

<https://piazza.com/upmc.fr/fall2015/m1lrc/>

LREC

Représentations sémantiques
Graphes conceptuels
Graphes de Sowa



ACASA - Université Pierre et Marie CURIE

Représentation des connaissances

1. Mémoire (externe et interne)

2. Représentation

3. Réseaux sémantiques, « Frames », ...

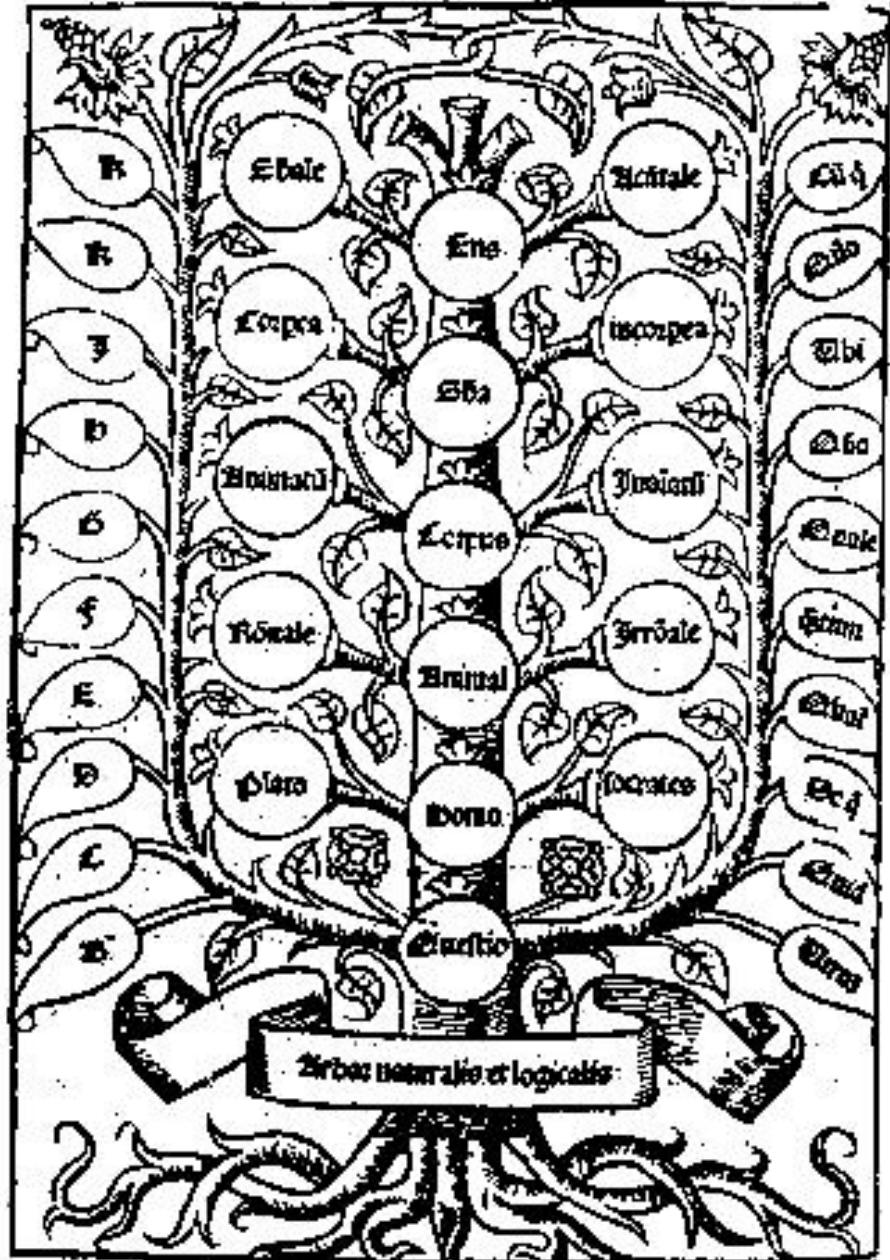
4. Graphes conceptuels



Graphes conceptuels

John Sowa 1984

- **Défis de la représentation:**
 - Adéquation psychologique
 - Traduction du langage naturel
 - Automatisation des inférences
 - **Représentation sous forme de graphes**
 - Visualisation
 - Traduction du langage naturel
 - Manipulables par une machine
 - Equivalence avec la logique
 - **Origine:**
 - IA
 - Réseaux sémantiques
 - Graphes « existentiels » de Charles Sanders Peirce (1839 - 1914)



Graphes existentiels

Charles Sanders Peirce 1839 – 1914

—homme *Il y a un homme*

$\exists x \text{ homme}(x)$



Charles Sanders Peirce was an American philosopher, logician, and scientist. He is considered one of the most important figures in the history of Western philosophy.

homme
mange —homme

*Un homme mange
un homme*

$\exists x \exists y \text{ homme}(x) \wedge \text{homme}(y) \wedge \text{mange}(x, y)$

Graphes existentiels C S Peirce (suite)

—homme *Il y a un homme*

Négation: grisé

—homme *Il est faux qu'il y a un homme*

—homme

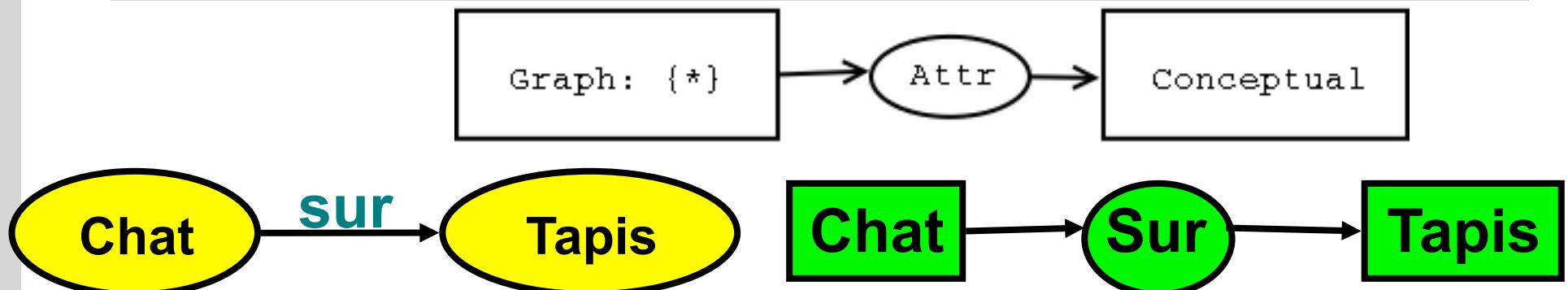
Il y a quelque chose qui n'est pas un homme



Introduction aux graphes conceptuels

« Un chat est sur un tapis »

- Une graphe conceptuel comporte deux types de noeuds:
 1. Des concepts, représentés avec des rectangles
 2. Des relations, représentées à l'aide d'ovales



Forme linéaire: [Chat] -> (Sur) -> [Tapis]

Traduction logique: $(\exists x: \text{Chat}) (\exists y: \text{Tapis}) \text{Sur}(x, y)$



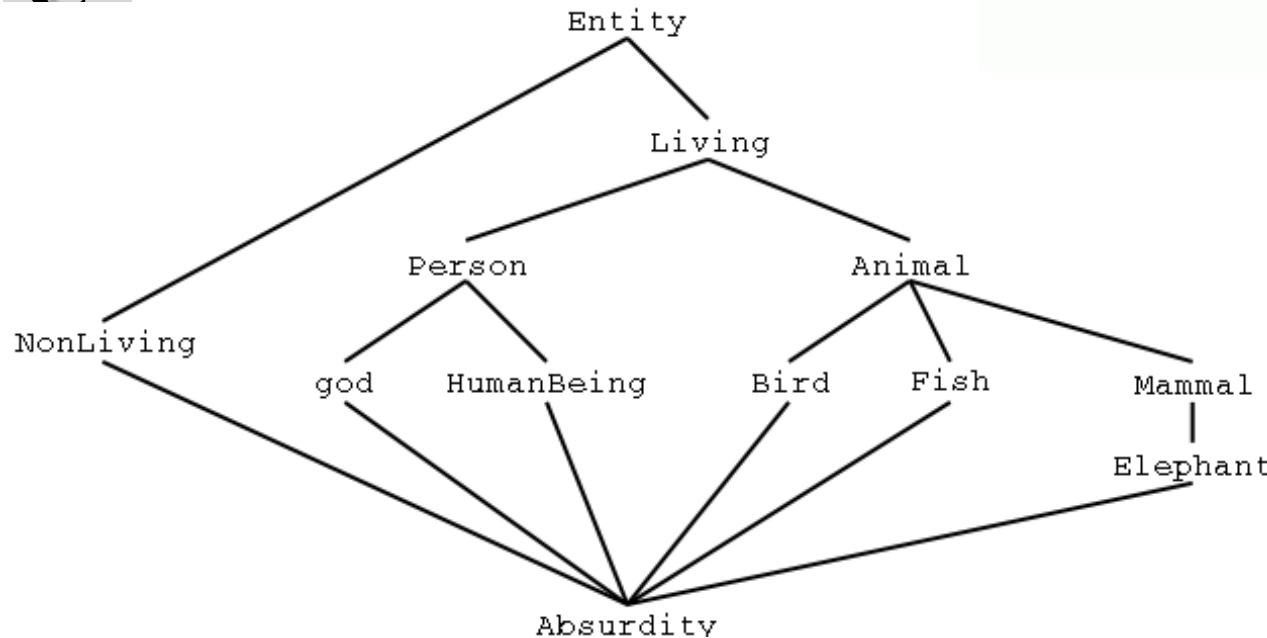
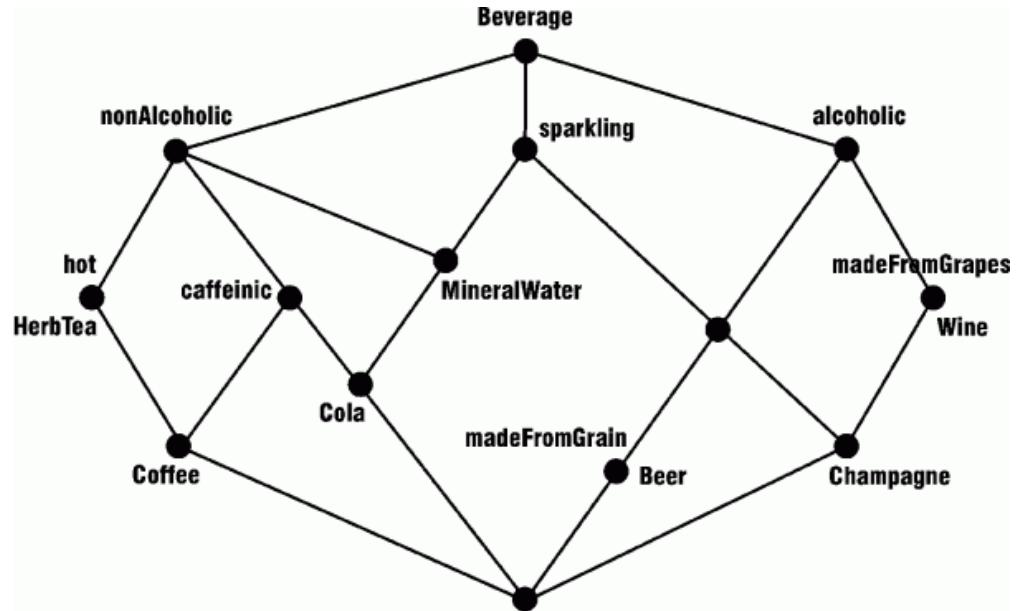
Les entités: Concepts, Types, Référents

- Concepts génériques:
 - [rectangle], [BUS], [PERSONNE], [VILLE], [CHAT]...
- Concepts individuels:
 - [PERSONNE : Jean], [VILLE : Paris],
 - Type de concept: [PERSONNE :], [VILLE :],
 - Référent: Jean, Paris, ...
- Référent:
 - Nom
 - Quantificateur universel ‘ \forall ’
 - Liste
 - Question ‘?’
 - Nombre ‘@n’
 - Quantité indéfinie ‘*’



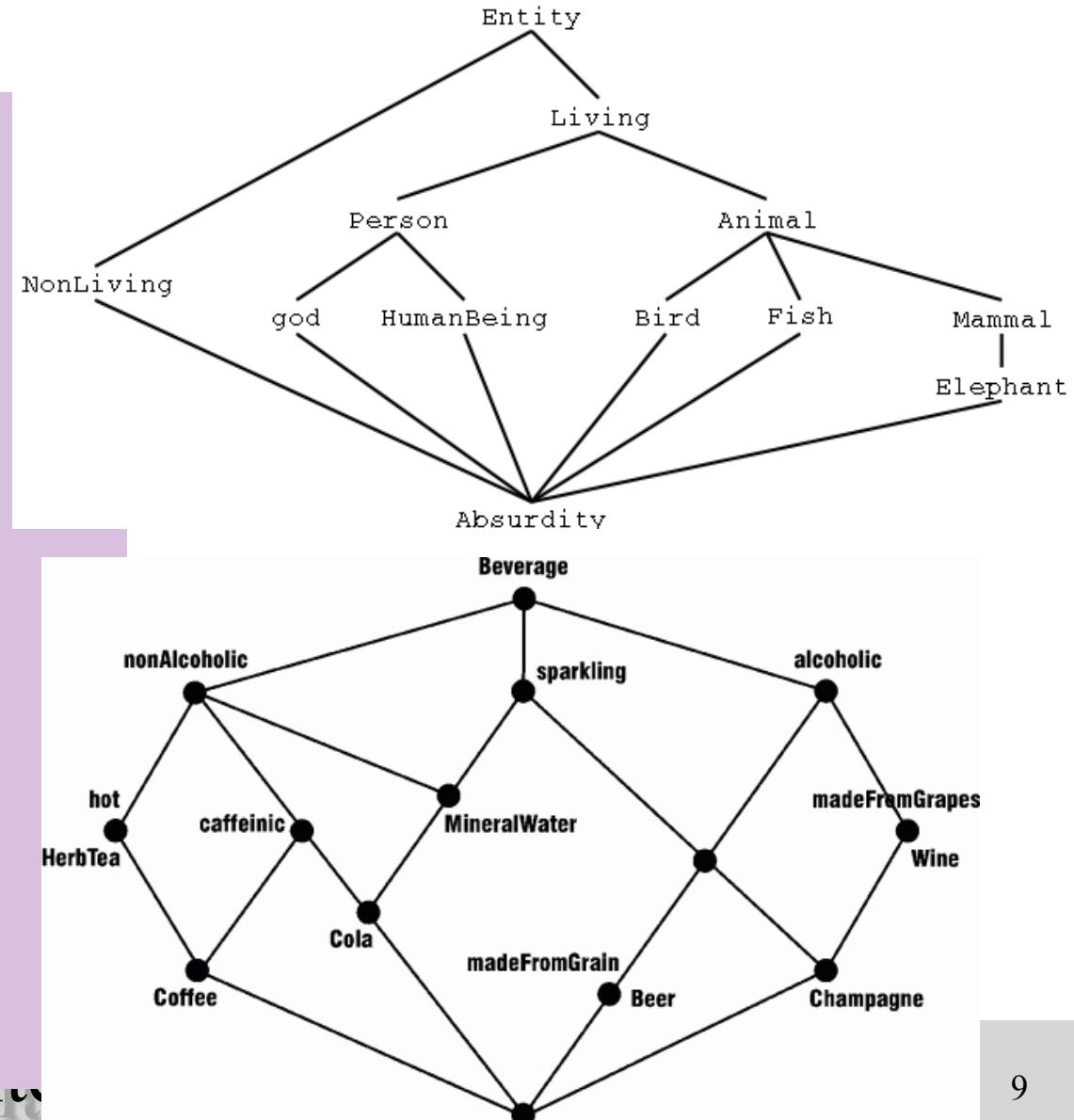
Hiérarchie des types - Ontologie

- Les types de concepts peuvent être organisés en « hiérarchies de types » qui sont en fait des treillis
- Relation de généralité entre type



Concepts: Individuels, Génériques, Types, Référents, Hiérarchie des types...

- Concepts génériques:
 - [rectangle], [BUS], [PERSONNE], [VILLE], [CHAT]...
- Concepts individuels:
 - [PERSONNE : Jean], [VILLE : Paris],
 - Type de concept: [PERSONNE :], [VILLE :],
 - Référent: Jean, Paris, ...
- Référent:
 - Nom
 - Quantificateur universel ‘ \forall ’
 - Liste
 - Question ‘?’
 - Nombre ‘@n’
 - Quantité indéfinie ‘*’



Types de relation

- Les relations constituent une hiérarchie (treillis) au sens où nous avons défini une « hiérarchie de relations »
- Les relations comportent toutes:
 - Une valence (arité): *nombre de flèches*
Exemple: DANS a pour valence 2
[PERSONNE: César] -> (DANS) -> [VILLE: Rome]
 - Une signature (type des arguments): *liste ordonnée de types <t₁, ...t_n>*
Exemple: MANGE a pour signature <ETRE-ANIME, OBJET-PHYSIQUE>



Modalités

« Tous les chats sont sur un tapis »



Forme linéaire

[Chat: \forall] -> (Sur) -> [Tapis]

ou

[Chat: @every] -> (Sur) -> [Tapis]

Traduction logique

$(\forall x:\text{Chat}) (\exists y:\text{Tapis}) \text{Sur}(x, y)$

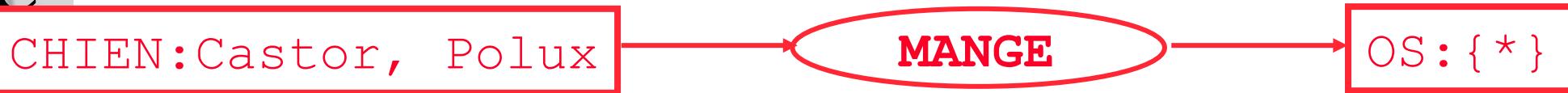


Référents (suite)

- Tous les chiens mangent de la viande - quantificateur
[chien: \forall] -> (mange) -> [viande]



- Castor et Polux mangent des os - énumération
[chien:Castor, Polux] -> (mange) -> [os : { * }]



- Quel chien mange la côtelette ? - interrogation - index
[chien:?] -> (mange) -> [côtelette:#]



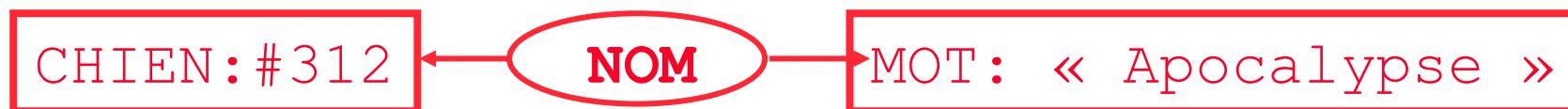
Graphes conceptuels - référents

- **Elle mange quatre os - cardinalité**
[femelle:#] → (mange) → [os: { * } @4]



- **Le chien nommé Apocalypse - marqueur individuel**

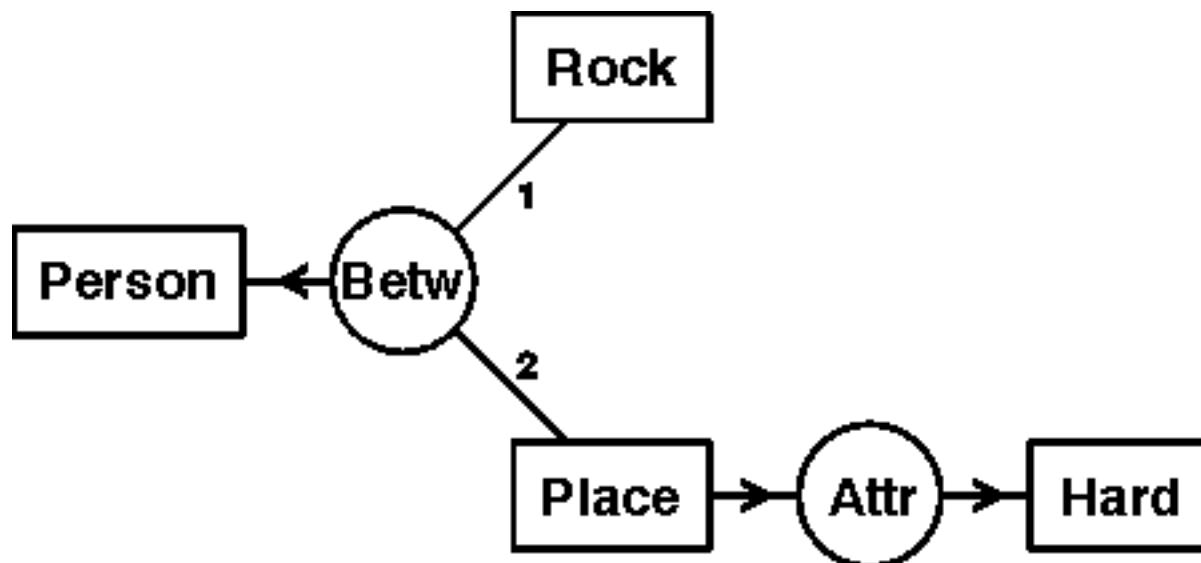
[chien:#312] ← (nom) → [mot: « Apocalypse »]



Prédicat n-aire

```
[Person] <- (Betw) -  
          <-1-[Rock]  
          <-2-[Place]->(Attr)->[Hard]
```

$(\exists x: \text{Person}) (\exists y: \text{Rock}) (\exists z: \text{Place}) (\exists w: \text{Hard})$
 $(\text{betw}(y, z, x) \wedge \text{attr}(z, w))$



Peut-on limiter le nombre de relation?

- Doit-il y avoir autant de relations que de verbes



- Elle mange quatre os - cardinalité
 $[femelle:\#] \rightarrow (\text{mange}) \rightarrow [\text{os: } \underline{\{ * \}} @4]$

- Idée:

- transformer les verbes en concepts
- Introduction de « primitives sémantiques »



- Polux mange quelques os - générique pluriel
 $[\text{chien: Polux}] \leftarrow (\text{Agent}) \leftarrow [\text{mange}] \rightarrow (\text{Patient}) \rightarrow [\text{os: } \underline{\{ * \}}]$

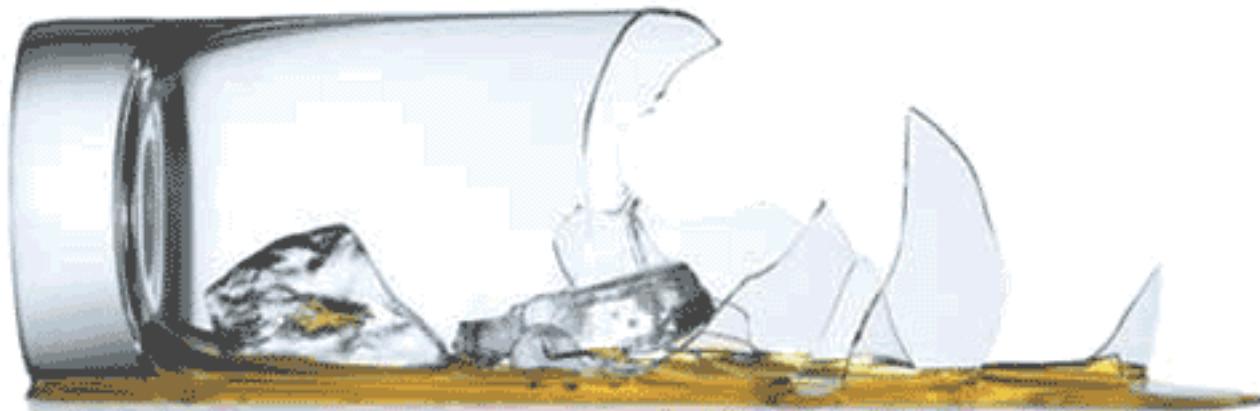


Primitives sémantiques

Représentation des actions:

un verre casse!

- **Un verre casse**
- **La pierre casse le verre**
- **Jean casse le verre**
- **Jean casse le verre avec une pierre**



Langues flexionnelles

| | | |
|---------------------|---------|-----------------------------|
| Nominatif singulier | Rosa | la <i>ou</i> une rose |
| Vocatif singulier | Rosa | rose, ô rose ! |
| Génitif singulier | Rosae | de la rose |
| Accusatif singulier | Rosam | la <i>ou</i> une rose |
| Datif singulier | Rosae | à la rose |
| Ablatif singulier | Rosa | de <i>ou</i> par la rose |
| Nominatif pluriel | Rosae | les <i>ou</i> des roses |
| Vocatif pluriel | Rosae | roses, ô roses ! |
| Génitif pluriel | Rosarum | des roses |
| Accusatif pluriel | Rosas | les <i>ou</i> des roses |
| Datif pluriel | Rosis | aux roses |
| Ablatif pluriel | Rosis | des <i>ou</i> par les roses |



- **Langues flexionnelles (ou casuelles):**

- Latin
- Russe
- Allemand
- ...

- **Les rapports grammaticaux sont indiqués**

- par des flexions (c'est-à-dire par des changements morphologiques) et
- Ni par l'ordre des mots, ni par des prépositions

Grammaires de « cas » (Fillmore 1971)

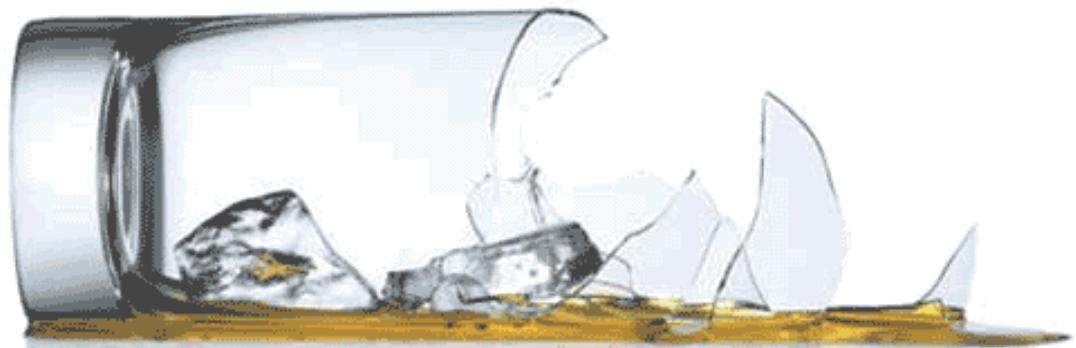
- Les **rapports sémantiques** sont indiqués
 - par des **cas sémantiques** (Fillmore en dénombre 8)
 - OBJET**: ce sur quoi porte l'action (ce qui est modifié, bougé...)
 - INSTRUMENT**: par quel intermédiaire l'action est accomplie
 - AGENT**: l'instigateur de l'action
 - CONTRE-AGENT**: la force contre laquelle l'action s'oppose
 - RESULTAT**: ce qui est créé par l'action
 - SOURCE**: lieu de départ
 - BUT**: lieu d'arrivée
 - PATIENT**: l'entité qui reçoit ou subit les effets de l'action
- Chaque verbe correspond à une action
- Il commande un certain nombre de cas





Grammaires de « cas » (Fillmore 1971)

- **Exemple:** le verbe *Casser* commande trois cas:
 - OBJET
 - INSTRUMENT
 - AGENT



- **Un verre casse**

[OBJET: *le verre*]

- **La pierre casse le verre**

[OBJET: *le verre*; INSTRUMENT: *la pierre*]

- **Jean casse le verre**

[OBJET: *le verre*; AGENT: *Jean*]

- **Jean casse le verre avec une pierre**

[OBJET: *le verre*; AGENT: *Jean*; INSTRUMENT: *la pierre*]

Représentation des actions dans les réseaux sémantiques

- L'action « vendre » commande quatre cas:

- **AGENT** celui qui vend
- **PATIENT** celui qui reçoit
- **OBJET** ce qui est vendu
- **INSTRUMENT** le paiement

- *Exemple:
une vente particulière*

(Vente-3218

 (ACTION Vente)

 (AGENT Eléonore)

 (PATIENT Madame Gilberte Dupuis)

 (OBJET appareil photo numérique)

 (INSTRUMENT 200))



Graphes de Sowa

John Sowa 1984

- **Graphe conceptuels**
 - **Primitives sémantiques**

OBJET: ce sur quoi porte l'action (ce qui est modifié, bougé...)

INSTRUMENT: par quel intermédiaire l'action est accomplie

AGENT: l'instigateur de l'action

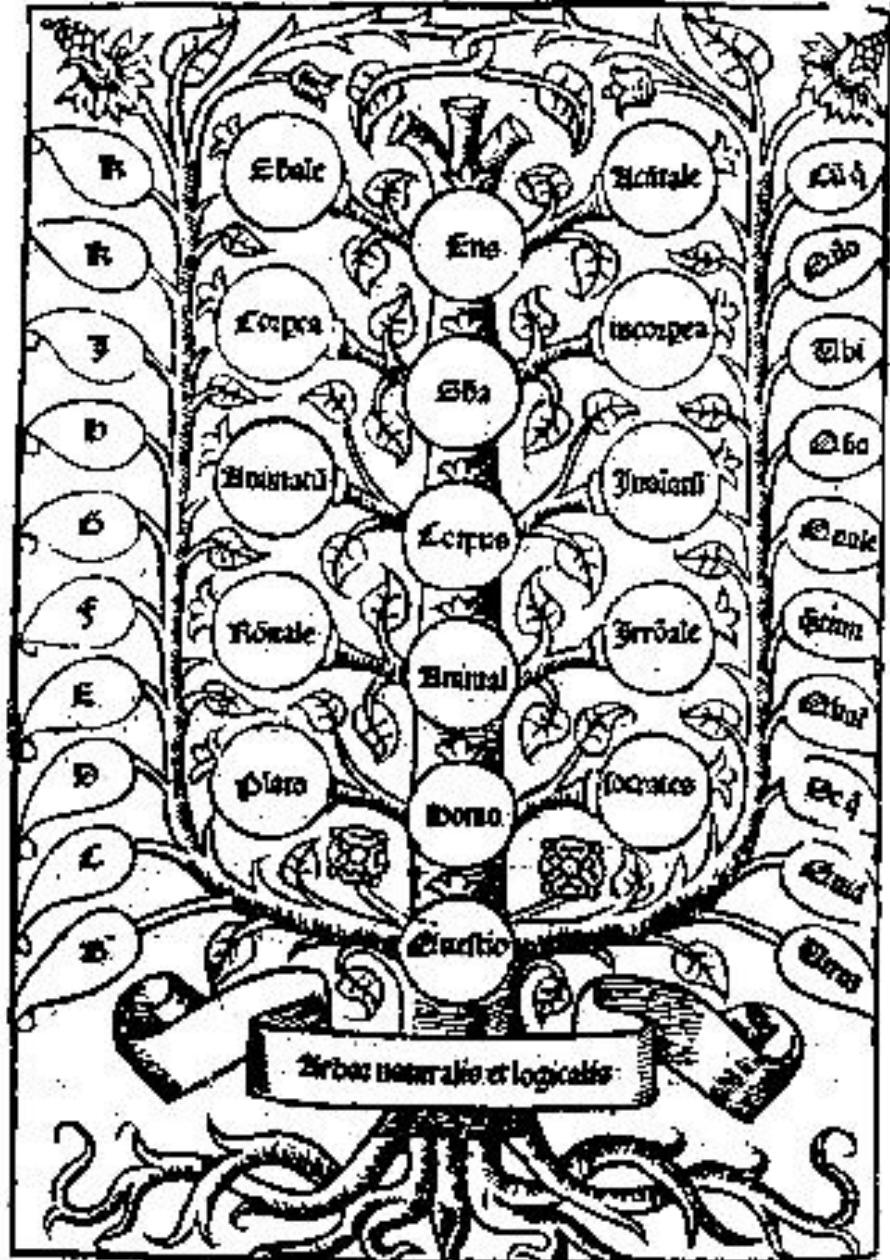
CONTRE-AGENT: la force contre laquelle l'action s'oppose

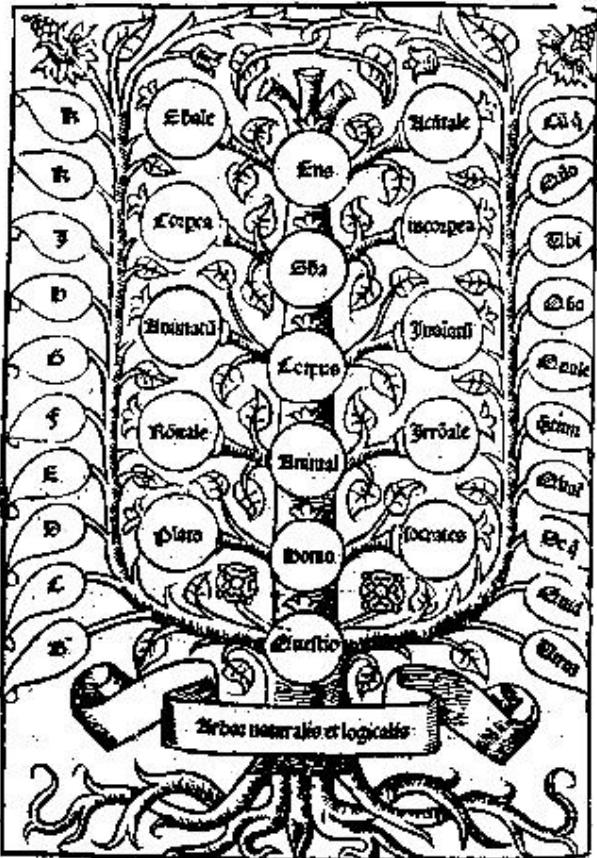
RESULTAT: ce qui est créé par l'action

SOURCE: lieu de départ

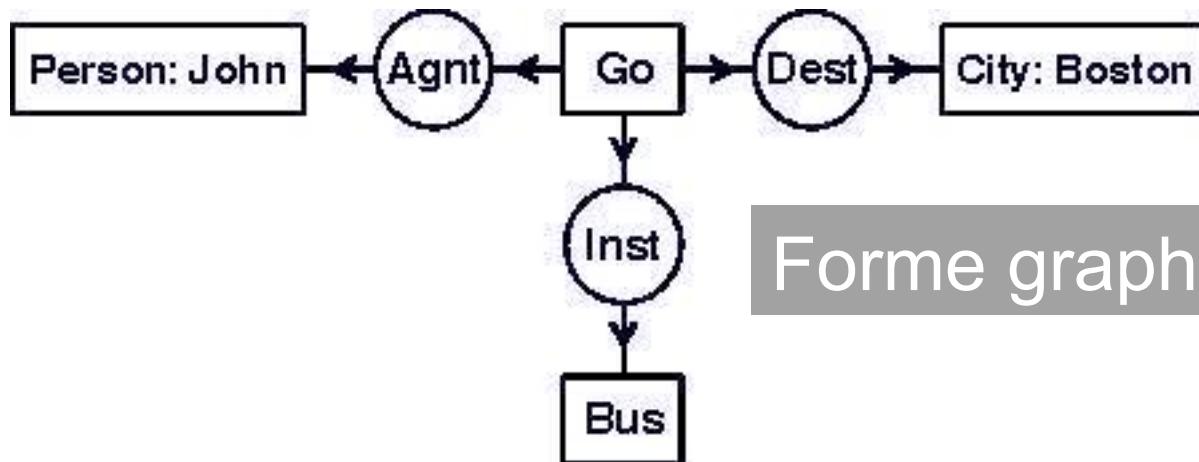
BUT: lieu d'arrivé

PATIENT: l'entité qui reçoit ou subit les effets de l'action





Un exemple de graphe de Sowa



Forme graphique

[GO] :

Forme linéaire

Traduction logique

(Agnt) → [PERSON: John]
(Dest) → [CITY: Boston]
(INST) → [BUS]

($\exists x: \text{Go}$) ($\exists y: \text{Person}$) ($\exists z: \text{City}$) ($\exists w: \text{Bus}$)

(name(y, 'John') \wedge name(z, 'Boston') \wedge agnt(x,y) \wedge
dest(x,z) \wedge inst(x,w))

Référents (sur les graphes de Sowa)

- Tous les chiens mangent de la viande - quantificateur

[chien: \forall]<- (Agent)<- [mange]->(Patient)->[viande]



- Castor et Polux mangent - énumération

[chien:Castor, Polux]<- (Agent)<- [mange]



- Quel chien mange la côtelette ? - interrogation - index

[chien:?]<- (Agent)<- [mange]->(Patient)->[côtelette:#]





Référents (graphes de Sowa)

- Polux mange quelques os - générique pluriel

[chien:Polux]<- (Agent)<- [mange]->(Patient)->[os:{*}]



- Elle mange quatre os - cardinalité

[femelle:#]<- (Agent)<- [mange]->(Patient)->[os:{*}@4]



Rôles thématiques – Sowa KR



- *Determinant*: le participant détermine la direction du processus, soit depuis le début, comme initiateur, soit par le fin, comme but (goal).
- Un participant *immanent* est présent depuis le début du processus, mais ne contrôle pas activement ce qui se produit
- Une *source* doit être présent au début du processus, mais n'a pas besoin de participer à l'ensemble du processus
- Un *produit* doit être présent à la fin du processus, mais n'a pas besoin de participer à l'ensemble du processus



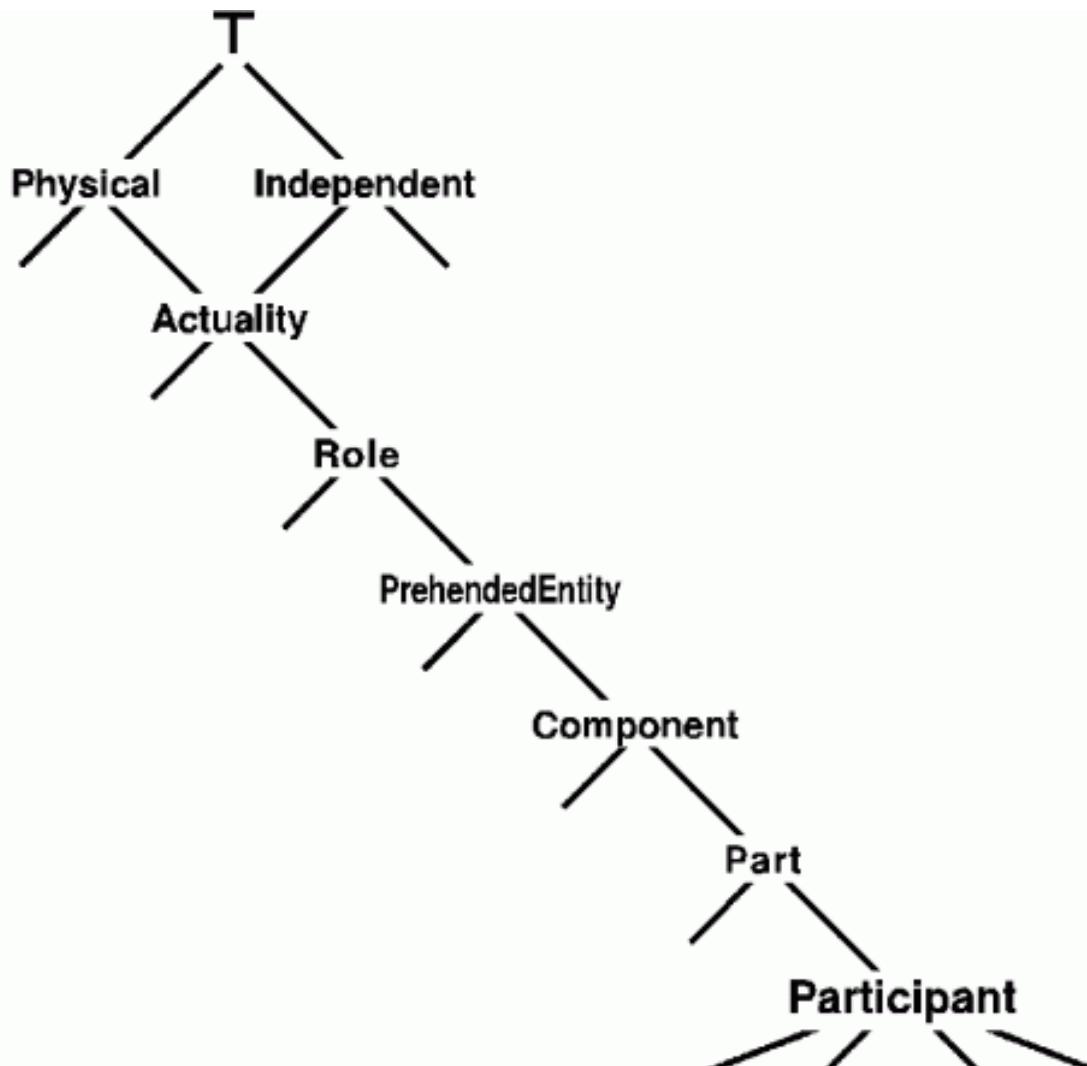
Rôles thématiques

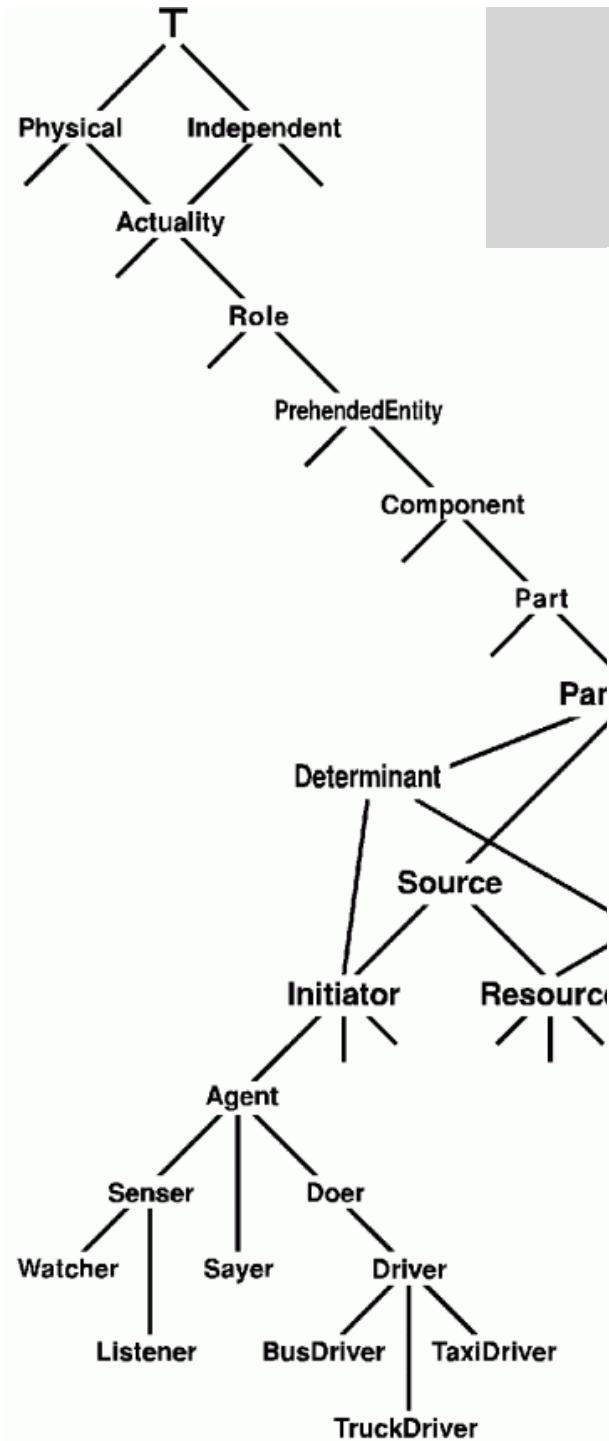


- ***Initiator:*** cause efficiente (Aristote) – par quoi le mouvement est initié
- ***Ressource:*** cause matérielle (Aristote) – quel est le substrat ou la matière
- ***Goal:*** cause finale (Aristote) – pour le bénéfice de qui ou de quoi le mouvement s'effectue
- ***Essence:*** cause formelle (Aristote) – ce que c'est

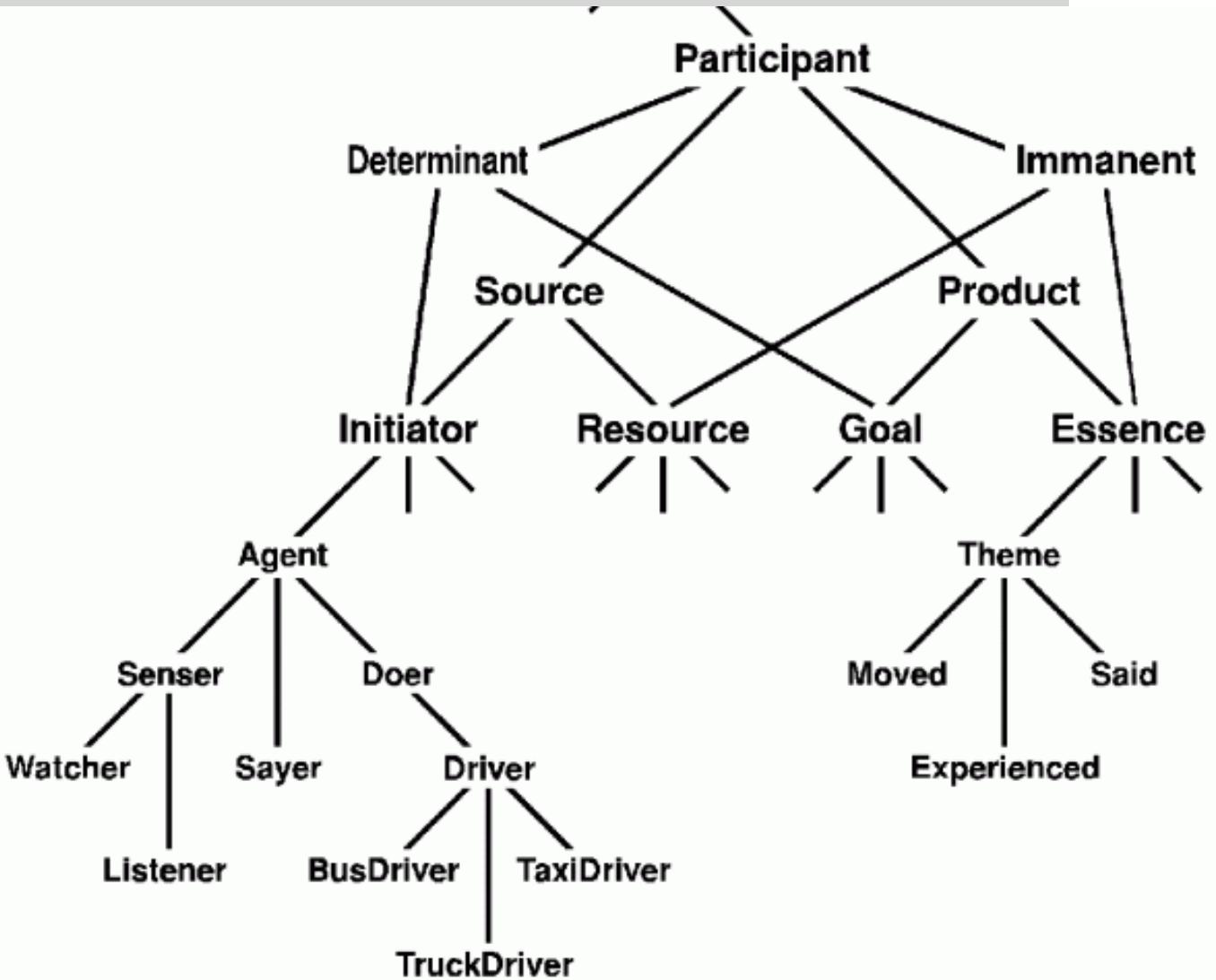


Rôles thématiques – suite





Rôles thématiques



Intentionnalité

- **Sens ancien (scholastique): application de l'esprit à un objet de connaissance et, par extension, le contenu de pensée correspondant.**
- **Brentano, Husserl (XIX^{ème}): distinction entre acte de pensée et l'objet visé par la pensée.**
- **Dennett (XX^{ème}): « système intentionnel » à qui l'on prête des « intentions », au sens précédent.**
 - Exemple: avatar affectifs
- **Représentation des connaissances: verbe comme croire, vouloir etc. où l'on distingue un sujet qui éprouve et un objet visé.**



Croyances et situation: intentionnalité

Tom believes that Mary wants to marry a sailor

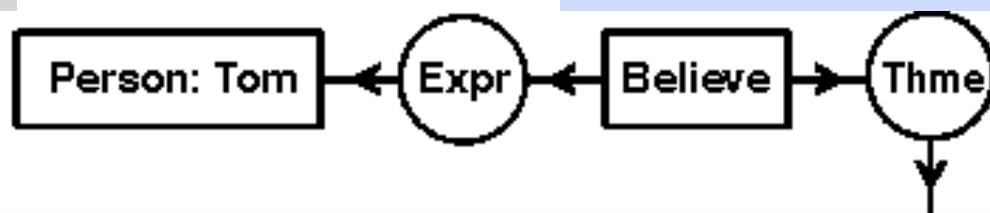
Relations: Expr (experienter celui qui éprouve), Thme (ce que l'on éprouve)

[Person: Tom] <- (Expr) <- [Believe] -> (Thme) -

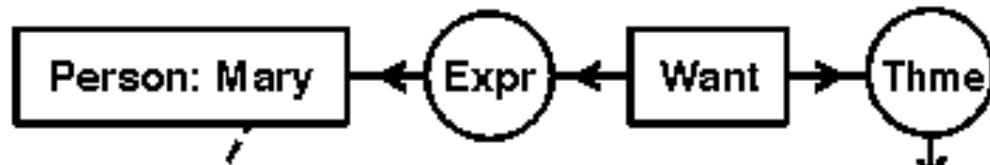
[Proposition: [Person: Mary *x] <- (Expr) <- [Want] -> (Thme) -

[Situation: [?x] <- (Agnt) <- [Marry] -> (Thme) -> [Sailor]]]

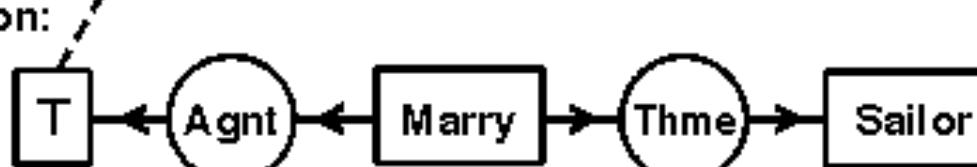
$(\exists x_1: \text{Person}) (\exists x_2: \text{Believe}) (\text{expr}(x_1, x_2) \wedge \text{thme}(x_2, (\exists x_3: \text{Person}) (\exists x_4: \text{Want}) (\exists x_8: \text{Situation}) (\text{name}(x_3, 'Mary') \wedge \text{expr}(x_4, x_3) \wedge \text{thme}(x_4, x_8) \wedge \text{dscr}(x_8, (\exists x_5: \text{Marry}) (\exists x_6: \text{Sailor}) (\text{agt}(x_5, x_3) \wedge \text{thme}(x_5, x_6))))))$



Proposition:

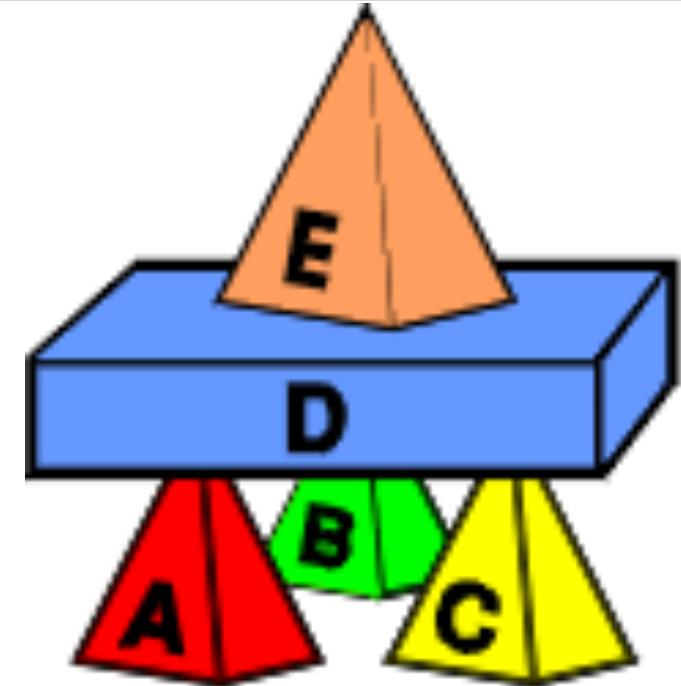


Situation:



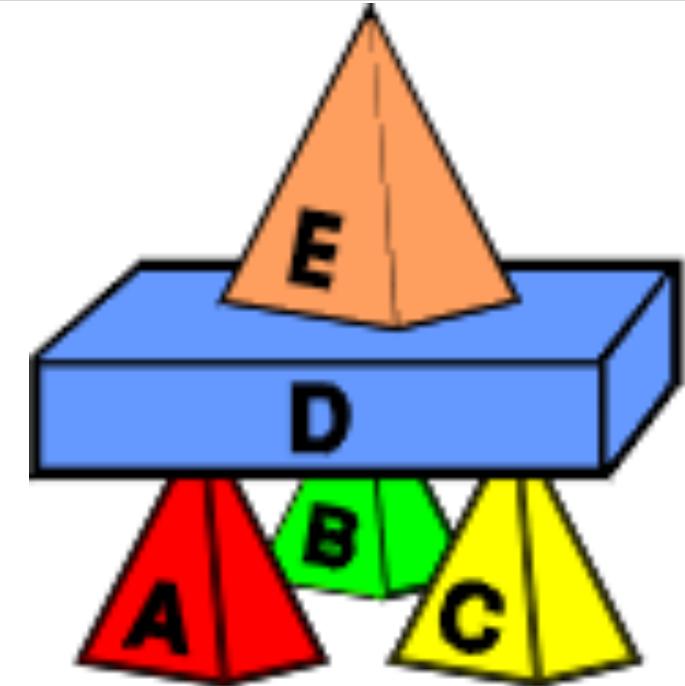
Pyramides

Existe-t-il une pyramide qui est soutenue par un bloc?

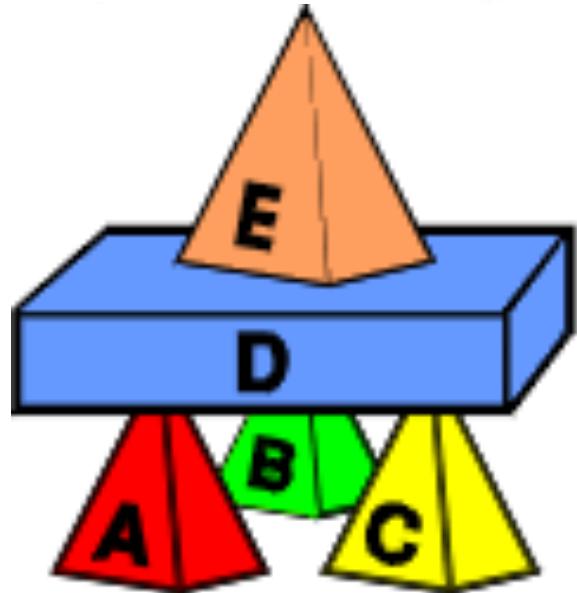


Pyramides: représentation logique

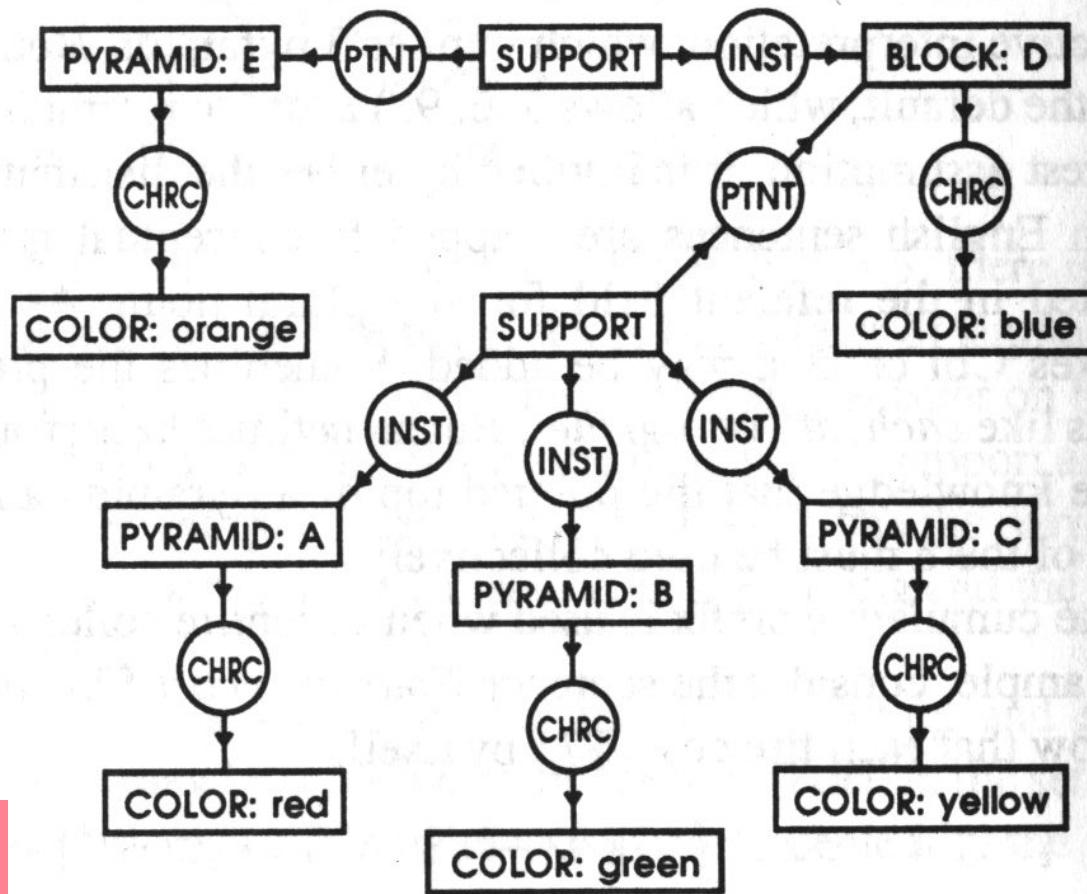
pyramide(E) \wedge bloc(D) \wedge
pyramide(A) \wedge pyramide(B) \wedge
pyramide(C) \wedge oranger(E) \wedge
bleu(D) \wedge rouge(A) \wedge vert(B) \wedge
vert(C) \wedge supporte(A, D) \wedge
supporte(B, D) \wedge supporte(C, D) \wedge
supporte(D, E)



Représentation & raisonnement sémantiques



PTNT → Patient
INST → Instrument
CHRC → Caractéristique



Raisonnement: appariement de sous-graphe

P

6

C

N

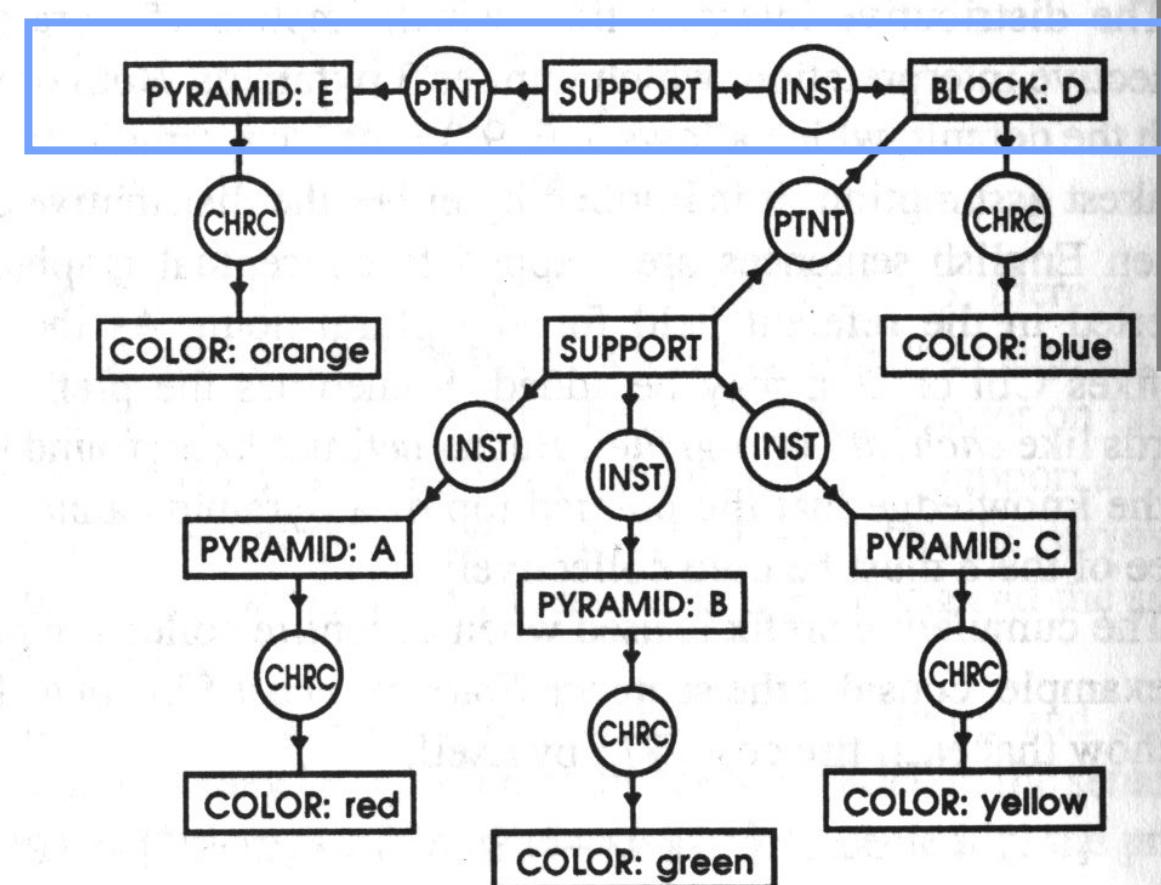
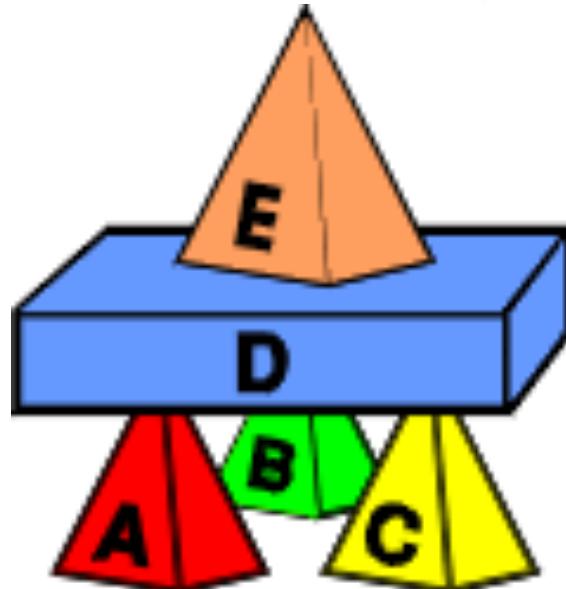
R

S



Exemples1: Quelle pyramide est soutenue par un bloc?

[PYRAMID: ?] ← (PTNT) ← [SUPPORT] → (INST) → [BLOCK]



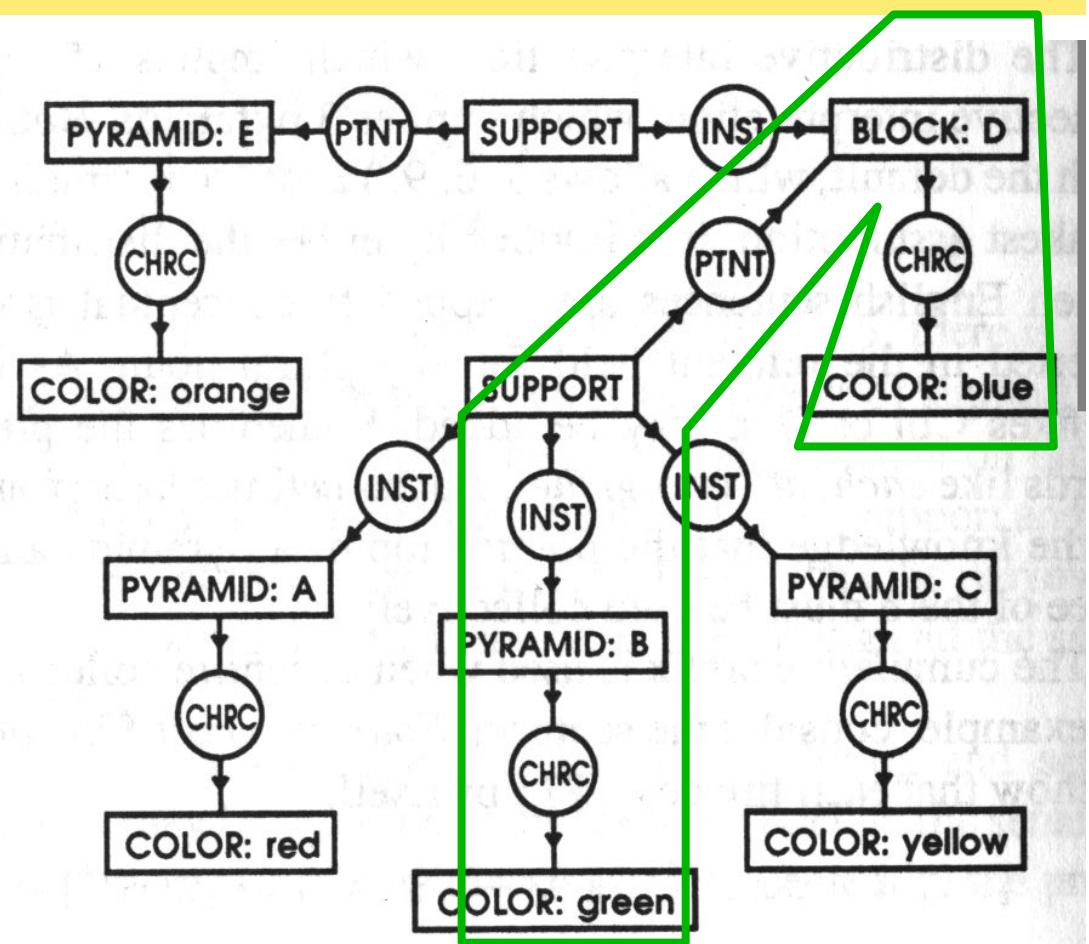
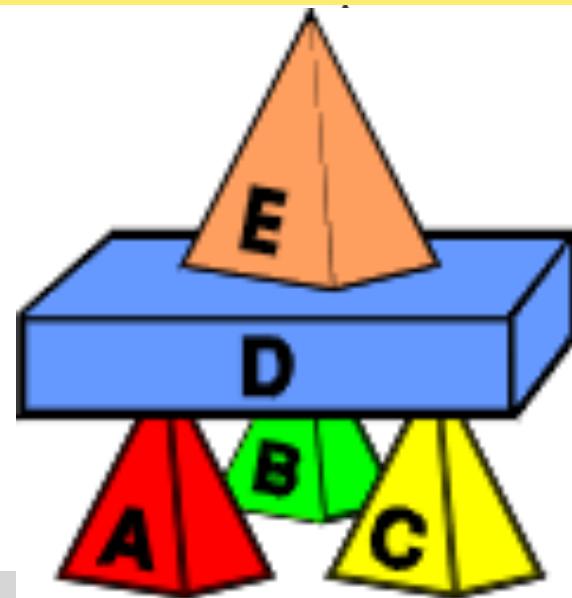
Raisonnement: appariement de sous-graphe

Exemples2: Quel *objet* coloré est soutenu par quelque chose de vert?

[COLOR: ?] ← (CHRC) ← [OBJECT] ← (PTNT) ← [SUPPORT] →
 (INST) ← [ENTITY] ← (CHRC) ← [COLOR:green] →

Utilisation du treillis des types:

- Un **bloc** est un **objet**
 - Une **pyramide** est une **entité**



Raisonnement: référents complexes

Exemples3: Quel *bloc* est soutenu par trois pyramides?

[BLOCK: ?] \leftarrow (PTNT) \leftarrow [SUPPORT] \rightarrow (INST) \rightarrow [PYRAMID: {*}@3]

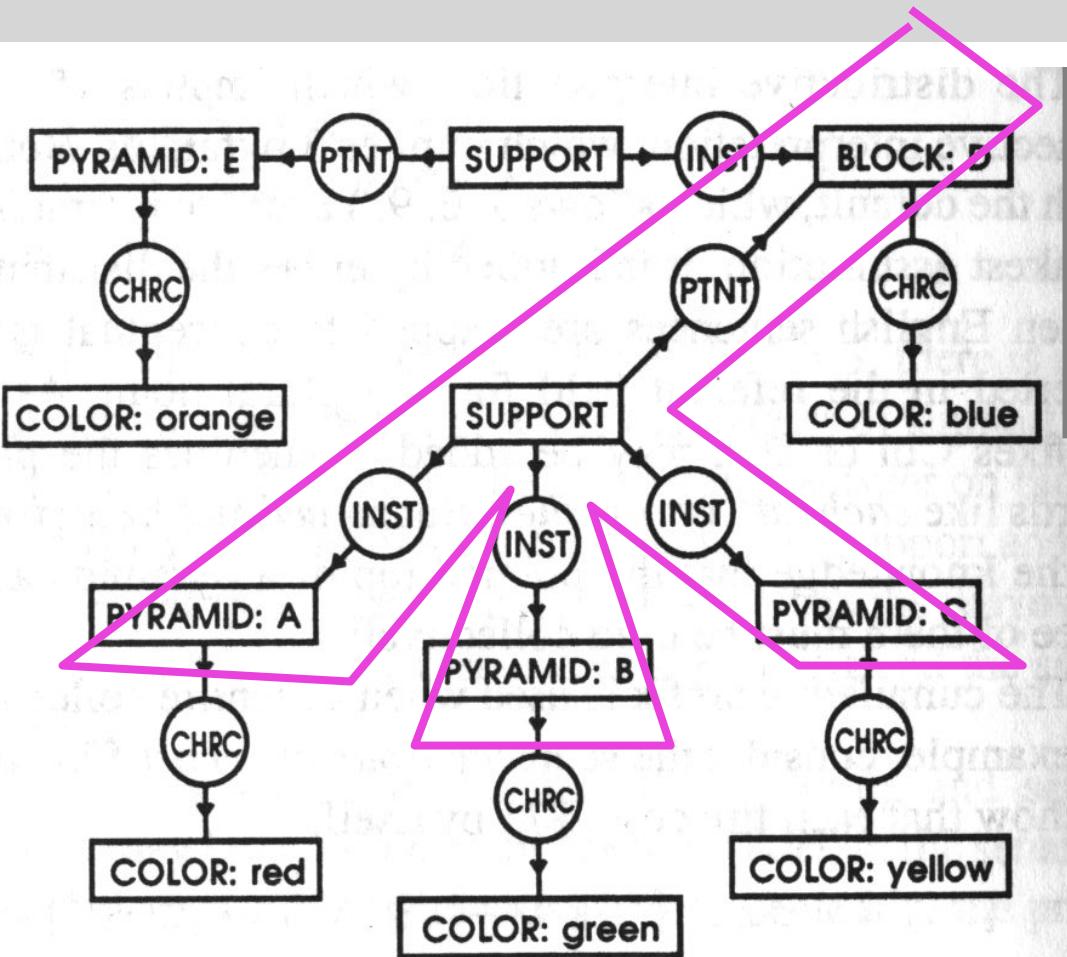
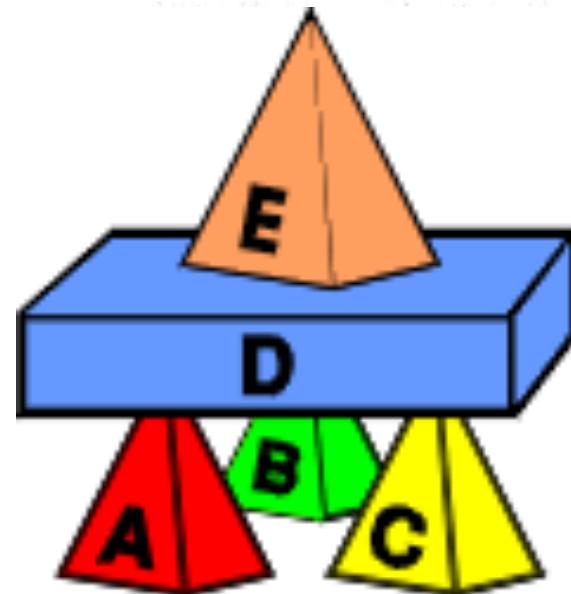
O

C

N

R

S



AC

Opérations sur les graphes

- **Jointure maximale:** construire le graphe qui contient toutes l'information de deux graphes, sans dupliquer d'information
- **Généralisation:** prend deux graphes et construit un graphe qui contient les informations communes aux deux graphes
- **Subsomption:** test si un graphe est plus général qu'un autre
- **Sous-graphe:** détermine si un graphe est une branche d'un autre



Jointure maximale et généralisation

```
g1 = [Drive] -  
      -obj->[Car],  
      -agnt->[Person],  
g2 = [Drive] -  
      -manr->[Fast],  
      -agnt->[Boy : John],
```

Jointure maximale de g1 et g2

```
g3 = [Drive] -  
      -agnt->[Boy : John],  
      -obj->[Car],  
      -manr->[Fast]
```

Généralisation de g1 et g2

```
g4 = [Drive]-agnt->[Person]
```



Sous-graphe et subsomption

```
g = [Drive] --agnt->x,  
h = [Drive] -  
      -manr->[Fast],  
      -agnt->[Boy : John],
```

Sous-graphe g h

Subsomption g h



Postérité des graphes conceptuels

- Couplage SQL (stockage et requête)
- Couplage PROLOG (raisonnement)
PROLOG+CG*
- Complétude des graphes conceptuel:
 - On démontre que tout ce qui est représentable en logique des prédictats du premier ordre l'est dans les graphes conceptuels
- Indexation base multimédia
- Notion de schémas RDF en XML
- Autres formalismes de représentation:
 - Logique terminologique*
 - Logique de description
 - ...

