

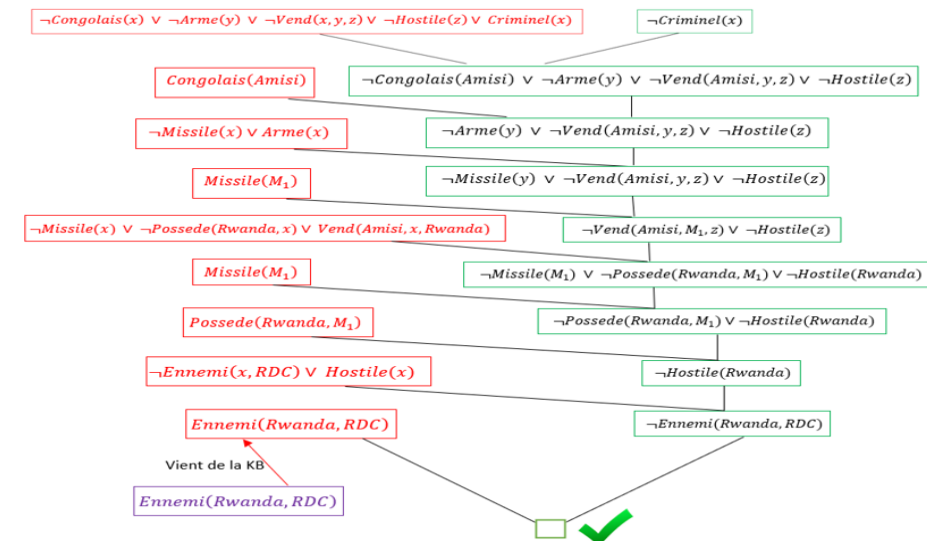
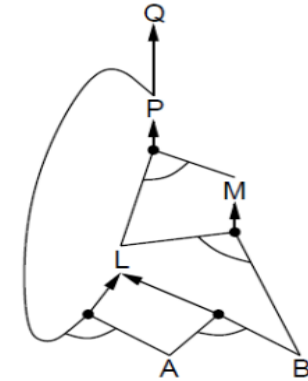


# Réseaux et cadres sémantiques

Dr. NSENGE MPIA HERITIER, Ph.D

# Précédemment : Logique

- Représentation de la logique propositionnelle
  - Traduire le langage naturel en une représentation formelle contrôlée
    - Syntaxe
    - Sémantique
- Raisonnement en logique propositionnelle
  - Principes de base pour tirer des conclusions à partir d'une base de connaissances
- Représentation de la logique du premier ordre
  - Représentation formelle plus souple
    - Prédicats
    - Quantificateurs
- Raisonnement dans la logique du premier ordre
  - Règles d'inférence
  - Inférence utilisant...
    - Preuves de théorèmes
    - Propositionnalisation
    - Chaînage avant
    - Chaînage arrière
    - Résolution



# Plan de la leçon

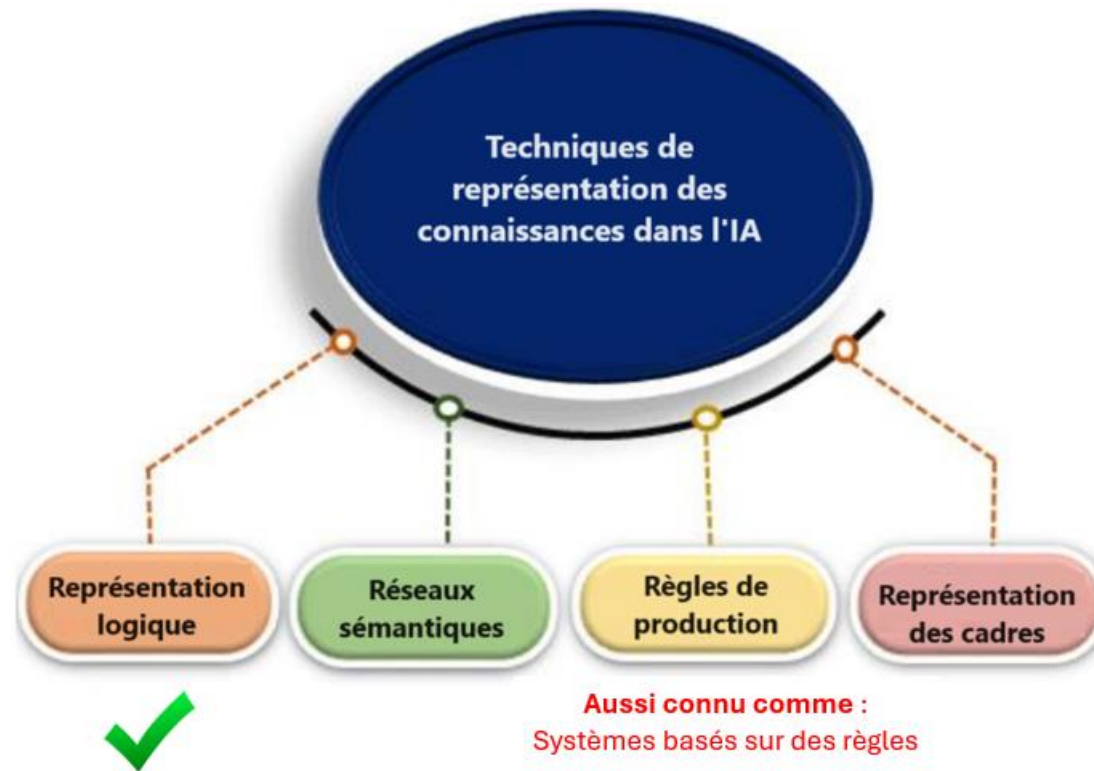


- Représentations des connaissances
- Réseaux sémantiques
  - Arbres ET/OU
  - Hiérarchie EST/PARTIE
  - Réseaux généraux
  - Interprétation
  - Héritage et inférence
  - Exemples biomédicaux
- Cadres
  - Slots
  - Défauts
  - Héritage et inférence
  - Exemples biomédicaux
- Scripts
- Autres représentations liées au réseau sémantique

# Représentation et base de connaissances

- L'expérience montre que les **systèmes intelligents** nécessitent une **connaissance large du monde réel** (probablement spécifique à un domaine) pour réaliser quoi que ce soit d'utile
  - Considérons que les humains s'attaquent souvent à des problèmes difficiles dans le monde réel en utilisant leurs ressources de connaissances (c'est-à-dire leur expérience et/ou leur formation).
- Dans l'idéal, la représentation et la base de connaissances devraient être
  - Interprétables par l'homme
  - capables d'encoder des informations dans la base de connaissances sans effort important
  - Permettre de comprendre ce que le système sait et comment il tire des conclusions.
- En fin de compte, les schémas de représentation des connaissances et les méthodes de raisonnement doivent être coordonnés et compatibles.

# Principales approches de représentation



# Représentations des connaissances dans l'IA

- **Représentation logique** : (déjà couvert)
  - Moins efficace lorsque nous voulons faire une classe restreinte de déductions
  - Difficulté de représentation :
    - Sens commun
    - Le temps, les croyances ou l'incertitude
  - Des logiques spéciales telles que les logiques temporelles et modales permettent la représentation.
    - Mais le raisonnement reste souvent inefficace
- **Réseaux et cadres sémantiques** : (c'est-à-dire des objets structurés)
  - Représentations plus simples et plus naturelles
  - Ne nécessitent pas de spécification formelle de la sémantique
  - Représentent des connaissances factuelles sur des **classes d'objets** et leurs **propriétés**
  - **Représentations déclaratives** et **statiques** des faits/rerelations
- **Règles (c'est-à-dire systèmes basés sur des règles)** :
  - Expressions SI/ALORS
  - Certaines similitudes avec les implications logiques
  - L'accent est mis davantage sur ce qui est fait avec les règles que sur ce qu'elles signifient.
    - C'est-à-dire que l'accent est mis sur les aspects **procéduraux** plutôt que sur les aspects déclaratifs.

# Objets structurés

- Contraires aux règles de production et à la logique formelle
- Formalismes de représentation des connaissances dont les composants sont essentiellement similaires aux **nœuds** et aux **arcs** que l'on trouve dans les graphes/réseaux basiques.
  - Tentative d'intégration des caractéristiques souhaitables de l'organisation de la **mémoire humaine** (association) **dans les représentations de la connaissance**
- Les représentations structurées d'objets comprennent:
  - les **réseaux sémantiques**
  - les **cadres**,
    - et la programmation orientée objet, dérivée des cadres
  - D'autres objets similaires, tels que les **scripts**



# Réseaux sémantiques



# Les bases du réseau sémantique

- **Aussi connu comme:** Réseaux d'association

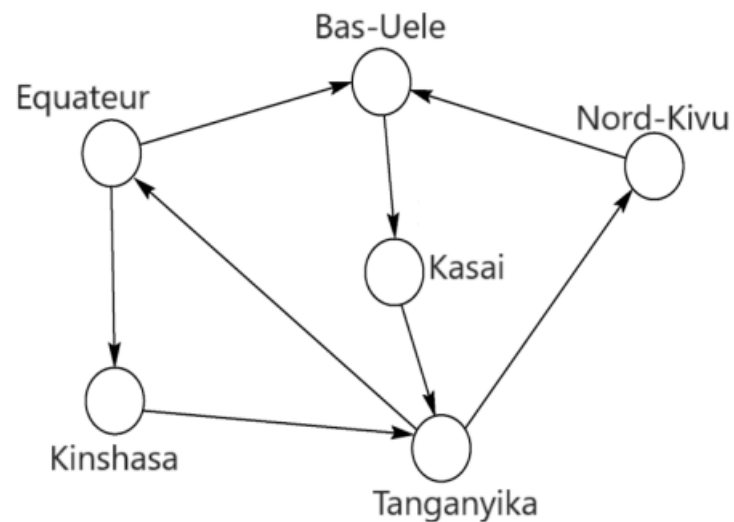
- Un **réseau sémantique** (net) est un graphe dirigé étiqueté

- Chaque **nœud** représente une proposition

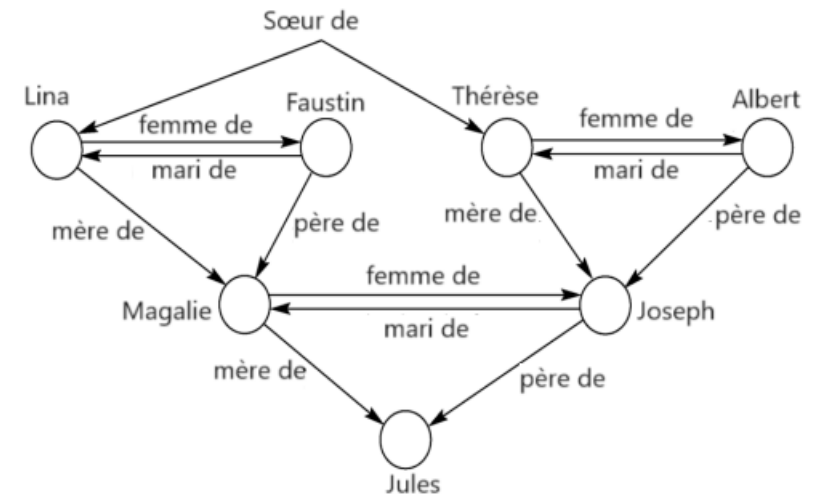
- Objet : **individu**, **concept**, **événement**

- Chaque **arc** (lien) représente une relation spécifique entre deux objets.

- Représentation graphique facile (un facteur important de leur popularité)



Réseau général



Réseau sémantique

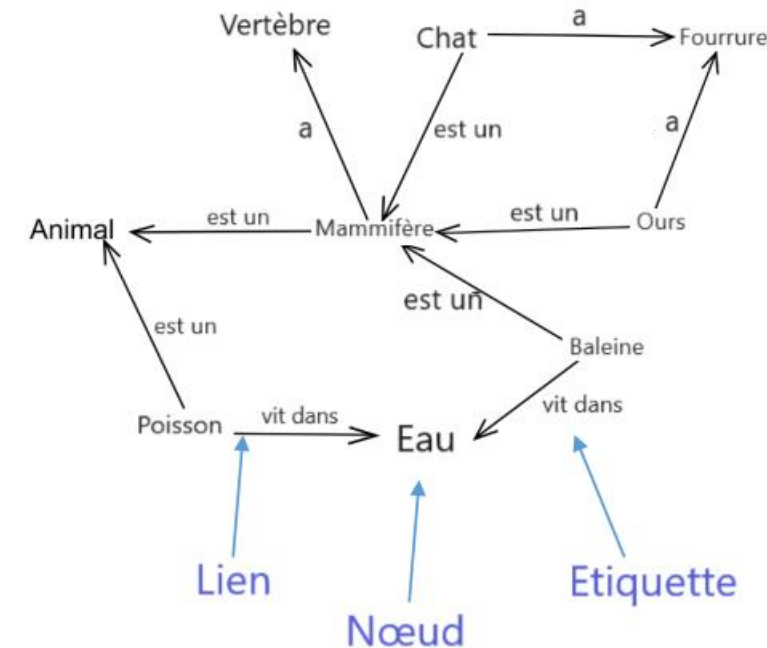
# Les réseaux sémantiques : bref historique

- Développé par Quillian en 1968
  - Un modèle pour la mémoire humaine (mémoire sémantique)
    - Quillian, M.R. (1968). "*Semantic Memories*", In M. M. Minsky, editor, *Semantic Information Processing*, pages 216-270. Cambridge, MA: MIT Press.

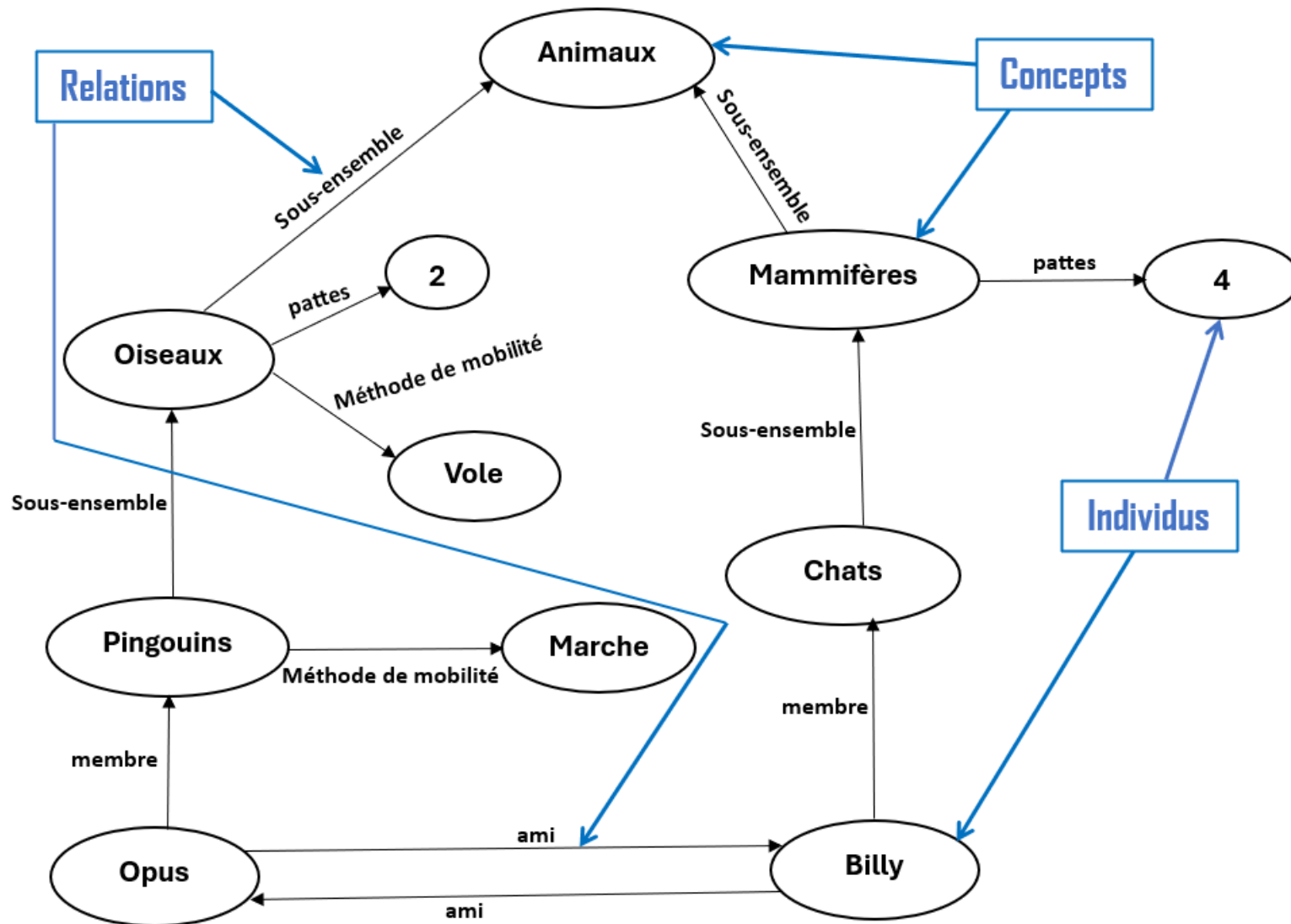
# Composantes des réseaux sémantiques



- **Partie lexicale :**
  - **Nœuds:** désignant des objets (individus et classes)
  - **Liens** (c'est-à-dire arêtes, arcs): désignant les relations entre les objets
  - **Étiquettes:** désignant des objets et des relations particuliers
- **Partie structurelle :**
  - Les liens et les nœuds forment des graphes dirigés
  - Les étiquettes sont placées sur les liens et les nœuds.
- **Partie sémantique :**
  - Les significations sont associées aux étiquettes des liens et des nœuds.
  - Les détails dépendent du domaine d'application
- **Partie procédurale :**
  - les **constructeurs** permettent de créer de nouveaux liens et nœuds
  - les **destructeurs** permettent de supprimer des liens et des nœuds
  - les **rédacteurs** (writers) permettent la création et la modification des étiquettes.-
  - les **lecteurs** (readers) peuvent extraire des réponses à des questions
- Une grande flexibilité est laissée dans la création de ces représentations



# Parties du réseau sémantique



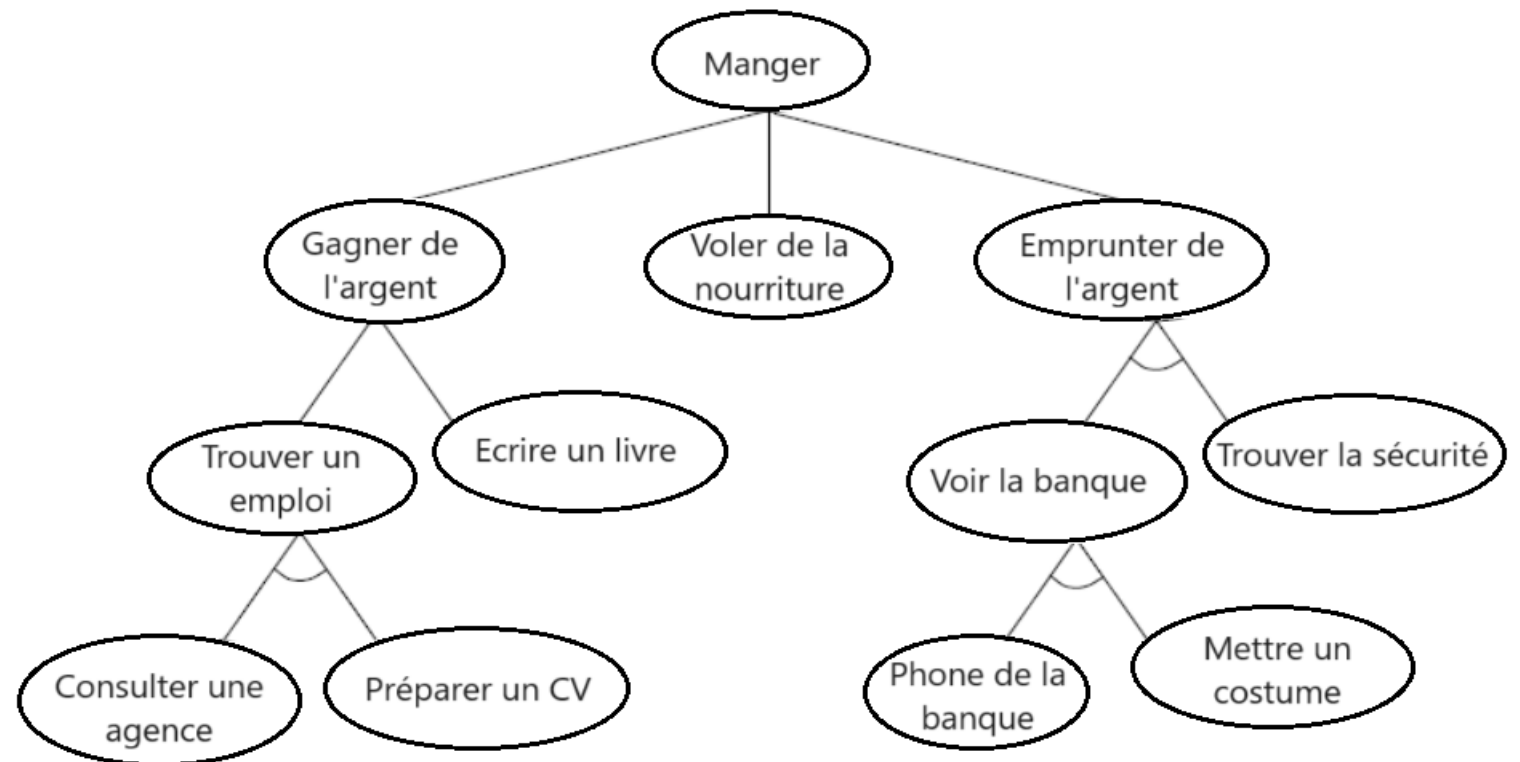
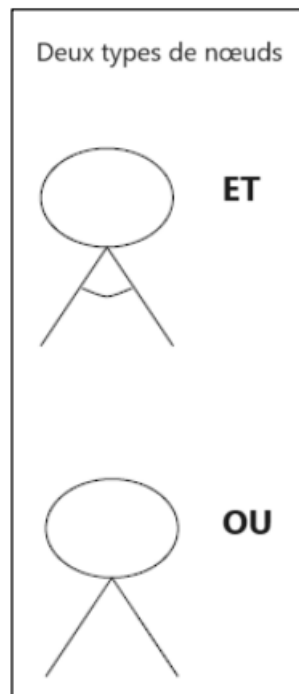
- N.B:

- Comme dit, les nœuds peuvent représenter les **concepts** ou les **individus**

- **Concepts:** sont des représentations générales comme Mammifères, Animaux
- **Individus:** sont des représentations spécifiques ou des valeurs uniques comme Billy, 4, 2,...

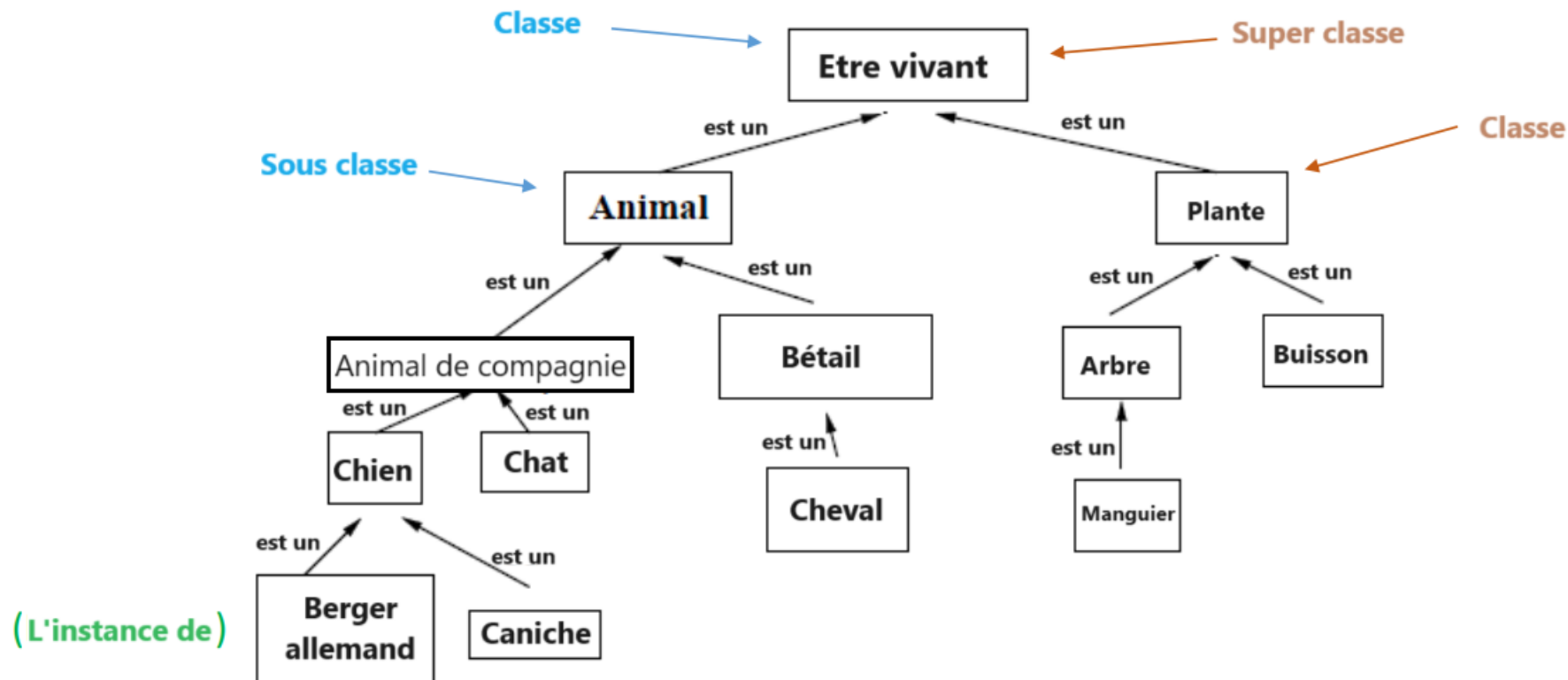
# Arbres ET/OU

- Une forme particulièrement simple de réseau sémantique
- Cette structure peut être reconnue à partir des **graphes ET/OU** utilisés pour illustrer les **chaînages avant et arrière**.



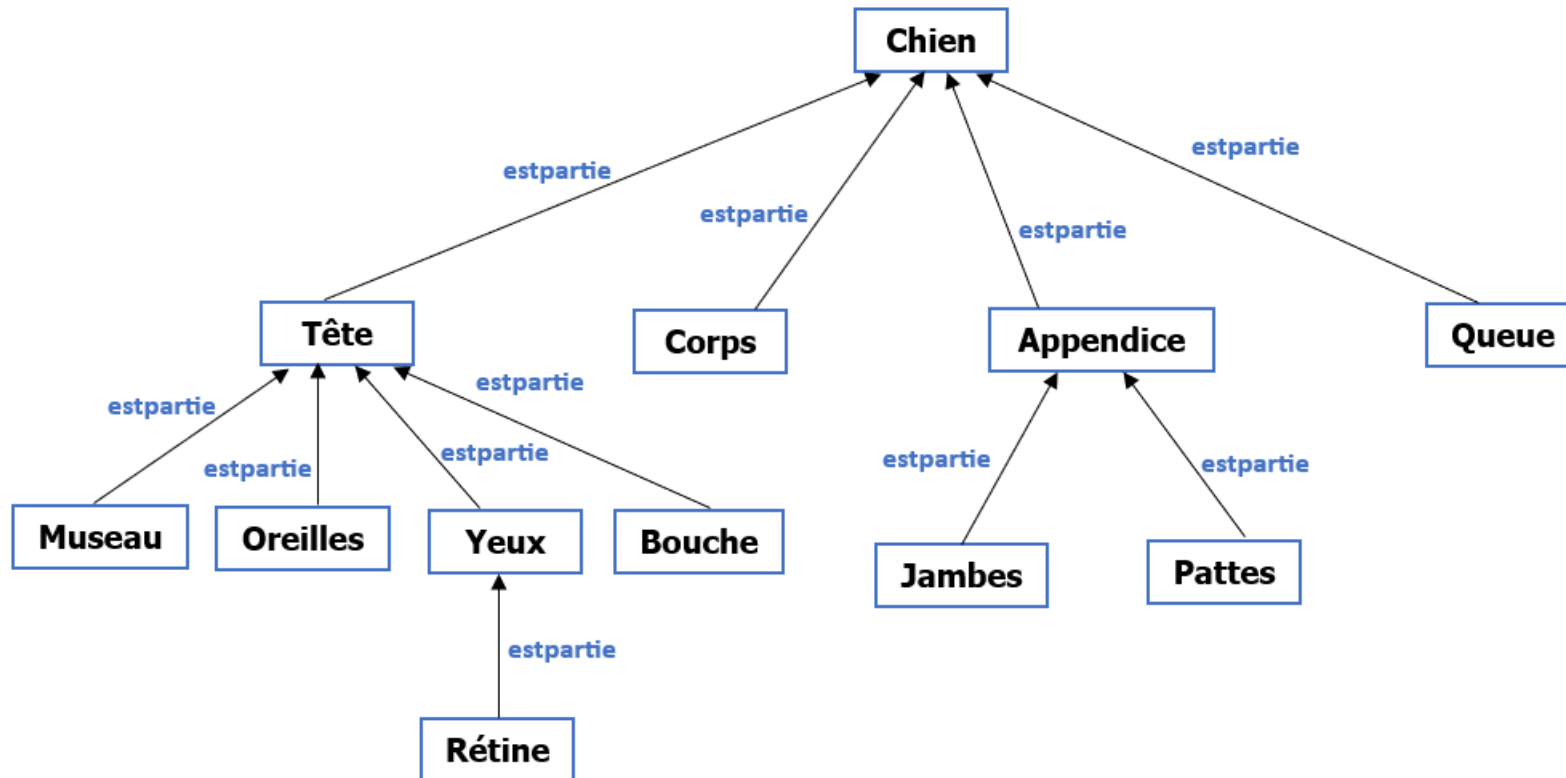
# Hiérarchie EST/UN

- Une autre forme simple de réseau sémantique (c'est-à-dire de taxonomie)
- En termes de théorie des ensembles, **est-un** correspond à la relation de sous-ensemble



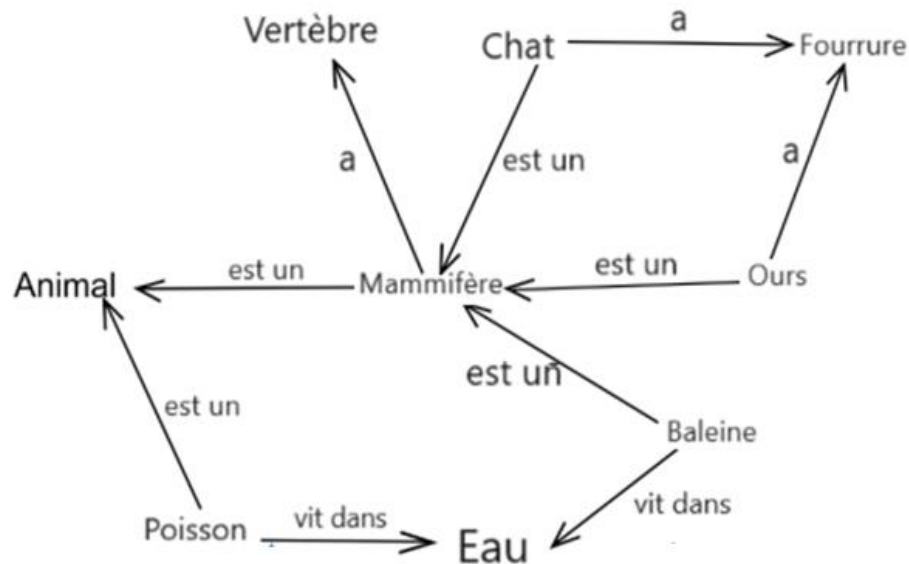
# Hiérarchie EST/PARTIE

- Nous pouvons faire descendre la hiérarchie jusqu'au niveau moléculaire ou "atomique"
- Naturellement, l'endroit où nous choisissons d'arrêter la hiérarchie dépend de ce que nous voulons représenter



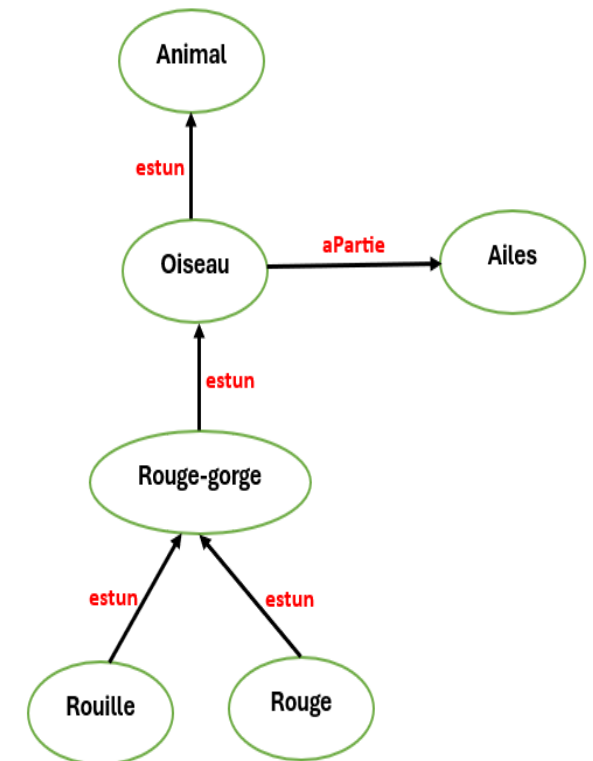
# Inférence par héritage

- L'héritage est l'un des principaux types de raisonnement effectués dans les réseaux sémantiques
- La relation **ESTUN** (EST/UN) est souvent utilisée pour relier une classe et sa superclasse.
- Certains liens (par exemple **a/partie** ou est/partie) sont hérités le long des **chemins ESTUN**.
- La sémantique d'un réseau sémantique peut être relativement informelle ou très formelle
  - Souvent définie au niveau de l'implémentation



## Exemple

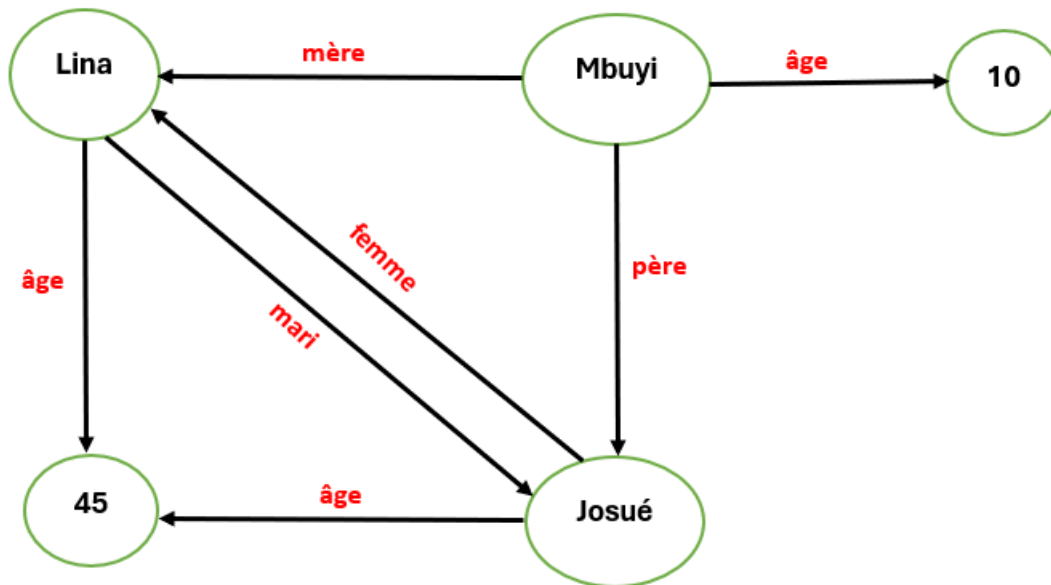
Chaque Chat **a une** colonne vertébrale, puisque chaque Chat **est un** Mammifère





# Réseaux sémantiques plus généraux

- Les **arcs** définissent des relations **binaires** entre les objets désignés par les nœuds.
  - Plus facile à représenter
- Relations au-delà de EST/UN, EST/PARTIE
  - "**CAUSE**": exprime une relation de cause à effet, par exemple "L'incendie provoque la fumée".



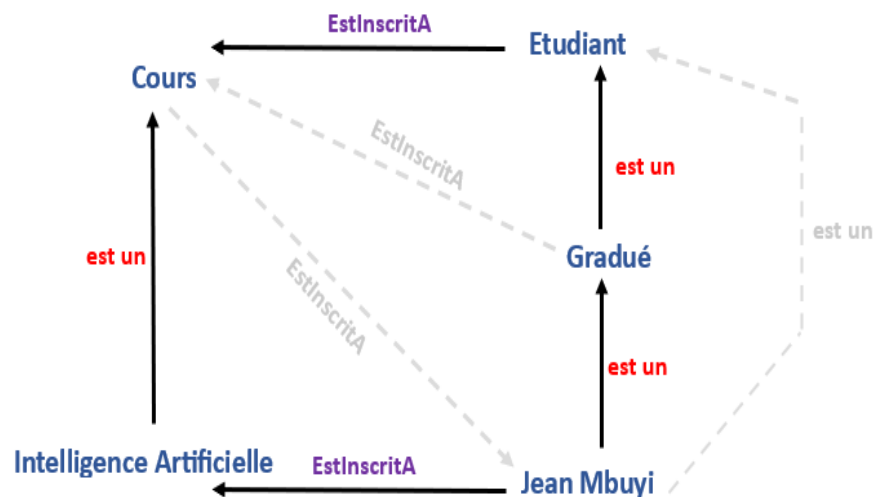
## Prédicats correspondants :

mère(Mbuyi, Lina)  
âge(Mbuyi, 10)  
femme(Lina, Josué)  
âge(Josué, 45)  
mari(Josué, Lina)  
âge(Lina, 45)  
père(Mbuyi, Josué)

# Types d'arcs importants



- **Membre** :
  - X **est un** Y : X est une *instance* de Y
  - Opus est membre de Penguin : De l'individu au Concept
- **Sous-ensemble** :
  - X **est un** *type* de Y
  - Pingouin est sous-ensemble d'Oiseau : De concept à concept
- **R-relation** :
  - X *nom-relation* Y
  - Opus est un ami de Bill ; Lina est un parent de Nsenge: Individu à Individu



- Dans l'exemple ci-contre, la relation entre l'individu **Jean Mbuyi** et la catégorie **Gradué** est un exemple **d'arc de type membre**
- La relation entre la catégorie **Gradué** et la catégorie **Etudiant** est un exemple **d'arc de type sous-ensemble**
- La relation entre l'individu **Jean Mbuyi** et la catégorie **Intelligence Artificielle** est un exemple **d'arc de type R-relation**.

# Relations non binaires



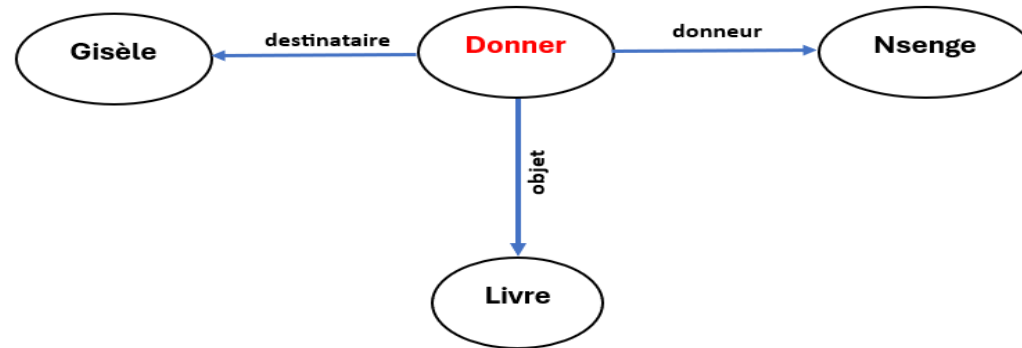
- **Réification :**

- Les relations non binaires peuvent être représentées en transformant la relation en objet.
- Réifier : *considérer un concept abstrait comme réel*

- Nous pouvons représenter un événement comme une relation impliquant trois éléments :

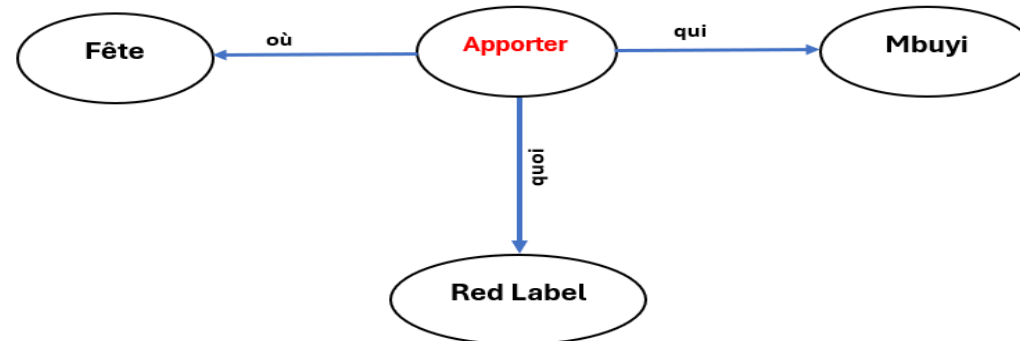
- Exemple de **Donner** :

- Un donneur
- Un destinataire
- Un objet



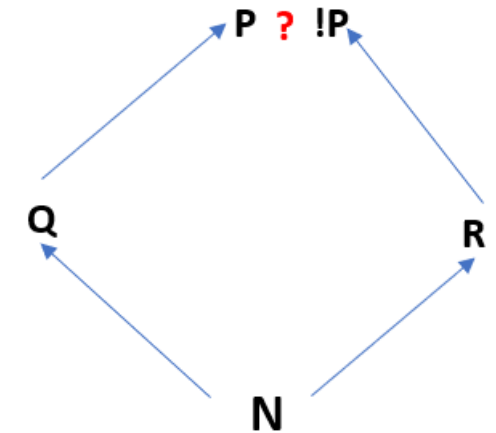
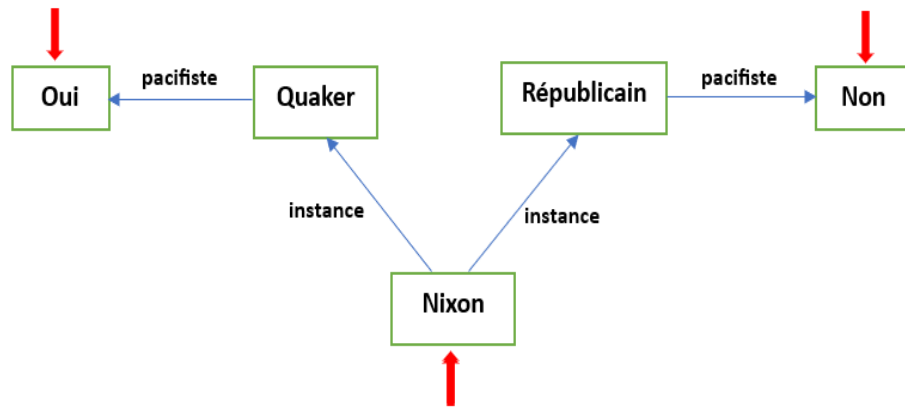
- Exemple de **Apporter** :

- Un apporteur (qui)
- Un lieu (où)
- Un objet (quoi)



# Héritage multiple

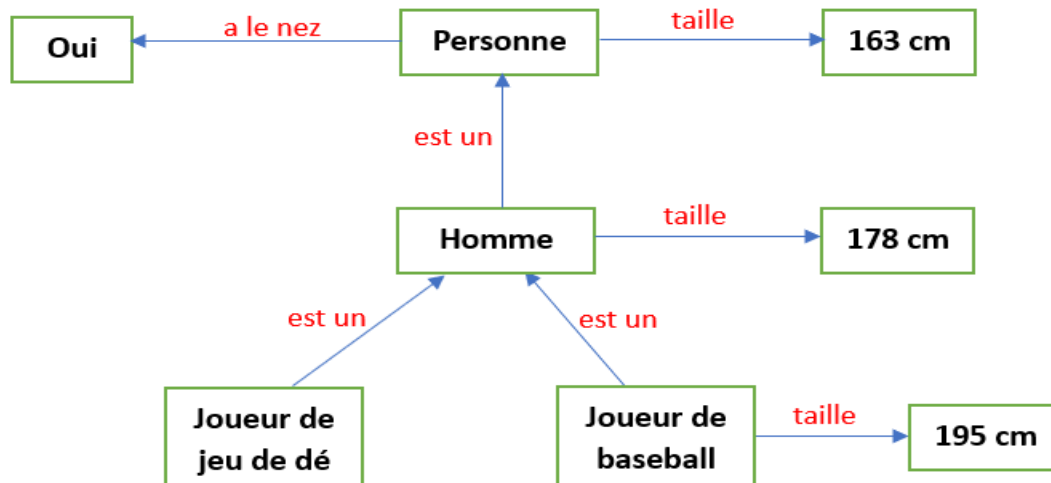
- Un nœud peut avoir un **nombre quelconque de superclasses** qui le contiennent
  - Permettre à un nœud d'hériter des propriétés de plusieurs nœuds parents et de leurs ancêtres dans le réseau
- Cela peut entraîner des conflits d'héritage



- Néanmoins, quand on définit un réseau sémantique qui peut hériter les objets de n'importe quel nœud parent ou superclasse, les situations peuvent survenir où on a le conflit d'héritage.
- Par exemple, imaginer on a la catégorie Républicain qui a la caractéristique pacifiste qui est égale à Non, alors que nous avons la catégorie Quaker qui a aussi la caractéristique pacifiste avec la valeur Oui.
- Connaissant seulement l'information dans le réseau sémantique, comment peut-on déterminer si le président Nixon était un pacifiste ou pas?
- L'image ci-haut à **droite** est une façon plus compacte d'illustrer cette situation.

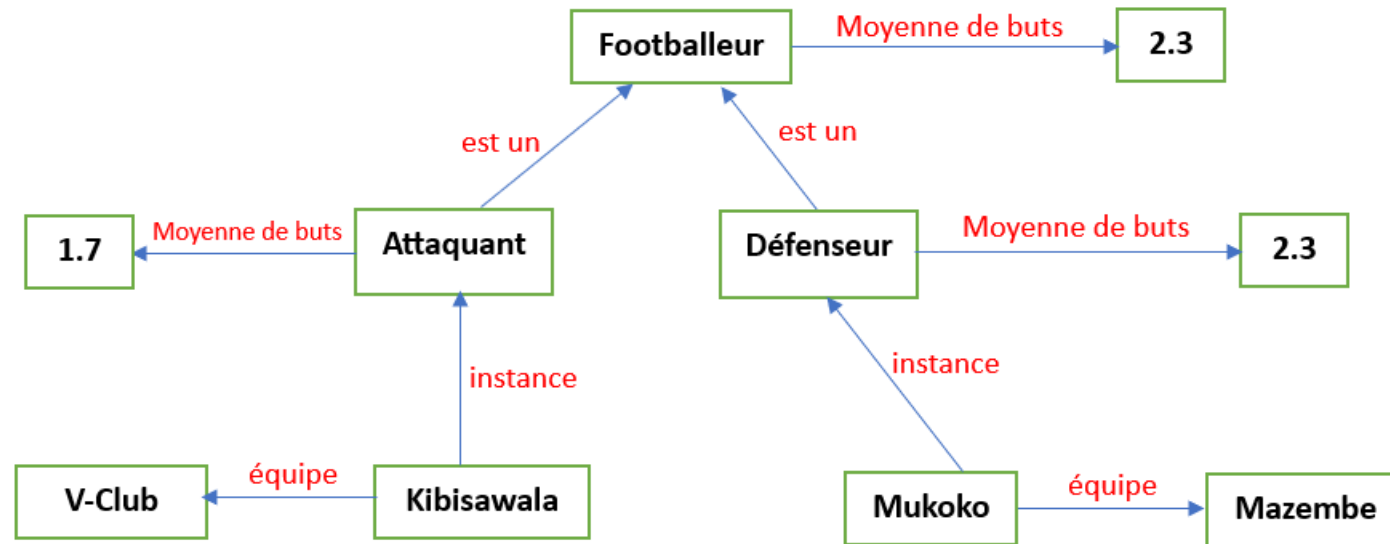
# Héritage : Valeurs par défaut et dérogations

- Les réseaux sémantiques se distinguent par **la manière dont ils traitent** le cas de **l'héritage de plusieurs valeurs différentes**
  - Toutes les valeurs possibles sont héritées (Mais ceci peut amener le problème qu'on a vu ci-haut)
  - Seule la ou les valeurs les plus spécifiques sont héritées (voir plus loin).
- Une autre stratégie est que les valeurs par défaut des relations peuvent être attribuées à chaque niveau
  - Plus efficace que l'énumération de tous les détails à chaque niveau
    - Exemple: pour la caractéristique **taille** dans l'image ci-dessous. Imaginer que la taille pour le joueur de baseball ne soit pas donnée, dans ce cas, sa taille sera la taille de la Personne qui est 163 cm.
- Des **overrides** peuvent être spécifiées
  - La catégorie la plus spécifique l'emporte sur la plus générale
    - Par exemple, joueur de base-ball vs. homme



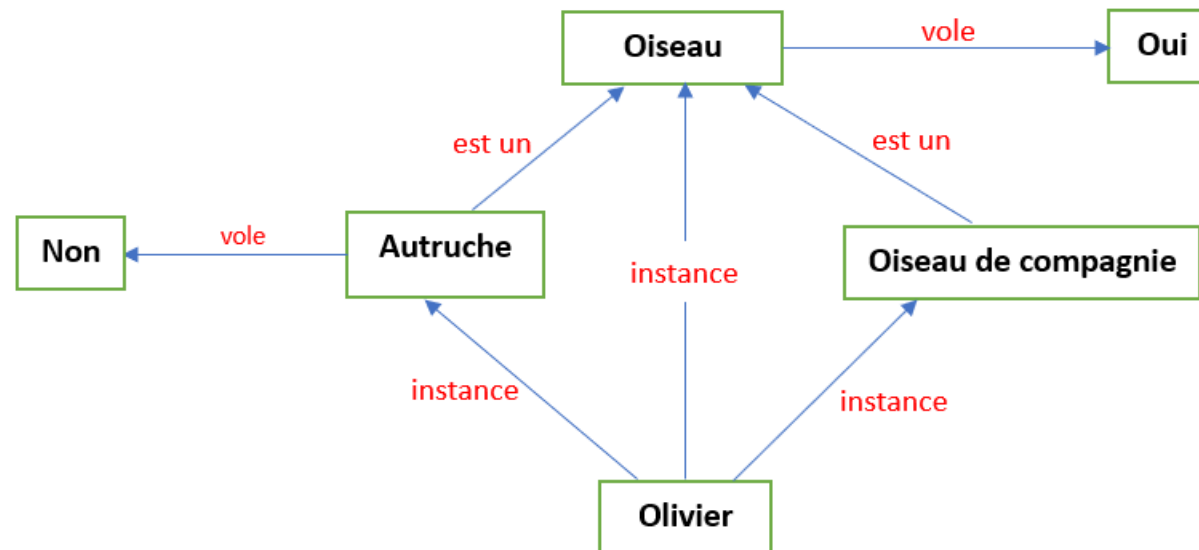
# Recherche d'interaction

- L'une des premières utilisations des réseaux sémantiques :
  - Trouver des relations entre les objets
    - Diffuser l'activation de chacun des deux nœuds et voir où les **activations** se rencontrent.
  - Ce processus est appelé **recherche d'interaction**
- Exemple :
  - Question : "Quelle est la relation entre les V-Club et Mazembe?"
  - Réponse : "Ce sont deux équipes de football congolaises".

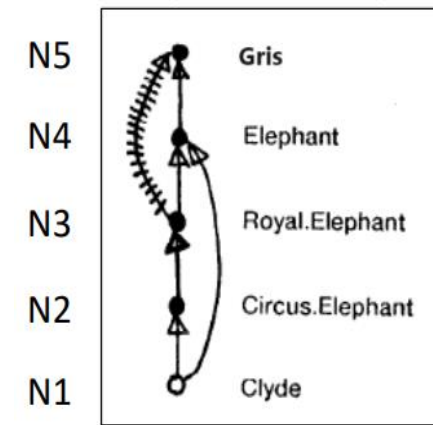


# Hiérarchies enchevêtrées

- Hiérarchies qui ne sont pas des arbres simples
  - Elles permettent un autre type de conflit d'héritage
- Question : "**Oliver peut-il voler** ?"
- Une meilleure solution que d'avoir un "vole non" spécifique pour toutes les instances individuelles d'une autruche :
  - Disposer d'un algorithme pour parcourir l'arbre qui garantit que la **connaissance spécifique dominera** toujours la connaissance générale.
  - Comment ? (Dans le slide qui suit)



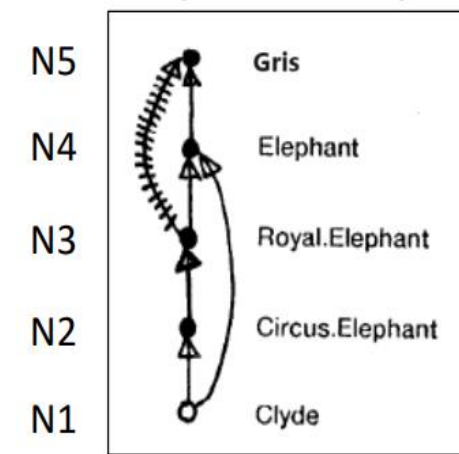
# Distance inférentielle



- Hériter de tout : "Clyde est gris et pas gris" (conclusions contradictoires)
- Arc avec lignes = "n'est pas un"
- **N.B:**
  - **Clyde** est un éléphant individuel qui est une instance de la catégorie **Circus** qui est à son tour une instance de la catégorie **Royal**, qui est l'instance de la catégorie **Eléphant**; et finalement Eléphant a la propriété couleur **grise** (Cf. image ci-haut)
  - Mais, l'Eléphant Royal n'est pas gris. Ce qui est représenté par l'arc à gauche barré. Notons que les  $N_i$  à gauche de l'image représentent le niveau de hiérarchie.
  - Dans cet exemple, nous supposons que Clyde hérite de toute chose au-dessus de lui et ainsi **Clyde** sera défini comme gris et comme non gris en même temps. Ce qui, bien sûr, est un conflit d'héritage.
  - On va essayer de résoudre ce problème avec la distance simple et après voir comment le faire avec la distance inférentielle.
- **Distance simple** : (compter les nœuds) le chemin le plus court donne "Clyde est gris". En d'autres termes, le chemin le plus court entre Clyde et la propriété **gris**. Dans ce cas, Clyde est le plus proche connecté à l'Eléphant qui serait alors hérité de la couleur grise si on se met à compter les nœuds(**ce qui est incorrect**).
- **Distance inférentielle** :
  - On a besoin ainsi de se focaliser sur le chemin le plus long car la distance simple ne tient pas.
  - Ainsi, dans ce cas, Clyde a soit les propriétés de **Royal** soit de **Eléphant**.
  - En se basant sur le chemin d'inférence le plus long, on va ignorer la connexion entre **Clyde** et **Eléphant** et on sera toujours en train d'aller sur le plus long chemin (la ligne droite de l'image) si nous allons de Clyde à Eléphant.



# Distance inférentielle (Cont.)

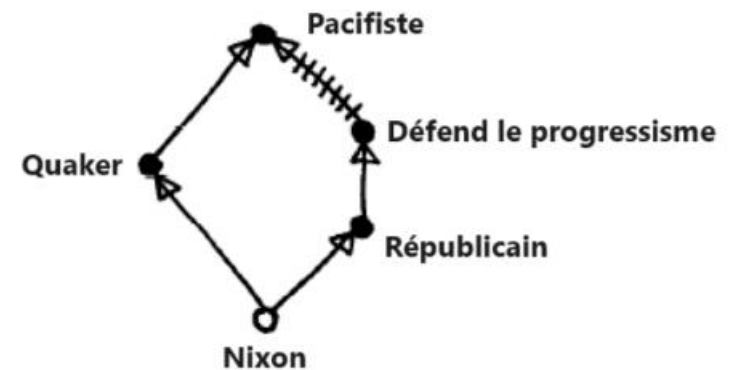
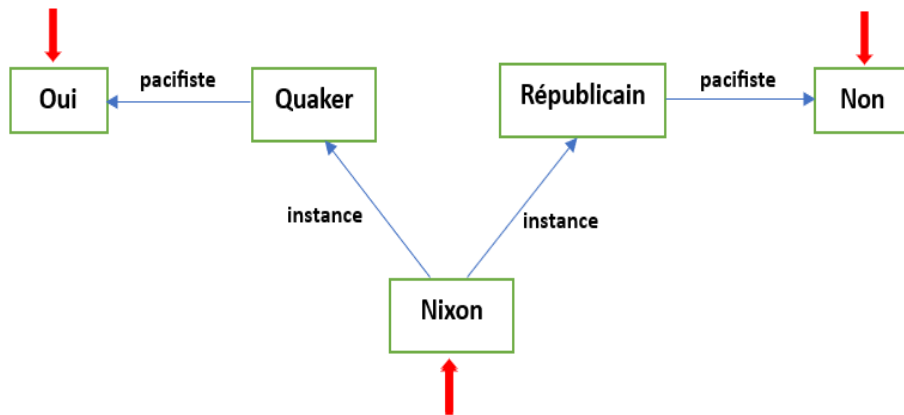


- **Distance inférentielle (Cont.):**

- Ainsi, maintenant pour déterminer la distance inférentielle, on ira sur N1, N2, et N3 (2 pas) pour obtenir **Royal** ou bien de N1 à N4 (3 pas) pour obtenir **Eléphant**.
- En utilisant la distance inférentielle, on ignore les liens redondants qui sont courts.
- Enfin, on déterminera quel nœud est plus proche en se basant sur la distance inférentielle.
- Dans ce cas, Clyde est plus proche de Royal que de Eléphant en se basant sur la distance inférentielle (2 pas).
- Pour ce faire, Clyde hérite de la propriété de **non gris** de Royal mais pas la propriété **gris** de Eléphant.
- Dans ce sens, les nœuds les plus proches seront **plus spécifiques** que les nœuds plus éloignés, et nous devrions donc **hériter des valeurs par défaut de ces nœuds**.

# Problème de distance inférentielle

- La distance inférentielle **définit seulement un ordre partiel concernant un ensemble de nœuds communs**
- Ainsi, elle ne sera pas d'aide avec notre exemple précédent du président Nixon
  - La raison est que parce que de Nixon, on a deux nœuds séparés de chemins d'héritage plutôt qu'un unique avec différentes distances à considérer comme dans l'exemple de l'Eléphant
  - Prenons l'exemple de l'image à droite similaire à celle de l'exemple de l'Eléphant:
    - Ici, il n'y a pas de voie d'héritage du Quaker au Républicain



- En général, un **moteur inférentiel** sera composé de nombreuses règles procédurales, comme l'utilisation de la distance inférentielle pour définir la manière dont le réseau sémantique doit être traité.
  - Doit être clairement spécifié afin de préserver le sens souhaité ou d'effectuer des inférences correctes.

# Réseaux sémantiques : Avantages

- Interprétation :
  - Facile à visualiser
  - Les connaissances connexes sont facilement regroupées
  - La sémantique, c'est-à-dire les significations du monde réel, est clairement identifiable.
- Représentation :
  - Souple et générale : les relations peuvent être définies arbitrairement par l'ingénieur des connaissances
  - Des définitions formelles des réseaux sémantiques ont été élaborées.
  - Efficace en termes d'exigences d'espace
  - Objets représentés une seule fois
  - Ils peuvent être utilisés pour représenter des événements et des phrases en langage naturel phrases en langage naturel
- Inférence :
  - Inférence réduite à la recherche (pas besoin de preuves comme en logique)

# Réseaux sémantiques : Désavantages

- Représentation :
  - La représentation générale peut être un problème à moins que nous soyons clairs sur la syntaxe et la sémantique dans chaque cas
  - Pas de normes sur les valeurs des nœuds et des arcs (pas de sémantique dans les réseaux sémantiques)
  - Expression limitée : peut nécessiter un certain nombre de procédures spécialement codées
    - Pas de moyen d'exprimer des connaissances non monotones (comme logique de premier ordre)
    - Pas de moyen facile d'exprimer des relations n-aires (réification nécessaire)
- Inférence :
  - L'héritage peut poser des problèmes
    - en particulier à partir de sources multiples et lorsque des exceptions dans l'héritage sont nécessaires
  - Les faits placés de manière inappropriée posent des problèmes
- Les problèmes susmentionnés rendent difficile:
  - la vérification et la validation des systèmes
  - Le partage des connaissances
  - La réutilisation des connaissances
  - L'acquisition des connaissances de manière méthodique

# Exemple : Construction d'un réseau sémantique

- **L'histoire d'Othello :**

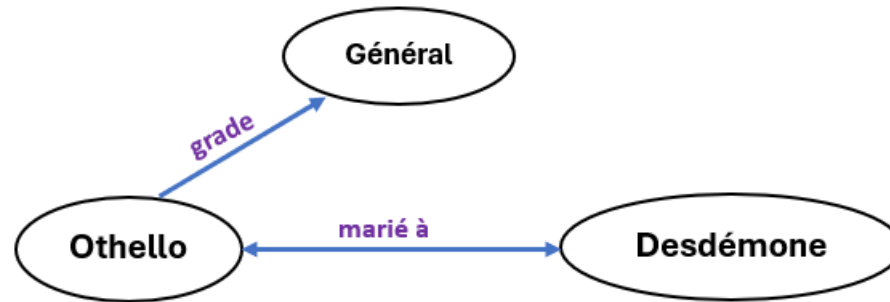
- Othello était un général marié à Desdémone
- Iago était un capitaine marié à Emilia ; il détestait Othello
- Iago a raconté à Othello des mensonges sur Desdémone
- Othello a tué Desdémone avec un oreiller. Il a eu des remords et s'est tué avec un poignard.



# Exemple : Construction d'un réseau sémantique (Cont.)



- Dans la première étape, on prend la première phrase. Et, ici nous avons les objets:
  - Othello
  - Général
  - Desdémone

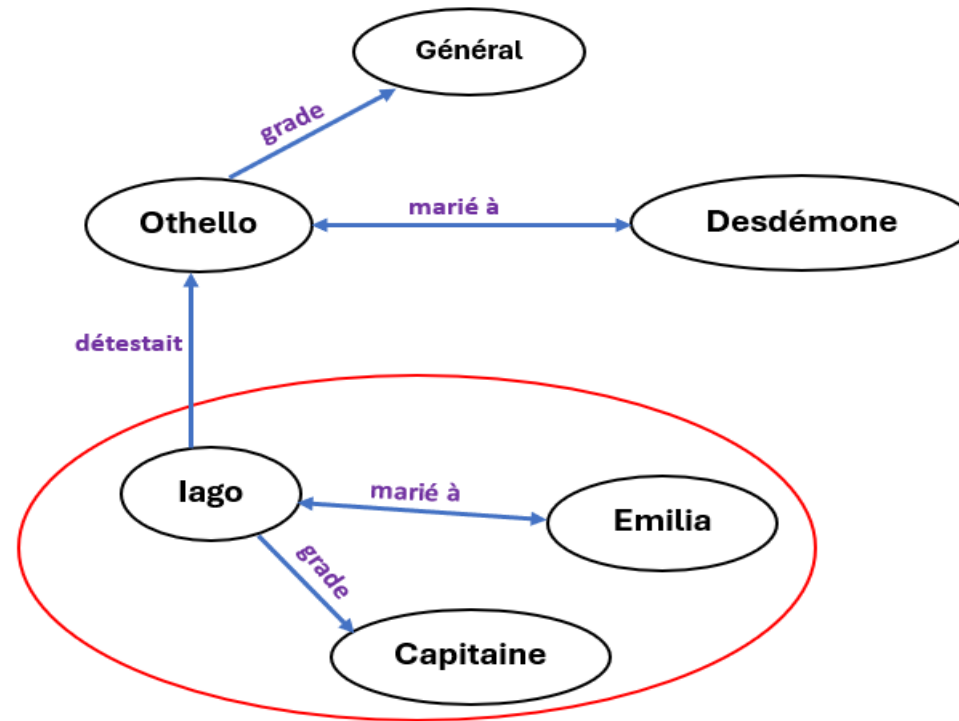


- Othello était un général marié à Desdémone

# Exemple : Construction d'un réseau sémantique (Cont.)



- Dans la deuxième étape, il y a le Capitaine Iago qui était marié à Emilia et qui détestait Othello.

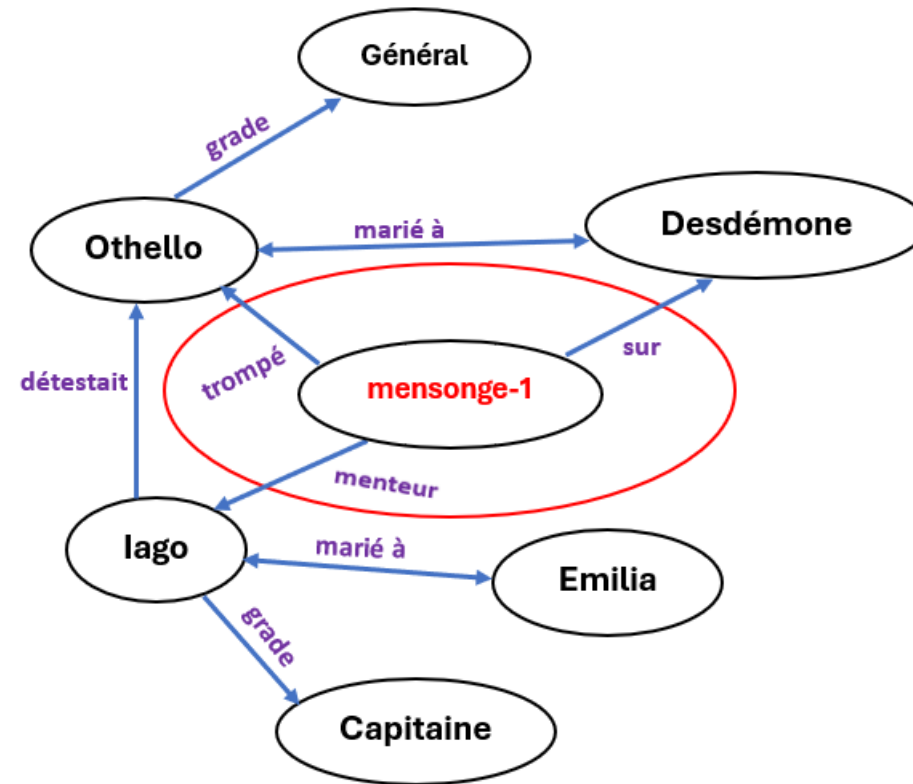


- Iago était un capitaine marié à Emilia ; il détestait Othello

# Exemple : Construction d'un réseau sémantique (Cont.)



- Dans la troisième étape, il y a l'élément **mensonge** qui est connecté à trois objets. En fait, **mensonge** n'est pas un objet mais c'est la réification (le menteur c'est Iago qui a menti Othello et ce mensonge a causé la mort de Desdémone)



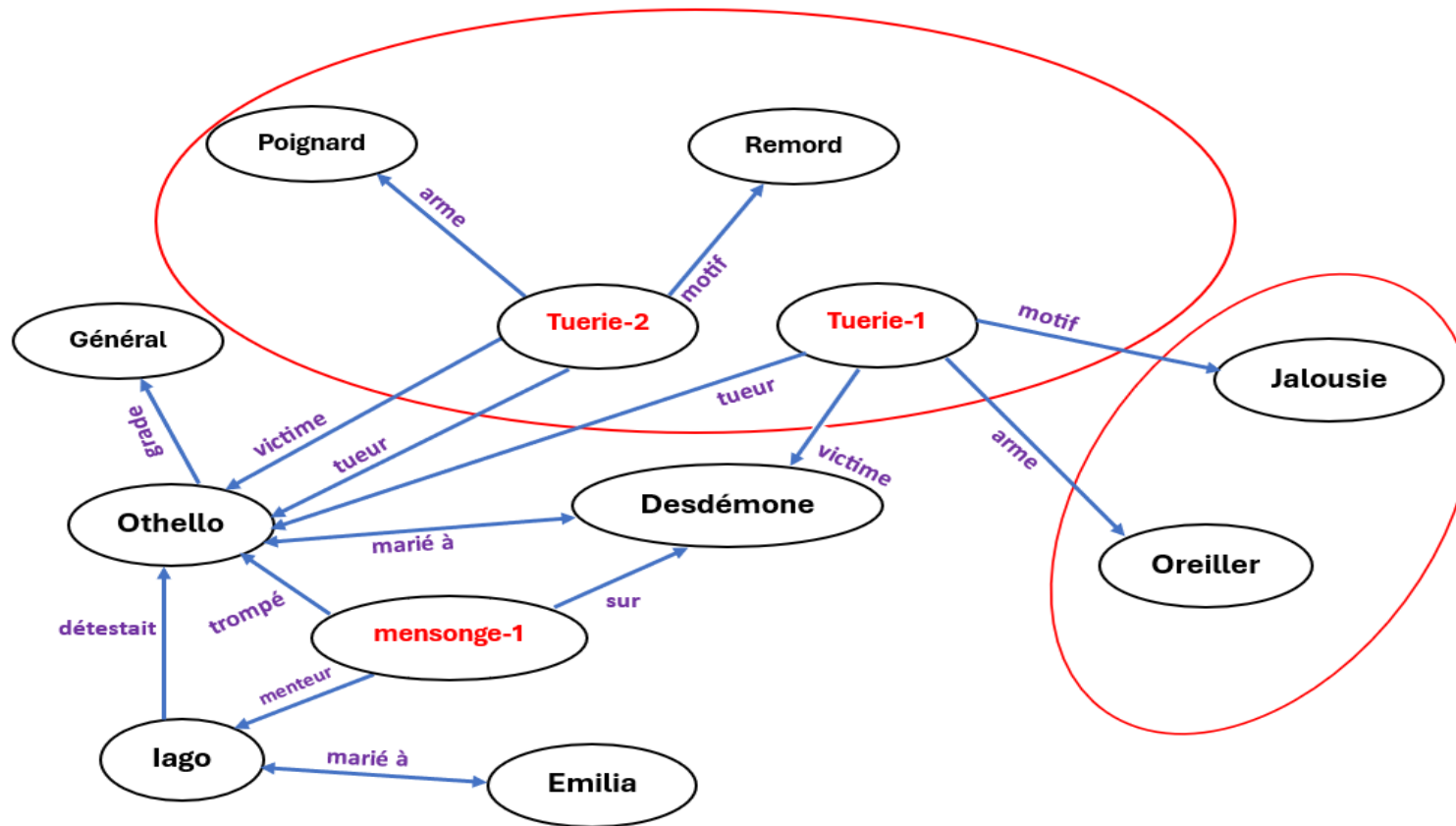
- Iago a raconté à Othello des mensonges sur Desdémone



# Exemple : Construction d'un réseau sémantique (Cont.)



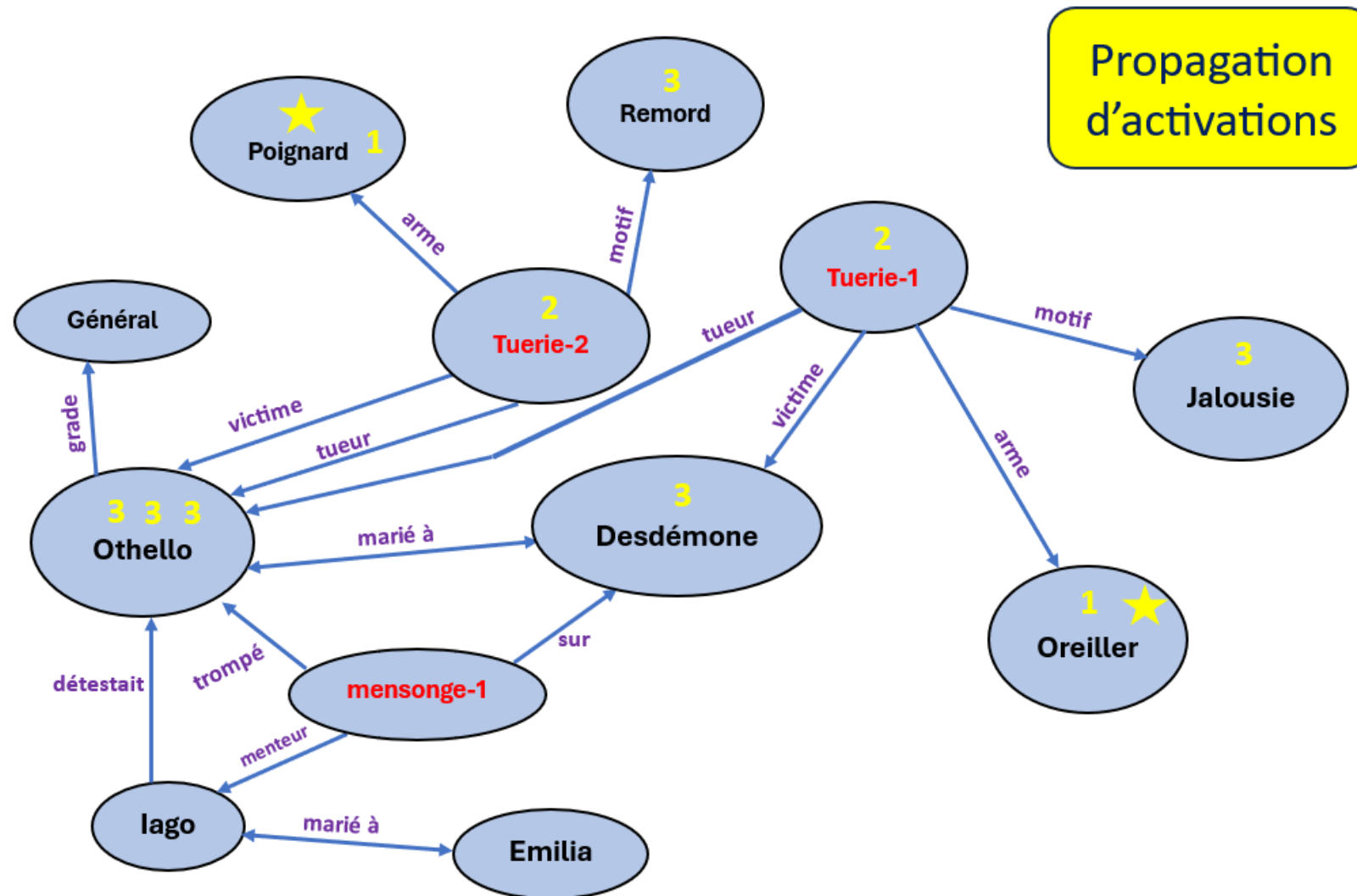
- Enfin nous avons la phase où Othello tue sa femme et se tue lui-même après. Ici, il y a deux réifications afin d'utiliser la relation tuer. La première c'est Othello qui tue sa femme par jalousie et la seconde c'est Othello qui se tue lui-même par le remord.



- Othello a tué Desdémone avec un oreiller. Il a eu des remords et s'est tué avec un poignard.

# Exemple d'inférence : Recherche d'interactions

- **Question:** Quel est le point commun entre l'oreiller et le poignard ?



# Exemple d'inférence : Recherche d'interactions (Cont.)

- Maintenant, nous avons construit le réseau en voyant comment on peut conduire l'inférence
- On peut utiliser l'idée de la question ci-haut pour propager l'activation
- On commence avec les deux objets qui constituent la question: **poignard** et **oreiller**.
  - 1 dans l'image du slide qui suit avec étoiles en jaune à côté sont des valeurs qui montrent les activations
  - On voit que la réification tuerie-1 se connecte avec l'oreiller, Desdémone et Othello à l'étape 2. En même temps, à l'étape 2, tuerie-2 est connectée avec poignard et Othello.
  - Enfin, à l'étape 3, toutes les flèches tombent sur les victimes mais il y a trois flèches qui sont connectées à Othello à cette étape
- **Réponse:** Armes utilisées par Othello dans les meurtres

# Des réseaux sémantiques aux cadres

- Les représentations des **réseaux sémantiques** sont devenues de plus en plus complexes afin de prendre en charge un plus grand nombre de tâches de résolution de problèmes.
  - L'information devient distribuée
- Il devient nécessaire d'attribuer plus de **structure aux nœuds** ainsi qu'aux liens
- Dans de nombreux cas, nous avons besoin **d'étiquettes de nœuds qui peuvent être calculées** (informatisées) plutôt que fixées à l'avance
- Les cadres adoptent des idées de base de données pour aider à garder une trace de tout
- La distinction entre les réseaux sémantiques et les cadres est floue dans les applications modernes car ils sont étroitement liés
  - Le cadre aura généralement plus de structure qu'un réseau sémantique.



Cadres

# Cadres : Origines



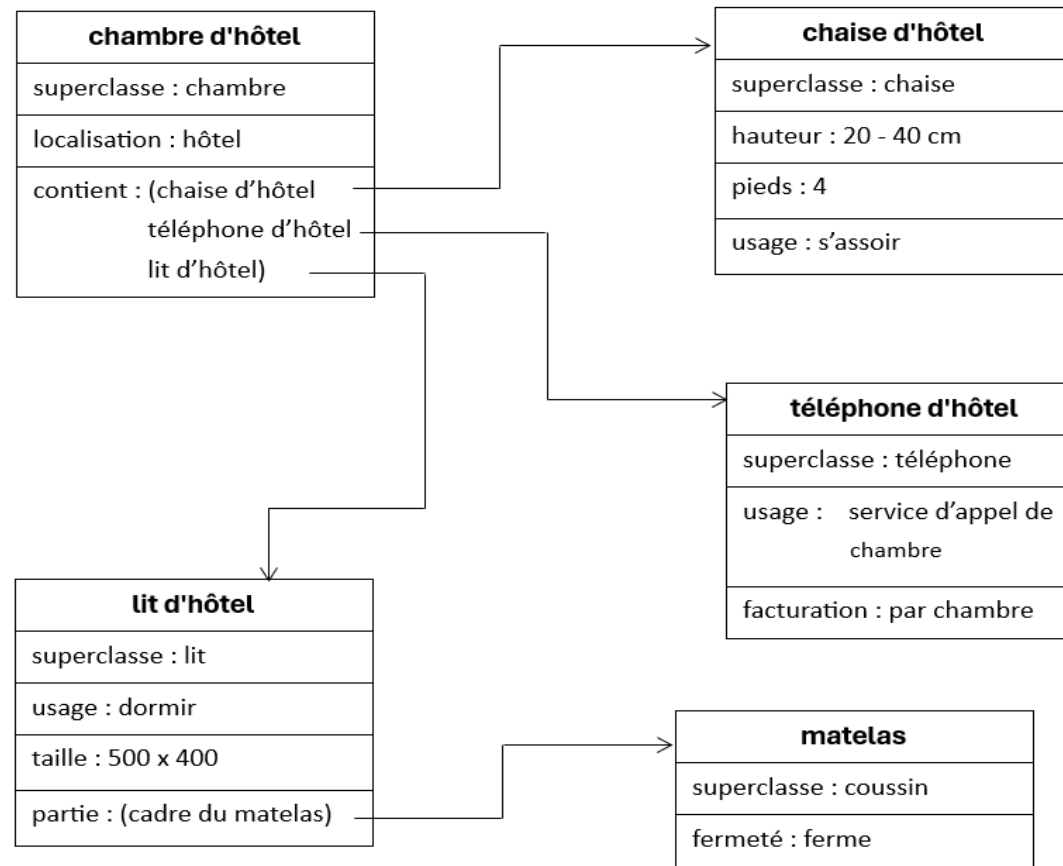
- Cadres : Conçu par Marvin Minsky, 1974
  - Minsky, M. (1974). A Framework for Representing Knowledge, MIT-AI Laboratory Memo 306
  - Une **variante des réseaux sémantiques**
  - Moyen populaire de représenter des faits dans un système expert
- Observation:
  - Les choses d'un type donné participent au même ensemble de relations
- Objectif : ***représentation de situations stéréotypées***
  - L'essence de cette forme de représentation des connaissances est la **typicité**, avec des **exceptions**, plutôt que de donner des définitions
  - Un **cadre** est un formalisme de représentation des connaissances basé sur l'idée d'un cadre de référence.

# Cadres

- Toutes les informations relatives à un concept particulier sont stockées dans une entité complexe unique (appelée **cadre**)
  - Les cadres ressemblent à des structures de données de type record
  - Toutefois, les cadres prennent en charge l'héritage comme les réseaux sémantiques
- Une **représentation complète basée sur un cadre** consistera en une **hiérarchie ou un réseau de cadres reliés** entre eux par des liens/pointeurs appropriés
- Cadres organisés dans une hiérarchie :
  - Forme de **programmation orientée objet** pour l'IA et les systèmes experts

# Cadre : Exemple simple

- Chaque boîte constitue un cadre et chaque cadre possède ses informations
- En plus, les parties d'un cadre peuvent être connectées à d'autres cadres.
- Exemple: le cadre chaise d'hôtel. Sa superclasse c'est chaise. La chaise d'hôtel possède par défaut une plage de valeur d'hauteur entre 20 et 40 cm, elles ont aussi les propriétés pieds et usage.





# Types de cadres



- Cadre de classe: il existe des cadres de type classe.

- Exemple:

1. Classe Mammifère qui est une sous classe d'animal
2. Eléphant qui est une sous classe de mammifère

## MAMMIFERE :

sous-classe : ANIMAL  
a\_partie : tête  
taille : grande

## ELEPHANT :

sous-classe : MAMMIFERE  
couleur : gris  
taille : grande

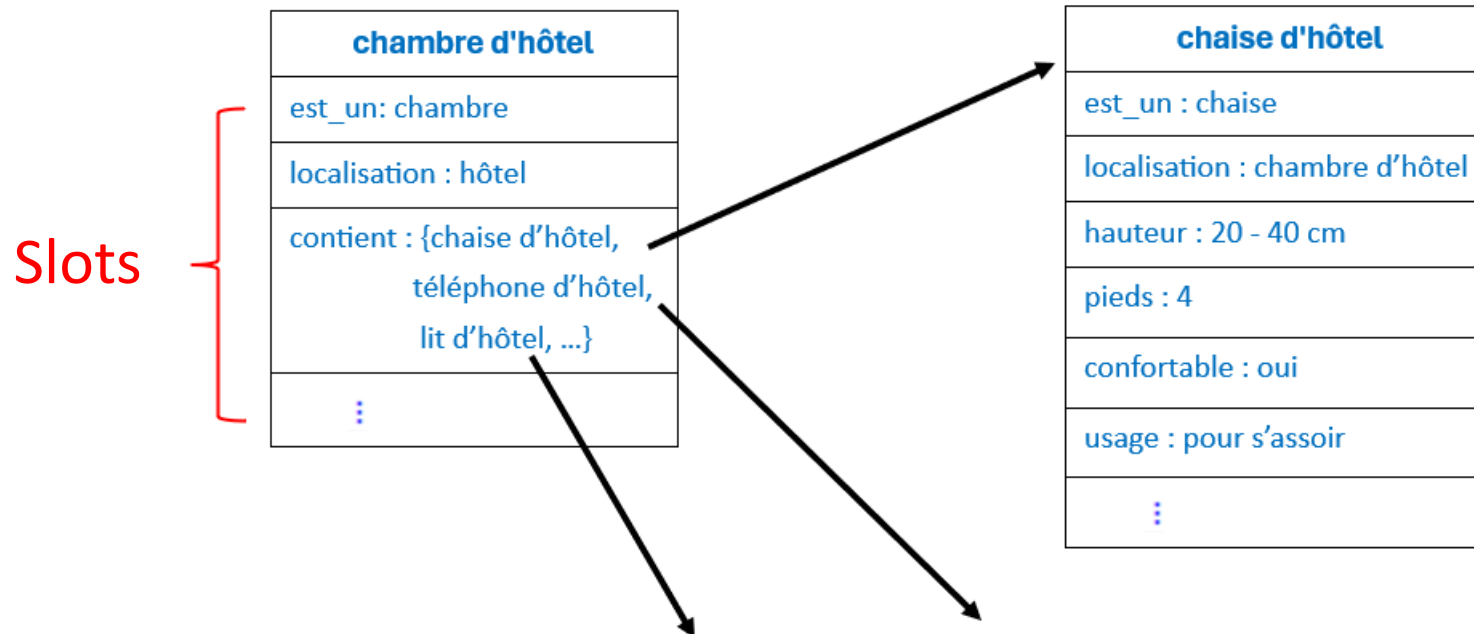
- Il y a aussi les cadres de type individus ou instances

## Nenette

instance : ELEPHANT  
aime : mangues

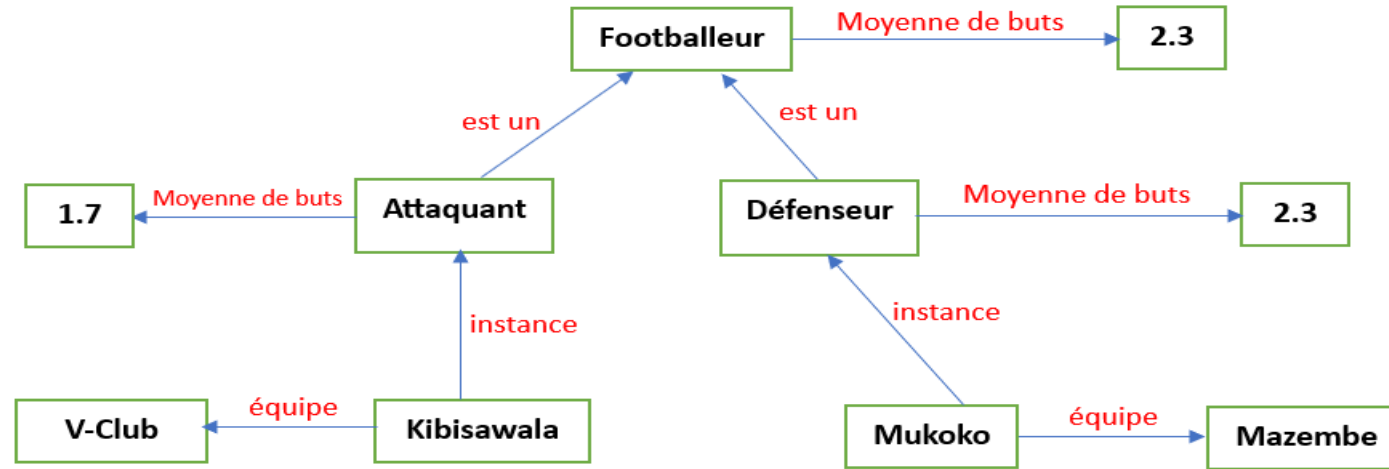
# Emplacements (Slots)

- Les cadres sont constitués d'une sélection d'**emplacements** qui peuvent être remplis par **des facettes/remplissages** :
  - **Valeurs** (connaissance statique) (peuvent représenter une valeur par défaut ou une exception)
  - **Jointures procédurales** (**démon**)
    - par exemple, activer une règle si la valeur dépasse un seuil (connaissance dynamique)
  - **Pointeurs** vers d'autres cadres (**hiérarchie d'héritage**)
- **Valeurs par défaut** fournies pour la plupart des slots



# Conversion entre réseaux et trames

- Il est facile de **construire des cadres pour chaque nœud** d'un **réseau sémantique** en lisant les liens, par exemple:



<b>Footballeur</b>
est_un: Homme adulte
moyenne de buts : 2.3
équipe :
⋮

<b>Défenseur</b>
est_un : footballeur
moyenne de buts : 2.3

<b>Mukoko</b>
instance : Footballeur
équipe : Mazembe

# Cadres : simples et au-delà



- Le type de **cadre le plus simple** est une structure de données dont les propriétés et les possibilités de représentation des connaissances sont similaires à celles **d'un réseau sémantique**, avec les mêmes idées d'**héritage** et de **valeurs par défaut**.
- Les cadres deviennent beaucoup plus puissants :
  - Les slots contiennent des instructions (procédures) pour calculer des choses à partir d'informations contenues dans d'autres slots ou dans d'autres cadres.

# En savoir plus sur les Emplacements (Slots)

- Chaque **slot** contient **une** ou **plusieurs facettes**
- Les facettes peuvent prendre les formes suivantes :
  - **Valeurs**
  - **Valeurs par défaut**
    - utilisées s'il n'y a pas d'autre valeur présente
  - **Plages**
    - quel type d'information peut apparaître dans le slot
  - **Si-ajouté**
    - **attachement (jointure) procédural** qui spécifie une action à entreprendre lorsqu'une valeur dans le slot est ajoutée ou modifiée (raisonnement guidé par les données, guidé par les événements ou ascendant)
  - **Si-nécessaire**
    - **attachement procédural** qui déclenche une procédure qui va chercher des informations que le slot ne possède pas (raisonnement guidé par les attentes ; raisonnement descendant)
  - **Autre**
    - peut contenir des cadres, des règles, des réseaux sémantiques ou d'autres types de connaissances.

# Slots comme des objets à part entière



- Les principales propriétés de remplissage que nous souhaitons généralement représenter sont les suivantes:
  1. Des détails sur le fait que le slot soit simple ou multivalué
  2. Contraintes sur les plages de valeurs ou les types de valeurs
  3. Valeurs par défaut simples pour l'attribut
  4. Règles d'héritage des valeurs de l'attribut
  5. Règles de calcul des valeurs indépendamment de l'héritage
  6. Les classes/cadres auxquels il peut être rattaché
  7. Les attributs inverses
- Naturellement, l'interprète du système de cadres doit savoir comment traiter ces cadres

# La logique à la base des cadres

- On peut représenter aussi les cadres sous forme logique
- $\forall x \text{ mammifère}(x) \Rightarrow a\_partie(x, \text{tête})$
- $\forall x \text{ éléphant}(x) \Rightarrow \text{mammifère}(x)$

éléphant(Clyde)

$\therefore$

mammifère(Clyde)

$a\_partie(\text{Clyde}, \text{tête})$

- Le fait de connaître que Clyde est un éléphant (partant de la règle d'héritage), signifie qu'on peut conclure que Clyde est un mammifère. Ainsi, conclure aussi que Clyde a une tête

<b>MAMMIFERE :</b> sous-classe : ANIMAL a_partie : tête *fourrure : oui
<b>ELEPHANT :</b> sous-classe : MAMMIFERE a_tronc : oui *couleur : gris *taille : grande *fourrure : non
<b>Clyde</b> instance : ELEPHANT couleur : rose propriétaire : Marcel
<b>Nenette</b> instance : ELEPHANT taille : petite

# Valeurs par défaut dans les cadres



- (\*) utilisé pour indiquer des valeurs par défaut qui peuvent être surchargées par :
  - une instance ou une sous-classe
- Les objets et les classes héritent des propriétés par défaut (typiques) des classes parentes, à moins qu'ils ne possèdent une propriété individuelle qui entre en conflit avec celle héritée.
- L'instance Nenette est petite bien qu'elle ait hérité d'une grande taille de la classe mère éléphant.

## MAMMIFERE :

sous-classe : ANIMAL  
a\_partie : tête  
\*fourrure : oui

## ELEPHANT :

sous-classe : MAMMIFERE  
a\_tronc : oui  
\*couleur : gris  
\*taille : grande  
\*fourrure : non

## Clyde

instance : ELEPHANT  
couleur : rose  
propriétaire : Marcel

## Nenette

instance : ELEPHANT  
taille : petite



# Héritage dans les cadres

- L'héritage est similaire à celui des réseaux sémantiques.
  - L'idée des hiérarchies de cadres est très similaire à l'idée des hiérarchies de classes que l'on trouve dans la **programmation orientée objet**.
- **Héritage multiple** :
  - Plus d'une classe parente autorisée
  - Spécifier l'ordre de **priorité** des classes parentes (**cirque-animal** d'abord)
  - Pas toujours suffisant !
  - Par exemple, nous voulons hériter de la taille de l'éléphant, mais de l'habitat du **cirque-animal**.
    - Normalement, il faut spécifier la classe **Cirque-Eléphant**.

## ELEPHANT :

sous-classe : MAMMIFERE  
a\_tronc : oui  
\*couleur : gris  
**\*taille : grande**  
\*habitat : jungle

## Clyde :

instance : **Cirque-Animal ELEPHANT**  
couleur : rose

## Nenette :

instance : Cirque-Animal

## Cirque-Animal :

sous-classe : Animal  
habitat : tente  
compétences : se balancer sur la balle  
**\*taille : petite**

# Construction et utilisation des cadres



- Nécessaire pour manipuler les instances et les classes
- Un **constructeur de classe** : crée un cadre qui a une superclasse directe
- Un **constructeur d'instance** : crée des cadres d'instance
  - L'entrée (input) est le nom de la classe à laquelle appartient l'instance
  - La nouvelle instance est reliée à la classe par un slot **est-un**
- Un slot de type **Writer**: installe des valeurs de slot
  - Ses données d'entrée sont le nom du cadre, le nom du slot et la valeur.
- Un slot de type **Reader**: récupère les valeurs des slots
  - Ses données d'entrée sont le nom du cadre, le nom du slot et la valeur correspondante en sortie.

# Cadres classés par application



- Un **cadre situationnel** contient des connaissances sur ce à quoi il faut s'attendre dans une situation donnée.
- Un **cadre d'action** contient des emplacements (slots) qui spécifient les actions à effectuer dans une situation donnée.
- La **combinaison des cadres de situation et d'action** peut être utilisée pour décrire les relations de cause à effet sous la forme de cadres de connaissances causales.
- En **général**, les cadres sont utiles pour représenter des hiérarchies de classes afin de modéliser des domaines dans lesquels :
  - les objets sont bien définis et la classification est claire

# Avantages des cadres



- Facilite la programmation en regroupant les connaissances connexes
  - Peut être facilement mis en œuvre à l'aide de techniques de programmation orientées objet
- Assez intuitif pour de nombreuses applications et pour les non-développeurs
  - Similaire à l'organisation des connaissances humaines
  - Convient aux connaissances causales
  - Plus facile à comprendre que la logique ou les règles
- Pouvoir d'expression, flexibilité
  - Facile à mettre en place des emplacements (slots) pour de nouvelles propriétés et relations
  - Facile à inclure des informations par défaut et à détecter les valeurs manquantes
  - Les facettes permettent un raisonnement non monotone
- L'héritage est facilement contrôlé
- Les cadres sont utiles pour simuler les connaissances de base, un domaine très difficile à maîtriser pour les ordinateurs.

# Inconvénients des cadres

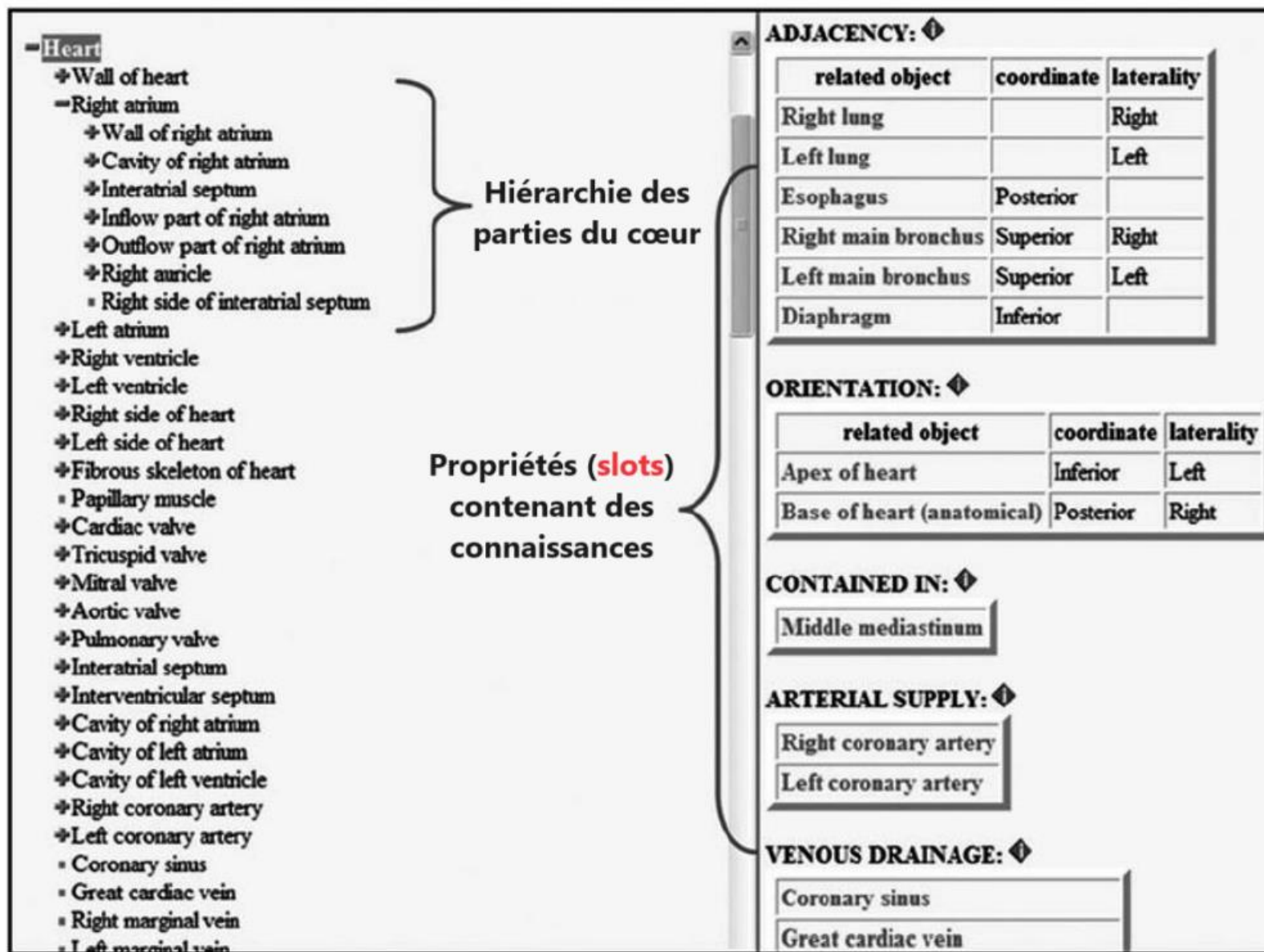


- Tout comme pour les **réseaux sémantiques**, il **n'existe pas de normes** pour définir les systèmes basés sur des cadres (sémantique).
  - Il s'agit davantage d'une méthodologie générale que d'une représentation spécifique :
    - Le cadre d'une salle de classe sera différent pour un professeur et pour un agent d'entretien.
- Pas de mécanismes formels de raisonnement/inférence
  - Toutes les propriétés d'un stéréotype de classe ne doivent pas être propagées aux sous-classes.
  - La modification des slots peut avoir des conséquences inattendues dans les sous-classes.
- Les cadres ne peuvent pas représenter des exceptions
  - Par définition, les cadres représentent des objets typiques

# Exemple de cadre dans le domaine médical



- Foundational Model of Anatomy (**FMA**), en Français Modèle fondamental d'anatomie est un cadre qui a été développé par l'Université de Washington (le lien de ce cadre est mis dans le bas de cette page)
- Le FMA est une ontologie de l'anatomie humaine qui fournit spécifiquement un son, une exactitude et une théorie formelle constante de l'ontologie humaine
- En pratique, il s'agit d'une source de connaissances informatisée évolutive pour les informations biomédicales.
  - Il concerne la représentation des classes ou des types et des relations nécessaires à la représentation symbolique de la structure phénotypique du corps humain sous une forme compréhensible par l'homme et également navigable, analysable et interprétable par des systèmes basés sur des machines.





# Scripts



# Scripts



- Conçu par Schank en 1977 :
  - Schank, R.C. & Abelson, R. (1977) Scripts, Plans, Goals and Understanding
- Séquences d'images ordonnées dans le temps
  - Schéma de représentation des connaissances décrivant une séquence stéréotypée d'événements, ainsi que les objectifs et les plans des acteurs concernés
  - Programmation axée sur les attentes
- Permet de déduire des relations implicites de cause à effet lors de l'interprétation des situations
- Chaque cadre est accompagné de plusieurs types d'informations :
  - Comment utiliser le cadre
  - Ce que l'on peut attendre de la suite des événements ou ce que l'on doit faire ensuite.
    - Certaines informations peuvent concerner ce qu'il faut faire si nos attentes ne sont pas confirmées.
  - Lorsque l'on est confronté à une nouvelle situation :
    - Sélectionner de mémoire un cadre approprié et l'adapter à la réalité en changeant certains détails si nécessaire.

# Script: Exemple



- Dans cet exemple, nous avons le Script Manger au restaurant avec les accessoires incluant argent, nourriture, etc.
- On voit comment le script présente les événements
- En parcourant ces événements, nous pouvons voir que d'autres scripts peuvent être appelés, chacun avec sa propre séquence d'événements et de propriétés. Par exemple, le script enter-restaurant, s'il y a un signe d'attente ou une réservation, continue avec le reste du script, sinon il s'arrête.

Script Manger au restaurant	
Accessoires :	(Restaurant, Argent, Nourriture, Menu, Tables, Chaises)
Rôles :	(Personnes affamées, personnes d'attente, chefs cuisiniers)
Point de vue :	Personnes affamées
Durée de l'événement :	(Durée de fonctionnement du restaurant)
Place de l'événement	(localisation du restaurant)
Séquences d'événement :	<p>Début : <b>Script</b> entrer-Restaurant</p> <p>puis : <b>Script</b> si (signe d'attente ou réservation) alors obtenir l'attention du chef du restaurant</p> <p>puis : <b>Script</b> veuillez-vous asseoir</p> <p>puis : <b>Script</b> commander de la nourriture</p> <p>puis : <b>Script</b> manger, sauf si l'attente est longue et que l'on quitte le restaurant en colère</p> <p>puis : <b>Script</b> si (la qualité de la nourriture est supérieure à la qualité gustative) alors compliments au chef</p> <p>puis : <b>Script</b> payer</p> <p>enfin: <b>Script</b> quitter le restaurant</p>

# Eléments du script

- Conditions d'entrée
  - Quels sont les descripteurs du monde qui doivent être vrais pour que le script soit appelé ?
  - C'est similaire à *si partie d'un si alors un argument ou proposition*
- Accessoires (props)
  - Quels sont les objets qui composent le contenu du script ?
- Rôles
  - Quelles sont les actions accomplies par les participants du script?
- Scènes
  - Décomposition temporelle du script en épisodes significatifs.
- Résultats
  - Quels sont les résultats à l'issue du script?



# Comprendre le langage naturel des scripts



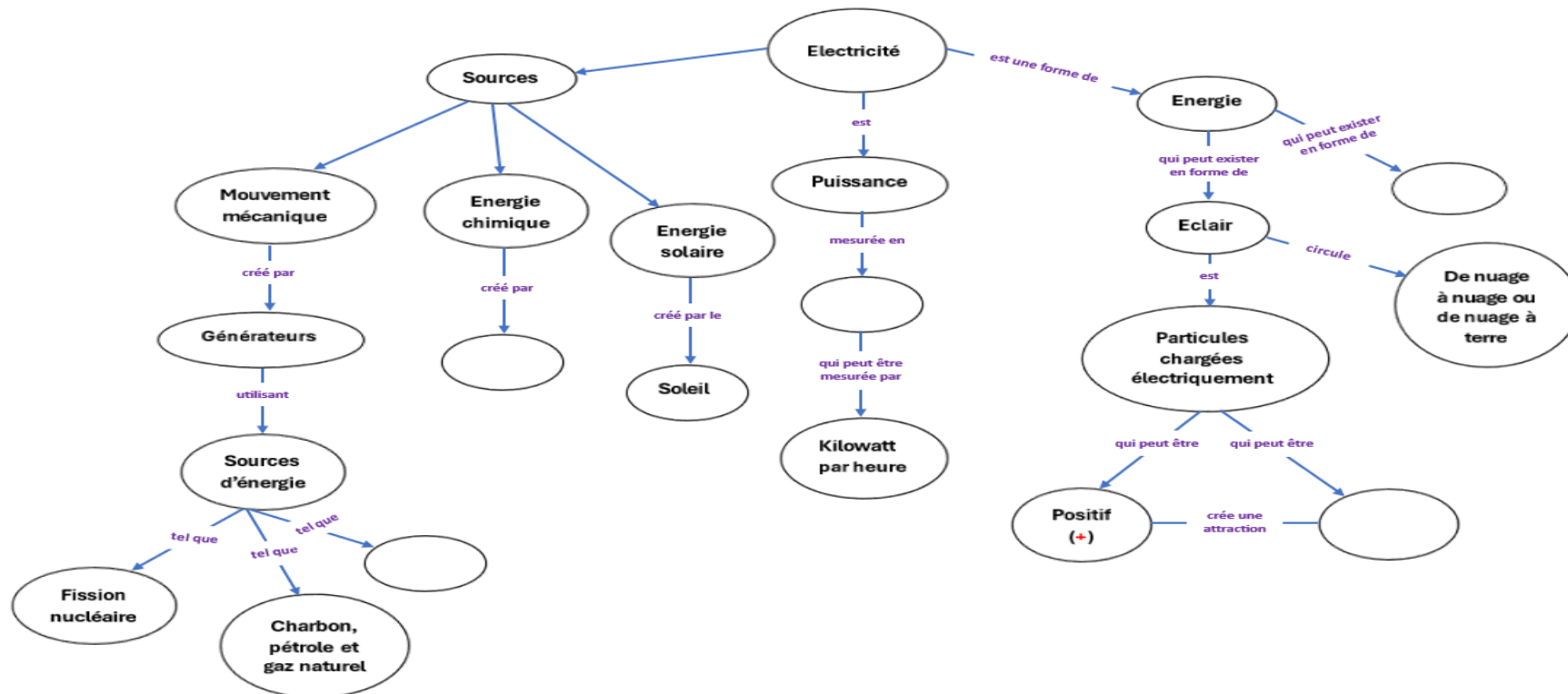
- Les scripts sont utilisés pour organiser une base de connaissances **en termes de situations à comprendre**
- "Dr. Nsenge est allé au restaurant. Il a décidé de commander un Poulet Mayo. Il s'est assis et a attendu longtemps. Finalement, il s'est mis en colère et est parti".
- On peut ainsi poser des questions suivantes dans le KB:
  - Qu'attendait Dr. Nsenge?
  - Pourquoi s'est-il mis en colère ?
- Le script est souvent efficace dans des domaines limités, peut être inflexible dans des domaines plus généraux



# Autres représentations liées aux réseaux sémantiques

# Concept Map

- Diagramme décrivant les relations suggérées entre les concepts
- Idée plus générale qu'un réseau sémantique
  - Moins formel
  - Certaines relations sont très complexes et ne suivent pas un ensemble très spécifique de relations sémantiques
- Un **concept map** pourrait être le point de départ de la création d'un réseau sémantique plus formel



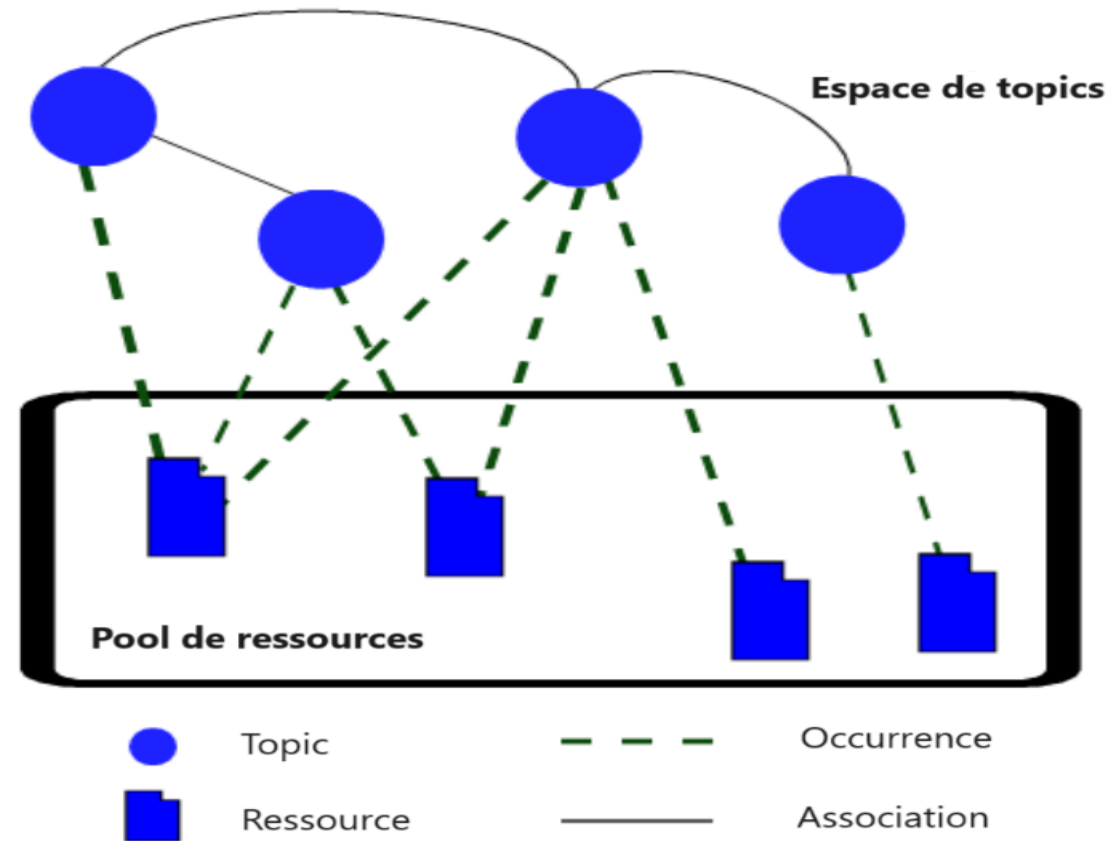
# Topic Maps

- Une norme pour la représentation et l'échange de connaissances
  - L'accent est mis sur la facilité de trouver des informations ou des ressources.
- Similaire à la **Concept Map**
  - Toutefois, les topic maps sont des normes de l'Organisation internationale de normalisation (ISO).
- Un topic map représente l'information à l'aide de:
  - **Topics** (thèmes/sujets): représentant n'importe quel concept, depuis les personnes, les pays et les organisations jusqu'aux modules logiciels, aux fichiers individuels et aux événements.
  - **Associations** : représentant les relations hypergraphiques entre les topics
  - **Occurrences** : représentent les ressources d'information pertinentes pour un topic particulier.



# Illustration d'un Topic Map

- En boules bleues sont des topics connectés par des associations avec ces lignes
- Les occurrences sont représentées par les dash dash
- Et les ressources spécifiques devraient être trouvées à l'intérieur d'une pool de ressources



# Graphes conceptuels

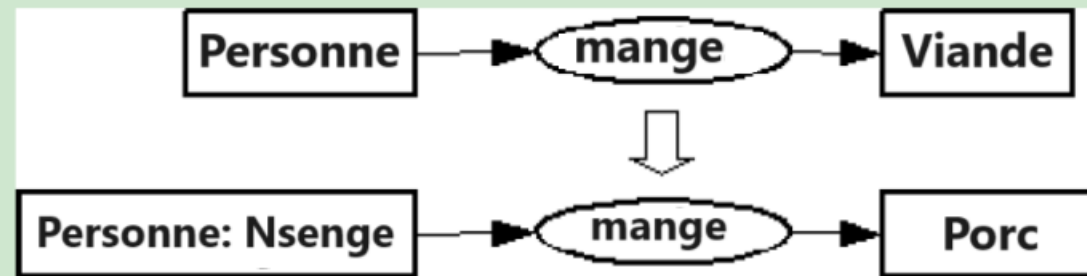


- Remarque : il ne s'agit pas de la même chose que les **Concept maps**
  - Les concept maps sont plus simples
- Un formalisme pour la représentation des connaissances
  - Représentation graphique de la **logique** basée sur les réseaux sémantiques
- Trois axes de développement :
  - Une interface graphique pour la logique du premier ordre
  - Un calcul diagrammatique des logiques
  - Un modèle de représentation des connaissances et de raisonnement basé sur les graphes

# Graphes conceptuels : Inférence

- L'**inférence** utilise ici plusieurs règles de **chaînage avant**
  - Généralisation du graphique
  - Spécialisation
  - Modifications équivalentes
- L'interrogation (query) est effectuée à l'aide de la **projection** :
  - Recherche d'un modèle de graphique conceptuel dans un autre graphique conceptuel en utilisant la hiérarchie des types conceptuels et les relations conceptuelles.

## Exemple - Projection



# Synthèse



- **Réseaux sémantiques :**
  - Facilité de mise en œuvre et d'interprétation
  - Très flexibles
  - Inférence simple grâce à l'héritage
- **Cadres :**
  - Extension des réseaux sémantiques basée sur le concept de typicité
  - Inférence plus efficace et plus claire
  - Fondement de nombreuses bases de connaissances du monde réel
  - Ni les réseaux sémantiques ni les cadres :
    - n'ont pas de sémantique standard
    - Peuvent limiter les applications et poser des problèmes d'inférence
- **Scripts :**
  - Étendent les cadres à des séquences de cadres ordonnées dans le temps
- **Autres représentations liées aux réseaux sémantiques**

**Je suis instinctivement un plombier  
d'infrastructure de réseau.**

(Audrey MacLean)

# Travail pratique 2: Rappel

- **Objectifs :**

- Logique propositionnelle et logique du premier ordre
  - S'entraîner à travailler avec la syntaxe et la sémantique de la logique
  - S'entraîner à manipuler des expressions logiques de manière "algébrique".
  - Appliquer la logique pour faire des déductions et déterminer la "vérité".
  - Comprendre deux des approches d'inférences les plus simples sur lesquelles nous nous concentrerons principalement dans ce cours :
    - Le chaînage avant
    - Le chaînage arrière