

Systemes Intelligents

CHAPITRE II



Les Ontologies (TD)

Octobre 2021



TD N°2

Exercice 1

Conceptualisation d'une ontologie

Méthode METHONTOLOGY

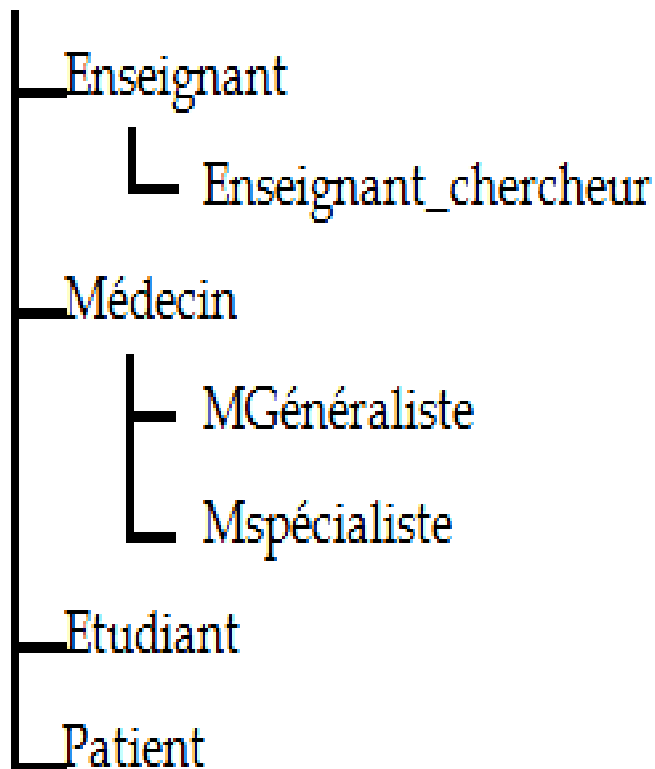
1. Construire un glossaire de **termes**.
2. Construire les arbres de **classification** des **concepts** (les hiérarchies).
3. Tracer le diagramme des **relations binaires** entre concepts.
4. Pour chaque arbre de classification construire :
 1. une table de concepts
 2. une table de relations binaires
 3. une table des **attributs**
 4. une table des **axiomes**
 5. une table des **instances**
 6. une table des **assertions**

1. Glossaire de termes candidats

Terme	Description (informelle)
Personne	Un être humain.
Habite	Une personne habite une ville
Enseignant	Une personne exerçant le métier d'enseignement à l'université
Est Chargé	Un enseignant est chargé d'un module ou plusieurs
Médecin	Une personne qui est docteur en médecine travaillant à l'hôpital : spécialiste ou généraliste
Soigne	Un médecin soigne un ou plusieurs malades
Etudiant	Personne engagé dans un cursus d'enseignement supérieur.
Patient	Une personne qui a été hospitalisée.
Ville	Un milieu urbain où se concentre une forte population humaine (plus de 20 000 habitants) et localisée dans un Pays.
Module	Matière d'enseignement.
Publication	une publication peut être un livre ou un article de recherche.
Ecrit	Une personne peut écrire un livre et les enseignants-chercheurs peuvent écrire des articles de recherche.
....	...

2. Arbres de **classification** des concepts

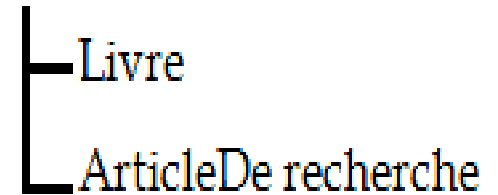
H1. Personne



H2. Lieu de travail



H3. Publication

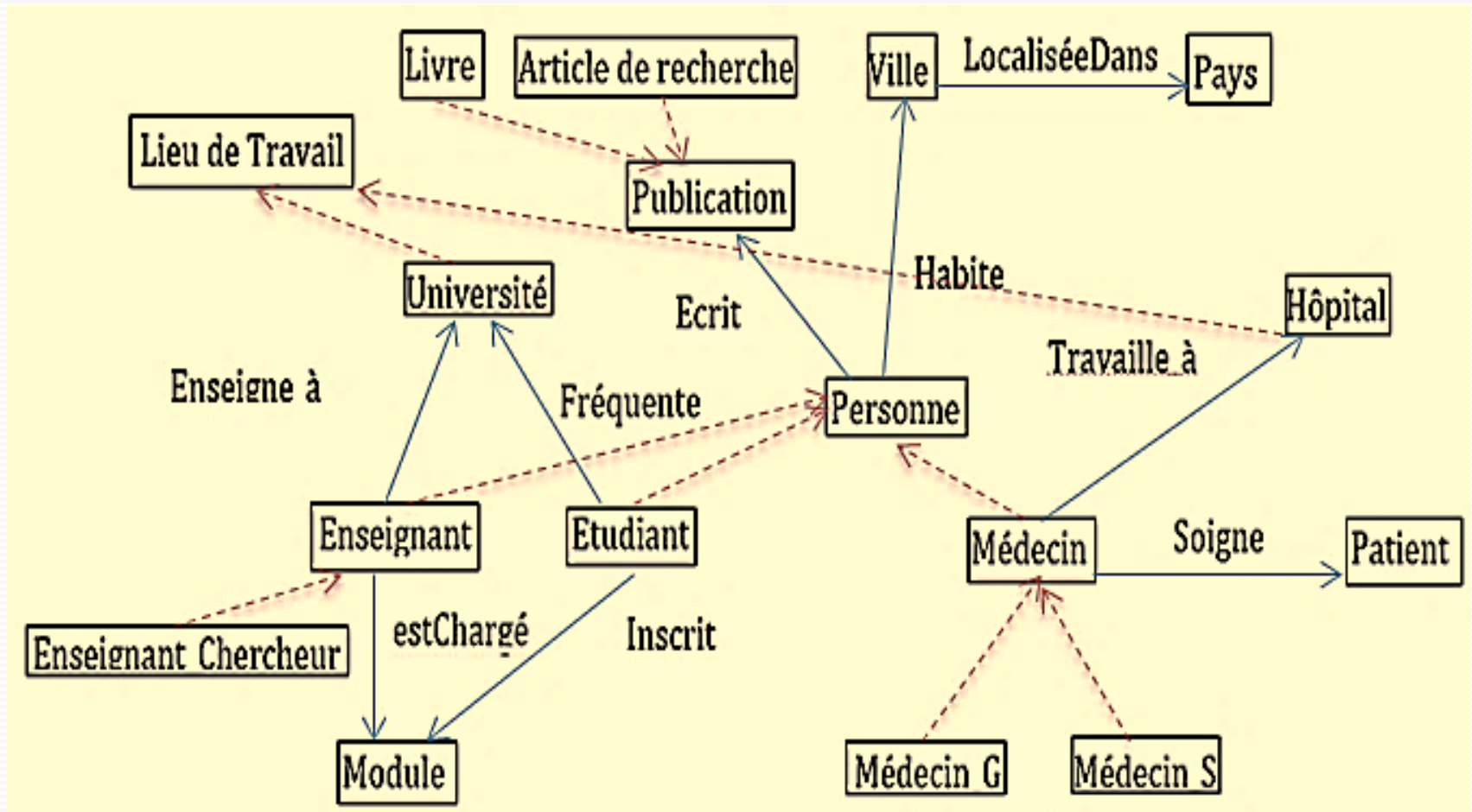


H4. Module

H5. Ville

H6. Pays

3. Diagramme des relations binaires entre concepts



3. Table de concepts

Concept	Attribut	Synonyme
Personne	Nom de famille, Prénom Date de naissance	Humain
Module	Intitulé	Matière d'enseignement
Enseignant	Grade	
Médecin		Docteur en Médecine
Médecin_S	Spécialité	
Lieu de Travail	Nom, Secteur	
Etudiant	Numéro d'inscription	
Patient	Date d'hospitalisation	Malade
Ville	Nom, Surface	
Pays	Nom, Code_ISO	
Publication	Titre Nombre page	
...		

4. Table de relations binaires

Relation	Concept source	Concept cible	Card Cible	Card Source	Relation inverse
Habite	Personne	Ville	1..1	200000..n	Habité_par
LocaliséeDans	Ville	Pays	1..1	1..n	Contient
Ecrit	Personne	Publication	0..n	1..n	Ecrit_Par
estChargé	Enseignant	Module	1..3	1..1	Enseigné_par
Enseigne_à	Enseignant	Université	1..1	1..n	aPourEnseignant
Travaille_à	Médecin	Hôpital	1..1	1..n	aPourMédecin
Soigne	Médecin	Patient	1..n	1..n	SoignéPar
Fréquente	Etudiant	Université	1..1	1..n	Fréquenté_par
Inscrit	Etudiant	module	1..n	1..n	Etudié_par
...

5. Table des attributs

Attribut	Type	Cardinalité	Domaine de valeurs
Nom de famille	String	1..1	
Prénom	String	1..2	
date de naissance	Date	1..1	
Intitulé	String	1..1	
Grade	String	1..1	MAB ,MAA, MCA,MCB,Prof
Nom	String	1..1	
Secteur	String	1..1	Santé, EnseignementSup
Numéro d'inscription	String	1..1	
Date d'hospitalisation	Date	1..n	
Surface	Integer	1..1	
Code	String	1..1	
Titre	String	1..1	
Nombre page	Integer	1..1	

6. Liste des axiomes

Les axiomes expriment la **sémantique du domaine** : ils déterminent comment les relations/concepts peuvent être utilisés d'une manière opérationnelle (aller au-delà de la terminologie):

- ✓ La propriété de subsomption entre concepts et/ou relation
- ✓ Les propriétés de cardinalité d'une relation
- ✓ L'incompatibilité entre concepts (i.e. la disjonction)
- ✓ Propriétés algébrique d'une relation (symétrie, réflexivité, transitivité)
- ✓ Etc.

Exemples d'axiomes dans notre ontologie

Axiome en langage naturelle

Une personne est soit un étudiant, un enseignant un médecin ou un patient

Une Ville ne peut être un pays

Un lieu de travail est soit un hôpital ou une université

Une publication est soit un livre ou un article de recherche

Un livre est écrit par **au moins** une personne

Un article de recherche est écrit par **au moins** un enseignant-chercheur

Un module est enseigné par **un et un seul** enseignant

Un étudiant est inscrit dans au moins un module et fréquente une université

Une personne habite **une et une seule** ville

Une ville est localisée dans **un seul** pays

....etc.

7. Table des Instances

Concept	Nom instance	Attribut	valeur
Patient	P_1	Nom de famille	f
		Prénom	Lila
		date de naissance	11-07-2005
		date hospitalisation	11-10-2015
MedecinG	MG_1	Nom de famille	ff
		Prénom	Ali
		date de naissance	15-05-1960
Hôpital	H_1	Nom	<i>MdSedikBenYahia</i>
		Secteur	Santé
Ville	V_1	Nom	Jijel
		Surface	2577 Km2
Pays	Pa_1	Nom	Algérie
		CodeISO	DZ

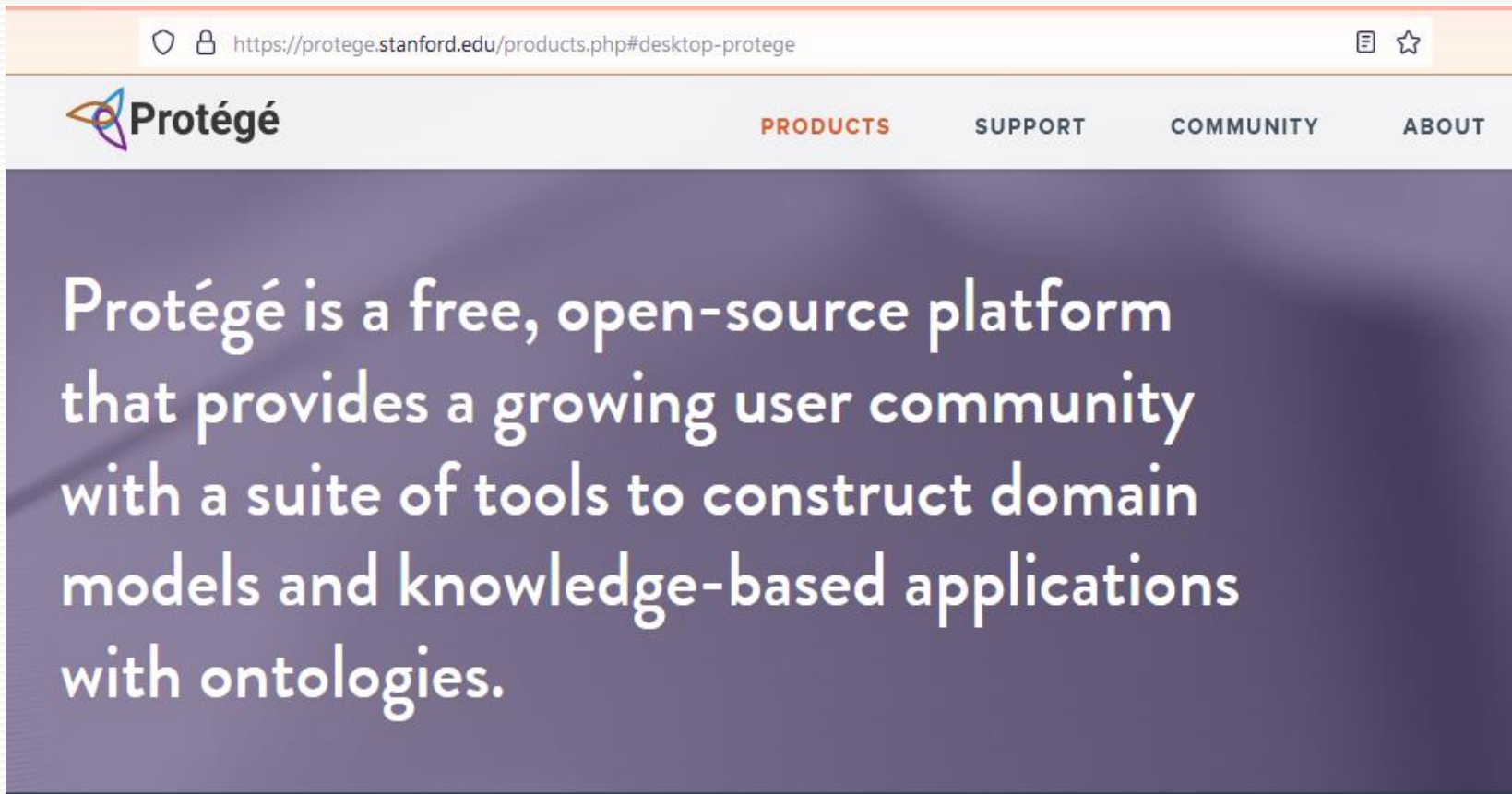
8. Table des assertions

Nom de relation	Instance source	Instance Cible
Soigne	MG_1	P_1
Habite	MG_1	V_1
	P_1	V_1
TravailleA	MG_1	H_1
LocaliséDans	V_1	Pa_1

TD N°2

Exercice 2/3

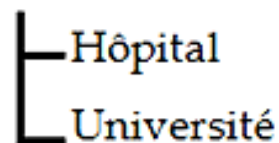
Formalisation/Opérationnalisation d'une ontologie



The image is a screenshot of the Protégé website. The browser's address bar shows the URL <https://protege.stanford.edu/products.php#desktop-protege>. The website's header features the Protégé logo on the left and a navigation menu with the items 'PRODUCTS', 'SUPPORT', 'COMMUNITY', and 'ABOUT' on the right. The main content area has a dark purple background with white text that reads: 'Protégé is a free, open-source platform that provides a growing user community with a suite of tools to construct domain models and knowledge-based applications with ontologies.'

Exemples sur la Subsumption

H2. Lieu de travail



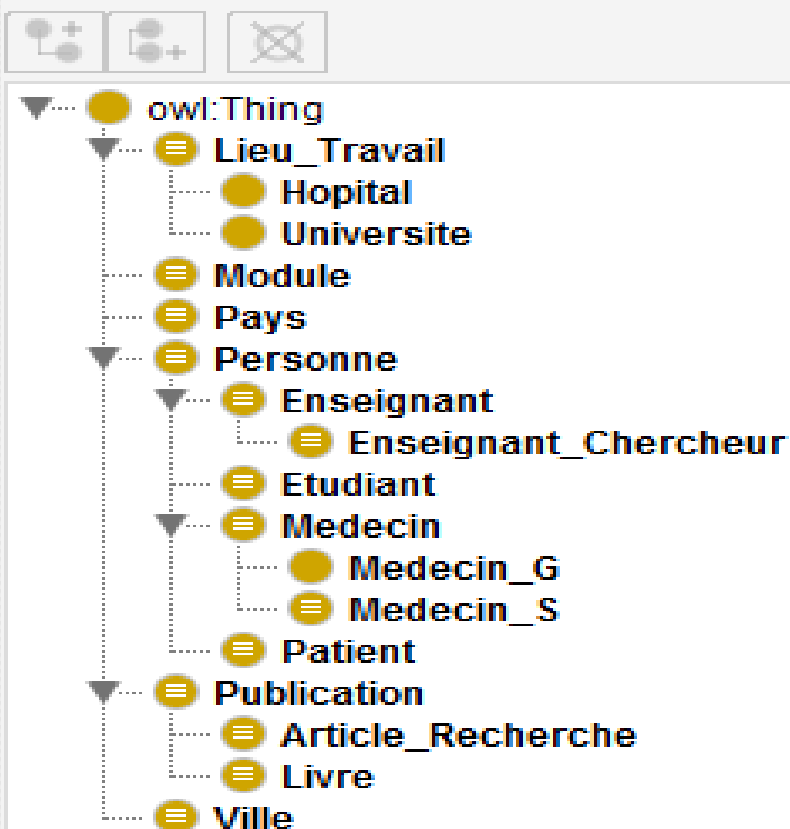
En LD



$H\hat{o}pital \sqsubseteq Lieu_Travail$

$Universit\acute{e} \sqsubseteq LieuTravail$

Class hierarchy:



Description: Hopital

Equivalent To +

SubClass Of +

Lieu_Travail

Exemples de définitions de concepts

$Personne \equiv (Enseignant \cup Etudiant \cup M\acute{e}decin \cup Patient) \sqcap \geq 0 \text{ Ecrit.Livre} \sqcap$
 $= 1 \text{ Habite.Ville} \sqcap = 1 \text{ Nom.String} \sqcap = 1 \text{ dateNaissance.Date} \sqcap$
 $\geq 1 \text{ Pr\acute{e}nom.String} \sqcap \leq 2 \text{ Pr\acute{e}nom.String}$

Les attributs sont traités comme des rôles

Description: Personne

Equivalent To 

- (Enseignant or Etudiant or Medecin or Patient)
and (Ecrit min 0 Livre)
and (Habite exactly 1 Ville)
and (Prenom min 1 xsd:string)
and (Date_Naissance exactly 1 xsd:dateTime)
and (Nom_F exactly 1 xsd:string)
and (Prenom max 2 xsd:string)

Exemples de définitions de concepts

$Module \equiv T \sqcap \exists \text{ EtudiéPar} . \text{Etudiant} \sqcap = 1 \text{ EnseignéPar} . \text{Enseignant}$
 $\sqcap = 1 \text{ Intitulé} . \text{String}$

Description: Module

Equivalent To 

 **owl:Thing**

and (Etudié_Par **min** 1 Etudiant)


and (Enseigne_Par **exactly** 1 Enseignant)


and (Intitulé **exactly** 1 xsd:string)

Exemple sur la Disjonction

$H\hat{o}pital \sqcap Universit\acute{e} \sqsubseteq \perp$


Description: Hopital

SubClass Of 


 Lieu_Travail

General class axioms 


SubClass Of (Anonymous Ancestor)

 (Hopital or Universite)
and (Nom exactly 1 xsd:string)
and (Secteur exactly 1 xsd:string)

Instances 

Target for Key 

Disjoint With 

 Universite

Exemple sur la définition du domaine des valeurs d'un attribut

$$Grade \equiv \{MAB\} \cup \{MAA\} \cup \{MCB\} \cup \{MCA\} \cup \{Prof\}$$

Description: Grade

Equivalent To 

SubProperty Of 

Domains (intersection) 

 Enseignant

Ranges 

 {"MAA", "MAB", "MCA", "MCB", "Prof"}

Exemples sur la définitions des rôles

$Ecrit \equiv (EcritPar)^-$

$Enseigne \equiv (EnseignéPar)^-$

Description: Ecrit

Equivalent To 

SubProperty Of 













Inverse Of 
 **Ecrit_Par**

Domains (intersection) 
 **Personne**

Ranges (intersection) 
 **Publication**

Exemples sur la définitions des rôles

$$aPourColl\grave{e}gue \equiv (aPourColl\grave{e}gue)^- \\ Tr(aPourColl\grave{e}gue)$$

Characteristic ?     	Description: aPourColleague
<input type="checkbox"/> Functional	Equivalent To 
<input type="checkbox"/> Inverse functional	SubProperty Of 
<input checked="" type="checkbox"/> Transitive	Inverse Of 
<input checked="" type="checkbox"/> Symmetric	Domains (intersection) 
<input type="checkbox"/> Asymmetric	 Personne
<input type="checkbox"/> Reflexive	Ranges (intersection) 
<input type="checkbox"/> Irreflexive	 Personne

Exemples sur la définitions des instances / assertions

MédecinG(MG_1)

Nom_F (MG_1, FF)

Soigne(MG_1, P_1)

Prenom (MG_1, Ali)

Habite (MG_1, V_1)

Date_Naissance (MG_1, 15-05-1960)

Travaille (MG_1, H_1)

Individuals: MG1



Annotations Usage

Annotations: MG1

Annotations +

Description: MG1



Property assertions: MG1

Object property assertions +

Travaille_a H1

Habite V1


Soigne P1

Data property assertions +

Nom_F "ff"^^xsd:string

Date_Naissance "15051960"^^xsd:dateTime

Prenom "Ali"^^xsd:string



Quelques exemples sur l'inférence avec Protégé.

Moteur d'inférence (Reasoner)

❑ Dans Protégé:

HermiT ou FaCT++

❑ Juste : Start Reasoner ...

```
Initializing the reasoner by performing the following steps:
```

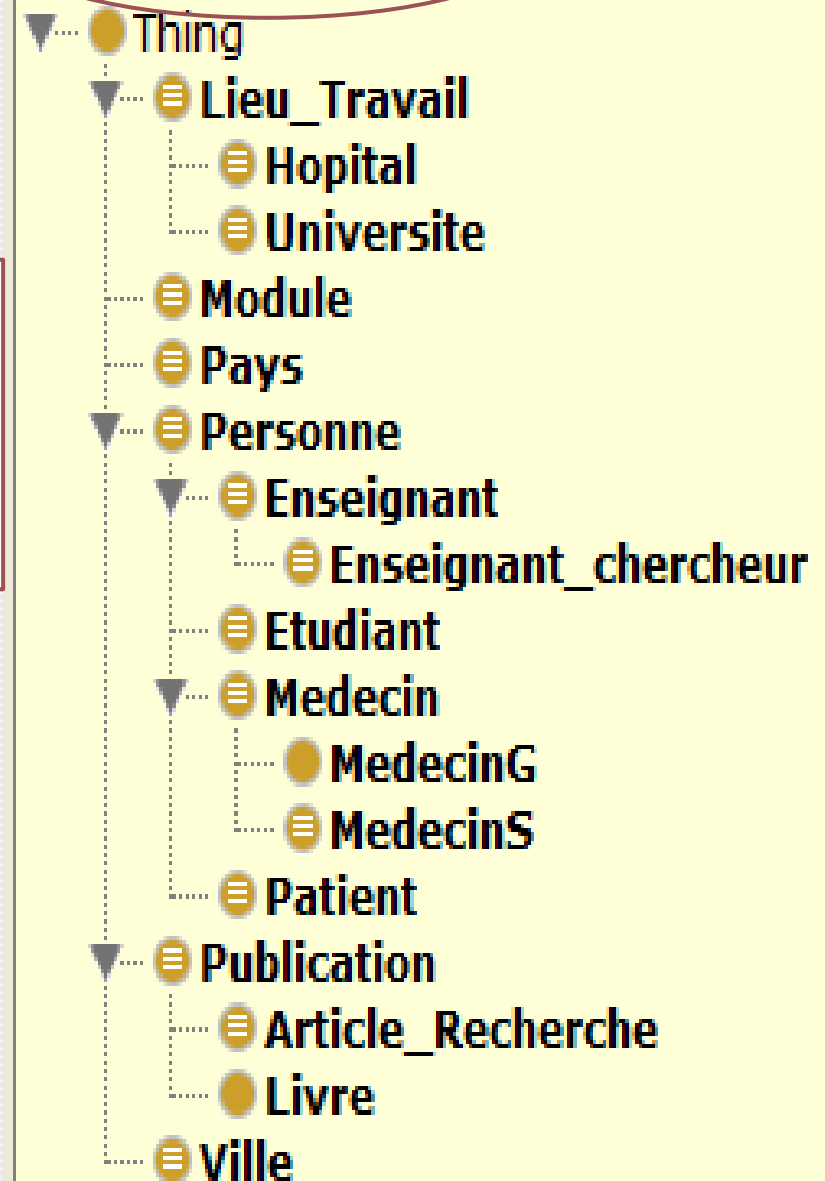
```
class hierarchy
object property hierarchy
data property hierarchy
class assertions
object property assertions
same individuals
```

```
HermiT 1.3.8 classified in 156ms
```


L'hiérarchie inférée est identique
à notre hiérarchie

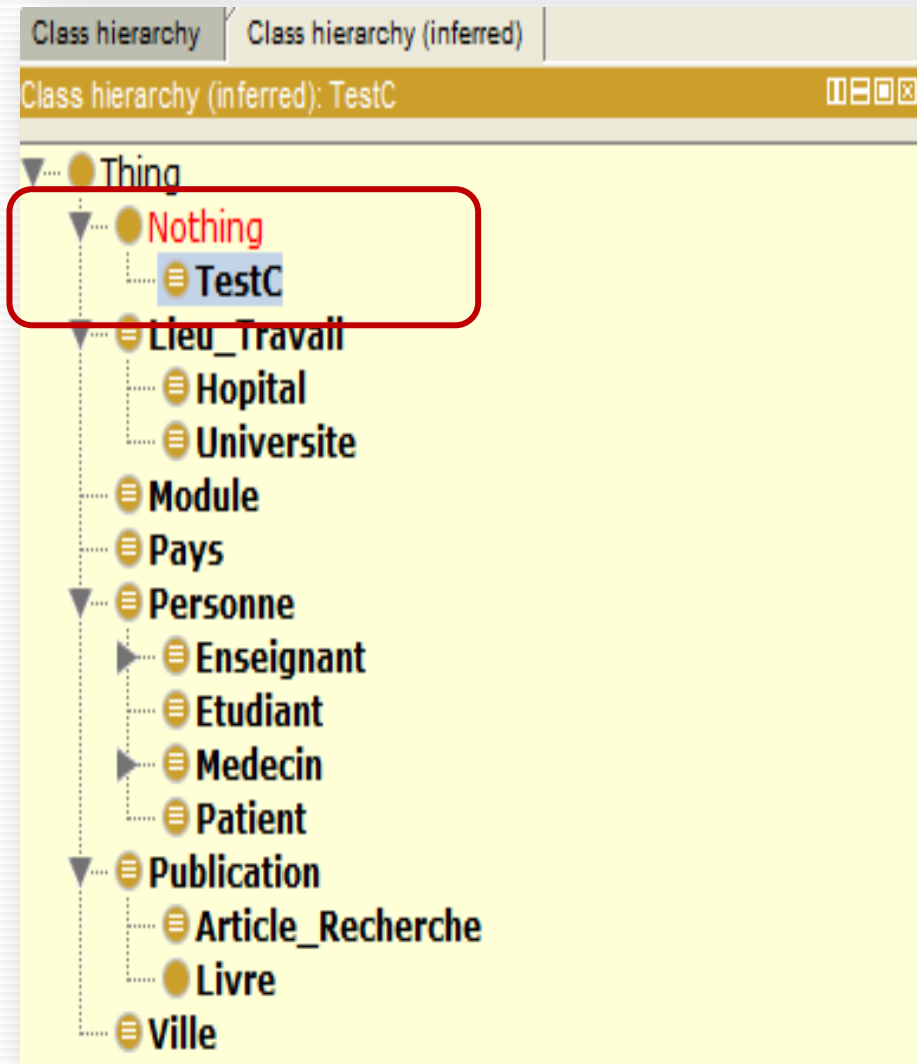
(Test de consistance)

Class hierarchy (inferred):



Ajoutant à l'ontologie le concept :

$$\textit{TestC} = \textit{Ville} \cap \textit{Pays}$$



Inférence concernant les individus

Description: Thing

Equivalent To +

SubClass Of +

SubClass Of (Anonymous Ancestor)

Members +

◆ H_1

◆ MG_1

◆ P_1

◆ Pa_1

◆ V_1

Inférence concernant les assertions

Property assertions: V_1

Object property assertions +

- LocaliseDans Pa_1
 - HabitePar P_1
 - HabitePar MG_1

Icons: ? @ X O

Icons for HabitePar: ? @

Ajoutant à l'ontologie l'assertion

« Publi_1 EcritPar H_1 »

The screenshot shows two panels of an ontology editor. The left panel, titled 'Description: Publi_1', shows a class 'Article_Recherche' with options to add types, same individuals, or different individuals. The right panel, titled 'Property assertions: Publi_1', shows two assertions: 'EcritPar H_1' under 'Object property assertions' and 'Titre "Architecture_des_ordinateurs"^^str' and 'Nombre_Page "300"^^positiveInteger' under 'Data property assertions'. Each assertion has control icons (help, edit, delete, add).

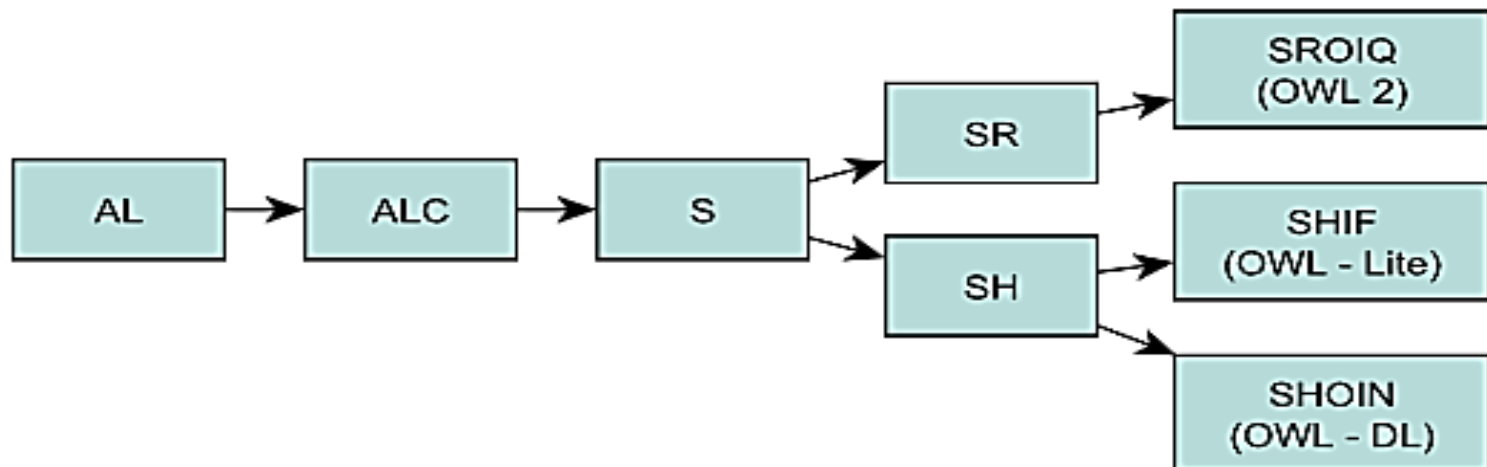
The dialog box is titled 'Inconsistent ontology explanation'. It has four radio buttons for justification types: 'Show regular justifications' (selected), 'All justifications', 'Show laconic justifications', and 'Limit justifications to' (with a value of 2). There is a checkbox for 'Display laconic explanation'. Below, under 'Explanation 1', is a list of six items explaining the inconsistency: 'Publi_1 EcritPar H_1', 'Lieu_Travail DisjointWith Personne', 'MG_1 Travaille H_1', 'Travaille Range Hopital', 'Hopital SubClassOf Lieu_Travail', and 'EcritPar Range Personne'. The first item is highlighted in green.

Quelques exemples du code OWL



OWL (Ontology Web Language)

- OWL un des langages du web sémantique développé par le W3C (World Wide Web Consortium)
- OWL standardisé pour la première fois en 2004, puis révisé en 2012 et nommé OWL 2.
- Divers Syntaxes (RDF/XML , OWL/XML, Turtle, etc.)
- La sémantique de OWL correspond à celle de la logique LD **SHOIN(D)**.
- La sémantique de OWL2 correspond à celle de la logique LD **SROIQ(D)**.



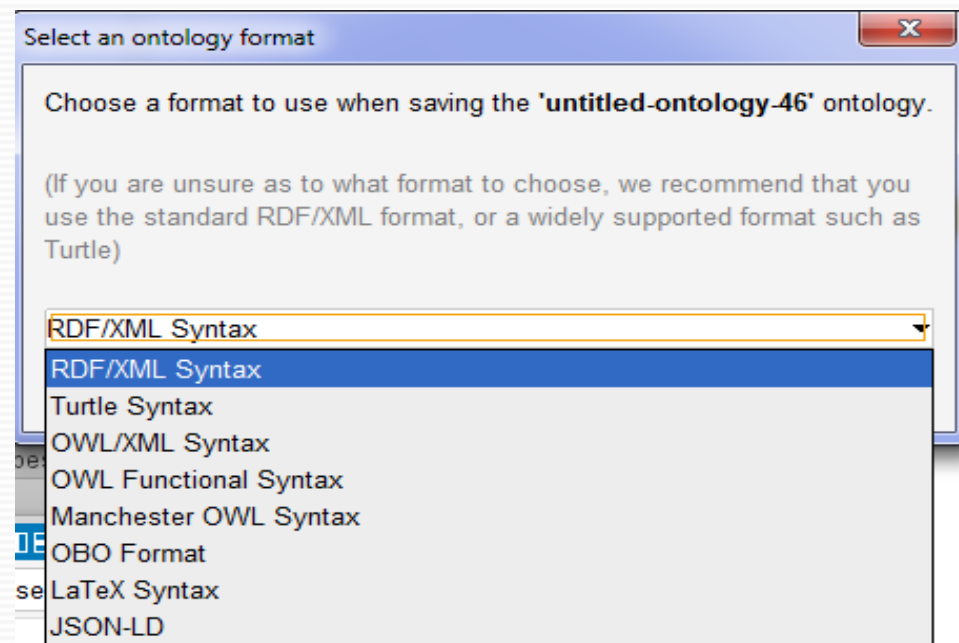
Logique SROIQ

- S = ALC + Rôle transitive
- O , I , Q : énumération, rôle inverse, restriction de nombre qualifiée
- R: ensemble d'axiomes sur les rôles (RBOX)
- BC = TBOX + ABOX + RBOX

Axiom	Syntax	Semantics
Complex role		
Inclusion (CRIA)	$R_1 \circ \dots \circ R_n \sqsubseteq S$	$R_1^{\mathcal{I}} \circ \dots \circ R_n^{\mathcal{I}} \subseteq S^{\mathcal{I}}$
Disjointness	$\text{Disj}(R, S)$	$R^{\mathcal{I}} \cap S^{\mathcal{I}} = \emptyset$
Transitivity	$\text{Trans}(R)$	$R^{\mathcal{I}} \circ R^{\mathcal{I}} \subseteq R^{\mathcal{I}}$
Reflexivity	$\text{Ref}(R)$	$\{(x, x) \mid x \in \Delta^{\mathcal{I}}\} \subseteq R^{\mathcal{I}}$
Irreflexivity	$\text{Irref}(R)$	$\{(x, x) \mid x \in \Delta^{\mathcal{I}}\} \cap R^{\mathcal{I}} = \emptyset$
Symmetry	$\text{Sym}(R)$	$(x, y) \in R^{\mathcal{I}} \Rightarrow (y, x) \in R^{\mathcal{I}}$
Antisymmetry	$\text{Asym}(R)$	$(x, y) \in R^{\mathcal{I}} \Rightarrow (y, x) \notin R^{\mathcal{I}}$

- **CRIA** (exemple: *aPourParent* \circ *aPourFrère* \sqsubseteq *aPourOncle*)
- Rôle universel U (ou Top tel que $U^{\mathcal{I}} = \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$)
- Des concepts du genre $\exists R.$ *Self* (exemple: \exists *aime.Self*)
- Négation des assertions sur les rôles (exemple: (*Ali, physique*): \neg *Enseigne*)

File >> Save As >> RDF/XML



Un document OWL est **un document RDF** avec l'élément racine `rdf:RDF` :

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns="http://www.semanticweb.org/toshiba/ontologies/2016/3/ex
  emple"

  xml:base="http://www.semanticweb.org/toshiba/ontologies/2016/3
  /exemple"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"

  xmlns:exemple="http://www.semanticweb.org/toshiba/ontologies/2
  016/3/exemple#">
  <owl:Ontology rdf:about="urn:absolute:monOnto"/>
```

Exemple1: attribut Code_ISO

```
<owl:DatatypeProperty rdf:about="Code_ISO">  
<rdfs:domain rdf:resource="Pays"/>  
<rdfs:range rdf:resource="&xsd:string"/>  
</owl:DatatypeProperty>
```

Exemple2: rang de l'attribut Secteur

```
<rdfs:range>  
  <rdfs:Datatype>  
    <owl:oneOf>  
      <rdf:Description>  
        <rdf:type rdf:resource="&rdf;List"/>  
        <rdf:first>Ens_Sup</rdf:first>  
        <rdf:rest>  
          <rdf:Description>  
            <rdf:type rdf:resource="&rdf;List"/>  
            <rdf:first>Sante</rdf:first>
```

Exemple 3: Relation a Pour collègue

```
<owl:ObjectProperty rdf:about="aPourCollegue">  
<rdfs:domain rdf:resource="Personne"/>  
<rdfs:range rdf:resource="Personne"/>  
<rdf:type rdf:resource="&owl;TransitiveProperty" />  
<rdf:type rdf:resource="&owl;SymmetricProperty" />  
</owl:ObjectProperty>
```

Exemple 4: Concept MedecinG

```
<owl:Class rdf:about="MedecinG">  
<rdfs:subClassOf rdf:resource="Medecin"/>  
<owl:disjointWith rdf:resource="Medecin_S"/>  
</owl:Class>
```

Exemple 5 : estCharge

```
<owl:ObjectProperty rdf:about="estCharge">  
<rdfs:domain rdf:resource="Enseignant"/>  
<rdfs:range rdf:resource="Module"/>  
<inverseOf rdf:resource="EnseignePar"/>  
</owl:ObjectProperty>
```

Exemple 6: Concept Etudiant

```
<owl:Class rdf:about="Etudiant">  
<owl:equivalentClass>  
<owl:Class><owl:intersectionOf rdf:parseType="Collection">  
<rdf:Description rdf:about="Personne"/>  
<owl:Restriction>  
<owl:onProperty rdf:resource="Inscrit"/>  
<owl:onClass rdf:resource="Module"/>  
<owl:minQualifiedCardinality rdf:datatype="&xsd;nonNegativeInteger">  
1  
</owl:minQualifiedCardinality></owl:Restriction>
```

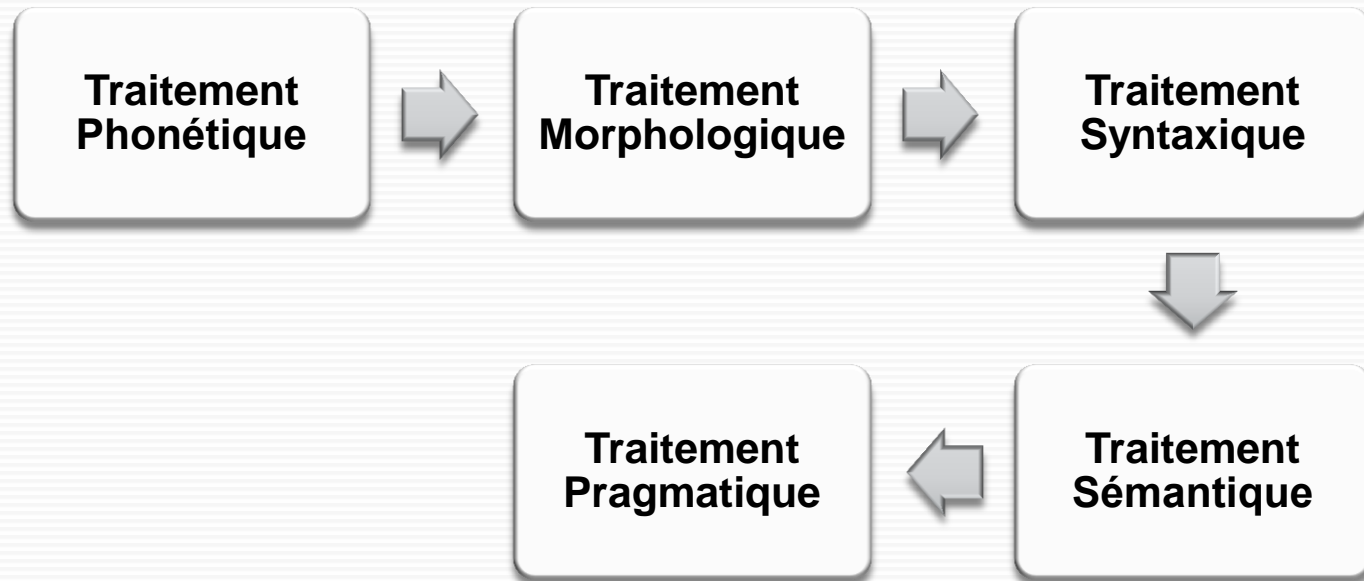
Exemple 7: Individu Pa_1

```
<owl:NamedIndividual rdf:about="Pa_1">  
  <Nom rdf:datatype="&xsd:string"> Algerie</Nom>  
  <Code_ISO rdf:datatype="&xsd:string"> DZ</Code_ISO>  
</owl:NamedIndividual>
```



Ontologies Linguistiques et TALN

Processus général du TALN



Sémantique lexicale

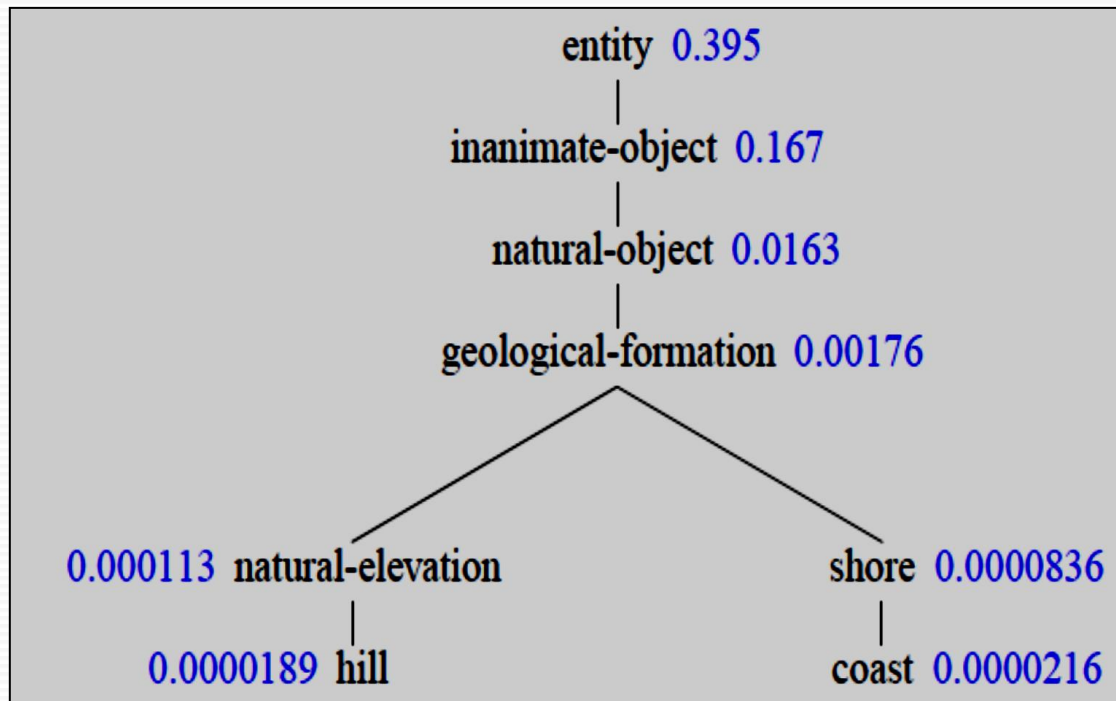
- ❑ La sémantique lexicale est l'étude du **sens** des **morphèmes** d'une langue. Un morphème est l'unité linguistique minimale ayant un sens (un morphème n'est donc pas forcément un mot).
- ❑ Le calcul de **similarité des mots** est utile pour de nombreuses applications du TALN comme : la recherche d'information, les systèmes questions-réponses, le résumé automatique, la traduction automatique, etc.
- ❑ Dans les métriques basées sur **thésaurus**, la distance entre deux mots est déterminée à partir d'une **ontologie linguistique** comme WordNet généralement en exploitant la **relation de subsomption** (est-un).

Similarité basée sur la longueur de chemin

- ❑ Idée: Plus le chemin entre deux mots dans le graphe défini par la hiérarchie du thésaurus est court, plus ils sont similaires.
- ❑ la similarité est inversement proportionnelle à la longueur du chemin.

$$sim_{chemin}(m_1, m_2) = -\log L(m_1, m_2)$$

- ❑ Calculer la similarité entre les mots « *hill* » et « *coast* »



Similarité de Resnik

- ❑ Idée: plus deux mots ont de contenu sémantique en commun, plus ils sont similaires
- ❑ Le **contenu sémantique** d'un mot est défini comme le logarithme négatif de sa **probabilité d'apparition** dans un **corpus**. (N est le nombre total des mots dans le corpus, mots(M) est l'ensemble des mots subsumés par M) :

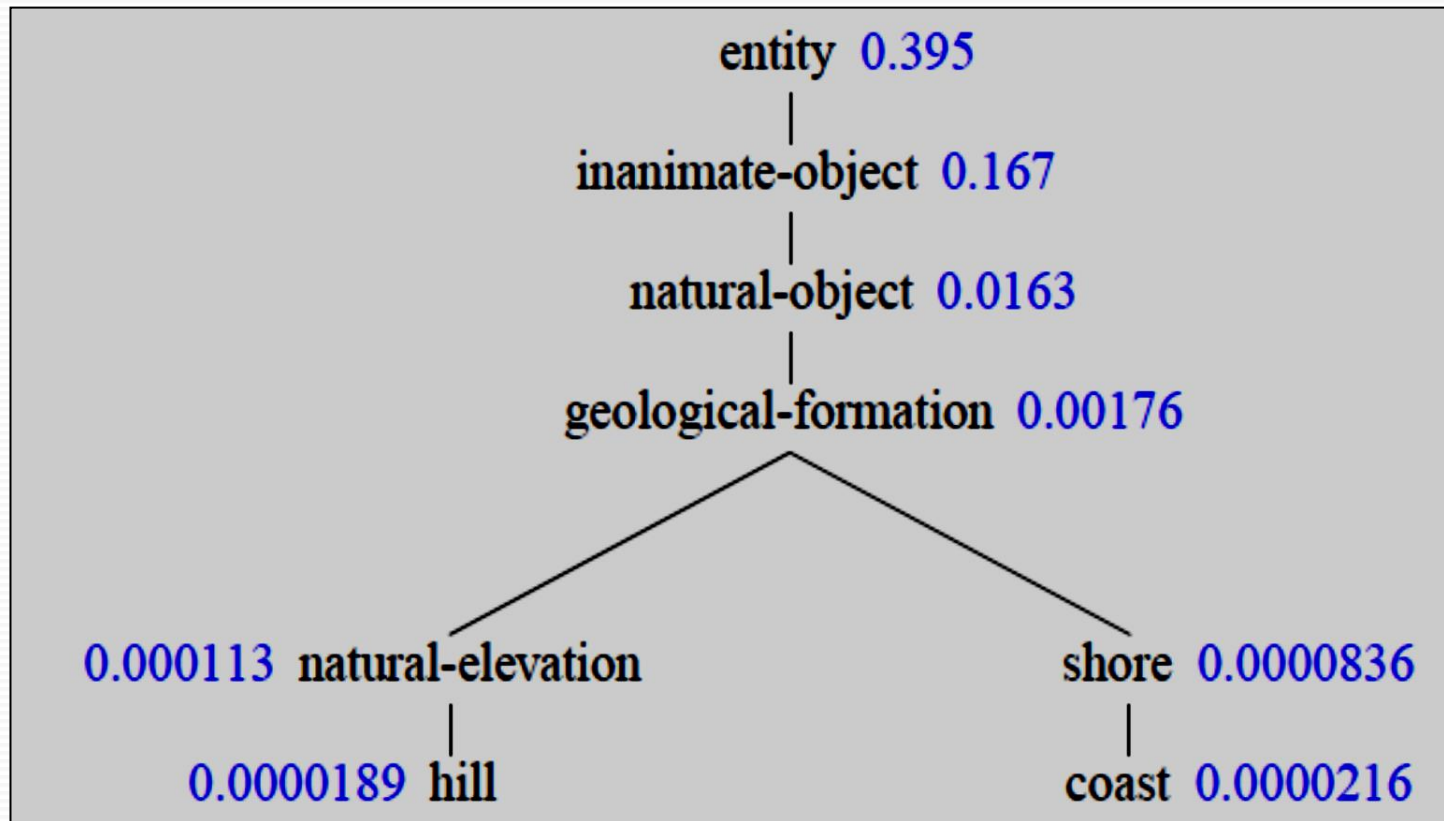
$$P(M) = \begin{cases} \frac{1}{N} \sum_{m \in \text{mots}(M)} \text{fréquence}(m) & \text{si } M = \text{Racine} \\ & \text{Sinon} \end{cases}$$

- ❑ La similarité entre deux mots est égale au contenu sémantique de leur LCS (LCS : Lowest Common Subsumer)

$$\text{sim}_{\text{Resnik}}(m_1, m_2) = -\log P(\text{LCS}(m_1, m_2))$$

- Calculer la similarité entre les mots « *hill* » et « *coast* »

$$sim_{Resnik}(m_1, m_2) = -\log P(LCS(m_1, m_2))$$



Exercice

Imaginons un moteur de recherche sémantique des images. Ce moteur est doté d'une base de données d'images annotées respectivement à une ontologie: les images joueront le rôle des individus.

Ce moteur de recherche exploite des mesures de similarité pour répondre à une requête utilisateur. Pour simplifier, considérons que la requête consiste en un seul mot. Les résultats d'une recherche sont organisés *par ordre décroissant de pertinence*.

- ❑ Supposons que l'utilisateur cherche le concept $C3$. Donner les résultats renvoyés suite à cette requête si la métrique basée sur la longueur de chemin est utilisée.
- ❑ Même question si on cherche $C6$ et la métrique utilisée est celle de Resnik.

$C1 \sqsubseteq T$	$C4 \sqsubseteq C5$
$C2 \sqsubseteq C1$	$C5 \sqsubseteq C1$
$C3 \sqsubseteq C1$	$C6 \sqsubseteq C5$
$C3 \sqsubseteq \perp$	
$C6 \sqsubseteq \perp$	$C1(image1)$
$C4 \sqsubseteq \perp$	$C2(image2)$
$C4 \sqsubseteq C3$	$C5(image3)$