

Ontologies et

Web Sémantique

Les Ontologies – Mise en Œuvre avec Protégé

Plan du cours

1. Protégé
2. Exemple: L'ontologie «African Wildlife »
3. Description de classes, propriétés, instances, en Protégé
4. Restrictions sur les propriétés en Protégé
5. Raisonnements sur les ontologies en Protégé

Présentation de Protégé

Protégé – IDE Ontologies OWL

- Protégé est un système permettant la création d'ontologies très populaire développé à l'Université de Stanford.
- Protégé est développé en Java.
- Protégé fonctionnant sur la machine virtuelle Java, est multiplateformes (linux, Windows, MacOS, ...)
- Protégé est gratuit et son code source est publié sous une licence libre (Mozilla Public License).
- Protégé peut lire et sauvegarder des ontologies dans la plupart des formats d'ontologies : RDF, RDFS, OWL, Turtle, JSON-LD, etc.
- Lien de téléchargement : <https://protege.stanford.edu/>
- Web Protégé : <https://webprotege.stanford.edu/>

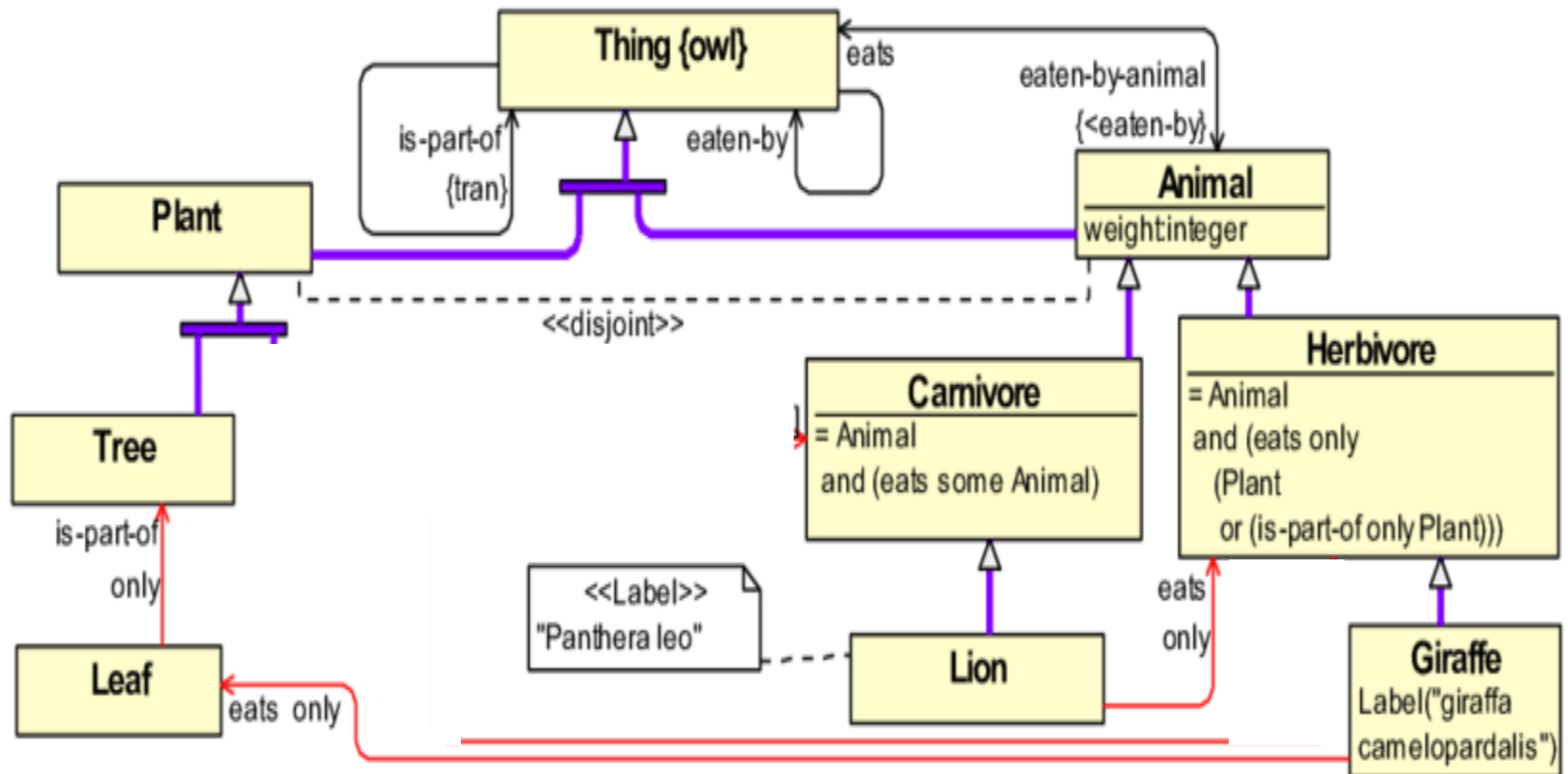
Présentation de Protégé

Protégé – IDE Ontologies OWL

- A télécharger depuis : <https://protege.stanford.edu/>
- Création classes / sous-classes.
- Création de propriétés.
- Restrictions classes / propriétés.
- Paramétrages .
- Création d'instances de classes.
- Vérifier l'ontologie. Reasoning.
- Visualisation en graphe. Interrogation SPARQL.
- Etc.

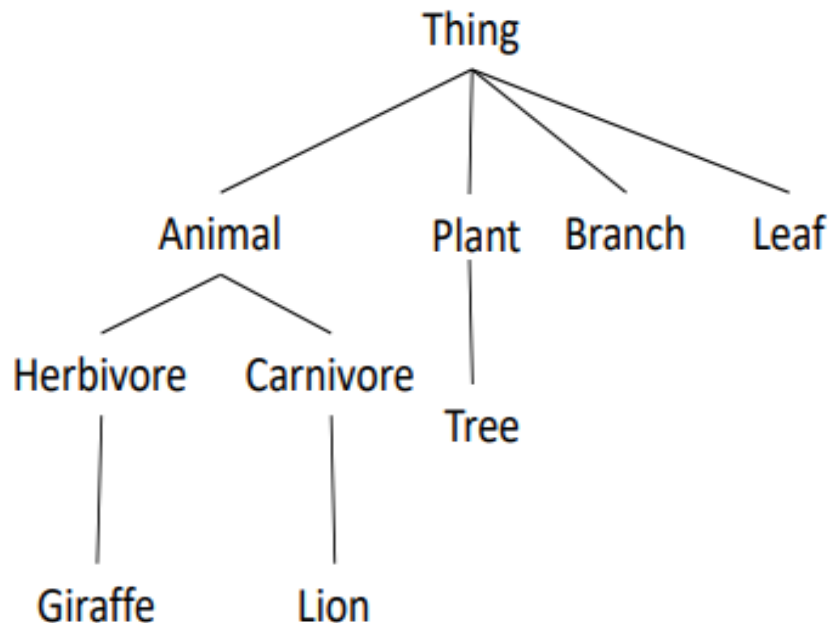
Exemple - African Wildlife Ontology

- **Source** : Antoniou, G, van Harmelen, F. A Semantic Web Primer. MIT Press, 2003.
- Réadaptée.

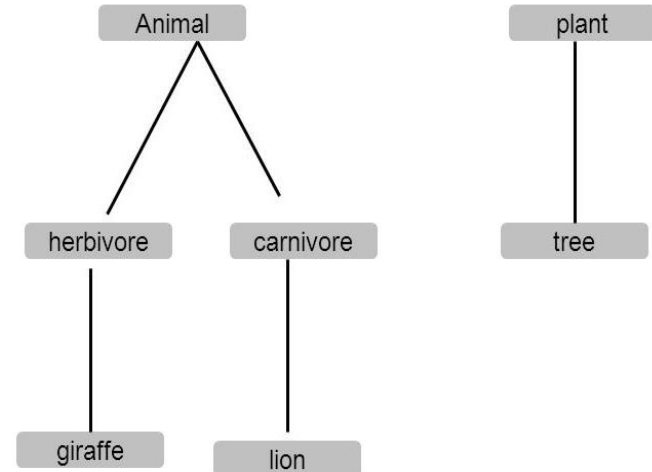


Exemple - African Wildlife Ontology

- **Source** : Antoniou, G, van Harmelen, F. A Semantic Web Primer. MIT Press, 2003. *Réadaptée*.
- **Objet** : Cette ontologie décrit la vie animale en Afrique avec :
 - des animaux: carnivores (des lions) et herbivores (des girafes)
 - des plantes telles que des arbres composés de branches et de feuilles



An African Wildlife Ontology – Class Hierarchy



Exemple - African Wildlife Ontology

▪ Déclaration de **Classes** :

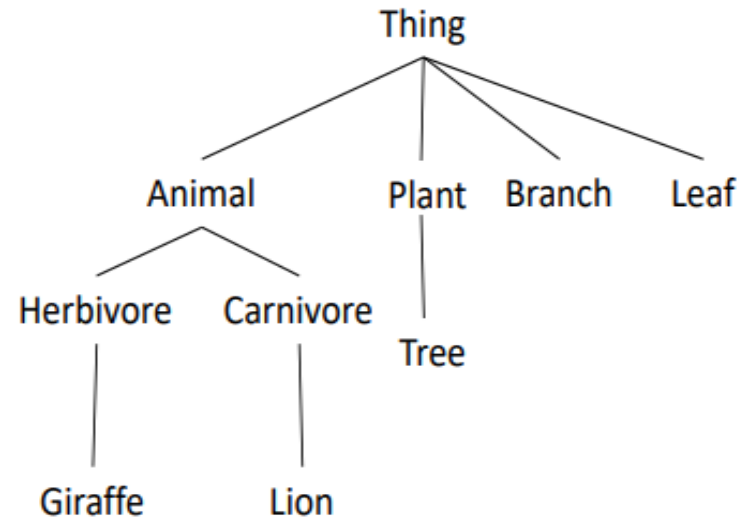
```
<owl:Class rdf:ID="Animal"/>
```

```
<owl:Class rdf:ID="Herbivore">  
  <rdfs:subClassOf rdf:resource="#Animal"/>  
</owl:Class>
```

```
<owl:Class rdf:ID="Carnivore">  
  <rdfs:subClassOf rdf:resource="#Animal"/>  
</owl:Class>
```

```
<owl:Class rdf:ID="Lion">  
  <rdfs:subClassOf rdf:resource="#Carnivore"/>  
</owl:Class>
```

```
<owl:Class rdf:ID="Giraffe">  
  <rdfs:subClassOf rdf:resource="#Herbivore"/>  
</owl:Class>
```



Exemple - African Wildlife Ontology

▪ Déclaration de **Classes** :

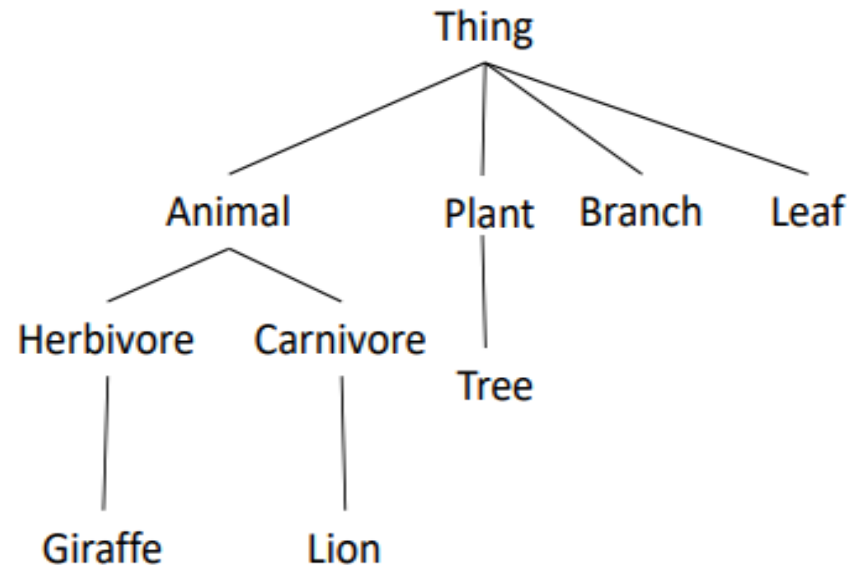
...

```
<owl:Class rdf:ID="Plant"/>
```

```
<owl:Class rdf:ID="Tree">  
  <rdfs:subClassOf rdf:resource="#Plant"/>  
</owl:Class>
```


```
<owl:Class rdf:ID="Branch"/>
```

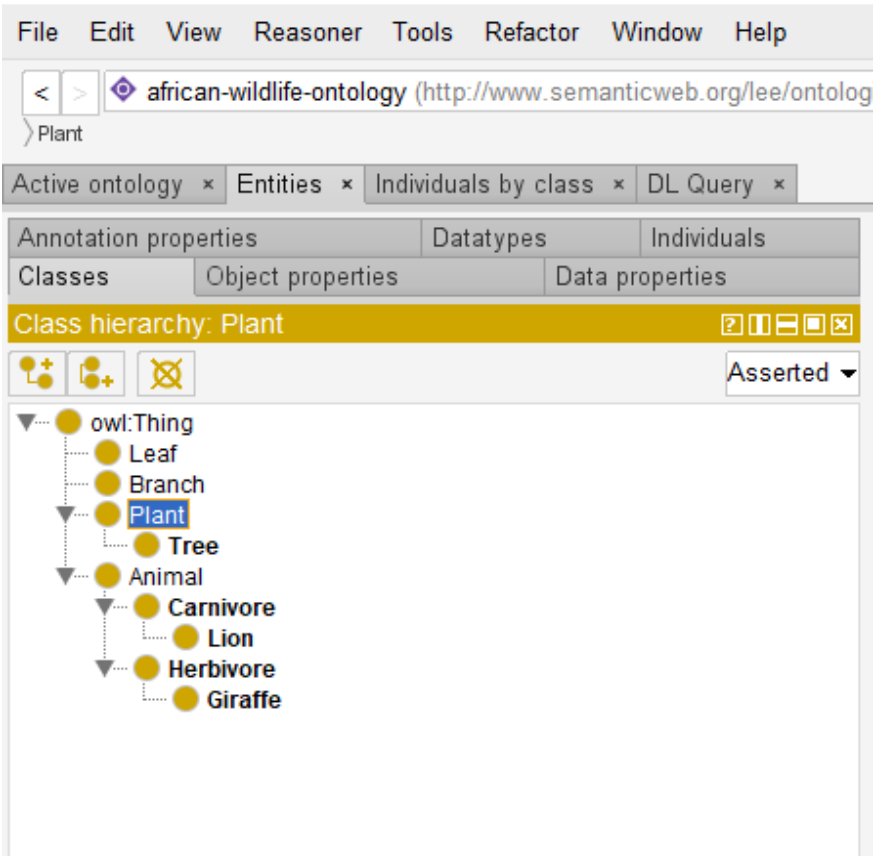
```
<owl:Class rdf:ID="Leaf"/>
```



Exemple - African Wildlife Ontology

▪ Déclaration de **Classes** : dans Protégé

 african-wildlife-ontology (<http://www.semanticweb.org>)



File Edit View Reasoner Tools Refactor Window Help

< > african-wildlife-ontology (<http://www.semanticweb.org/lee/ontology>)

Plant

Active ontology x Entities x Individuals by class x DL Query x

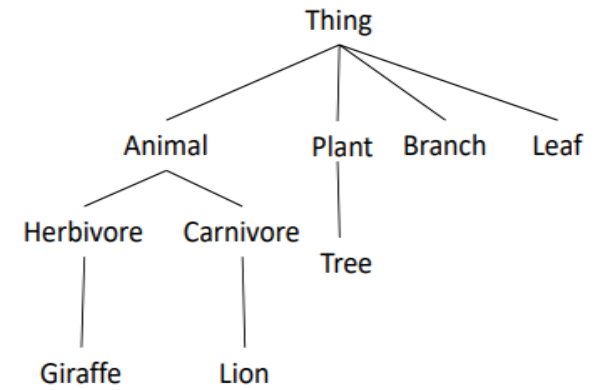
Annotation properties Datatypes Individuals


Classes Object properties Data properties

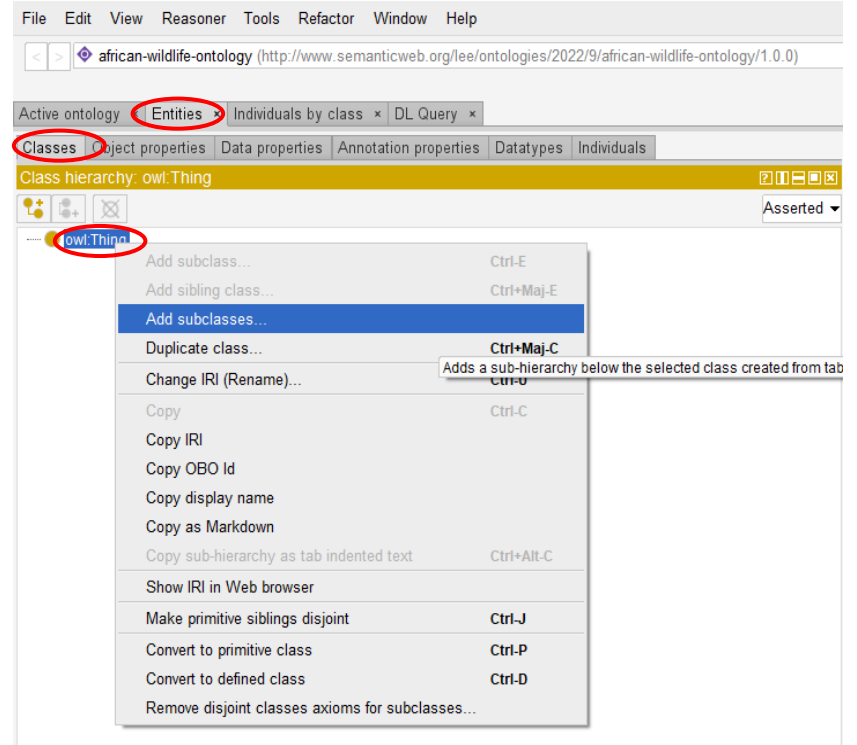
Class hierarchy: Plant

Asserted

- owl:Thing
 - Leaf
 - Branch
 - Tree
 - Animal
 - Carnivore
 - Lion
 - Herbivore
 - Giraffe



 african-wildlife-ontology (<http://www.semanticweb.org/lee/ontologies/2022/9/african-wildlife-ontology/1.0.0>)



File Edit View Reasoner Tools Refactor Window Help

< > african-wildlife-ontology (<http://www.semanticweb.org/lee/ontologies/2022/9/african-wildlife-ontology/1.0.0>)

Active ontology x Entities x Individuals by class x DL Query x

Classes Object properties Data properties Annotation properties Datatypes Individuals

Class hierarchy: owl:Thing

Asserted

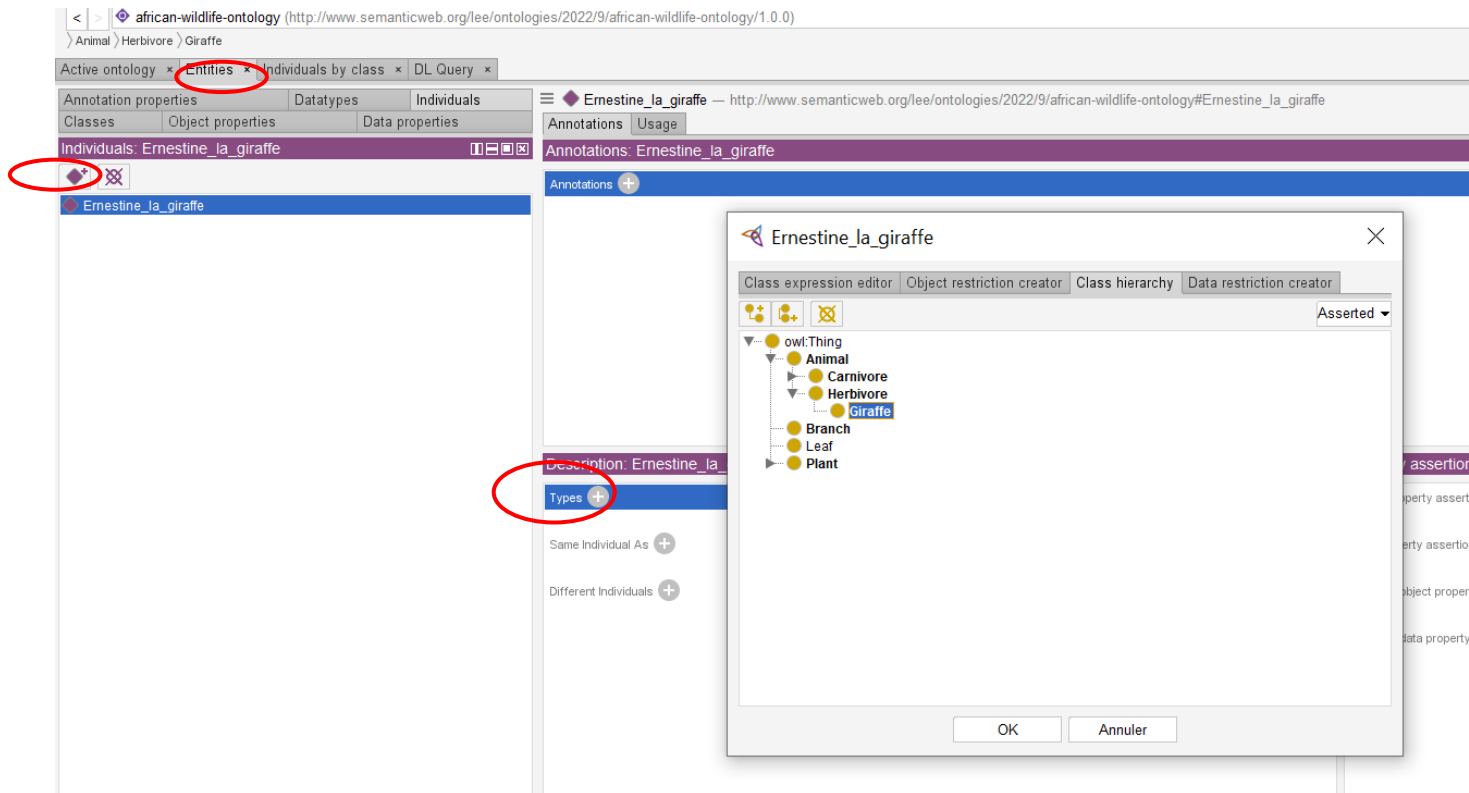
- owl:Thing
 - Leaf
 - Branch
 - Tree
 - Animal
 - Carnivore
 - Lion
 - Herbivore
 - Giraffe

Exemple - African Wildlife Ontology

- Déclaration des **instances (individuals)** : dans Protégé

<Giraffe rdf:about="Ernestine_la_girafe"/>

<Lion rdf:about="Leon_le_lion"/>



Exemple - African Wildlife Ontology

- Déclaration de **Propriétés** : **ObjectProperty**

```
<owl:ObjectProperty rdf:ID="is-part-of"/>
```

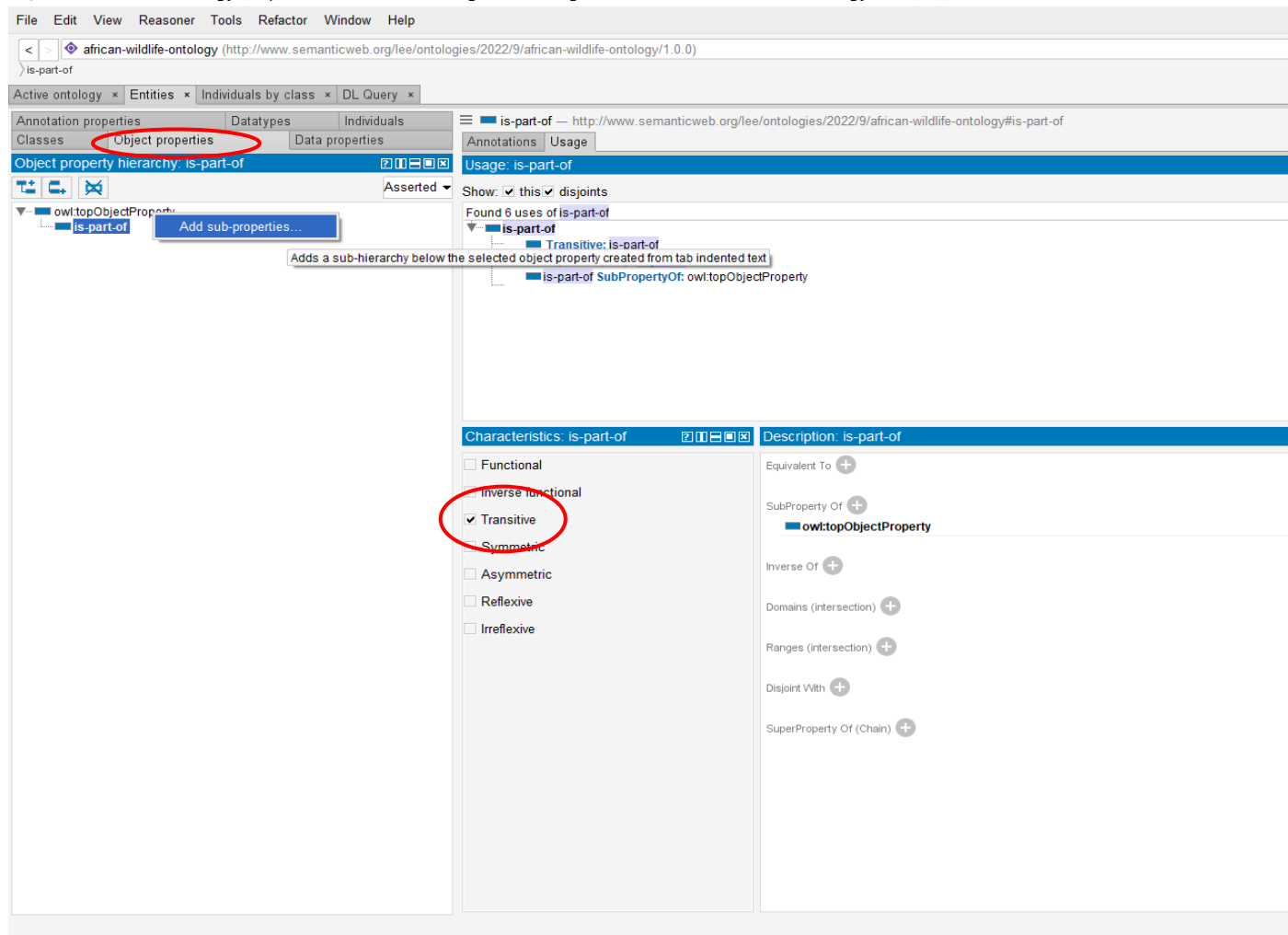
```
<owl:TransitiveProperty rdf:ID="is-part-of"/>
```

```
<owl:ObjectProperty rdf:ID="eats">  
  <rdfs:domain rdf:resource="#Animal"/>  
</owl:ObjectProperty>
```

```
<owl:ObjectProperty rdf:ID="eaten-by">  
  <owl:inverseOf rdf:resource="#eats"/>  
</owl:ObjectProperty>
```

Exemple - African Wildlife Ontology

■ Déclaration de **Propriétés** : dans Protégé



Exemple - African Wildlife Ontology

■ Déclaration de **Propriétés** : dans Protégé

The screenshot displays the Protégé ontology editor interface for the African Wildlife Ontology. The top navigation bar includes tabs for 'Active ontology', 'Entities', 'Individuals by class', and 'DL Query'. Below this, a secondary bar shows 'Annotation properties', 'Datatypes', 'Individuals', 'Classes', 'Object properties' (highlighted with a red circle), and 'Data properties'. The main workspace is divided into several panels:

- Object property hierarchy: eats**: A tree view on the left showing the hierarchy of the 'eats' property. It includes 'owl:topObjectProperty', 'eaten-by', 'eats' (highlighted), and 'is-part-of'.
- Usage: eats**: A panel on the right showing the usage of the 'eats' property. It includes a 'Show' checkbox for 'this' and 'disjoints', and a list of 10 uses of 'eats'. The uses include 'eaten-by InverseOf eats', 'eaten-by', 'eats InverseOf eats', 'ObjectProperty: eats', 'eats SubPropertyOf: owl:topObjectProperty', and 'eats Domain Animal'.
- Characteristics: eats**: A panel on the left showing the characteristics of the 'eats' property. It includes checkboxes for 'Functional', 'Inverse functional', 'Transitive', 'Symmetric', 'Asymmetric', 'Reflexive', and 'Irreflexive'.
- Description: eats**: A panel on the right showing the description of the 'eats' property. It includes fields for 'Equivalent To', 'SubProperty Of' (set to 'owl:topObjectProperty'), 'Inverse Of' (set to 'eaten-by'), 'Domains (intersection)' (set to 'Animal', highlighted with a red circle), 'Ranges (intersection)', 'Disjoint With', and 'SuperProperty Of (Chain)'.

Exemple - African Wildlife Ontology

■ Déclaration de **Propriétés** : dans Protégé

The screenshot displays the Protégé ontology editor interface. The top navigation bar includes tabs for 'Active ontology', 'Entities', 'Individuals by class', and 'DL Query'. The left sidebar shows the 'Object property hierarchy' for 'eaten-by', with a tree structure including 'owl:topObjectProperty', 'eaten-by', 'eats', and 'is-part-of'. The main area is divided into two panes. The top pane, titled 'Usage: eaten-by', shows 'Found 4 uses of eaten-by' and lists 'ObjectProperty: eaten-by' and 'eaten-by SubPropertyOf: owl:topObjectProperty'. The bottom pane, titled 'Description: eaten-by', contains a list of characteristics (Functional, Inverse functional, Transitive, Symmetric, Asymmetric, Reflexive, Irreflexive) and a list of relationships (Equivalent To, SubProperty Of, Inverse Or, Domains (intersection), Ranges (intersection), Disjoint With, SuperProperty Of (Chain)). The 'Inverse Or' relationship is highlighted with a red circle, showing 'eaten-by Inverse Or eats'.

Active ontology × Entities × Individuals by class × DL Query ×

Annotation properties | Datatypes | Individuals | Classes | Object properties | Data properties

Object property hierarchy: eaten-by [?] [] [] [] []

Annotations | Usage

Usage: eaten-by

Show: ☒ this ☒ disjoints

Found 4 uses of eaten-by

owl:topObjectProperty

- eaten-by
- eats
- is-part-of

Characteristics: eaten-by [?] [] [] [] []

- ☐ Functional
- ☐ Inverse functional
- ☐ Transitive
- ☐ Symmetric
- ☐ Asymmetric
- ☐ Reflexive
- ☐ Irreflexive

Description: eaten-by

Equivalent To +

SubProperty Of +

- owl:topObjectProperty

Inverse Or +

- eats

Domains (intersection) +

Ranges (intersection) +

Disjoint With +

SuperProperty Of (Chain) +

Exemple - African Wildlife Ontology

- Déclaration de **Propriétés** : DatatypeProperty

```
<owl:DatatypeProperty rdf:about="hasAge">
```

```
  <rdfs:range rdf:resource=  
    "http://www.w3.org/2001/XMLSchema#nonNegativeInteger"/>
```

```
</owl:DatatypeProperty>
```

```
<owl:FunctionalProperty rdf:about="hasAge"/>
```

Propriété fonctionnelle : propriété qui a au plus une valeur pour chaque objet

Exemple - African Wildlife Ontology

- **Déclaration de Propriétés** : dans Protégé

The screenshot displays the Protégé ontology editor interface. The top navigation bar includes tabs for 'Active ontology', 'Entities', 'Individuals by class', and 'DL Query'. Below this, a secondary bar shows 'Annotation properties', 'Datatypes', 'Individuals', 'Classes', 'Object properties', and 'Data properties'. The 'Data properties' tab is selected and circled in red. The main workspace is divided into several panels:

- Data property hierarchy: hasAge**: Shows a tree view with 'owl:topDataProperty' as the parent and 'hasAge' as the child. A context menu is open over 'hasAge' with the option 'Add sub-properties...'.
- Annotations: hasAge**: A panel for adding annotations to the selected property.
- Characteristics: hasAge**: A panel where the 'Functional' checkbox is checked.
- Description: hasAge**: A panel for defining the property's characteristics, including:
 - Equivalent To**: A button to add equivalent properties.
 - SubProperty Of**: A button to add superproperties, with 'owl:topDataProperty' listed below.
 - Domains (intersection)**: A button to add domain restrictions.
 - Ranges**: A button to add range restrictions, with 'xsd:nonNegativeInteger' listed below.
 - Disjoint With**: A button to add disjoint properties.

Exemple - African Wildlife Ontology

▪ Déclaration de **Classes** :

ies/2022/9/african-wildlife-ontology/1.0.0)

Animal — <http://www.semanticweb.org/lee/ontologies/2022/9/african-wildlife-ontology#Animal>

Annotations Usage

Usage: Animal

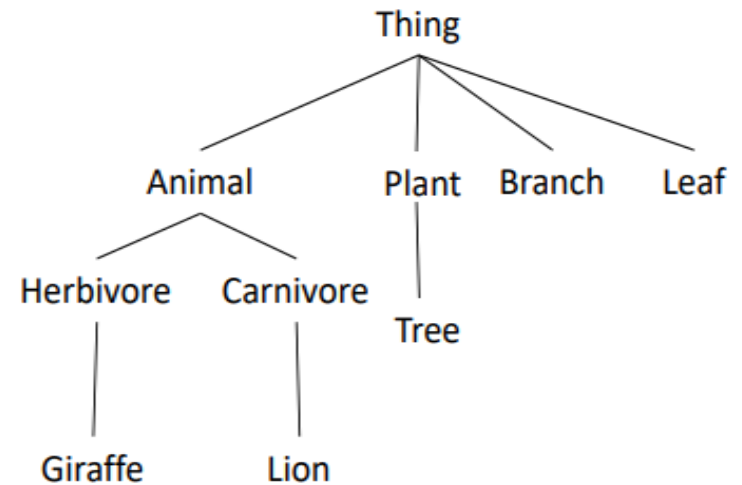
Show: ☒ this ☒ disjoints ☒ named sub/superclasses

Found 6 uses of Animal

- Animal
 - Class: Animal
- Carnivore
 - Carnivore SubClassOf Animal
- Herbivore
 - Herbivore SubClassOf Animal

Description: Animal

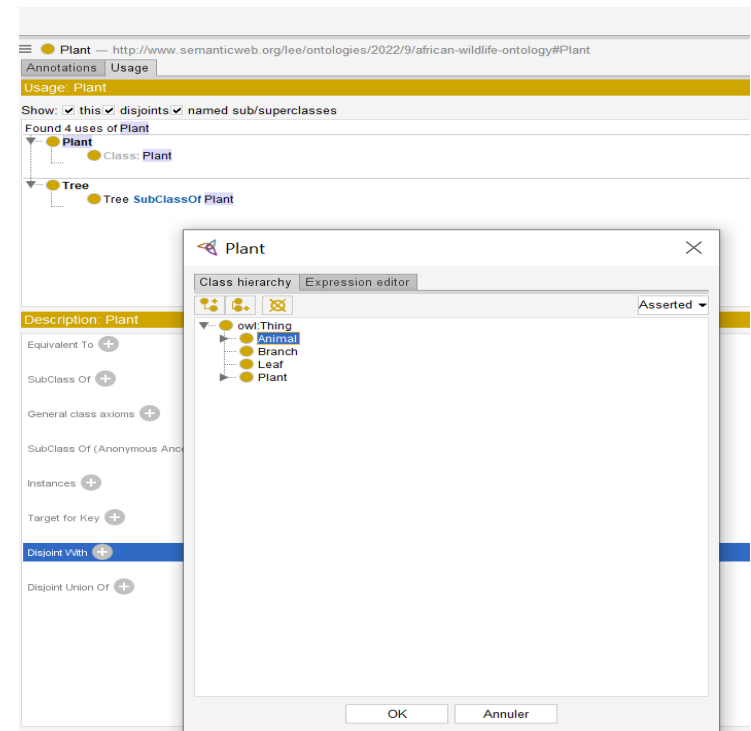
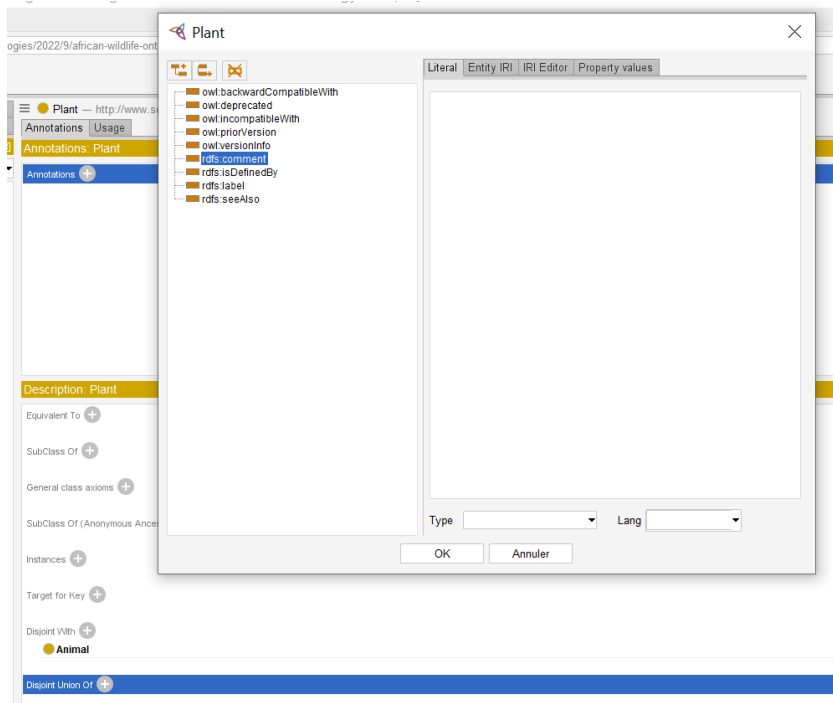
- Equivalent To +
- SubClass Of +
- General class axioms +
- SubClass Of (Anonymous Ancestor)
- Instances +
- Target for Key +
- Disjoint With +
- Disjoint Union Of +



Exemple - African Wildlife Ontology

▪ Déclaration de **Classes** : sur Protégé

```
<owl:Class rdf:ID="Plant">  
  <rdfs:comment>Plants are disjoints from animals  
</rdfs:comment>  
  <owl:disjointWith rdf:resource="#Animal"/>  
</owl:Class>
```



Exemple - African Wildlife Ontology

- **Déclaration de **Classes**** : restrictions sur les propriétés

```
<owl:Class rdf:ID="Carnivore">
```

```
  <rdfs:subClassOf>
```

```
    <owl:Restriction>
```

```
      <owl:onProperty rdf:resource="#eats"/>
```

```
      <owl:someValuesFrom rdf:resource="#Animal"/>
```

```
    </owl:Restriction>
```

```
  </rdfs:subClassOf>
```

```
  <rdfs:comment>
```

```
    Carnivores are those animals that eat also animals.
```

```
  </rdfs:comment>
```

```
</owl:Class>
```

Exemple - African Wildlife Ontology

- **Déclaration de **Classes**** : restrictions sur les propriétés – dans Protégé

The screenshot displays the Protégé interface for the African Wildlife Ontology. The main window shows the class hierarchy for 'Carnivore' (http://www.semanticweb.org/lee/ontologies/2022/9/african-wildlife-ontology#Carnivore). The hierarchy includes 'owl:Thing', 'Leaf', 'Branch', 'Plant', 'Tree', 'Animal', 'Carnivore', 'Lion', 'Herbivore', and 'Giraffe'. The 'Carnivore' class is highlighted, and its description is shown as 'Carnivores are those animals that eat also animals'.

On the left, a 'Description: Carnivore' panel shows the class hierarchy and the 'SubClass Of' relationship with 'Animal' and 'eats some Animal'.

On the right, the 'Object restriction creator' dialog is open, showing the 'Restricted property' (eaten-by) and the 'Restriction filler' (Animal). The 'Restriction type' is set to 'Some (existential)' and the 'Cardinality' is 1.

The 'Object restriction creator' dialog is a window with a close button (X) in the top right corner. It has four tabs: 'Data restriction creator', 'Class expression editor', 'Class hierarchy', and 'Object restriction creator' (which is selected and circled in red). The 'Restricted property' section shows a tree structure with 'owl:topObjectProperty', 'eaten-by', and 'is-part-of'. The 'Restriction filler' section shows a tree structure with 'owl:Thing', 'Animal' (circled in red), 'Branch', 'Leaf', and 'Plant'. The 'Restriction type' section has a dropdown menu with 'Some (existential)' (circled in red) and a 'Cardinality' field with the value '1'. At the bottom are 'OK' and 'Annuler' buttons.

Exemple - African Wildlife Ontology

- **Déclaration de **Classes**** : restrictions sur les propriétés

```
<owl:Class rdf:ID="Branch">
```

```
  <rdfs:comment>
```

```
    Branches are parts of trees.
```

```
  </rdfs:comment>
```

```
  <rdfs:subClassOf>
```

```
    <owl:Restriction>
```

```
      <owl:onProperty rdf:resource="#is-part-of"/>
```

```
      <owl:allValuesFrom rdf:resource="#Tree"/>
```

```
    </owl:Restriction>
```

```
  </rdfs:subClassOf>
```

```
</owl:Class>
```

Exemple - African Wildlife Ontology

- **Déclaration de Classes** : restrictions sur les propriétés – dans Protégé

The screenshot displays the Protégé ontology editor interface for the 'African Wildlife Ontology'. The main window shows the 'Class hierarchy' for the 'Branch' class, which is a subclass of 'Plant'. The hierarchy includes 'Leaf', 'Branch', 'Plant', 'Tree', 'Animal', 'Carnivore', 'Lion', 'Herbivore', and 'Giraffe'. The 'Description' tab for 'Branch' is active, showing a list of axioms: 'Equivalent To', 'SubClass Of', 'General class axioms', 'SubClass Of (Anonymous Ancestor)', 'Instances', 'Target for Key', 'Disjoint With', and 'Disjoint Union Of'. A dialog box titled 'Branch' is open, showing the 'Restricted property' tab. The 'Restricted property' is 'is-part-of' (a subclass of 'owl:topObjectProperty'). The 'Restriction filler' is 'Plant' (a subclass of 'owl:Thing'). The 'Restriction type' is 'Only (universal)' and the 'Cardinality' is '1'. The dialog also has 'OK' and 'Annuler' buttons.

Active ontology × Entities × Individuals by class × DL Query ×

Annotation properties | Datatypes | Individuals | **Classes** | Object properties | Data properties

Class hierarchy: Branch [Icons] [Asserted]

Annotations: Branch

Annotations +

rdfs:comment [language: en]
Branches are parts of trees.

Description: Branch

Equivalent To +

SubClass Of +

General class axioms +

SubClass Of (Anonymous Ancestor)

Instances +

Target for Key +

Disjoint With +

Disjoint Union Of +

owl:Thing
Leaf
Branch
Plant
Tree
Animal
Carnivore
Lion
Herbivore
Giraffe

Branch

Data restriction creator | Class expression editor | Class hierarchy | Object restriction creator

Restricted property

owl:topObjectProperty
eaten-by
eats
is-part-of

Restriction filler

owl:Thing
Animal
Branch
Leaf
Plant

Restriction type

Only (universal) Cardinality 1

OK Annuler

Description: Branch

Equivalent To +

SubClass Of +

is-part-of only Plant

Exemple - African Wildlife Ontology

- **Déclaration de **Classes**** : restrictions sur les propriétés

```
<owl:Class rdf:about="#Animal">

  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="hasAge"/>
      <owl:cardinality
        rdf:datatype=&xsd;nonNegativeInteger>
        1
      </owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>

</owl:Class>
```

Exemple - African Wildlife Ontology

- **Déclaration de Classes** : restrictions sur les propriétés – dans Protégé

The screenshot shows the Protégé ontology editor interface. The main window displays the class hierarchy for the 'Animal' class, which includes subclasses like 'Carnivore', 'Lion', 'Herbivore', and 'Giraffe'. The 'Description: Animal' dialog box is open, showing the 'hasAge' property restricted to 'xsd:nonNegativeInteger' with an exact cardinality of 1. A 'Data restriction creator' dialog box is also open, showing the 'Restricted property' as 'owl.topDataProperty' and the 'Restriction filler' as 'xsd:nonNegativeInteger'.

Class hierarchy: Animal

- owl:Thing
 - Leaf
 - Branch
 - Plant
 - Tree
 - Animal**
 - Carnivore
 - Lion
 - Herbivore
 - Giraffe

Description: Animal

Equivalent To +

SubClass Of +

General class axioms +

SubClass Of (Anonymous Ancestor) +

Instances +

Target for Key +

Disjoint With +

Disjoint Union Of +

hasAge exactly 1 xsd:nonNegativeInteger

Animal

Data restriction creator | Class expression editor | Class hierarchy | Object restriction creator

Restricted property

owl.topDataProperty

hasAge

Restriction filler

- xsd:name
- xsd:NCName
- xsd:negativeInteger
- xsd:NMTOKEN
- xsd:nonNegativeInteger**
- xsd:nonPositiveInteger
- xsd:normalizedString
- xsd:positiveInteger
- xsd:short
- xsd:string
- xsd:token
- xsd:unsignedByte
- xsd:unsignedInt
- xsd:unsignedLong
- xsd:unsignedShort

Restriction type

Exactly (exact cardinality) Cardinality 1

OK Annuler

Exemple - African Wildlife Ontology

- **Déclaration de **Classes**** : intersectionOf

```
<owl:Class rdf:ID="Omnivore">
```

```
  <owl:equivalentClass>
```

```
    <owl:Class>
```

```
      <owl:intersectionOf rdf:parseType="Collection">
```

```
        <owl:Class rdf:about="#Herbivore"/>
```

```
        <owl:Class rdf:about="#Carnivore"/>
```

```
      </owl:intersectionOf>
```

```
    </owl:Class>
```

```
  </owl:equivalentClass>
```

```
</owl:Class>
```

Exemple - African Wildlife Ontology

- **Déclaration de Classes** : intersectionOf – dans Protégé

The screenshot displays the Protégé interface for the African Wildlife Ontology. The main window shows the class hierarchy for 'Omnivore', which is a subclass of 'Animal'. The 'Omnivore' class is highlighted in blue. The 'Description' tab for 'Omnivore' is active, showing the 'Equivalent To' section with the expression 'Carnivore and Herbivore'. A red circle highlights the 'Equivalent To' section. A modal dialog box titled 'Omnivore' is open, showing the 'Class expression editor' tab. The dialog contains the text 'Carnivore and Herbivore' and has 'OK' and 'Annuler' buttons at the bottom. The 'Description' tab for 'Omnivore' also shows the 'Equivalent To' section with the expression 'Carnivore and Herbivore'.

afican-wildlife-ontology (<http://www.semanticweb.org/lee/ontologies/2022/9/afican-wildlife-ontology/1.0.0>)

Animal > Carnivore > Omnivore

Active ontology x Entities x Individuals by class x DL Query x

Annotation properties Datatypes Individuals
Classes Object properties Data properties

Class hierarchy: Omnivore Annotations: Omnivore

Annotations +

owl:Thing
Leaf
Branch
Plant
Tree
Animal
Carnivore
Omnivore
Lion
Herbivore
Omnivore
Giraffe

Description: Omnivore

Equivalent To +
Carnivore and Herbivore

SubClass Of +

General class axioms +

SubClass Of (Anonymous Ancestor)
eats some Animal
hasAge exactly 1 xsd:nonNegativeInteger

Instances +

Target for Key +

Omnivore

Object restriction creator Class expression editor Data restriction creator Class hierarchy

Carnivore and Herbivore

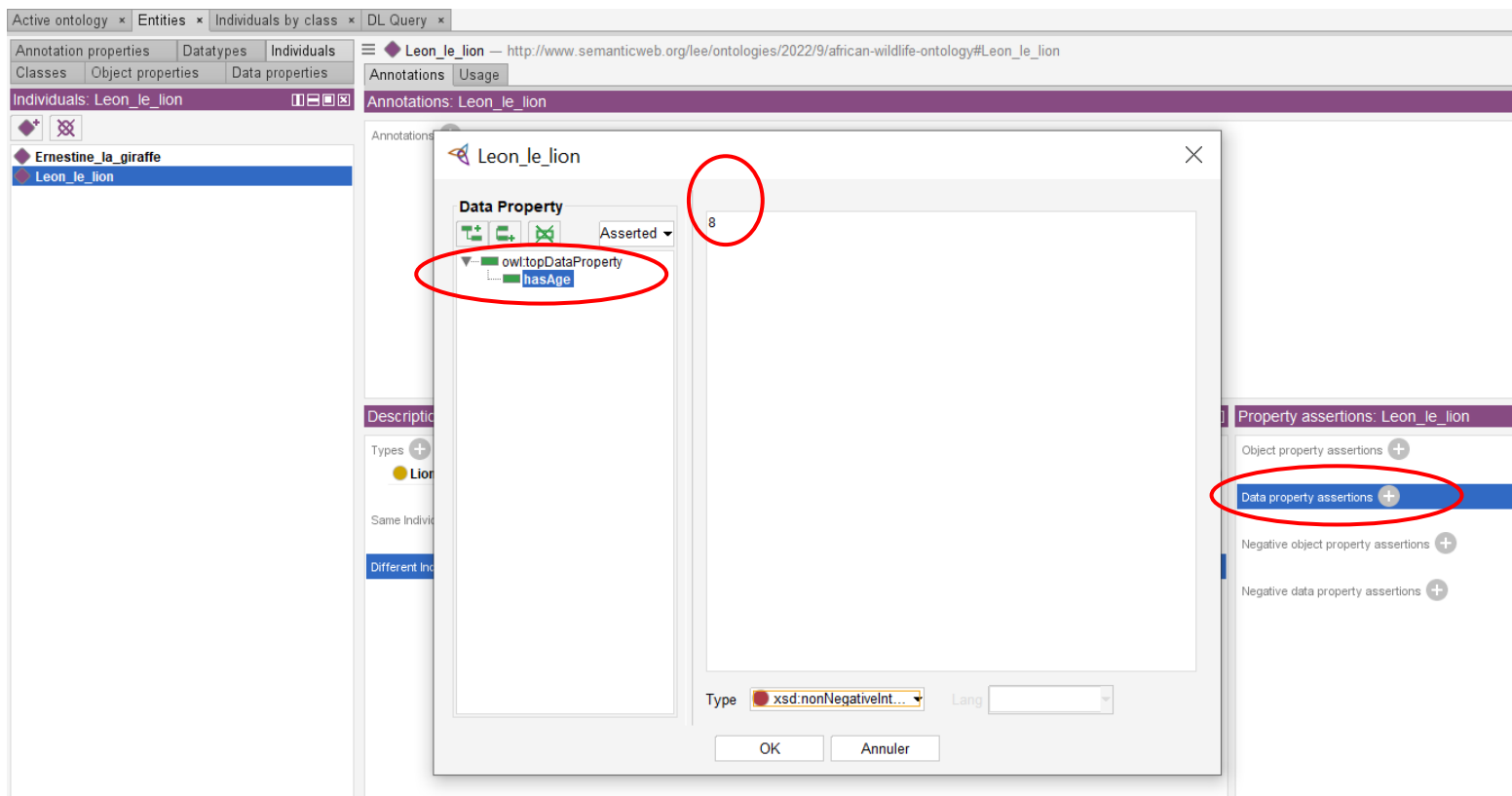
Help...

OK Annuler

Exemple - African Wildlife Ontology

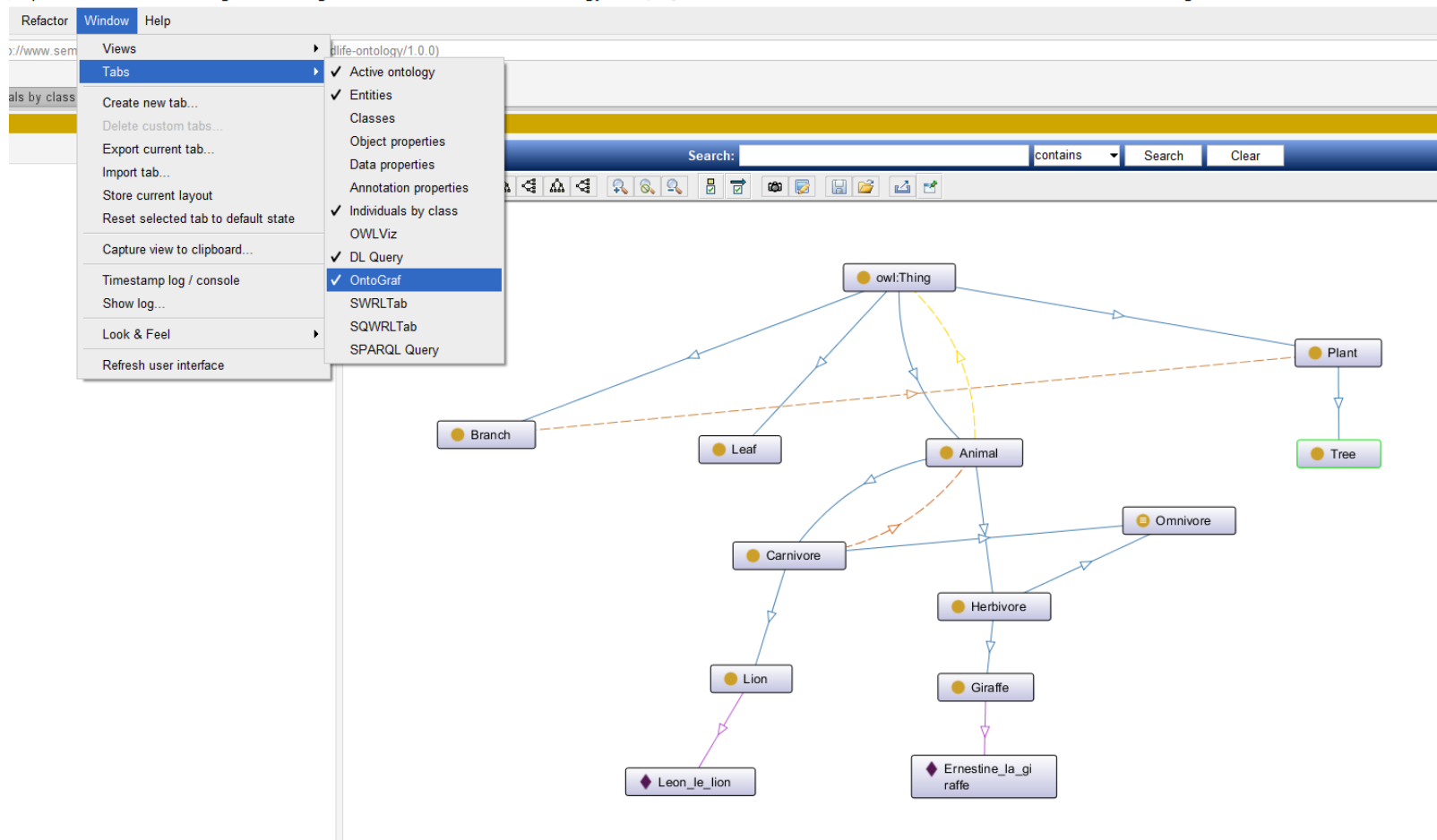
- **Déclaration des instances** : hasAge - dans Protégé

```
<Lion      rdf:about="Leon_le_lion"/>  
  <hasAge  rdf:datatype="&xsd;integer">8</hasAge>  
</Lion>
```



Exemple - African Wildlife Ontology

■ Visualisation de l'ontologie : OntoGraph - dans Protégé



Exemple - African Wildlife Ontology

▪ Requête: SPARQL Query - dans Protégé

Active ontology x Entities x Individuals by class x DL Query x OntoGraf x SPARQL Query x

SPARQL query:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX local: <http://www.semanticweb.org/lee/ontologies/2022/9/african-wildlife-ontology#>
```

```
SELECT ?lions ?giraffes
```

```
WHERE {
    ?lions rdf:type local:Lion .
    ?giraffes rdf:type local:Giraffe
}
```

PREFIX local:

<http://www.semanticweb.org/lee/ontologies/2022/9/african-wildlife-ontology#>

SELECT ?lions ?giraffes

WHERE {

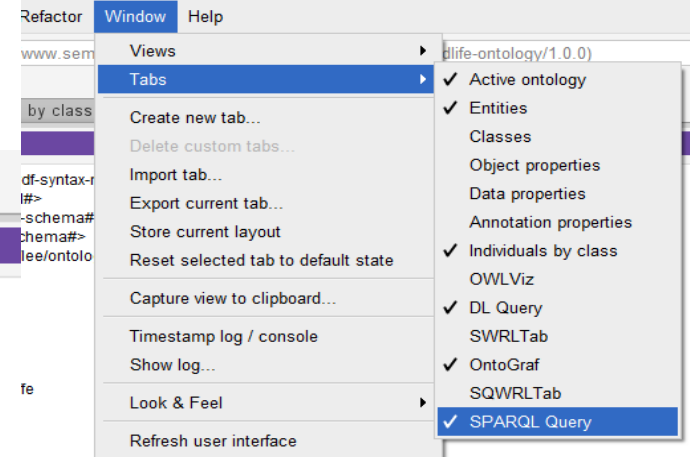
?lions rdf:type local:Lion .

?giraffes rdf:type local:Giraffe

}

lions		giraffes	
Leon_le_lion		Ernestine_la_giraffe	

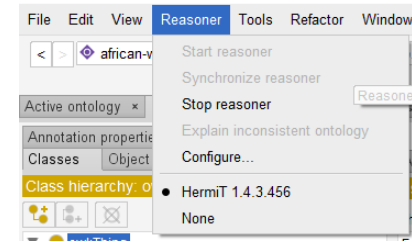
http://www.semanticweb.org/lee/ontologies/2022/9/african-wildli



Raisonnement sur les ontologies

- Pourquoi raisonner sur les ontologies :
 - Pour **vérifier** la cohérence d'une ontologie.
 - Pour **inférer** de nouvelles connaissances, on peut utiliser un raisonneur.
- Exemple : *Vérifier*
- Supposons qu'on crée une nouvelle instance *Test_Instance*, de type Carnivore et Herbivore, qui sont déclarées disjointes.
- Si on lance le « raisonneur » de Protégé, on obtient:

Raisonnement sur les ontologies



A screenshot of the Protege interface showing an inconsistent ontology. The main window displays the 'Test_Instance' ontology. The 'Individuals' tab is active, showing a list of individuals: Ernestine_la_giraffe, Leon_le_lion, and Test_Instance. The 'Test_Instance' individual is highlighted. The 'Annotations' tab is also visible, showing 'Annotations: Test_Instance'. A red circle highlights the 'Test_Instance' individual in the list. Another red circle highlights the 'Help for inconsistent ontologies' dialog box, which is open and displays the following text:

Help for inconsistent ontologies

Your ontology is inconsistent which means that the OWL reasoner will no longer be able to provide any useful information about the ontology.

You have several options at this point:

- Click the Explain button to try the Protege explanation facility.
- If you think you know what the problem is, click Cancel to fix the ontology yourself.
- Some reasoners come with command line tools that will provide complete explanations for inconsistent ontologies.

At the bottom of the dialog box are two buttons: 'Explain' and 'Cancel'.

In the background, the 'Types' section of the 'Test_Instance' individual is visible, showing 'Carnivore' and 'Herbivore' as types, which are also circled in red.

Raisonnement sur les ontologies

- Pourquoi raisonner sur les ontologies :
 - Pour **vérifier** la cohérence d'une ontologie.
 - Pour **inférer** de nouvelles connaissances, on peut utiliser un raisonneur.
- Exemple : *inférer*
- Supposons qu'*Ernestine_la_girafe* mange (*eats*) *Leon_le_lion*.
- Si on lance le « raisonneur » de Protégé, on obtient:



Raisonnement sur les ontologies

Property assertions: Leon_le_lion

Object property assertions +

eaten-by Ernestine_la_giraffe ? @

Data property assertions +

hasAge "8"^^xsd:nonNegativeInteger ? @ x o

Negative object property assertions +

Negative data property assertions +

Inferred

Description: eats

Equivalent To +

SubProperty Of +

owl:topObjectProperty

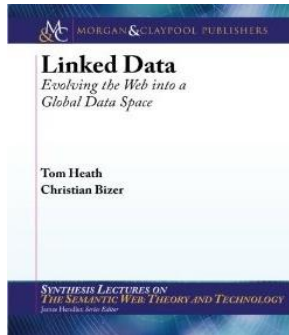
Inverse Of +

eaten-by

Domains (intersection) +

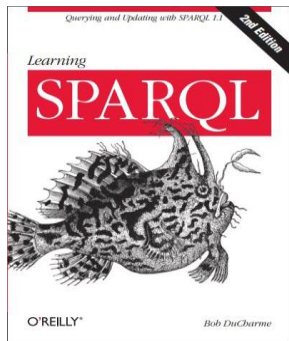
Animal

Références



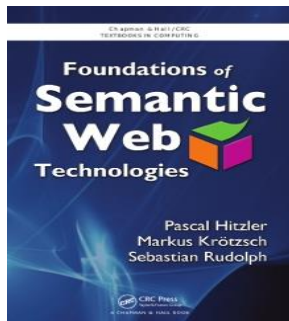
Linked Data: Evolving the Web into a Global Data Space

- ✓ Auteur : Christian Bizer, Tom Heath
- ✓ Éditeur : Morgan & Claypool Publishers
- ✓ Edition : Février 2011 - 136 pages - ISBN 9781608454310



Learning SPARQL : Querying and Updating with SPARQL

- ✓ Auteur : Bob DuCharme
- ✓ Éditeur : O'Reilly Media
- ✓ Edition: Juillet 2013– 386pages -ISBN : 9781449306595



Foundations of Semantic Web Technologies

- ✓ Auteur : Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph
- ✓ Éditeur : CRC Press/Chapman and Hall
- ✓ Edition : 2009 - 455 pages - ISBN : 9781420090505

Références

- W3C – OWL Web Ontology Language
 - ✓ <https://www.w3.org/Submission/SWRL/>
- INRIA MOOC - Fabien Gandon – Web Sémantique et Web de Données
 - ✓ https://www.canal-u.tv/producteurs/inria/cours_en_ligne/web_semantique_et_web_de_donnees
- Noy et McGuinness - Ontology Development 101: A Guide to Creating Your First Ontology.
 - ✓ https://protege.stanford.edu/publications/ontology_development/ontology101.pdf/
- Cours – Bernard ESPINASSE
 - ✓ <https://pageperso.lis-lab.fr/bernard.espinasse/wp-content/uploads/2021/12/8-Cours-OWL-Protege-4P-1.pdf> /