

# Ontologies et Web sémantique

---

## Cours 2: Représentation des ontologies

Dr. TA Tuan Anh  
[ttanh@ciid.vast.vn](mailto:ttanh@ciid.vast.vn)

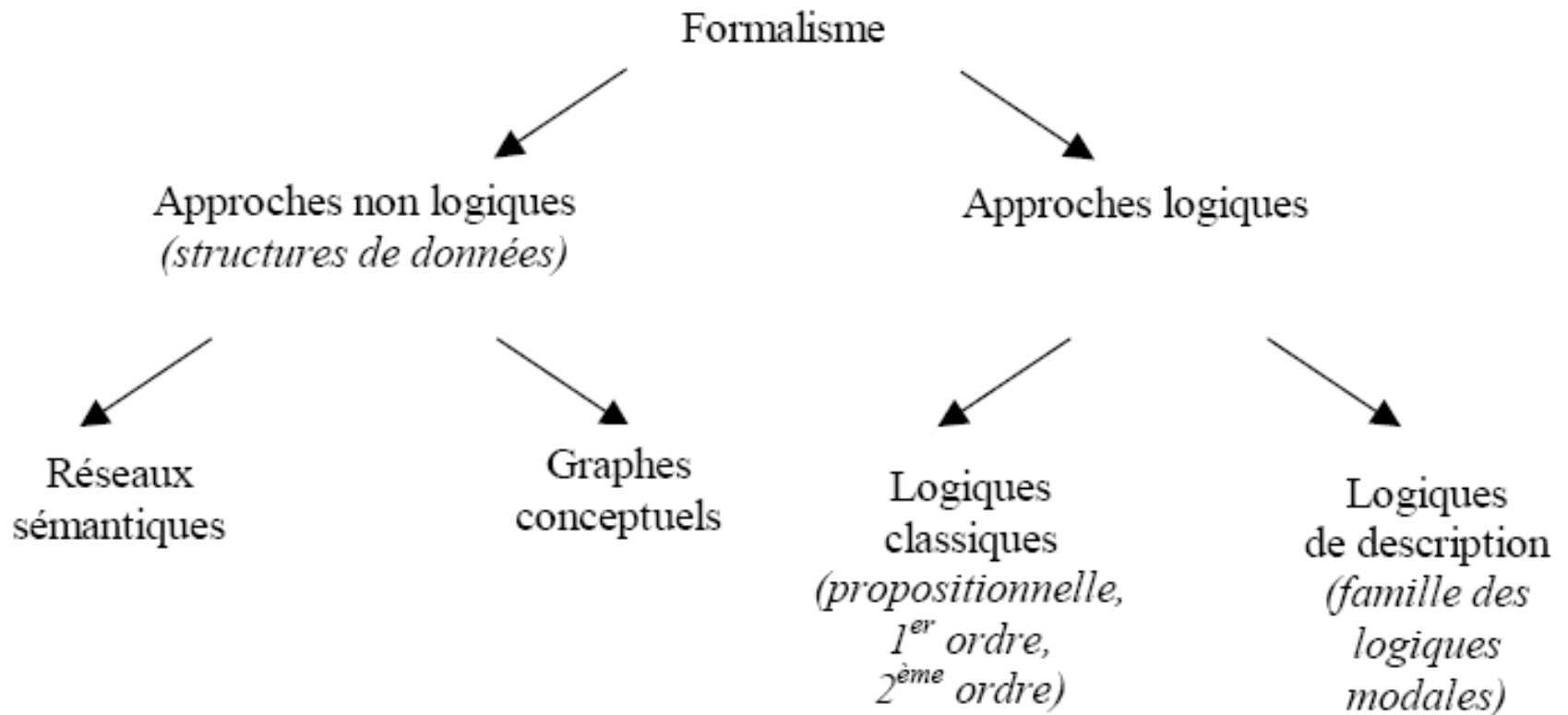
# Historique (rappel)

---

- ❑ Réseaux sémantiques (Brachman, Levêques)
- ❑ Logique du 1er ordre : CycL, KIF
- ❑ Frames : Frame Logic, Ontolingua
- ❑ Logiques de description
- ❑ Graphes conceptuels de Sowa
- ❑ Standards : DAML (Drapa) et OIL (Europe), OWL (W3C),  
...

# Une classification

---



# Caractéristiques des approches

---

- Approches non logiques
  - le sens intrinsèque attribué aux structures (objet en soi)
  - l'existence de procédures adaptées et souvent efficaces et une plus grande facilité de révision
- Approches logiques
  - le sens externe (interprétation, modèle)
  - des procédures génériques plus ou moins adaptées selon le choix de la logique
  - la possibilité de caractériser des propriétés du formalisme et des procédures (expressivité, complétude,...)

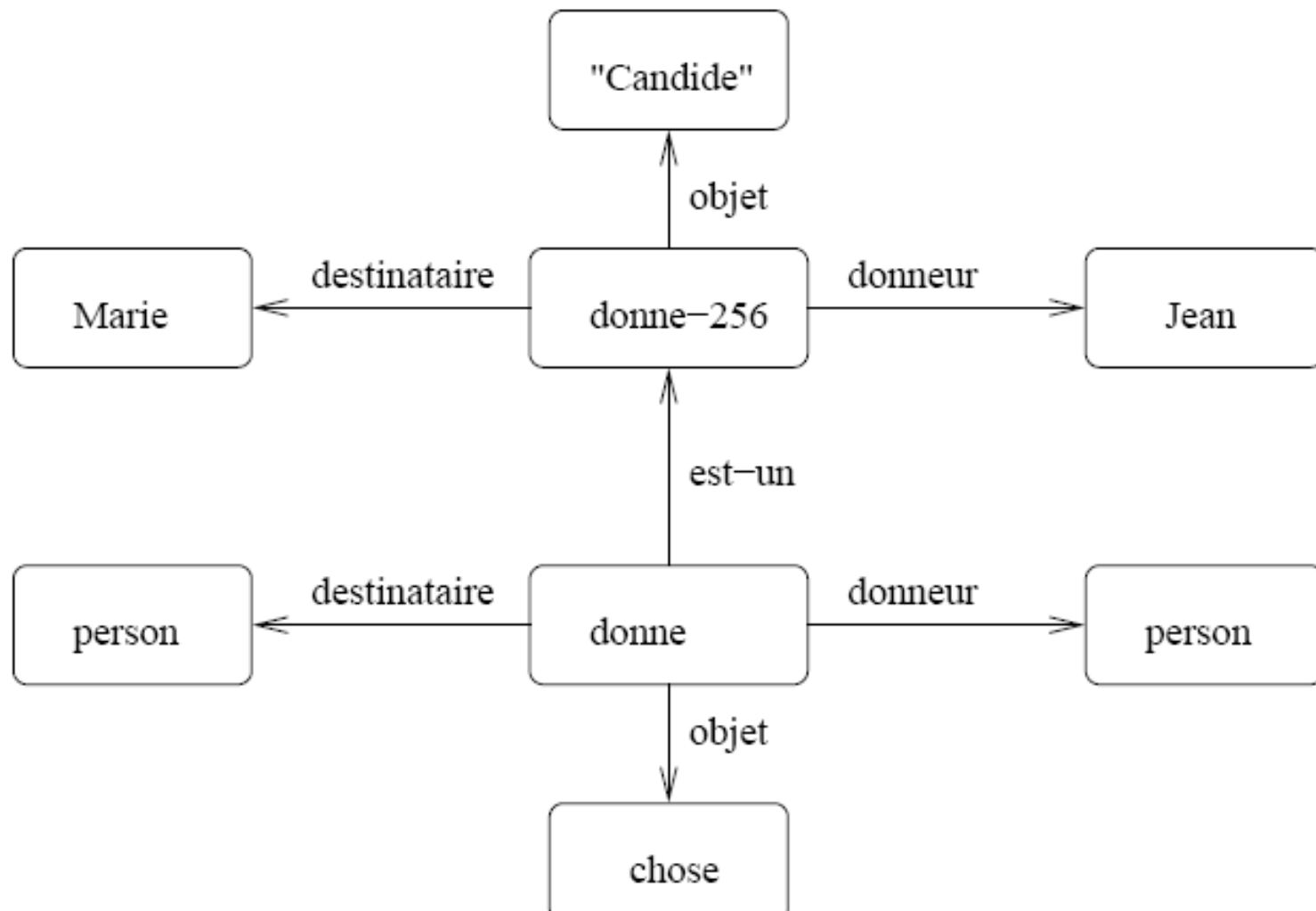
# Réseau sémantique

---

- Un graphe orienté acyclique dont :
  - les noeuds et les arcs sont étiquetés
  - les noeuds sont concepts,
  - les arcs représentent des relations entre concepts
- Tentative pour représenter le raisonnement de façon symbolique, sans règles

# Exemple

---



# Caractéristiques des réseaux sémantiques

---

## □ Intérêts

- définition d'un noeud par ses relations avec d'autres, capacités de définir un noeud comme une instance d'un autre
- héritage de propriétés, et plus généralement déductions de propriétés par parcours de graphe

## □ Inconvénients

- Une difficulté à comprendre la sémantique des nœuds et des liens (inadéquation logique) :
  - impossible d'exprimer simplement certaines distinctions logiques (un objet particulier, tous les objets, un objet quelconque, aucun objet...)
  - la signification des liens est mise dans l'implémentation
- Manque de stratégie dans la recherche d'information et de connaissances

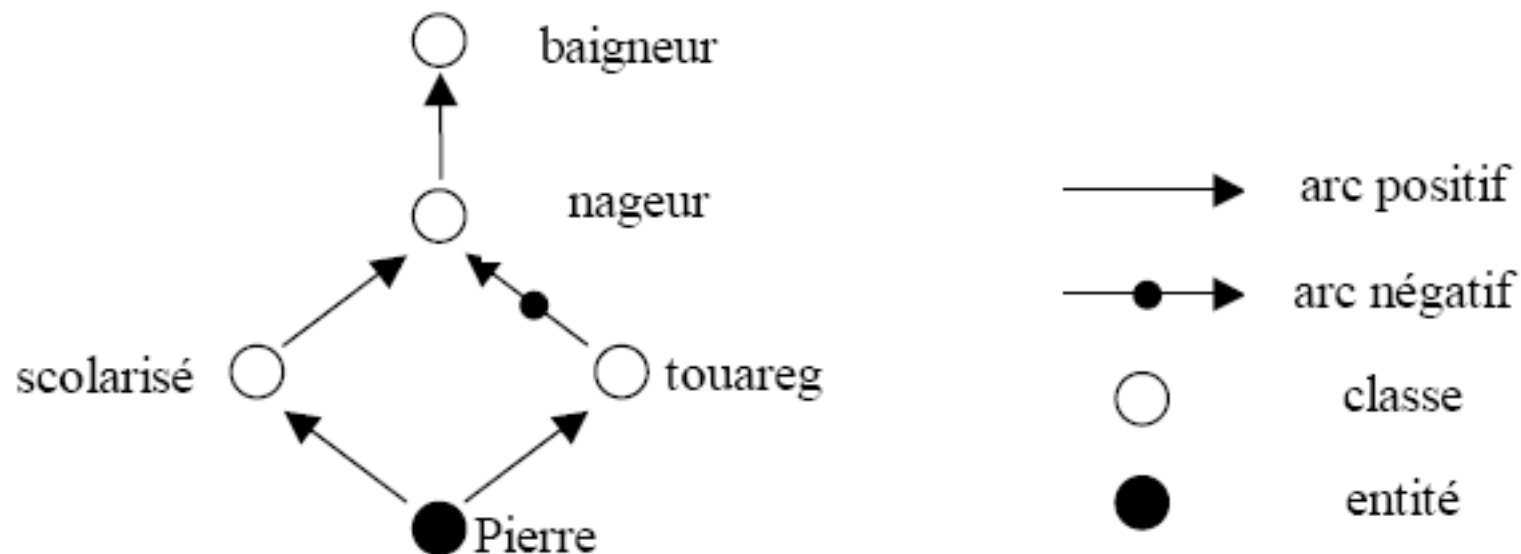
# Un modèle de réseaux sémantiques

---

- ❑ Les noeuds sont soit des classes d'entités, soit des entités spécifiques
- ❑ Les arcs sont de deux types arcs positifs et arcs négatifs
- ❑ Un arc positif reliant un noeud C à un noeud D signifie : C est généralement un D
- ❑ Un arc négatif reliant un noeud C à un noeud D signifie : C n'est généralement pas un D



# Exemple



- ❑ Un nageur est généralement un baigneur
- ❑ Un élève scolarisé sait généralement nager puisqu'il a suivi une ou deux années d'initiation à la natation
- ❑ Un touareg ne sait généralement pas nager
- ❑ Pierre est (selon toute vraisemblance) un touareg qui a été scolarisé

# Questionnement du réseau

---

- Effectuer la fermeture transitive du graphe selon les arcs positifs
  - A est généralement un B, B est généralement un C, alors A est généralement un C
  - E.g., Pierre est un élève scolarisé, touareg, nageur et baigneur
- Si un arc négatif relie E à D et C est généralement un E (transitive ou non), alors C n'est pas généralement un D
  - E.g., Pierre n'est pas généralement un nageur
- Impossible de conclure à partir d'un chemin composé de deux arcs négatifs

# Ambiguïté

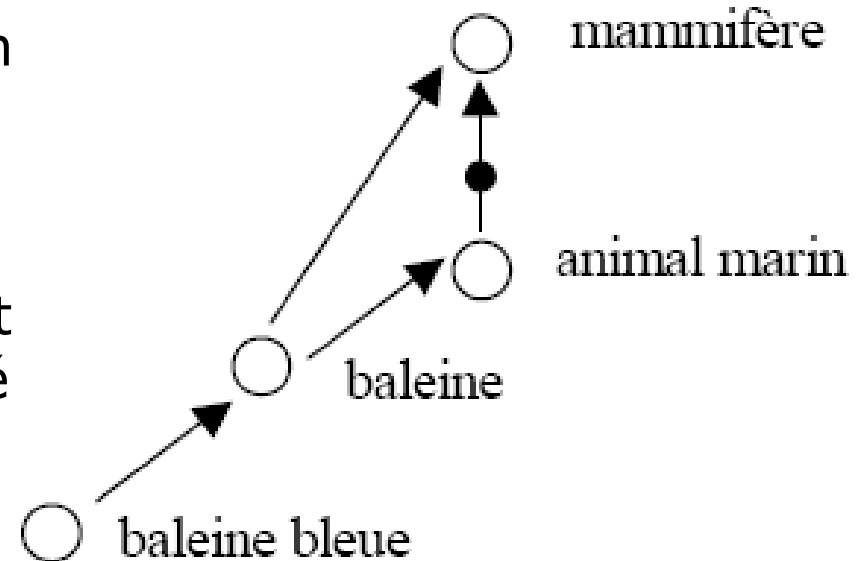
---

- ❑ Nouvel algorithme pour traiter correctement l'ambiguïté. Il importe de traiter simultanément les conclusions négatives et positives
- ❑ Quatre classifications de C vis à vis de D sont possibles :
  - Manque d'information.
  - C est généralement un D.
  - C n'est généralement pas un D
  - Ambiguïté
- ❑ E.g., Pierre est un élève scolarisé et un touareg, il y a ambiguïté quant à son statut de nageur et absence d'informations pour celui de baigneur

# Spécificité

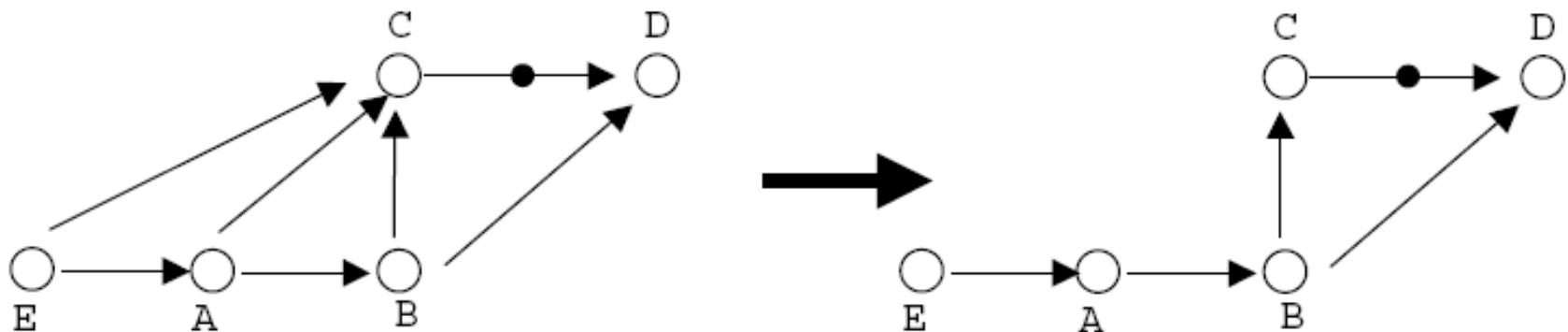
---

- ❑ L'ambiguïté apparente est levée en considérant que l'arc baleine-mammifère est plus spécifique que le chemin baleine-animal marin-mammifère
- ❑ Pour conclure "les animaux marins ne sont généralement pas des mammifères excepté les baleines"



# Redondance

---



- ❑ Aucun des chemins E-C-D et A-B-D n'est plus spécifique
- ❑ Si on supprime maintenant les liens redondants (E-C, A-C), l'ambiguïté disparaît
- ❑ Le chemin E-A-B-D est plus spécifique que le chemin E-A-B-C-D. E est donc généralement un D.

# Exercice

---

- Ecrire les algorithmes de raisonnement
  - Naïf
  - Détectant l'ambiguïté
  - Détectant la spécificité
  - Supprimant les redondances

# Graphe conceptuel

---

- ❑ Introduit par John-F Sowa (Sowa, 1984)
- ❑ Conçu pour représenter la sémantique du langage naturel
- ❑ Défini comme un graphe qui a deux sortes de noeuds
  - Les noeuds concepts qui représentent des entités, des attributs, des états, des événements...
  - Les noeuds relations conceptuelles qui symbolisent les liens qui existent entre deux concepts.
- ❑ Syntaxe
  - $[\text{Concept}] \rightarrow (\text{Relation}) \rightarrow [\text{Concept}]$

# Exemple

---

« Accident mortel » (accident dans lequel une personne est morte)





# Caractéristiques

---

- ❑ Orienté (arcs mono-directions)
- ❑ Fini (avoir un nombre fini de nœuds)
- ❑ Connexe (tous les nœuds sont connectés)
- ❑ Bipartie (deux sortes de nœuds - les concepts et les relations conceptuelles- chaque arc reliant une sorte de nœud à l'autre sorte de nœud)

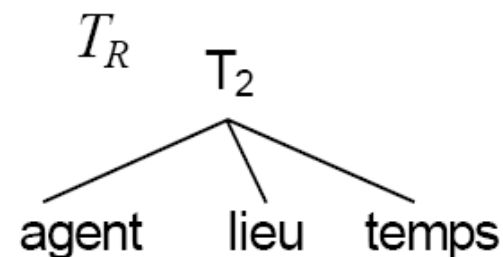
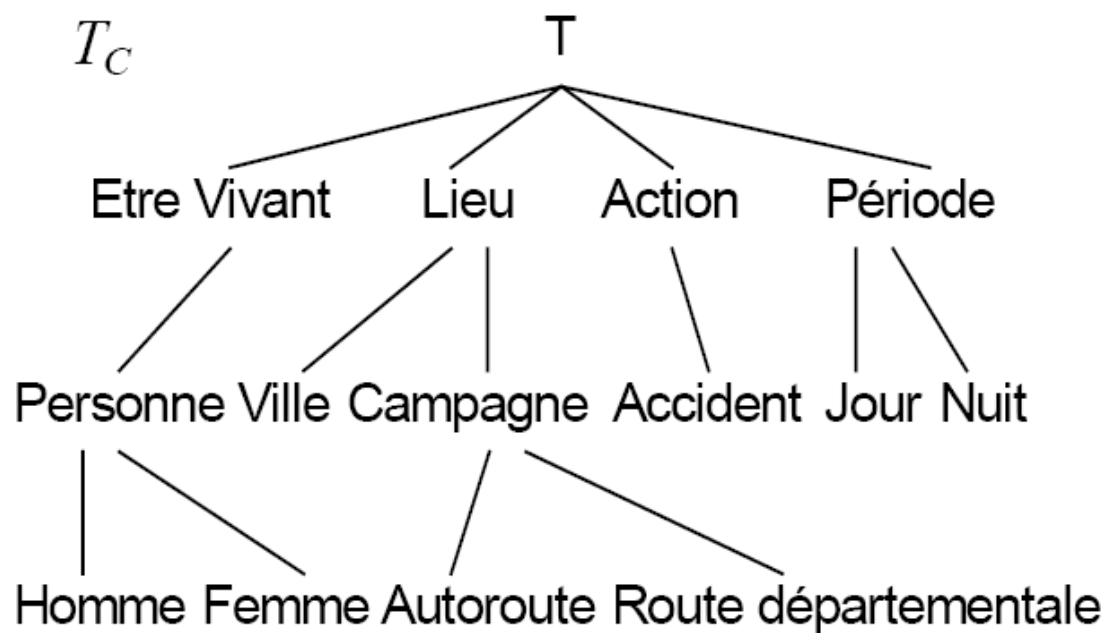
# Définition : support

---

- Un support  $S$  est un couple  $(TC, TR)$  tel que :
  - $TC$  est un ensemble des types de concepts partiellement ordonné.  $TC$  admet un plus grand élément (noté  $T$ ) appelé type universel.
  - $TR$  est un ensemble des types de relations partiellement ordonné, partitionné en sous-ensembles de types de relations de même arité.  $TR = TR_1 \cup \dots \cup TR_n$ , où  $TR_i$  est l'ensemble des types de relations d'arité  $i$ . Tout  $TR_i$  admet un plus grand élément (noté  $T_i$ ).

# Exemple : support

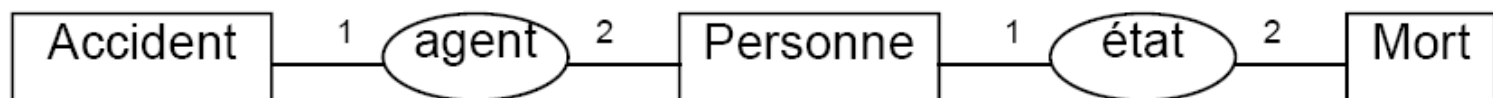
---



# Définition : graphe conceptuel

---

- Un *graphe conceptuel*  $G = (CG, RG, EG, \text{étiq}G)$  défini sur un support  $S$  est un graphe où :
  - $CG$  est l'ensemble des *sommets concepts* et  $RG$  l'ensemble des *sommets relations*
  - $EG$  est l'ensemble des *arcs*. Toutes les arcs d'un graphe conceptuel  $G$  ont une extrémité dans  $CG$  et l'autre dans  $RG$ .
  - $\text{étiq}G$  est une application qui à tout sommet de  $CG$  et à toute arête de  $EG$  associe une étiquette : si  $r \in RG$ ,  $\text{étiq}G(r) \in TR$  ; si  $c \in CG$ ,  $\text{étiq}G(c) \in TC$  ; si  $e \in EG$ ,  $\text{étiq}G(e) \in \{1, 2\}$ . L'ensemble des arcs adjacentes à tout sommet relation  $r$  est totalement ordonné, ce que l'on représente en étiquetant les arcs de 1 au degré de  $r$ .



# Marqueur du concept

---

- Un nœud de concept du graphe conceptuel peut être étiqueté par
  - [**<Type>**: **<Réfèrent>**]
- Le réfèrent peut être
  - « \* » qui indique un individu quelconque de la classe du concept, c'est-à-dire comme un concept générique
    - E.g. [HOMME: \*]
  - « # » suivi d'un numéro indique un concept individuel
    - E.g. [HOMME: #124], pour désigner l'homme dont l'identification est 124
  - Soit une instanciation du concept, un individu est donné par son nom
    - E.g. [HOMME: JEAN]



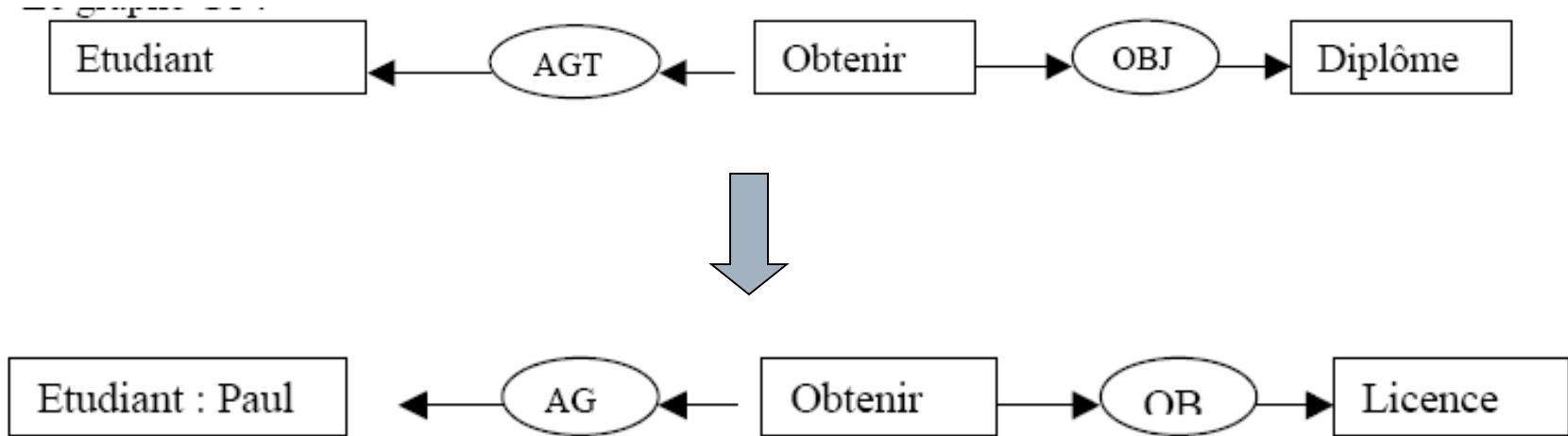
# Les opérations

---

- ❑ Copie d'un graphe consiste à construire un graphe conceptuel identique à celui de départ
- ❑ Restriction d'un graphe consiste à remplacer un type de concept du graphe conceptuel par un de ses sous-types
- ❑ Jointure des deux graphes conceptuels à regrouper des relations sur les concepts identiques
- ❑ Simplification d'un graphe consiste à enlever des relations dupliquées

# Restriction

---



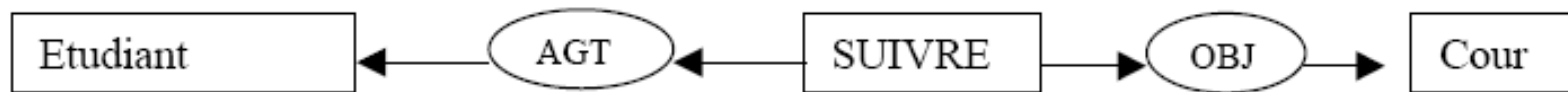
Le type de concept [DIPLOME] a été remplacé par un de ses sous-type [LICENCE]

Le concept générique [ETUDIANT] a été individualisé au concept [ETUDIANT: PAUL]

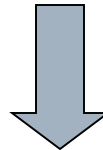
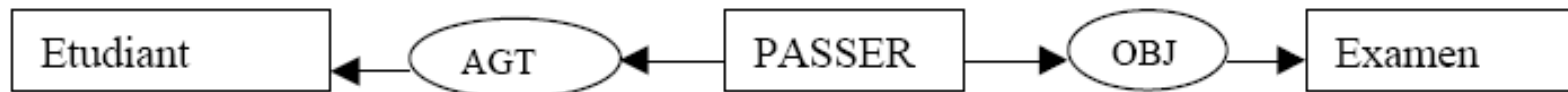
# Jointure (1)

---

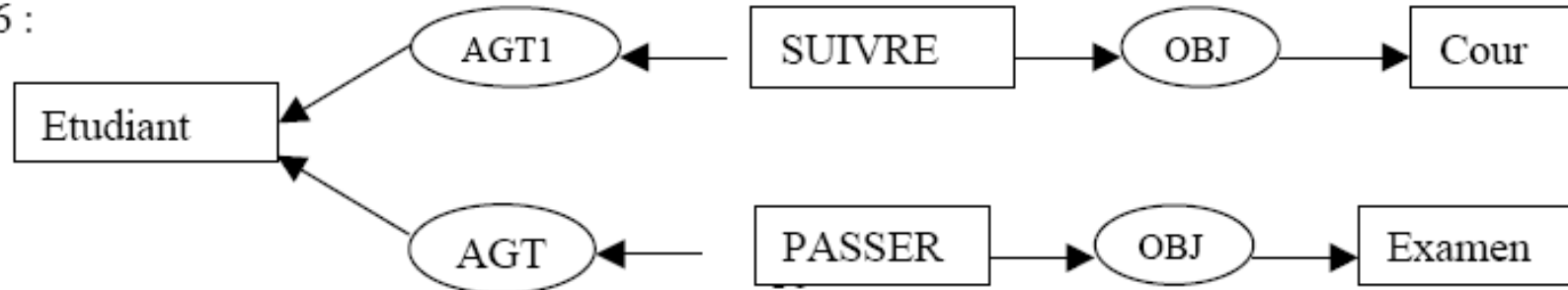
G4 :



G5 :



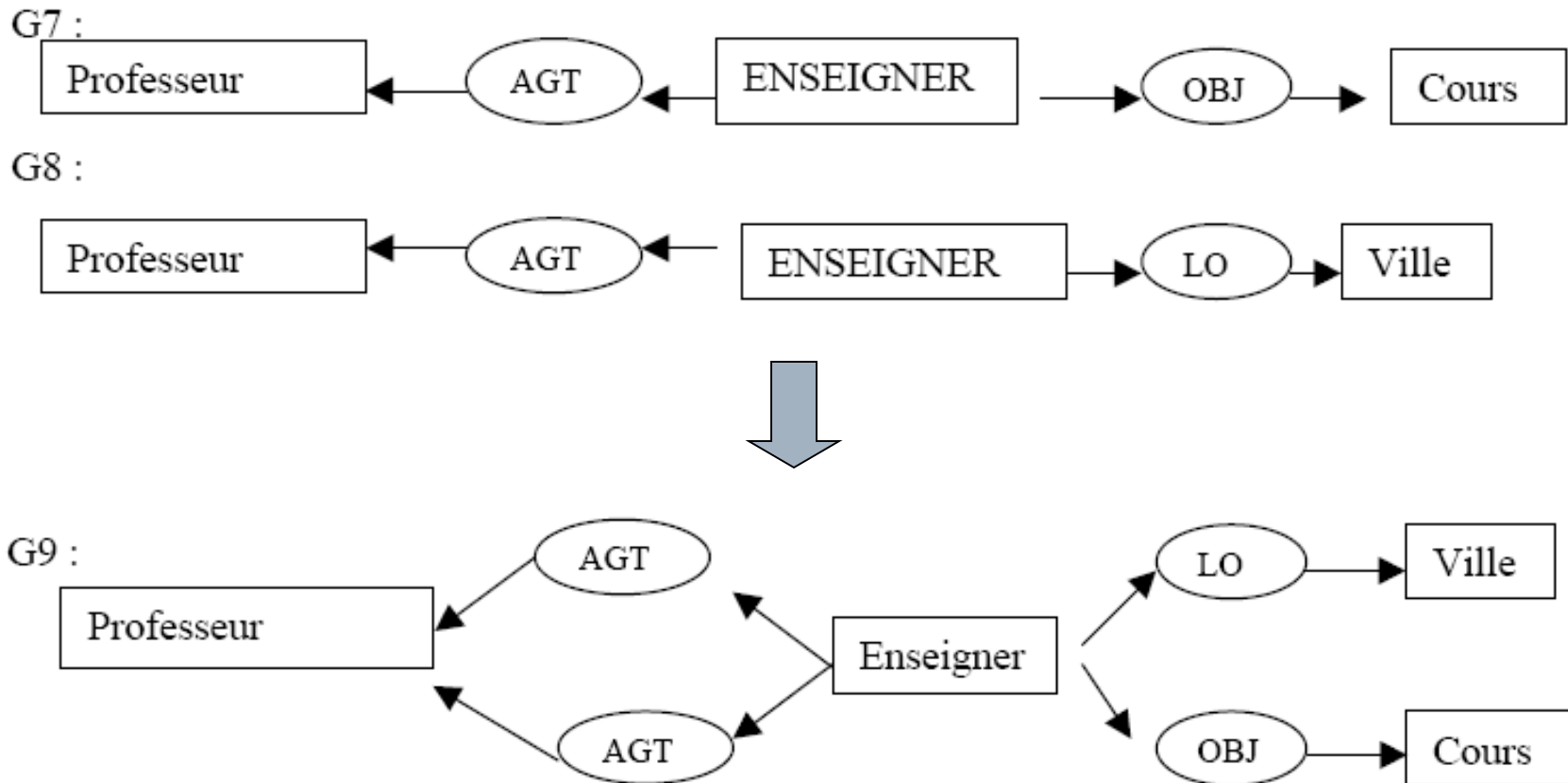
G6 :





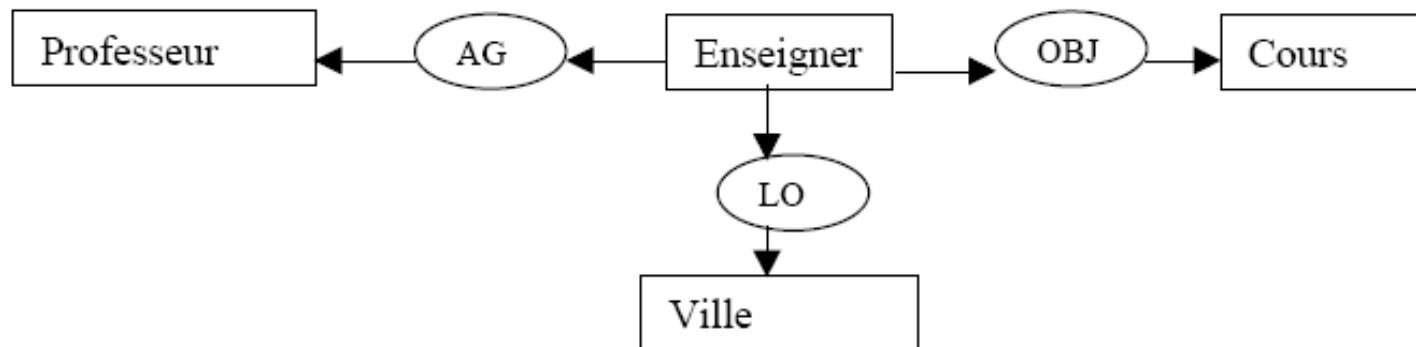
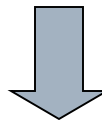
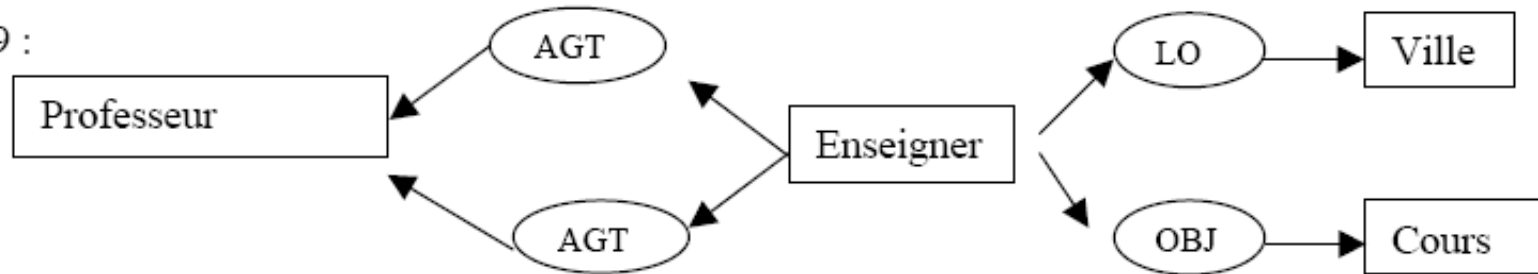
# Jointure (2)

---



# Simplification

G9 :



# Spécialisation / Généralisation

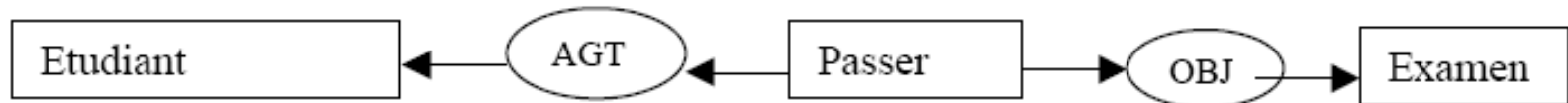
---

- U est une spécialisation de V (ou V est une généralisation de U, noté  $U \leq V$ ) si U est dérivable à partir V par une restriction ou une jointure
- Propriétés de spécialisation/généralisation
  - réflexivité :  $U \leq U$
  - transitivité : si  $U \leq V$  et  $U \leq W$  alors  $U \leq W$
  - antisymétrie : si  $U \leq V$  et  $V \leq U$  alors  $U = V$
- Le test de spécialisation entre deux graphes admet un algorithme NP-complet.

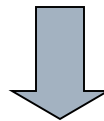
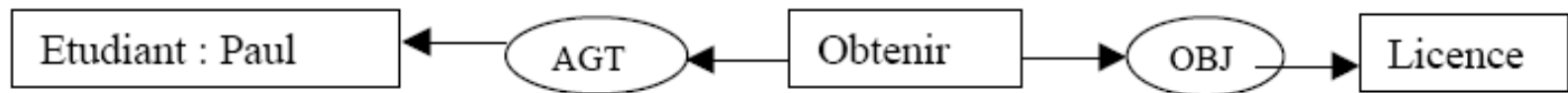
# Jointure maximale

---

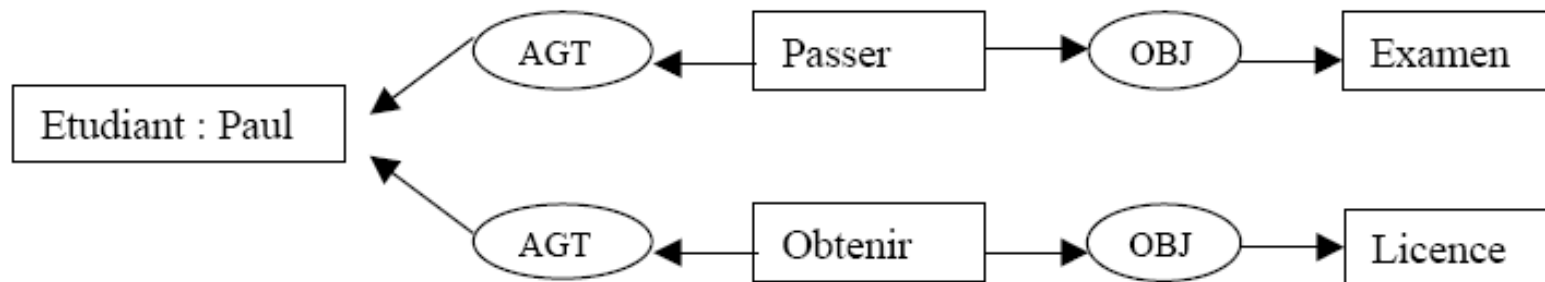
G17 :



G18 :



G19 :



# Lecture

---

- "Online Course in Knowledge Representation using Conceptual Graphs", Aalborg University's Department of Communication, <http://www.huminf.aau.dk/cg/>

# Frame

---

- ❑ Proposé par Minsky (1975)
- ❑ Plusieurs idées sous-jacentes :
  - intégrer à la fois des notions déclaratives (comme pour les réseaux sémantiques) et des notions procédurales (comment exploiter les déclarations)
  - hiérarchiser les objets à l'aide de relations is\_a

# Structure d'une frame

---

- Une structure de données pour représenter un objet, une situation caractéristique ou prototypique
  - les champs sont appelés slots ou éléments terminaux
  - chaque slot peut être lui-même une frame
  - chaque slot dispose d'une ou plusieurs facettes qui présente les propriétés du slot
  - chaque facette dispose d'une valeur

# Structure à 3 niveaux

---

(Frames

    (slot-1 (facette)

        (facette)

    ...)

    (slot-2 (facette)

        (facette)

    ...)

)

Les facettes peuvent être descriptives ou comportementales.



# Facettes

---

- ❑ Facettes de typage
  - \$un, \$sorte-de, \$domaine, \$sauf, \$card-min,...
- ❑ Facettes de valeurs
  - \$valeur, \$default,...
- ❑ Facettes procédurales
  - \$si-besoin, \$si-succès, \$si-ajout, \$avant-mod,...
- ❑ Facettes de visualisation
  - \$avant-vi, \$apres-vi, \$en-vi,...

# Exemple

---

```
(date
  (sorte-de ($valeur objet))
  (jour ($un entier)
    ($intervalle [1,31]))
  (mois ($un chaine)
    ($domaine
      "janvier","fevrier","mars","avril",...,"décembre"))
  (année ($un entier))
)
(date-23
  (est-un ($valeur date))
  (jour ($valeur 15))
  (mois ($valeur "avril"))
  (année ($valeur 1987))
)
```

# Hiérarchie des objets

---

```
(personne
  (nom ($un chaîne))
  (sexe ($un entier)
    ($domaine 0, 1))
)
(homme
  (sorte-de ($valeur personne))
  (sexe ($default 0))
)
(femme
  (sorte-de ($valeur personne))
  (sexe ($default 1))
)
(marc
  (est-un ($valeur homme))
  (nom ($valeur "marc"))
)
```

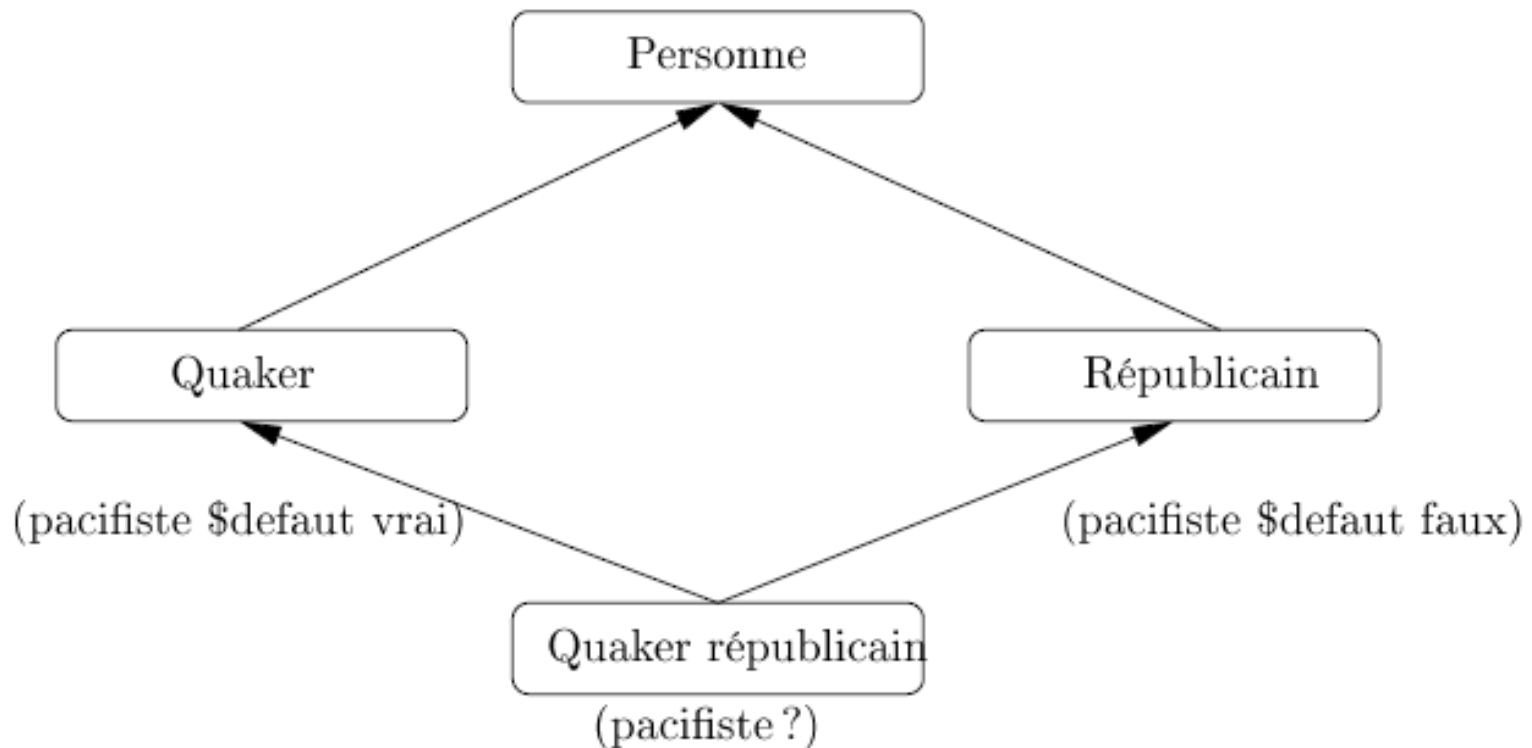
# Lecture d'une valeur

---

- ❑ La valeur d'un slot dépend des facettes \$valeur, \$défaut en prenant en compte l'héritage
  - la facette \$valeur est d'abord recherchée sur toute la hiérarchie puis \$défaut
- ❑ Quel est le sexe de Marc?
- ❑ Template slots (\$défaut) vs. Own slots (\$valeur)

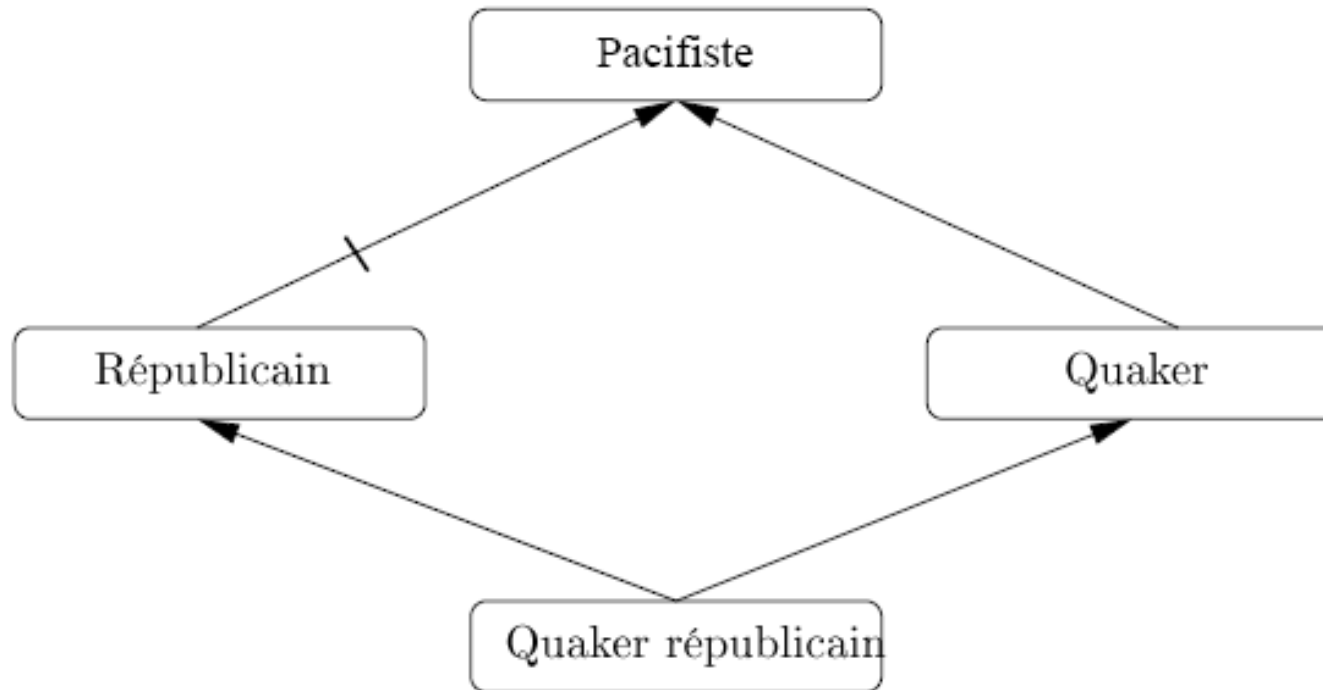
# Héritage multiple : diamant de Nixon

---



# Diamant de Nixon

---



- ❑ Ce problème peut apparaître pour tout réseaux sémantiques (avec un lien est un et un lien n'est pas un), et plus généralement dans tout modèle de raisonnement non-monotone

# Intérêt des frames

---

- Par rapport aux réseaux sémantiques, les frames permettent un travail de simplification
  - en attachant plus de signification aux noeuds (y compris des contenus procéduraux, pour mieux gérer les données)
  - en limitant les types de liens nécessaires.

# Lecture

---

- "The Generic Frame Protocol", P.D. Karp et T. Gruber,  
<http://www.ai.sri.com/~gfp/spec/paper/paper.html>