en règles.**
 - **Deux types de formalisme :**
 - **Règles de production (RP)**
 - **Logique des prédicats**

III. Approche logique

□ 3.3. Notation logique

■ Alphabet :

- lettres propositionnelles : **p, q, r, s, t,...**;
- opérateurs logiques : **$\neg, \Rightarrow, \Leftrightarrow, \wedge, \vee, \forall, \exists$**
- parenthèses : **()**

■ Formation des mots :

- une lettre propositionnelle est un mot
- si m est un mot alors (m) est un mot
- si m est un mot alors $\neg m$ est un mot
- si m_1 et m_2 sont des mots alors $m_1 \wedge m_2$ et $m_1 \vee m_2$ sont des mots
- si m_1 et m_2 sont des mots alors $m_1 \Rightarrow m_2$ est un mot

■ Axiomes :

- $m_1 \Rightarrow (m_2 \Rightarrow m_1)$
- $(m_1 \Rightarrow (m_2 \Rightarrow m_3)) \Rightarrow ((m_1 \Rightarrow m_2) \Rightarrow (m_1 \Rightarrow m_3))$
- $(\neg m_2 \Rightarrow \neg m_1) \Rightarrow (m_1 \Rightarrow m_2)$

III. Approche logique

■ 3.4. Principales lois logiques

■ Lois de tautologie

Formule	Explication
$p \Leftrightarrow p$	Loi d'identité
$p \Rightarrow p$	Tautologie élémentaire
$p \Leftrightarrow (p \vee p)$ $p \Leftrightarrow (p \wedge p)$	Loi d'impotence des connecteurs logique
$\neg (p \wedge \neg p) \Leftrightarrow \text{Vrai}$	Loi de non contradiction, deux formules P et $\neg P$ ne peuvent être toutes les deux vraies
$p \vee \neg p \Leftrightarrow \text{Vrai}$	Loi du tiers exclu, deux formules P et $\neg P$ ne peuvent être toutes les deux fausses
$\neg \neg p \Leftrightarrow p$	Loi de la double négation

■ Autres lois logiques

Formule	Explication
$(p \vee q) \Leftrightarrow (q \vee p)$ $(p \wedge q) \Leftrightarrow (q \wedge p)$	Commutativité
$((p \vee q) \vee r) \Leftrightarrow (p \vee (q \vee r))$ $((p \wedge q) \wedge r) \Leftrightarrow (p \wedge (q \wedge r))$	Associativité
$(p \vee (q \wedge r)) \Leftrightarrow (p \vee q) \wedge (p \vee r)$ $(p \wedge (q \vee r)) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$	Double distribué
$(p \wedge (p \vee q)) \Leftrightarrow p$ $(p \vee (p \wedge q)) \Leftrightarrow p$	Absorption
$(p \Rightarrow q) \Leftrightarrow (\neg q \Rightarrow \neg p)$ $p \Rightarrow q \Leftrightarrow \neg p \vee q$	Contraposition
$\neg(p \vee q) \Leftrightarrow \neg p \wedge \neg q$ $\neg(p \wedge q) \Leftrightarrow \neg p \vee \neg q$ $(p \Rightarrow q) \Leftrightarrow (\neg p \vee q)$ $(p \Rightarrow q) \Leftrightarrow \neg(p \wedge \neg q)$ $(p \Leftrightarrow q) \Leftrightarrow (p \Rightarrow q) \wedge (q \Rightarrow p)$ $(p \vee q) \Leftrightarrow (\neg p \Rightarrow q)$ $(p \wedge q) \Leftrightarrow \neg(p \Rightarrow \neg q)$	Lois de dualité ou lois de Morgan

III. Approche logique

□ 3.5. Règles de déduction

■ Modus ponens

- SI a ET $(a \Rightarrow b)$ ALORS on peut déduire b
- Formellement : $((a) \wedge (a \Rightarrow b)) \Rightarrow b$

■ Modus tollens

- SI $\neg b$ ET $a \Rightarrow b$ alors on peut déduire $\neg a$
- Formellement : $((\neg b) \wedge (a \Rightarrow b)) \Rightarrow \neg a$

III. Approche logique

□ 3.6. Principe de résolution

■ Pour démontrer un théorème :

1. Supposer que la négation du théorème est vraie.
2. Démontrer que les axiomes et la négation du théorème déterminent quelque chose de vrai et qui ne peut l'être.
3. Conclure que la négation ne peut être vraie car elle conduit à une contradiction.
4. Conclure que le théorème est vrai.

■ Justification : H = Hypothèse, C = Conclusion

- Si $H \Rightarrow C$ est vrai alors $\neg(H \Rightarrow C)$ est fausse
- Or d'après la loi de Morgan : $(p \Rightarrow q) \Leftrightarrow (\neg p \vee q)$
- donc $\neg(H \Rightarrow C) \Leftrightarrow \neg(\neg H \vee C)$
- Or d'après la loi de Morgan : $\neg(p \vee q) \Leftrightarrow \neg p \wedge \neg q$
- Donc $\neg(\neg H \vee C) \Leftrightarrow \neg(\neg H) \wedge \neg C \Leftrightarrow H \wedge \neg C$
- $H \wedge \neg C$ est contradictoire car $H \Rightarrow C$ si H est vrai C l'est aussi

III. Approche logique

■ Exemple du principe de résolution

□ Axiomes de départ :

- $\neg \text{PlumesPerdrix} \vee \text{OiseauPerdrix},$
 PlumesPerdrix

□ Prouver que la perdrix est un oiseau :

- **OiseauPerdrix**

□ Ajout de la négation du théorème :

- $\neg \text{PlumesPerdrix} \vee \text{oiseauPerdrix},$
 $\text{PlumesPerdrix}, \neg \text{OiseauPerdrix}$

□ Or $(p \Rightarrow q) \Leftrightarrow (\neg p \vee q)$: Loi de Morgan

- $(\neg \text{PlumesPerdrix} \vee \text{OiseauPerdrix}) \Leftrightarrow$
 $(\text{PlumesPerdrix} \Rightarrow \text{OiseauPerdrix})$

□ Or $((p) \wedge (p \Rightarrow q)) \Rightarrow q$: Modus ponens

- $((\text{PlumesPerdrix}) \wedge (\text{PlumesPerdrix} \Rightarrow \text{OiseauPerdrix})) \Rightarrow \text{OiseauPerdrix}$

□ Récapitulatif des Axiomes :

- $\neg \text{plumesPerdrix} \vee \text{oiseauPerdrix},$
 $\text{PlumesPerdrix}, \neg \text{OiseauPerdrix},$
OiseauPerdrix
- Contradiction : **$\neg \text{OiseauPerdrix}$**
et OiseauPerdrix

□ Conclusion :

- **OiseauPerdrix est donc un théorème**

III. Approche logique

■ Exercice

- **Axiomes : $D \Rightarrow (S \vee P), \neg S, \neg P$**
- **Théorème : $\neg D$**
- **Remarque 1 : principe de résolution**
 - $(S \vee P)$ et $\neg S$ on déduit P
- **Remarque 2 : notion de clause**
 - Une clause est une formule bien formé qui a la forme d'une disjonction de littéraux Cas particulier : un littéral isolé est une clause.

IV. Règles de production

□ 4.1. Présentation

- **Formalisme classique de RC, souvent utilisé dans les systèmes experts (SE).**
- **Semblable aux raisonnements humains.**
- **Beaucoup de SE possèdent une RC par règle de production :**
 - **MYCIN, DENDRAL, PROSPECTOR...**
- **Système de production :**
 - **Systèmes utilisant des Règles de production**

IV. Règles de production

□ 4.2. Représentation de la connaissance

■ Syntaxe :

□ **SI P1 ET P2 ET ... ET Pm ALORS C1 ET C2 ... ET Cn.**

□ **Prémisses (Pi) :**

- décrivent une certaine situation.
- Sous forme de conjonction de conditions, de négations ou de disjonctions.

□ **Conclusions (Ci) :**

- ensemble d'actions à entreprendre si les prémisses sont satisfaites.
- Conclusions toujours sous forme de conjonction (ET)

■ **Formalisme sous forme de RP :**

- **Représentation externe de la connaissance.**
- **La plupart des systèmes "compilent" l'ensemble des règles pour obtenir une représentation interne efficace.**
- **La représentation interne a pour effet de structurer la connaissance en groupant en listes (au sens de LISP) les règles et les prémisses.**

IV. Règles de production

- **Exemple de règles de production :**

- **Règle 1 : SI vertige ET maux de tête ET malaise général ET bourdonnement d'oreille ALORS hypertension artérielle**
- **Règle 2 : SI hypertension artérielle ALORS Anacardium occidentale**

- **Règles mises sous forme déclarative**

- **Indispensable aux procédures d'interprétation**

IV. Règles de production

□ 4.3. Classification des systèmes de production

■ Systèmes d'ordre 0

- Utilise la logique des propositions seule.
- Pas d'utilisation de variable
- Actions autorisées : ajout et effacement de faits
- Toute règle appliquée est éliminée
- Exemples :

Exemple 1	Exemple 2
SI fièvre ET maux de tête ET vomissement ALORS paludisme	SI le moteur cale ET allumage correct ET réservoir d'essence non vide ALORS vérifier carburation

■ Système d'ordre 0+

- Utilisation de variables
- par exemple un compteur que l'on incrémente.
- Une règle appliquée n'est pas nécessairement effacée.
- Exemple :

Exemple 1	Exemple 2
SI âge > 65 ALORS retraité	SI température < 30°C ET consigne = 50 ALORS consigne = consigne + 2

IV. Règles de production

■ Système d'ordre 1

- Utilise la logique des prédicats.
- Autorise des variables dans les règles mais pas dans les faits (notion de variable et d'appariement).
- Une règle appliquée reste toujours possiblement applicable.
- Exemple :

SI X est un homme
ALORS X est mortel
 $\forall x, \text{homme}(x) \Rightarrow \text{mortel}(x)$

■ Système d'ordre 2

- Utilisation de variables dans les règles et les faits.
- $\forall N \forall x, y (\text{Nationalité}(N) \wedge \text{Mariés}(x, y)) \Rightarrow N(x) \Leftrightarrow N(y)$
- Fait N autorise une variable

IV. Règles de production

□ 4.4. Prise en compte du raisonnement incertain

■ Les SE avec des RP permettent la prise en compte de raisonnement incertain.

- on ne peut pas toujours évaluer ses connaissances avec une échelle à deux valeurs : vrai ou faux.

■ Causes de l'incertitude :

- Impossibilité de traduire sous forme de RP une connaissance.
- Les règles enregistrées sont imprécises.
- Connaissances manquantes, erronées ou probables.

INP-HB/K. M. BROU

■ Solution

- Ajout aux faits d'un nombre compris entre -1 et +1
 - **-1 = faux, +1 = vrai**
- Ce nombre est appelé coefficient de vraisemblance ou de plausibilité.

■ Exemple

- **SI gros nuage blanc ALORS pluie (0.8)**
- **SI fumée ALORS feu (0.9)**

IV. Règles de production

□ 4.5. Les méta-règles ou méta-connaissance

- Connaissance sur la connaissance.
- Règles de la BC pour réduire l'espace de recherche
 - Choisir la règle à activer et prendre en compte le contexte d'application.
- Indiquer un enchaînement des règles à utiliser de préférence.

■ Plusieurs catégories :

- Utiliser certaines règles de préférence :
 - **SI paludisme ALORS considérer les règles R50, R61, R63 ensuite les règles R15, R12.**
- Exclure certaines règles :
 - **SI diabète ALORS ne pas utiliser les règles R20 et R23.**
- Proposer un plan de recherche :
 - **SI rougeole ALORS considérer les règles R3, R4**

IV. Règles de production

■ Exemple de méta-règle dans MYCIN

SI le patient est un hôte à risque
ET s'il existe des règles mentionnant le pseudo-monias dans leurs prémisses
ET s'il existe d'autres règles mentionnant le klebsiellas dans une prémisse
ALORS il est préférable d'utiliser les premières règles avant les secondes

IV. Règles de production

□ 4.7. L'apprentissage

□ Enrichissement de la connaissance, deux cas possibles :

- Enrichissement sans apprentissage : l'expert ajoute des règles à la BC.
- Enrichissement avec apprentissage : les règles sont ajoutées à la BC automatiquement à partir d'exemples. Dans le SE Meta-Dendral, un module donne de nouvelles règles à partir d'exemples (problème d'induction).

□ Amélioration des performances du SE

- Il s'agit de trouver des heuristiques performantes ou des méta-règles associées à certains faits de la base de faits.
- Le système LEX conçu pour calculer des primitives trouve des heuristiques à partir d'exemples traités.

IV. Règles de production

□ 4.8. Critiques

■ Avantages :

- Simplicité, uniformité et modularité
- Exemple : on peut rajouter une nouvelle règle sans remettre en cause la structure globale du système.
- Dans l'approche classique, le rajout de cette nouvelle règle est possible en modifiant la procédure correspondante :
 - remise en cause de la procédure (ordre d'ajout, ...).

- Règles sont assez proches du modèle de RC humaines, qui est largement utilisé dans les approches de modélisation des connaissances.

IV. Règles de production

■ Inconvénients

- Insuffisant pour décrire des phénomènes complexes
- Fournit une base un peu brouillonne difficilement maintenable
- Perte de cohérence de la base si le nombre de règle est trop important (circularité, redondance, incompatibilité).
- Problème de la circularité :
 - **R1 : SI a ALORS c**
 - **R2 : SI c ALORS a.**
 - Le SE tourne alors en rond.

□ Problème de redondance :

- **R1 : SI a ALORS c**
- **R2 : SI a et b ALORS c**
- BC volumineuse, temps de réponse long

□ Problème de l'incompatibilité :

- **R1 : SI a ALORS c**
- **R2 : SI a ALORS non c**
- Réponse contradictoire

IV. Règles de production

□ Exercice 1 :

- On veut mettre en place un système de production permettant de réparer les pannes d'ordinateur. On vous demande de trouver les règles de production.

■ Exemples :

Panne	Causes
Redémarrage intempestif	<ul style="list-style-type: none">• Présence de virus Ouf• Défaillance de la boîte d'alimentation• Bouton de redémarrage enfoncé• Incompatibilité matérielle
Absence de son	
Ecran noir	
Ecran bleu	
Système hors service	
Impossible de booter sur le CD	

IV. Règles de production

□ Exercice 2

- On veut mettre en place un système de production permettant de diagnostiquer les maladie en médecine traditionnelle.
- On vous demande de trouver les règles de production.

■ Exemples de RP :

SI FIÈVRE ET MAUX DE TÊTE ET VOMISSEMENTS BILEUX ALORS PALUDISME
SI URINES FRÉQUENTES SURTOUT LA NUIT ET INFECTIONS CUTANÉES INEXPLIQUÉES ET NOTION D'HÉRÉDITÉ ALORS DIABÈTE
SI temperature > 38 ET toux ET perte d'odorat ALORS COVID'19

V. La logique des prédicats

□ 5.1. Présentation

■ Limites de la logique d'ordre 0 ou logique des propositions

- Impossible d'exprimer certaines assertions du type :
 - Syllogisme de Socrates
 - «**Tous les hommes sont mortels Or Socrates est un homme donc Socrates est mortel**»
- D'où la logique du 1^{er} ordre.
- Reprend l'ensemble des éléments de la logique propositionnelle et ajoute des nouveautés.

□ Exemple : «**Si x est un homme, alors il est mortel**» donne

- $\forall x, \text{homme}(x) \Rightarrow \text{mortel}(x)$

■ **Prédicat : expression logique dont la valeur peut être vraie ou fausse selon la valeur de ses arguments.**

■ Exemples :

- homme(socrate) est vrai
- mortel(socrate) est vrai
- femme(socrate) est faux

V. La logique des prédicats

■ Logique d'ordre 1

- Ou Logique de premier ordre ou logique des prédicats du 1^{er} ordre
- Généralisation de la logique des proposition
- 1^{er} ordre signifie que les prédicats contiennent des variables quantifiés universellement, qui peuvent être remplacés par n'importe quelle expression bien formée du langage.
 - Utilisation de variables
 - Utilisation de quantificateurs (\forall et \exists)

- Les bases de données et particulièrement des langages comme SQL se fondent sur le calcul des prédicats.

V. La logique des prédicats

□ 5.2. Notation logique

■ Alphabet :

□ **Connecteurs** : \neg , \wedge , \vee , \Rightarrow , \Leftrightarrow

□ **Quantificateur** :

- \forall : Universel (pour tout, quel que soit)
- \exists : Existentiel (il existe au moins un . . . tel que)

■ Expressions courantes :

Expression	Formule
tous les A sont B	$\forall x, A(x) \Rightarrow B(x)$
seuls les A sont B	$\forall x, B(x) \Rightarrow A(x)$
aucun A n'est B	$\forall x, A(x) \Rightarrow \neg(B(x))$
quelques A sont B	$\exists x, A(x) \Rightarrow B(x)$

V. La logique des prédicats

- **5.2. Représentation de la connaissance**
 - **Pour résoudre un problème on le représente par :**
 - **des formules-axiomes $A_1 \dots A_n$ et une formule conjecture C**
 - **et on tente de prouver par un enchaînement fini d'applications des règles d'inférence que la formule $A_1 \wedge \dots \wedge A_n \wedge \neg C$ est inconsistante, i.e. toujours fausse.**

V. La logique des prédicats

■ Exemple du principe de résolution

□ Axiomes de départ :

$\neg \text{Plumes}(\text{perdrix}) \vee \text{Oiseau}(\text{perdrix}), \text{Plumes}(\text{perdrix})$

□ Prouver que la perdrix est un oiseau :

■ $\text{Oiseau}(\text{perdrix})$

□ Ajout de la négation du théorème :

$(\neg \text{Plumes}(\text{perdrix}) \vee \text{oiseau}(\text{perdrix})) \wedge \text{Plumes}(\text{perdrix}) \wedge \neg \text{Oiseau}(\text{perdrix})$

□ Or $(p \Rightarrow q) \Leftrightarrow (\neg p \vee q)$: Loi de Morgan

$(\neg \text{Plumes}(\text{perdrix}) \vee \text{Oiseau}(\text{perdrix})) \Leftrightarrow (\text{Plumes}(\text{perdrix}) \Rightarrow \text{Oiseau}(\text{perdrix}))$

□ Or $((p) \wedge (p \Rightarrow q)) \Rightarrow q$: Modus ponens

$((\text{Plumes}(\text{perdrix})) \wedge (\text{Plumes}(\text{perdrix}) \Rightarrow \text{Oiseau}(\text{perdrix}))) \Rightarrow \text{Oiseau}(\text{perdrix})$

□ Récapitulatif des Axiomes :

$\neg \text{plumes}(\text{perdrix}) \vee \text{oiseau}(\text{perdrix}), \text{Plumes}(\text{perdrix}), \neg \text{Oiseau}(\text{perdrix}), \text{Oiseau}(\text{perdrix})$

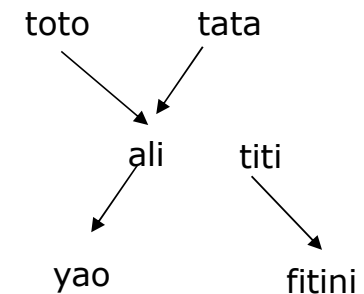
□ Contradiction : $\neg \text{Oiseau}(\text{perdrix})$ et $\text{Oiseau}(\text{perdrix})$

□ Conclusion : $\text{Oiseau}(\text{perdrix})$ est donc un théorème

V. La logique des prédicats

■ Exemple de programme en Prolog

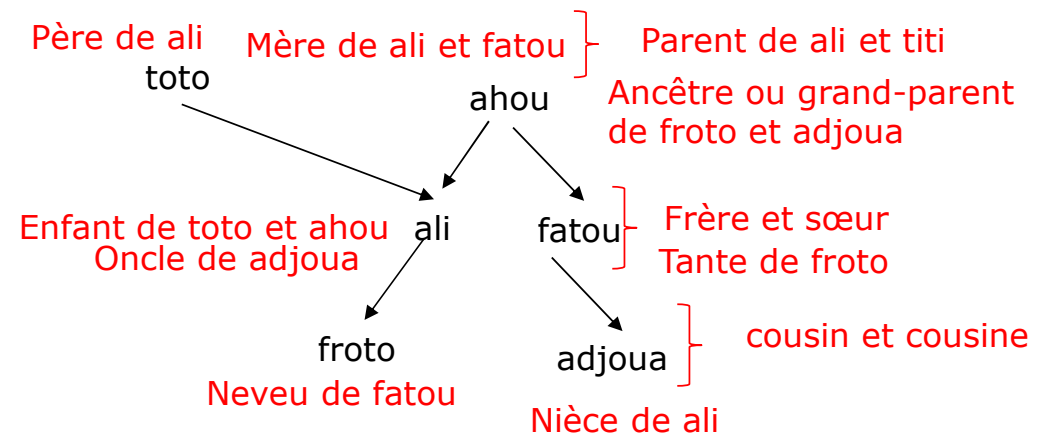
<pre>/*Définition des faits*/ homme(toto). homme(ali). homme(yao). femme(tata). femme(titi). femme(fitini). parent(toto, ali). %toto est le père de ali parent(tata, ali). %tata est la mère de ali parent(ali, yao). %ali est le père de yao parent(titi, fitini). /* Définition des règles */ enfant(X, Y) :- %X est enfant de Y si parent(Y, X). %Y est parent de X</pre>	<pre>/*Question*/ ?- parent(toto, ali). ?- parent(X, ali). ?- enfant(X, toto).</pre>
---	--



V. La logique des prédicats

■ Exercice : traduire les assertions suivantes en prédicats :

- **toto est un homme, ali est un homme, froto est un homme.**
- **fatou est une femme, adjoua est une femme, ahou est une femme.**
- **toto est le père de ali, ahou est la mère de ali**
- $\forall x, y, x$ est parent de y si x est le père ou la mère de y .
- $\forall x, y, x$ est enfant de y, x est frère de y, x est sœur de y
- $\forall x, y, x$ est cousin de y, x est neveu de y, x est oncle de y



VI. Réseaux sémantiques

□ 6.1. Présentation

■ Fondateur

- Quillian (1966)
- Issue de ses travaux sur les modèles de mémoires associatives.

■ Utilisation

- A l'origine, ils sont utilisés dans des recherches liées à la compréhension du langage naturel.

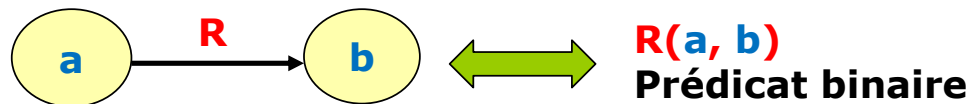
■ Idées principales

- Signification d'un concept vient de ses relations avec d'autres concepts.
- Information représentée en interconnectant des nœuds par des arcs étiquetés.
- Les réseaux sémantiques se présentent comme des ensembles de points ou nœuds étiquetés.

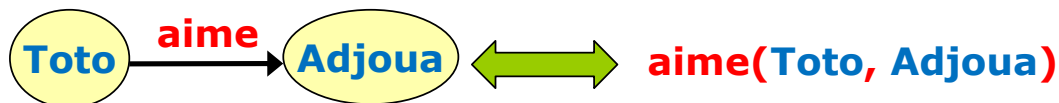
VI. Réseaux sémantiques

□ 6.2. Représentation de la connaissance

- Utilisation d'un graphe orienté
 - Nœud = concept
 - Arc = propriétés des relations ou actions possibles sur les sommets.



■ Exemple :



- Formalisme déclaratif complété par un ensemble de règles d'inférences représentées à l'aide du calcul des prédicats.
- Relations de n'importe quel type et portent la sémantique de l'arc.

VI. Réseaux sémantiques

■ Deux sortes de liens entre les nœuds :

□ **est-un (isa)** :

- décrit le fait qu'un concept est considéré comme une instance d'une famille d'objets.
- Correspond à l'appartenance (\in) en théorie des ensembles.

□ **sorte-de (a kind of – ako)** :

- décrit le fait que le réseau considéré est un sous-réseau d'un réseau donné.
- Correspond à l'inclusion (\subset) en théorie des ensembles.

■ Héritage des propriétés

- Chaque fois que le lien est-un est présent.

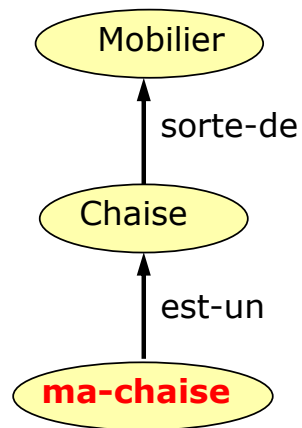
■ Représentation liée au langage LISP et les SE EMYCIN, PROSPECTOR, SRL.

- Exemples de systèmes : ATN, SCRIPTS et MOPS.

VI. Réseaux sémantiques

■ Exemple 1 :

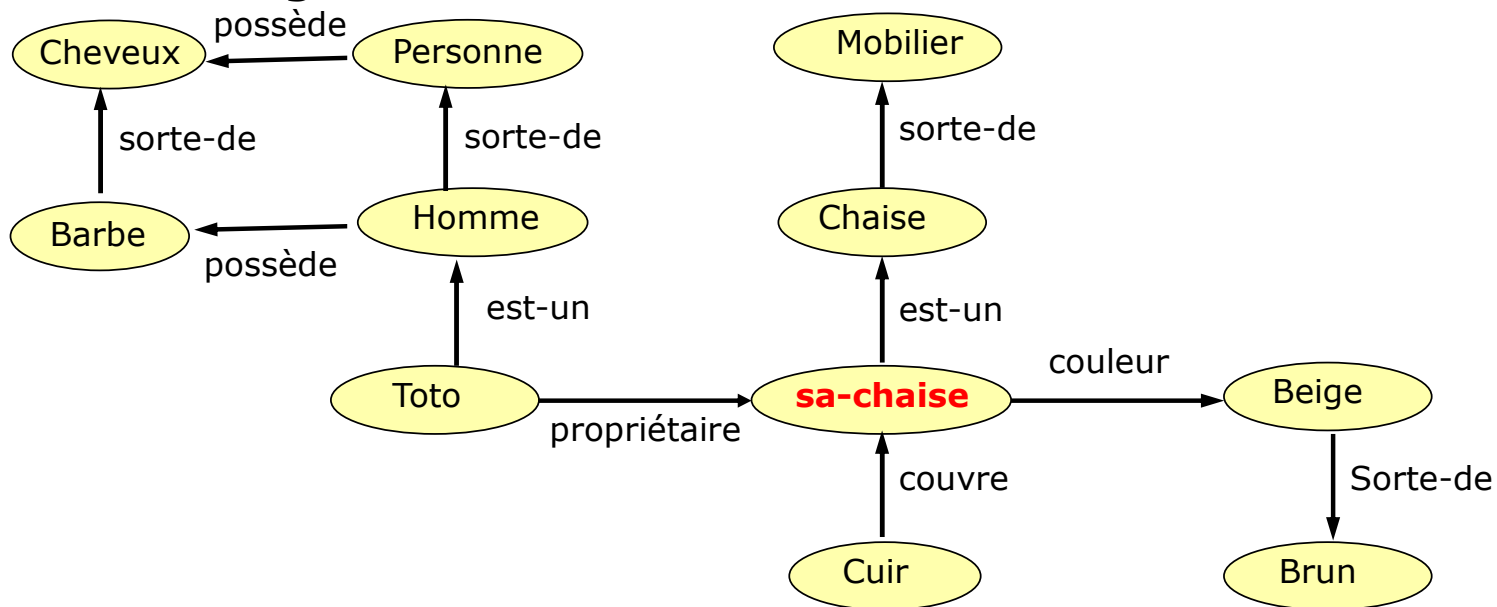
□ Représentation de l'objet "ma chaise"



VI. Réseaux sémantiques

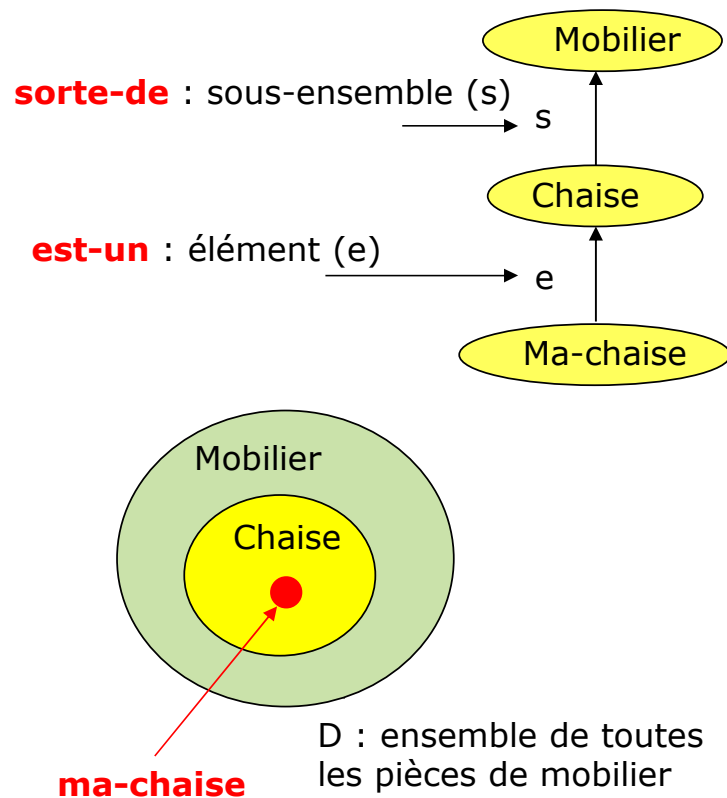
■ Exemple 2 :

- Représentation de « La chaise en cuir de Toto est de couleur beige ».



VI. Réseaux sémantiques

■ Interprétation

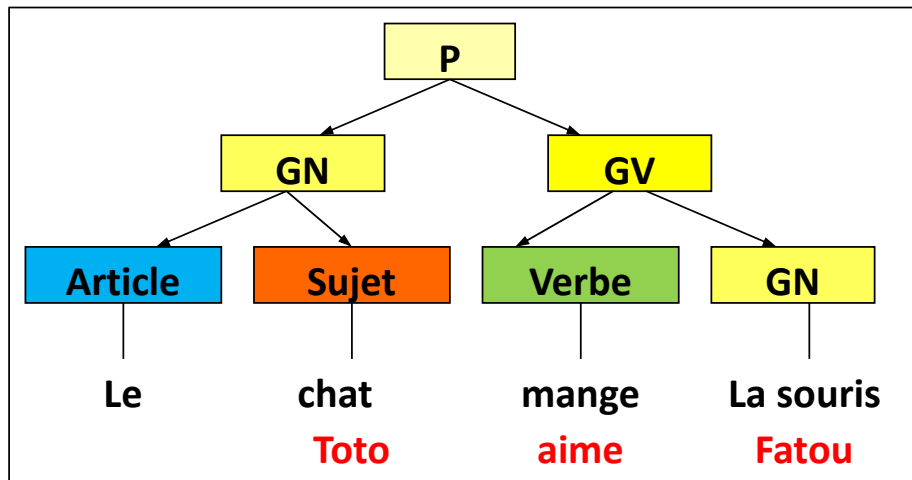


- **D** : domaine d'interprétation
- **I(A)** : interprétation de A
- "**Chaise** → **s** → **Mobilier**" est vrai ssi **$I(\text{Chaise}) \subseteq I(\text{Mobilier})$**
 - Logique des Prédicats du 1^{er} Ordre : **$\forall x \text{Chaise}(x) \Rightarrow \text{Mobilier}(x)$**
- "**Ma-chaise** → **e** → **Chaise**" est vrai ssi **$I(\text{Ma-chaise}) \in I(\text{Chaise})$**
 - Logique des Prédicats du 1^{er} Ordre : **$A(a)$**
 - **$\text{Chaise}(\text{Ma-chaise})$, $\text{Chaise}(\text{Sa-chaise})$**

VI. Réseaux sémantiques

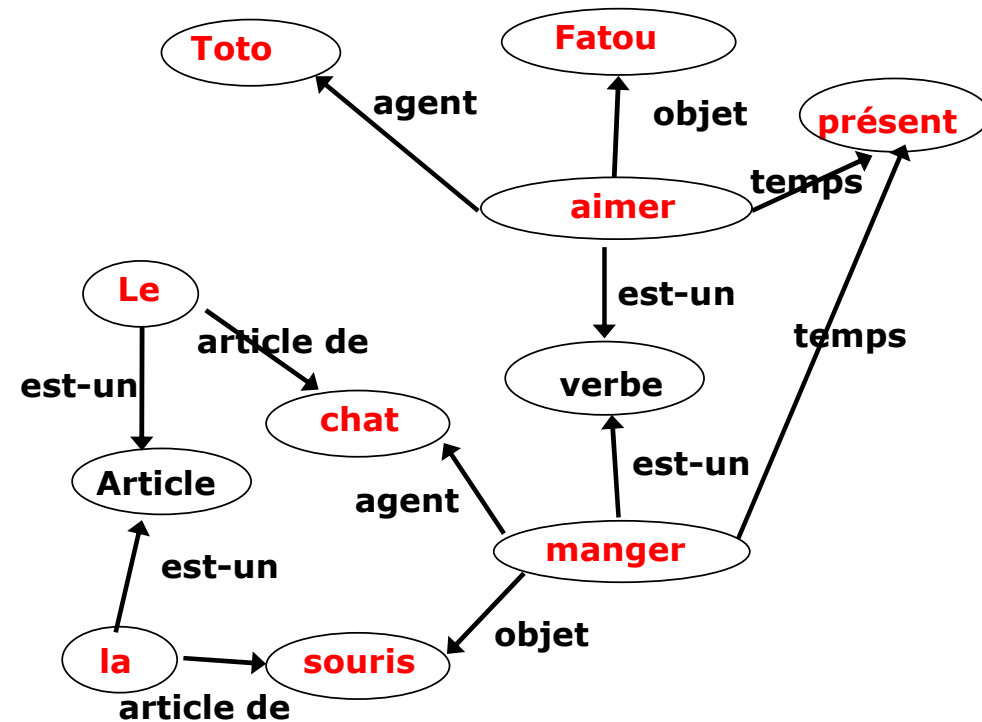
□ 6.3. Représentation d'une phrase

■ Syntaxe



■ Agent et objet d'un verbe

- Agent : fait l'action
- Objet : subit l'action



VI. Réseaux sémantiques

□ 6.4. Inférence dans les réseaux sémantiques

■ Mécanisme :

- Suivre les arcs reliant les nœuds.

■ 2 méthodes pour réaliser l'inférence :

□ par recherche d'intersection :

- On propage l'activation à partir de 2 nœuds et en trouvant les intersections des activations on trouve les relations entre objets.

□ par héritage :

- les relations "est-un" et "sorte-de" permettent de suivre les liens d'héritage dans une taxonomie hiérarchique.

- L'héritage permet aussi de faire du raisonnement par défaut (on peut remplacer une instance du sommet hérité par une instance du sommet héritier).
- On peut déduire que « machine » est un mobilier
- On peut déduire que « Toto est un homme qui peut avoir une barbe qui est une sorte de cheveux. »
- On peut déduire que « Toto peut avoir une barbe et des cheveux. »

VI. Réseaux sémantiques

□ 6.5. Critiques

■ **Avantage :**

- **Représentation aisée des connaissances**

■ **Inconvénients :**

- **Difficulté d'explication**
- **Les connaissances procédurales ne sont pas exprimées dans le réseau.**
- **Les réseaux sémantiques deviennent vite très complexes et peu lisibles.**

VI. Réseaux sémantiques

- **Exercices : Représenter les assertions suivantes :**
 - **« Hier, à Abidjan, l'ASEC a joué contre l'AFRICA. »**
 - **« La personne qui s'appelle Toto Ali possède une voiture appelée Peugeot 605. »**
 - **Avec les réseaux sémantiques ;**
 - **Avec la logique des prédicats.**

VII. Les schémas

□ 7.1. Présentation

■ Développé par M. Minsky (1970).

- C'est le fruit des réflexions de Minsky sur les systèmes de perception et de vision.
- Exemple : représenter les différentes perspectives d'un cube.

■ Schéma ou frame

- C'est une méthode, un modèle de représentation.
- Ensemble d'informations représentant des entités et leurs instances basées sur le concept de schéma.

■ Concept de schéma

- Correspond à une structure d'informations stéréotypées, associée à un concept donné.
- Elle s'oppose aux représentations relationnelles dans lesquelles un objet est décrit à travers ses propriétés disséminées dans des règles ou formules logiques.

VII. Les schémas

□ 7.2. Représentation de la connaissance

- Un frame est décrit par un ensemble d'attributs (slots).
- Chaque attribut est lui même décrit par un ensemble de facettes et leurs valeurs.
 - En LPO, les slots seraient des fonctions.
- Facettes décrivent la sémantique de l'attribut :
 - type, manière d'obtenir une valeur, etc....

- Contraintes de types et valeurs par défaut
- Systèmes de frames :
 - Ensemble de frames interconnectés

VII. Les schémas

■ Liaisons avec d'autres frames à travers des hiérarchies

□ Mécanisme d'inférence :

- **Subsumption** : un concept A subsume un concept B si l'ensemble des instances de A contient l'ensemble des instances de B.
 - A est appelé le subsumant et B le subsumé.
- **Classification** : opération qui permet de placer un concept donné dans un graphe d'héritage.

■ Systèmes de frames

- Représentent souvent la même entité vue sous différentes perspectives.

VII. Les schémas

□ 7.3. Structure d'un frame

- **Frame = restriction des réseaux sémantiques**
- **Structure à trois niveaux :**
 - **1^{er} niveau : nom du frame**
 - **2^{ème} niveau : attributs (ou slots)**
 - **3^{ème} niveau : facettes**

(Nom du frame		
(Attribut ₁	(Facette ₁	Valeur ₁)

	(Facette _n	Valeur _n))
(Attribut _k	(Facette ₁	Valeur ₁)

	(Facette _m	Valeur _m))
)		

■ **Slot :**

- **Décrit les différentes propriétés d'un frame**

■ **Facette :**

- **Modalité descriptive ou comportementale d'un slot.**
- **A une facette est toujours associée une valeur.**

■ **Attribut sorte-de :**

- **Tout frame contient un attribut « sorte-de » qui traduit le lien d'héritage entre frames.**
- **Sa valeur est une liste de nom de schémas hérités directement.**

VII. Les schémas

- **Exemple : Schéma**
personne, décrit par un nom, un
prénom, ...

```
(Personne
  (sorte-de      ($valeur Objet))
  (nom          ($est-un chaîne))
  (prénom       ($est-un chaîne))
  (dateNaissance ($est-un chaîne))
  (âge          ($est-un entier)
    ($intervalle [1 200])
    ($si-besoin (calculeAge âge)))
  (profession   ($est-un chaîne)
    ($default "chômeur"))
)
```


VII. Les schémas

■ Instance

- Une instance peut-être complète ou partielle. Elle hérite du schéma de sa classe à l'aide de la liaison « **sorte de** ».

■ Exemples :

```
(PersToto
  (sorte-de      ($valeur Personne))
  (nom           ($valeur "Toto"))
  (prénom       ($valeur "Ali"))
  (dateNaissance ($valeur "07/04/1961"))
  (âge          ($valeur 54))
  (profession    ($valeur "étudiant"))
)
```

```
(PersFatou
  (sorte-de      ($valeur Personne))
  (nom           ($valeur "Fatou"))
  (prénom       ($valeur "Fatou"))
  (dateNaissance ($valeur "20/10/1980"))
  (âge          ($valeur 35))
)
```

VII. Les schémas

□ 7.4. Les facettes

- Les facettes d'un attribut décrivent les diverses connaissances sur cet attribut :

- nature ;
- valeur ;
- valeur par défaut ;
- moyens d'obtenir sa valeur.

■ Facette de type

- Type simple : entier, réel, booléen et chaîne.
- Type défini par un frame : référence à une instance de ce frame ou d'un frame plus spécifique.

Facette	Explication
\$est-un	Précise le type

VII. Les schémas

■ Facette de valeur

- Décrit un moyen d'obtenir la valeur d'un attribut.

Facette	Explication
\$valeur	Précise la valeur de l'attribut
\$default	Permet d'associer une valeur par défaut qui sera retenue en cas d'absence d'autres informations sur la valeur de l'attribut
\$card-min	Valeur minimale
\$card-max	Valeur maximale

■ Facette procédurale

- Décrit un moyen d'obtenir la valeur d'un attribut par calcul.
- On parle d'attachement procédural.

Facette	Explication
\$si-besoin	Permet d'associer des méthodes de calcul des valeurs
\$a-verifier	Vérifier une condition

VII. Les schémas

■ Facette réflexe

- Appelées aussi "démon", les réflexes sont déclenchées lorsqu'une valeur est effectivement donnée à un attribut.
- Elles permettent de maintenir la cohérence de la base d'instances en propageant les modifications.

Facette	Explication
\$si-ajout	Que faire si la valeur est ajoutée (pour un attribut multi-valué),
\$si-supprime	Que faire si la valeur est supprimée
\$si-modification	Que faire si la valeur est modifiée

□ Exemple

```
(Personne
  (sorte-de      ($valeur Objet))
  (nom           ($est-un chaîne))
  (prénom        ($est-un chaîne))
  (dateNaissance ($est-un chaîne)
    ($si-modification ($calculeAge âge)))
  (âge           ($est-un entier)
    ($intervalle (1, 200))
    ($si-besoin calculAge))
  (profession    ($est-un chaîne)
    ($default "chômeur"))
)
```

VII. Les schémas

■ Facette de restriction de type

- Permettent de décrire des contraintes auxquelles la valeur de l'attribut doit obéir.

Facette	Explication
\$domaine	une liste de prédicats décrivant des contraintes sur la valeur de l'attribut
\$restriction	une liste de prédicats décrivant des contraintes sur la valeur de l'attribut
\$intervale	intervalle des valeurs admissibles
\$sauf	un ensemble de valeurs possibles

□ Exemple

```
(Jour
  (sorte-de ($valeur Objet))
  (nom ($domaine (dimanche, lundi, mardi,
                  mercredi, jeudi, vendredi, samedi))
  (numéro ($est-un entier)
    ($restriction ((>= 1) (<=7)))
  )
```

VII. Les schémas

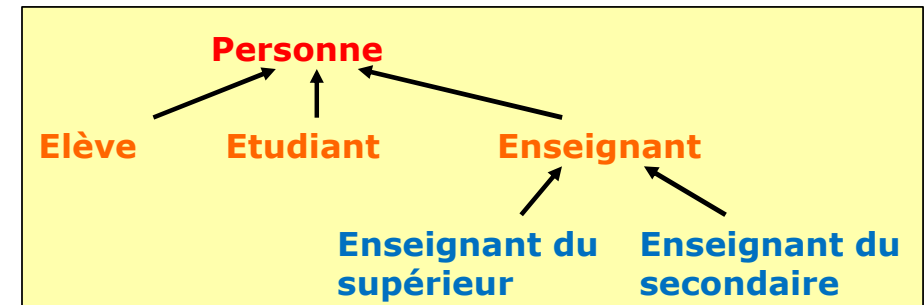
□ 7.5. Hiérarchie de schémas

- **Schémas organisés dans une structure de treillis appelée "hiérarchie de schémas".**
- **Attribut sorte-de**
 - Traduit une relations entre objets dans une taxonomie hiérarchique.
- **Dans la hiérarchie :**
 - **Pas de cycle.**
 - **principe de spécialisation**
 - Relation de de haut vers le bas, affinement de la description des schémas de niveau supérieur.

□ Principe de généralisation

- Relation de bas en haut, généralisation des schémas de niveau inférieur;

■ Exemple : hiérarchie personne.



VII. Les schémas

■ Exemple :

- Schéma Etudiant, décrit par un matricule et une classe
- toutes les propriétés de Personne sont aussi les propriétés de Etudiant.

```
(Etudiant
  (sorte-de ($valeur Personne))
  (matricule ($est-un entier))
  (classe ($est-un chaîne))
)
```

□ Une instance de Etudiant

```
(EtudiantToto
  (sorte-de ($valeur Etudiant))
  (nom ($valeur "Toto"))
  (prénom ($valeur "Ali"))
  (dateNaissance ($valeur "07/04/61"))
  (âge ($valeur 47))
  (matricule ($valeur 1234))
  (classe ($valeur "6e"))
)
```

- Les systèmes basés sur le formalisme "schéma" les plus connus sont KRL (Bobrow), FRL (Goldstein), SHIRKA et UNITS (Stefik).

VII. Les schémas

■ **Avantages :**

- **Représentation naturelle des connaissances :**
 - On représente les objets en tant que tel.
- **Flexibilité : possibilité d'utiliser la représentation pour résoudre des problèmes de natures différentes.**
- **Modularité : objets considérés comme des boîtes noires que l'on peut temporairement déconnecter du reste du monde.**
 - implication dans le problème des seuls objets y intervenant.

□ **Connaissance hiérarchisées (héritage)**

- Ce qui facilite la spécialisation des connaissances.

■ **Inconvénients :**

□ **Mécanisme de raisonnement pauvre**

- Implémenter des connaissances dynamiques pour inférer et utiliser la sémantique des objets : leurs contenus plutôt que leurs noms.

VII. Les schémas

- **Exercice**

- **Représenter les schémas :
Enseignant (spécialité, compte) et
Enseignant du supérieur
(ingénieur, docteur)**

VIII. Graphes conceptuels

□ 8.1. Présentation

- Graphe Conceptuel (GC)
- Trouvent leurs fondements en linguistique, en psychologie, en philosophie et en logique.
- Systèmes logiques développés pour la représentation des sens des phrases en langage naturel.
- Offrent une notation de la logique plus proche des propositions en langage naturel que la logique des prédicats du 1^{er} ordre.

■ Exemple : affirmation « un chat est sur le toit »

□ Logique du 1^{er} ordre :

- $(\exists x) (\exists y) (\text{chat}(x) \wedge \text{toit}(y) \wedge \text{sur}(x, y))$

□ Graphes conceptuels

- [CHAT] > (SUR) > [TOIT]

VIII. Graphes conceptuels

□ 8.2. Représentation de la connaissance

■ GC comporte deux types de nœuds :

□ Concepts et relations conceptuelles

- Toute relation conceptuelle a un ou plusieurs arcs chacun lié à un concept.
- Un concept simple peut être considéré comme un graphe.

[Concept] —→ (Relation) —→ [Concept]



- Possible de représenter les phrases du langage naturel et expliciter le sens des composants de la phrase.

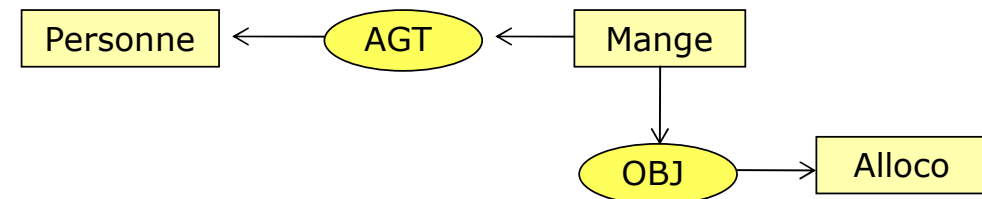
VIII. Graphes conceptuels

■ GC est :

- **Orienté** : sens de lecture de la relation ;
- **Fini** : tout graphe dans une mémoire d'un ordinateur ne peut avoir qu'un nombre fini de nœuds ;
- **Connexe** : si deux parties n'étaient pas connectées entre elles on aurait deux graphes conceptuels ;
- **Bipartie** : il ne possède que deux sortes de nœuds :
 - Les concepts et les relations conceptuelles.
 - Chaque arc reliant une sorte de nœud à l'autre sorte de nœud.

□ Exemple : Représenter la phrase suivante :

- "Une personne mange de l'alloco."



VIII. Graphes conceptuels

□ 8.3. Éléments du modèle

■ Concepts

- « Toute idée, toute pensée, ou toute construction mentale au moyen de laquelle l'esprit appréhende les choses ou parvient à les reconnaître ».

■ Référent du concept :

- Un concept est formé par deux éléments principaux :
 - Le type du concept et le référent du concept.

- Le type du concept est une abstraction de l'ensemble des référents du concept.

- Par exemple "MEDECIN" est un type qui représente la classe de tous les médecins.

□ Notation

- [**<Type>**: **< Référent>**]

□ Exemple :

- [PERSONNE : MAX]
 - [PERSONNE : # 804]

VIII. Graphes conceptuels

■ Concept générique

- Représente un individu quelconque du type donné
 - variables en logique
- Le référent du concept n'est donc pas indiqué.
- [**< Type générique >**]
 - Exemple : [HOMME]

□ Le référent peut être :

*	indique un individu de la classe du concept (comme un concept générique). Exemple : [HOMME: *] idem [HOMME]
#	suivi d'un numéro indique un concept individuel (constante en logique). Exemple : [HOMME: #123], l'homme dont le numéro est 123.
instance	Soit une instanciation du concept : un individu est donné par son nom. Exemple : [HOMME: TOTO]
@	indique une mesure
SET	indique un ensemble. Exemple : SET (X1&X2 & ...Xn) : un ensemble de conjonctions X1,..., Xn. [PERSONNE: SET ('TOTO', 'ALI')] SET (X1/X2/.../Xn) : un ensemble de disjonctions de X1,..., Xn.

VIII. Graphes conceptuels

□ 8.4. Hiérarchie des concepts

- Permet de généraliser ou de spécialiser les concepts.
- Notion fondamentale pour tout raisonnement sur les GC :
 - Relation de spécialisation/généralisation (ou subsomption).

■ Hiérarchie (ou treillis) des concepts regroupe les concepts dépendant du même hyperonyme.

- Exemple : Meuble désigne l'ensemble des concepts relatifs au mobilier.
 - Meuble est un hyperonyme de chaise, bureau, armoire, etc....

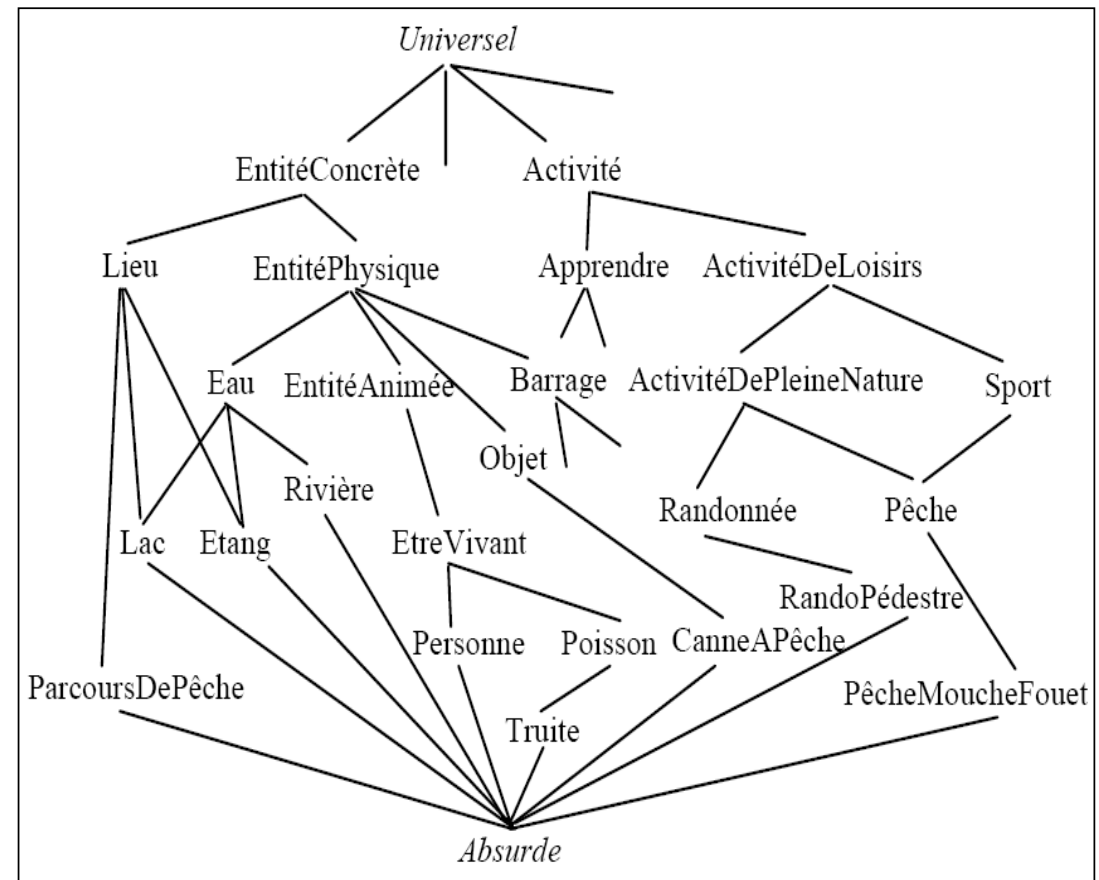
VIII. Graphes conceptuels

- **Concepts ordonnés par une relation d'ordre partiel : \leq**
 - **Réflexive, antisymétrique, transitive**
 - **Exemple :**
 - « Table \leq Meuble » signifie « une table est une sorte de Meuble ».
- **Sur-type et sous-type : position de deux concepts dans la hiérarchie**
 - **$\forall t$ et s deux types de concepts, si $t \leq s$ alors**
 - t est un sous-type de s
 - s est un sur-type de t

VIII. Graphes conceptuels

■ Exemple de hiérarchie

- **Concept le plus générique : UNIVERSELLE**
- **Tout concept qui n'a pas de sous-type est relié à ABSURDE.**



VIII. Graphes conceptuels

□ 8.5. Relation conceptuelle

- Définit les liens et spécifient les rapports qui existent entre les concepts du graphe.

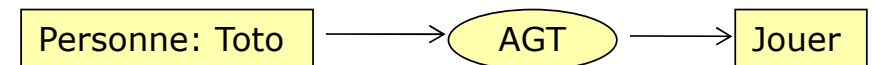
■ Notation

- Une relation se lit toujours dans le sens des flèches :

[C1] —→ (Relation) —→ [C2]

- Signifie que « C1 a pour RELATION C2 »

■ Exemple :

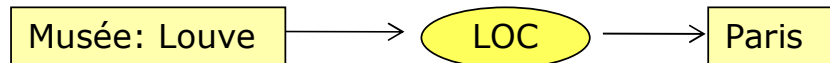


- Relation (AGT) lie l'action «jouer» à la personne qui l'exécute «Toto».
- Le graphe se lit «Jouer a pour agent Toto».
- La relation (AGT) du graphe lie l'action «jouer» à la personne qui l'exécute «Toto». Et le graphe se lit «Jouer a pour agent Toto».

VIII. Graphes conceptuels

- **D'une manière générale, une relation entre concepts spécifie le rôle joué par ces concepts dans le graphe.**

- **Exemple :**



- **S'interprète comme : la ville de Paris est la localisation du musée du Louvre.**

- **Une relation conceptuelle peut avoir un nombre quelconque d'arguments.**

- **Types de relations :**

Relation	Explication
AGT	Agent (entité intervenant de façon active et directement dans le procès)
PAT	Patient (entité intervenant de façon passive dans le procès)
OBJ	Objet (entité affectée par le procès)
INST	Instrument (moyen par lequel un agent agit pour un résultat ou une cause)
LOC	Lieu
TEM	Temps
DEST	Destination (aboutissement qui peut être de nature spatiale)
ORIG	Origine (provenance spatiale ou abstraite)
CRC	Caractéristique
MNR	Manière
APP	Appartenance
POSS	Possession

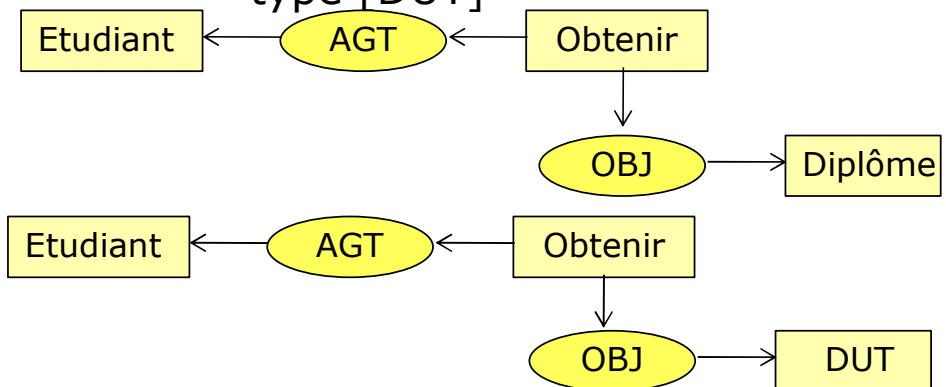
VIII. Graphes conceptuels

□ 8.6. Opérations sur les graphes conceptuels

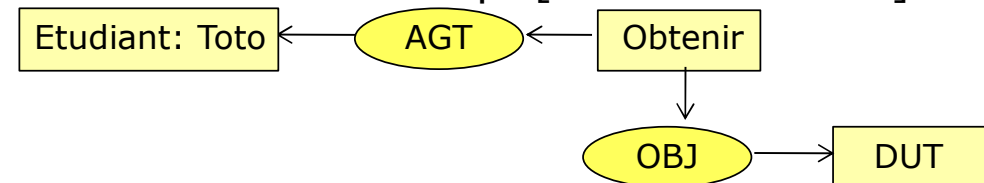
- **Combiner les GC en utilisant diverses opérations.**
- **Copie d'un graphe :**
 - **consiste à construire un GC identique à celui de départ.**
- **Restriction :**
 - **Consiste à remplacer un type de concept GC par un de ses sous-types ;**
 - **s'il s'agit d'un concept générique, son référent peut devenir un marqueur individuel.**

□ Exemple :

- Le type de concept [DIPLOME] a été remplacé par un de ses sous type [DUT]



- Le Concept générique [ETUDIANT] a été individualisé au concept [ETUDIANT: Toto]



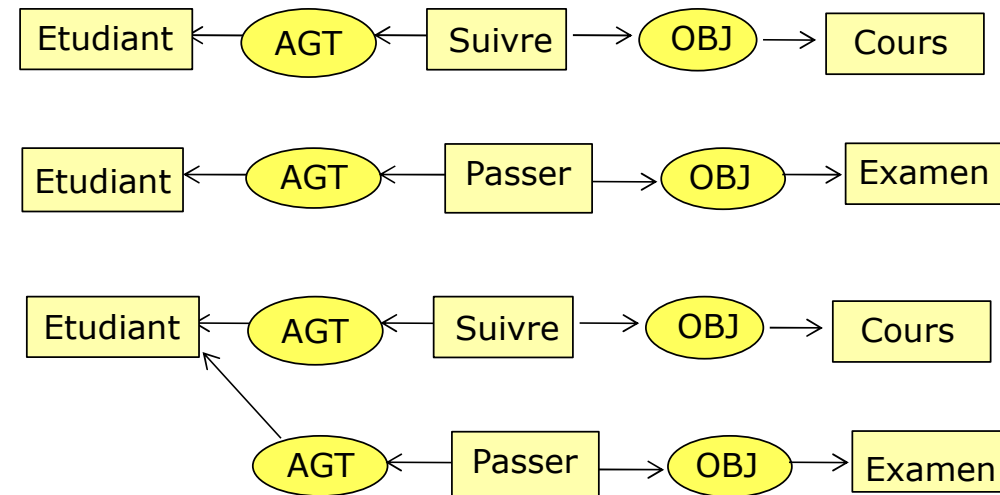
VIII. Graphes conceptuels

■ Jointure des deux GC

□ Se définit formellement comme suit :

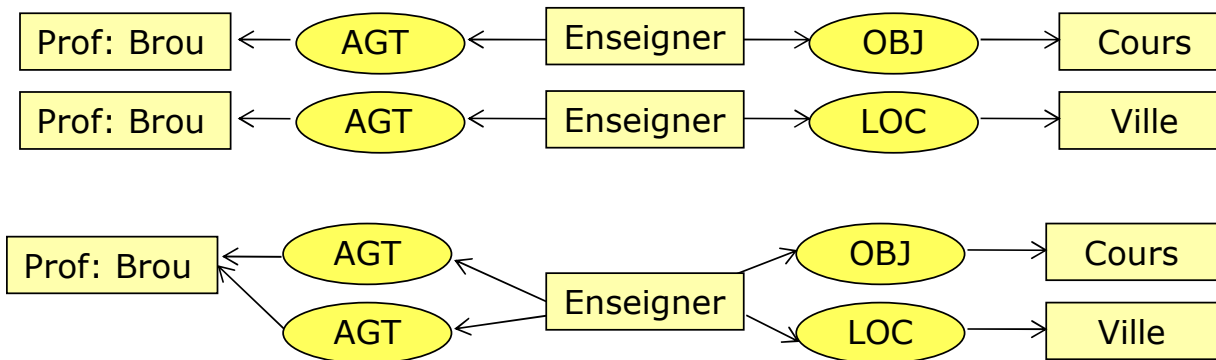
- Soit u et v deux graphes conceptuels tels que :
- il existe au moins $i \in [1...n]$ et au moins $j \in [1...p]$ tels que $c_{ui} = c_{vj}$ alors on peut réaliser une opération de jointure de u et de v dont le résultat est le GC

□ Exemple 1: La jointure sur le concept commun [Etudiant]



VIII. Graphes conceptuels

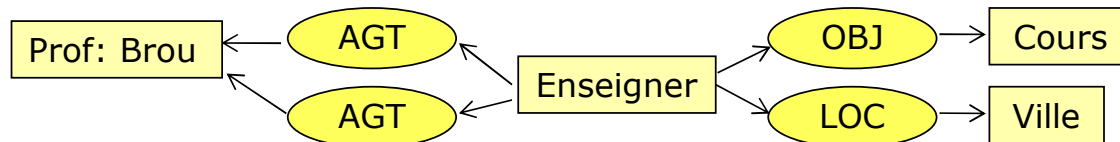
- **Exemple 2 : deux concepts communs [Professeur : Brou] et [Enseigner]**



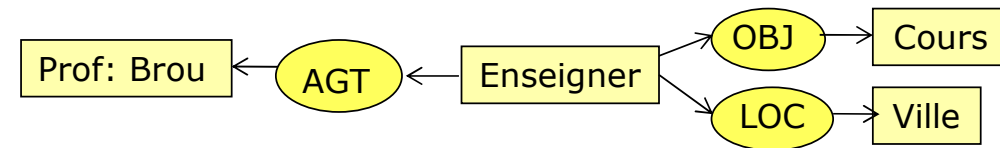
VIII. Graphes conceptuels

■ Simplification

- Si deux relations conceptuelles de u sont dupliquées, c'est-à-dire que deux relations identiques relient deux mêmes concepts (suite par exemple à une opération de jointure), alors l'une d'elles est enlevée ainsi que les arcs reliés à celle-ci.



- On peut faire une simplification sur le GC ci-dessus car la relation conceptuelle (AGT) est dupliquée entre les concepts [Professeur] et [Enseigner].



VIII. Graphes conceptuels

- **Exercice : Représenter la phrase suivante :**
 - **"Le lac Bayard résulte d'un barrage sur une rivière ; il contient des truites que l'on peut pêcher avec une canne à pêche."**

Bibliographie

□ Livres

- **"Les Systèmes Experts, Principes et exemples", H Farreny, CEPADUES Edition**

Bibliographie

□ Webographie

- <http://www.loria.fr/~napoli/CNAM/regles-041007-4.pdf>
- <http://www-poleia.lip6.fr/~jfp/insia/Cours/Cours6.pdf>
- <http://cui.unige.ch/DI/cours/1815/slides/12-representationStructuree.pdf>
- <http://www.lirmm.fr/~chein/chapGC.pdf>
- http://www.revue-texto.net/marges/marges/Documents%20Site%206/doc0004_chawk_m/graphc.pdf
- <ftp://ftp.inrialpes.fr/pub/sherpa/theses/marino.ps.gz>
- <http://brassens.upmf-grenoble.fr/IMSS/dciss/Enseignements/SCRC/RC/coursrpo.pdf>
- http://liris.cnrs.fr/amille/enseignements/DEA-ECD/site_ia_emiage/session3/syst%E8mes_experts_%E0_r%E8gles.htm