

# Tutoriel *protégé* 5.2 - Création d'une *petite* ontologie

[Guy Lapalme](#)

Vous devez d'abord [installer Protégé](#) [Ce document suppose l'installation de Protégé 5.2]. Télécharger la version *desktop* et **non** WebProtégé.

Le Site de Protégé explique [comment explorer une ontologie existante](#), mais ici vous allez créer une nouvelle ontologie.

Informations à exprimer dans l'ontologie

Cette ontologie est adaptée de l'exemple 3.1 (p 17) d'un texte de [Grigoris Antoniou and Frank van Harmelen](#). Elle était aussi présentée dans la deuxième édition [Semantic Web Primer](#) (p 135 et suivantes).

Pour simplifier, nous indiquons déjà les mots qui désigneront une classe avec cette police et avec sa première lettre en majuscule; les propriétés, elles, auront cette police avec sa première lettre en minuscule et en «bossesDeChameau» pour les mots différents.

Ces choix devraient en principe être faits suite à une analyse complète des besoins, mais nous les imposons ici pour les besoins de cet exercice.

1. Un Animal est une classe.
2. Une Plante est une classe, mais disjointe d'Animal.
3. Un Arbre est une sous-classe de Plante.
4. Une Branche est une partie d'un Arbre.
5. Une Feuille est une partie d'une Branche
6. Un Herbivore est un Animal qui ne mange qu'une Plante ou une partie d'une Plante.
7. Un Carnivore est un Animal qui mange aussi un Animal.
8. Une Girafe est un Herbivore qui ne mange que des Feuilles.
9. Un Lion est un Carnivore qui ne mange que des Herbivores.
10. Une PlanteSavoureuse est une Plante qui est mangée par un Herbivore et aussi par un Carnivore.

Étapes de création de l'ontologie

Lancer Protégé et vous devriez obtenir une configuration semblable à la prochaine figure qui utilise le *Look & Feel* Protégé qui peut différer légèrement sur différentes plateformes (on peut le modifier via le menu Window). Pour ce tutoriel, vous pouvez ignorer les mises à jour de *plug-in* offertes au lancement du programme.

*Sur MacOSX*: il faudra peut-être autoriser le lancement d'une application venant d'un *développeur non enregistré* en passant par les Préférences Système dans la section Sécurité et confidentialité.

Définir l'Ontology IRI pour un URI qui vous *ressemble*.

Dans le menu système File/Save, indiquer le format de sauvegarde de l'ontologie, je suggère RDF/XML et l'endroit sur votre disque où sera conservée l'ontologie.

En principe, votre écran devrait ressembler à celui-ci:

The screenshot shows the Protégé software interface for the 'AnimauxPlantes' ontology. The main window has a menu bar (File, Edit, View, Reasoner, Tools, Refactor, Window, Help) and a toolbar. The 'Active Ontology' tab is selected, showing the 'Entities' view. The 'Ontology header' pane displays the IRI and Version IRI. The 'Ontology metrics' pane shows various counts and the DL expressivity (AL). The 'Imported ontologies' pane shows direct and indirect imports. The status bar at the bottom indicates 'No Reasoner set' and 'Show Inferences' is checked.

L'écran est divisé en *vues* (*views* en anglais) qui dépendent des *onglets* (*tabs* en anglais) accessibles en cliquant sur un des éléments d'une liste de titres Active Ontology, Entities, .... Appuyer sur chacun de ses onglets pour constater les changements d'affichage des vues. Certaines vues (par exemple Entities) possèdent eux-mêmes des sous-onglets qui ont eux-mêmes plusieurs vues. On peut configurer l'écran en ajoutant des vues et des onglets (*tabs*) dans les menus Window>View et Window>Tabs. Voir [ce document](#) pour plus de détails sur la manipulation de l'interface. Il y a des vidéos expliquant comment manipuler [les vues](#) et [les onglets](#)

#### Définition des classes simples

Dans l'onglet Entities, choisir le sous-onglet Classes. Cliquer sur le mot owl:Thing dans la vue Class hierarchy qui est la racine de toutes les classes que nous allons créer. En appuyant sur le premier bouton en haut à gauche de cette vue, un dialogue apparaît qui permet de créer une classe qui sera un enfant de la classe sélectionnée. On peut ainsi définir des sous-classes. Le deuxième bouton permet de créer une classe *soeur* de la classe sélectionnée. Le troisième bouton détruit la classe sélectionnée ainsi que ses sous-classes.

Faire de même pour créer la hiérarchie des classes dont les noms sont en gras dans la liste des énoncés [ci-haut](#) pour obtenir une figure semblable à celle-ci (ne pas tenir compte de l'ordre des classes et sous-classes). Ne définir ici que les noms des classes et leur hiérarchie, sans tenir des autres types de contraintes pour le moment.

**Attention:** Ne pas oublier d'indiquer que les classes Plante et Animal sont disjointes en choisissant une de celles-ci et en indiquant l'autre dans Disjoint With de la vue Description.

The screenshot shows the Protégé software interface. The top menu bar includes File, Edit, View, Reasoner, Tools, Refactor, Window, and Help. The address bar shows the URL for the ontology: <http://www.iro.umontreal.ca/lapalme/AnimauxPlantes.owl>. The main window is divided into several panes:

- Class hierarchy: Plante:** A tree view showing the hierarchy of classes. The root is `owl:Thing`, which has subclasses `Feuille`, `Branche`, and `Plante`. `Plante` has subclasses `Arbre` and `PlanteSavoureuse`. `Animal` is also a subclass of `owl:Thing`, with subclasses `Carnivore` and `Herbivore`. `Carnivore` has subclasses `Lion` and `Girafe`. `Herbivore` has subclass `Girafe`.
- Annotations: Plante:** A pane for adding annotations to the selected class.
- Description: Plante:** A pane for defining the class's characteristics. It shows:
  - SubClass Of: `owl:Thing`
  - General class axioms: `owl:Thing`
  - SubClass Of (Anonymous Ancestor): `owl:Thing`
  - Instances: `owl:Thing`
  - Target for Key: `owl:Thing`
  - Disjoint With: `Animal`

The bottom status bar indicates "No Reasoner set. Select a reasoner from the Reasoner menu" and "Show Inferences" is checked.

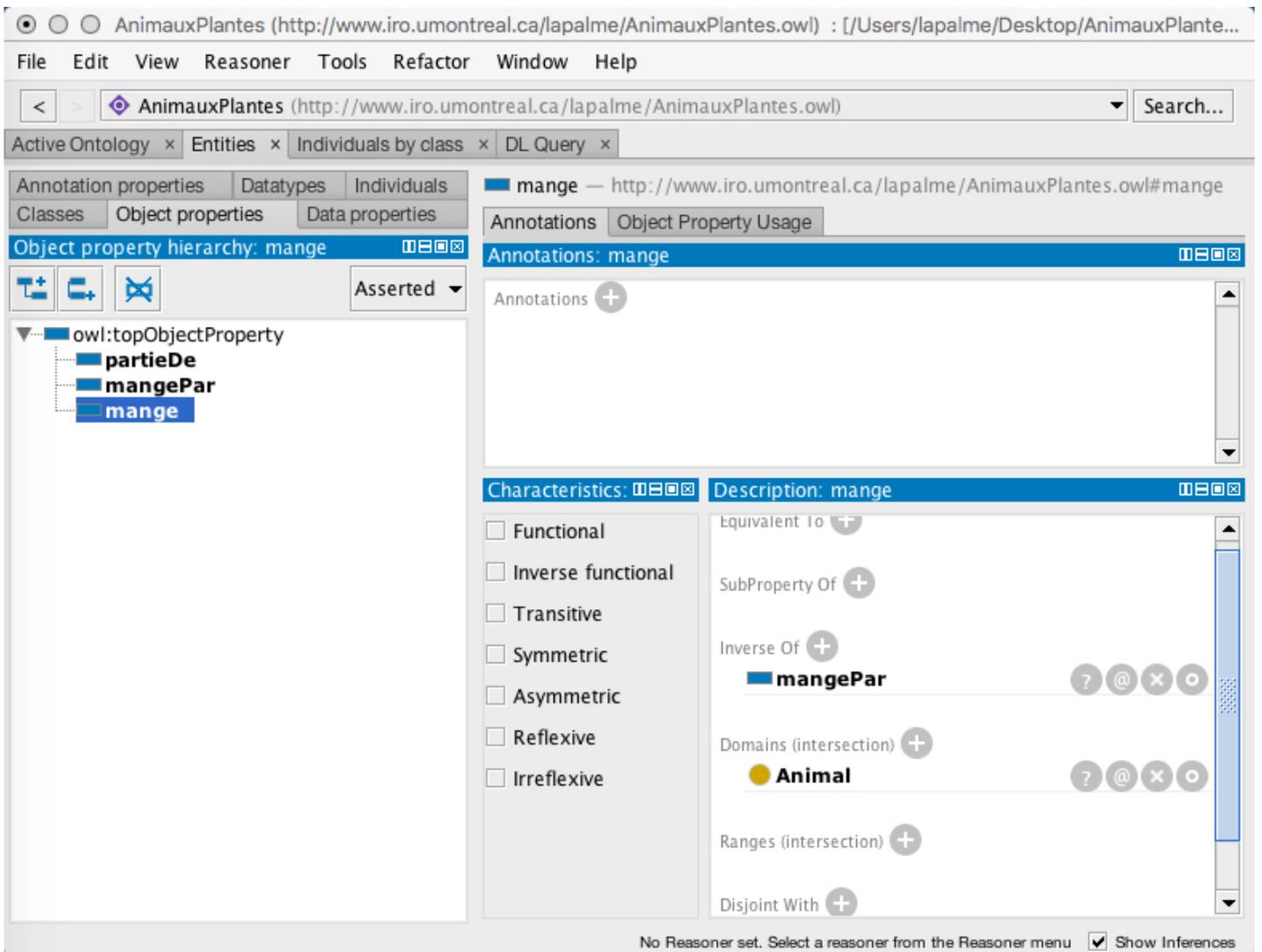
La partie droite de l'écran permet de voir des informations sur les classes et d'en changer les caractéristiques.

### Définition des propriétés

Appuyer sur l'onglet Object Properties et, en utilisant le bouton en haut à gauche, créer les 3 propriétés suivantes:

- `mange` dont le domaine est un `Animal`: choisir l'onglet Class hierarchy dans la fenêtre qui apparaît en cliquant sur le + à l'item Domains (intersections) de la vue Description
- `mangePar inverse` de `mange` en choisissant la propriété dans la fenêtre qui apparaît en cliquant sur le + à l'item inverse Of de la vue Description
- `partieDe` transitive indiquée en cochant Transitive dans la vue Characteristics

Vous devriez obtenir une figure semblable à celle-ci:



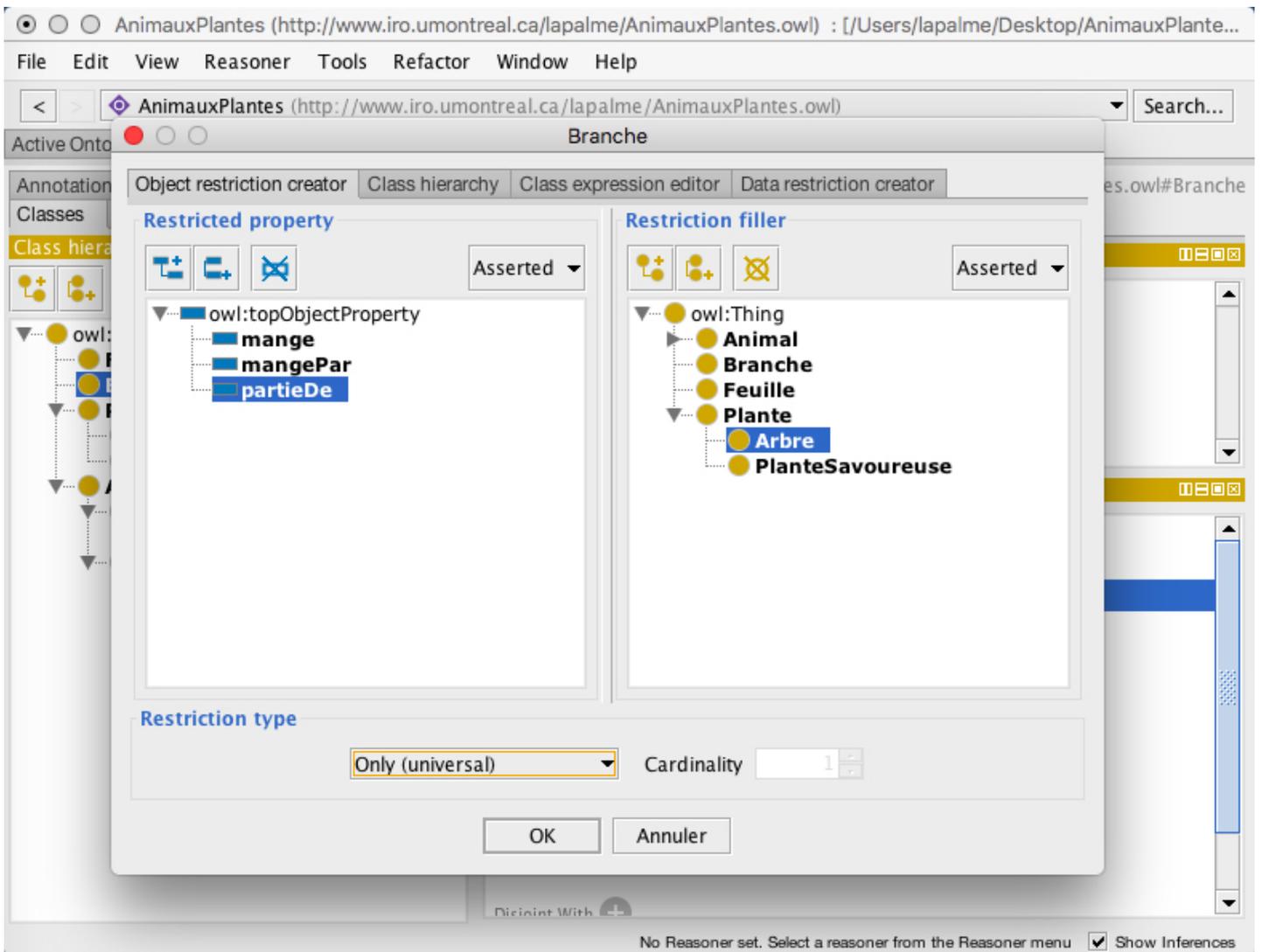
Sauver l'ontologie avec l'item Save du menu système File. Ceci créera un fichier dans le format RDF/XML qu'il peut être instructif d'ouvrir avec un éditeur de texte ou XML pour voir le code généré par Protégé.

Définition des classes complexes

Revenir dans l'onglet Classes pour compléter les descriptions de classes complexes. Pour l'énoncé IV

«Une Branche est une partie d'un Arbre»

il faut choisir la classe Branche dans la vue Class hierarchy et cliquer sur SubClass Of dans la vue Description pour faire apparaître une fenêtre permettant de définir une expression définissant la superclasse. Dans cette fenêtre, choisir l'onglet Object restriction creator et sélectionner la propriété partieDe à gauche, Arbre à droite et Only (universal) dans le menu en bas comme dans cette figure:



Ceci permet d'exprimer qu'une branche fait toujours partie d'un arbre et lorsqu'on clique OK, on obtient sous la rubrique SubClass Of de la vue Description l'expression correspondante en [syntaxe de Manchester](#): `partieDe only Arbre` [[plus de détails sur la syntaxe des expressions de classe](#)].

Effectuer une manipulation similaire pour l'énoncé V

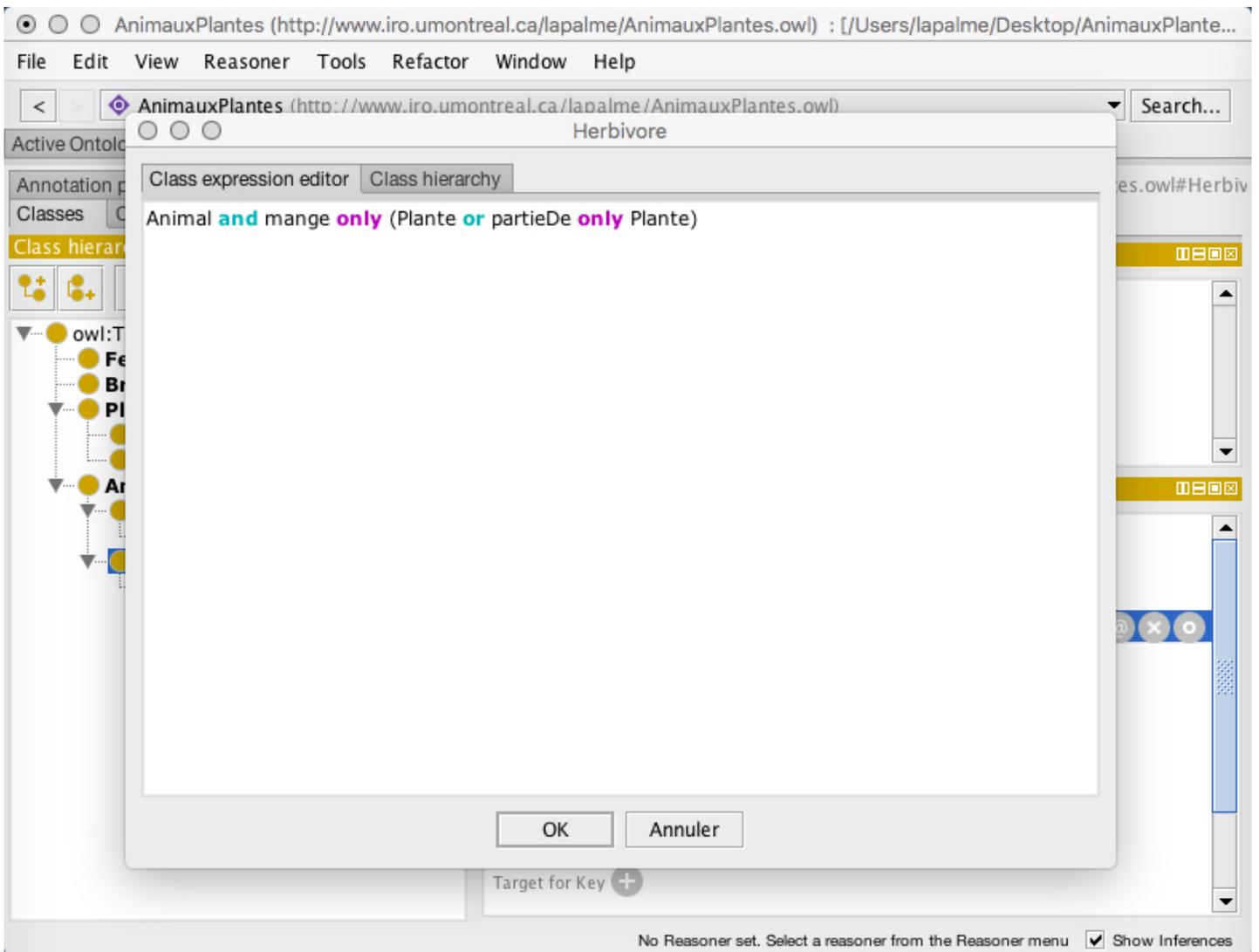
«Une Feuille est une partie d'une Branche»

de façon à obtenir l'expression `partieDe only Branche` comme SubClass Of de Feuille.

Pour l'énoncé VI

«Un Herbivore est un Animal qui ne mange qu'une Plante ou une partie d'une Plante»

choisir Herbivore dans la vue Class hierarchy et, dans la vue Description à l'item SubClass Of, cliquer sur le dernier bouton à droite (*Edit*) pour écrire `Animal and mange only (Plante or partieDe only Plante)` directement l'expression en syntaxe de Manchester dans l'onglet Class expression editor de la fenêtre qui apparaît, comme dans la figure suivante:



Au lieu de taper les noms des classes ou propriétés, on peut les glisser à partir d'autres vues et ainsi éviter des erreurs de frappe. L'éditeur effectue une vérification syntaxique de l'expression à mesure de la frappe. Il est possible d'obtenir une liste de suggestions en tapant sur CTRL-espace ou TAB en tout temps.

Pour l'énoncé VII

«Un Carnivore est un Animal qui mange aussi un Animal.»

modifier la définition de la superclasse pour indiquer qu'un carnivore peut aussi manger un autre carnivore pour obtenir la figure suivante:

The screenshot shows the Protégé interface for the ontology 'AnimauxPlantes'. The left pane displays a class hierarchy under 'owl:Thing':

- owl:Thing
  - Feuille
  - Branche
  - Plante
    - Arbre
    - PlanteSavoureuse
  - Animal
    - Carnivore** (highlighted)
    - Lion
    - Herbivore
      - Girafe

The right pane shows the details for the selected class 'Carnivore'. The 'Description' section contains the following axiom:

SubClass Of +  
 ● **Animal and mange some Animal**

Other sections include 'Annotations', 'Equivalent To', 'General class axioms', 'SubClass Of (Anonymous Ancestor)', 'Instances', and 'Target for Key'. The status bar at the bottom indicates 'No Reasoner set. Select a reasoner from the Reasoner menu' and 'Show Inferences' is checked.

Pour VIII

«Une Girafe est un Herbivore qui ne mange que des Feuilles.»

ajouter une superclasse pour obtenir la figure suivante:

The screenshot shows the Protégé OWL editor interface. The top menu bar includes File, Edit, View, Reasoner, Tools, Refactor, Window, and Help. The browser address bar shows the URL: <http://www.iro.umontreal.ca/lapalme/AnimauxPlantes.owl>. The main window displays the class hierarchy on the left and the description of the selected class, Girafe, on the right.

**Class hierarchy: Girafe**

- owl:Thing
  - Feuille
  - Branche
  - Plante
    - Arbre
    - PlanteSavoureuse
  - Animal
    - Carnivore
    - Lion
    - Herbivore
      - Girafe

**Description: Girafe**

- Annotations: Girafe
- Equivalent To: +
- SubClass Of:
  - Herbivore
  - mange only Feuille
- General class axioms: +
- SubClass Of (Anonymous Ancestor):
  - Animal and mange only (Plante or partieDe only Plante)
- Instances: +

At the bottom of the interface, it states: "No Reasoner set. Select a reasoner from the Reasoner menu" and "Show Inferences" is checked.

Pour IX

«Un Lion est un Animal qui ne mange que des Herbivores.»

Ajouter une superclasse pour obtenir la figure suivante:

The screenshot shows the Protégé OWL editor interface. The main window displays the class hierarchy for 'Lion'. The hierarchy is as follows:

- owl:Thing
  - Feuille
  - Branche
  - Plante
    - Arbre
    - PlanteSavoureuse
  - Animal
    - Carnivore
      - Lion
    - Herbivore
      - Girafe

The 'Lion' class is highlighted in blue. The right-hand pane shows the 'Description: Lion' section, which includes the following information:

- Equivalent To: (empty)
- SubClass Of:
  - Carnivore
  - mange **only** Herbivore
- General class axioms: (empty)
- SubClass Of (Anonymous Ancestor):
  - Animal **and** mange **some** Animal
- Instances: (empty)

At the bottom of the editor, there is a status bar that reads: "No Reasoner set. Select a reasoner from the Reasoner menu" and a checked checkbox for "Show Inferences".

Pour X

«Une PlanteSavoureuse est une Plante qui est mangée par un Herbivore et aussi par un Carnivore»

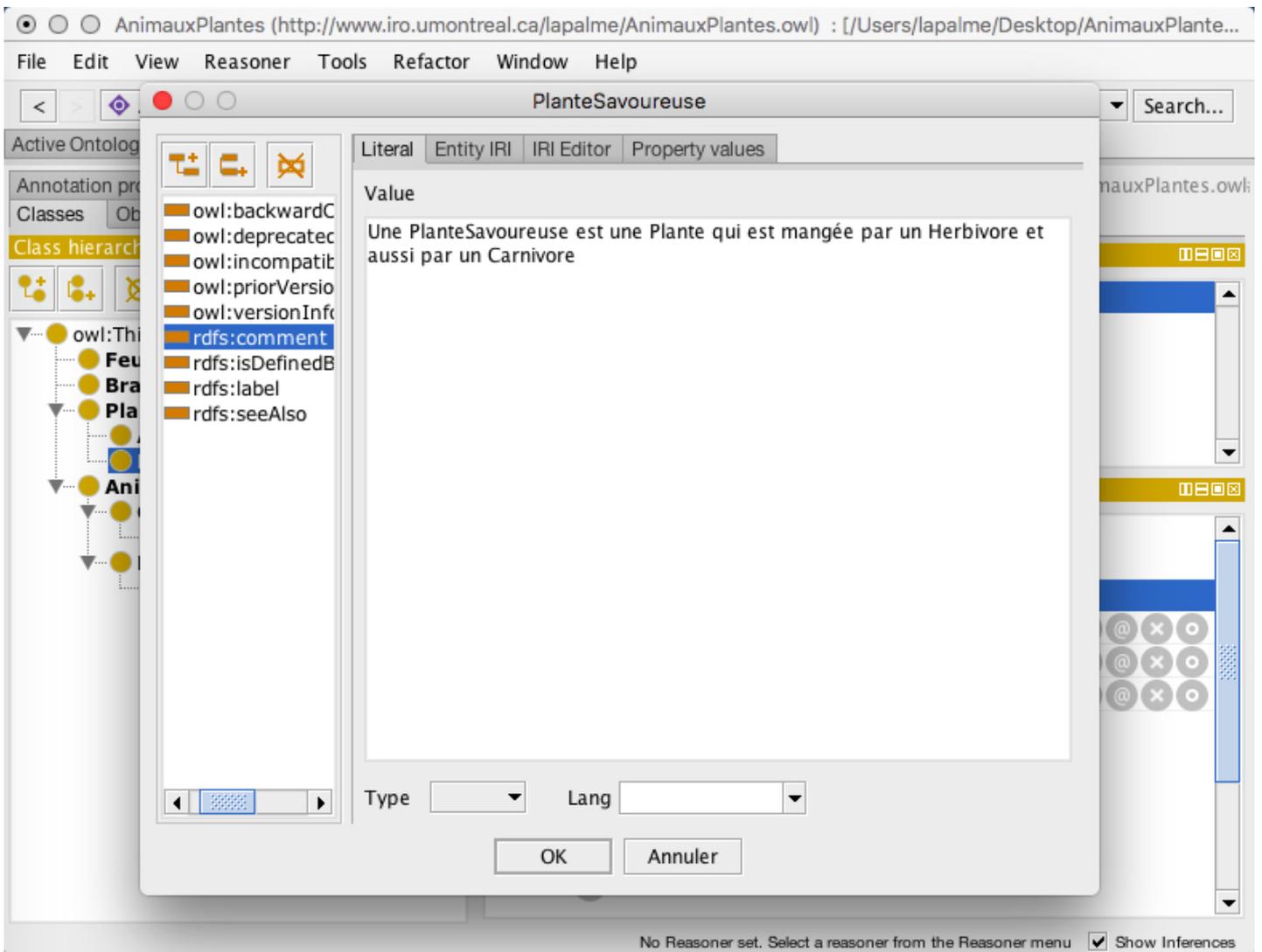
ajouter deux superclasses à PlanteSavoureuse pour obtenir la figure suivante:

The screenshot shows the Protégé OWL editor interface. The top menu bar includes File, Edit, View, Reasoner, Tools, Refactor, Window, and Help. The address bar shows the URL for the ontology: <http://www.iro.umontreal.ca/lapalme/AnimauxPlantes.owl>. The main window is divided into several panes:

- Class hierarchy:** A tree view showing the ontology structure. The classes are: owl:Thing, Feuille, Branche, Plante, Arbre, PlanteSavoureuse (highlighted), Animal, Carnivore, Lion, Herbivore, and Girafe.
- Annotations:** A pane for adding annotations to the selected class (PlanteSavoureuse). It currently shows no annotations.
- Description:** A pane showing the class description for PlanteSavoureuse. It includes:
  - Equivalent To: (empty)
  - SubClass Of:
    - mangePar some Carnivore
    - mangePar some Herbivore
    - Plante
  - General class axioms: (empty)
  - SubClass Of (Anonymous Ancestor): (empty)
  - Instances: (empty)

At the bottom of the interface, there is a status bar that reads: "No Reasoner set. Select a reasoner from the Reasoner menu" and a checkbox for "Show Inferences" which is checked.

Lorsque les expressions de classe deviennent complexes, il est utile d'ajouter une annotation de type comment en appuyant sur le signe + à côté de Annotations dans la vue Annotations ce qui fait apparaître une fenêtre semblable à la suivante:



Sauvez votre ontologie dans le format RDF/XML. Il peut être instructif d'ouvrir le fichier que vous venez de sauver avec un éditeur de texte ou XML pour voir le code généré par Protégé.

Il est possible de voir le rendu RDF/XML dans l'onglet Active Ontology la vue RDF/XML Rendering qui affiche le contenu de l'ontologie dans cette syntaxe. Si cette vue n'apparaît pas, choisissez l'item de menu système Window>Views>Ontology Views>RDF/XML Rendering et placer le *point* qui apparaît au milieu de la vue du bas et cliquer pour y ajouter cette nouvelle vue. Cette vue est générée à partir des informations conservées par Protégé, elle **ne peut pas** servir à modifier l'ontologie *à la main*.

Il peut être intéressant d'explorer la hiérarchie à l'onglet global DL Query. L'interrogation de l'ontologie est effectuée avec un langage spécial inspiré de la syntaxe de Manchester décrit [dans ce document](#).

Il est aussi possible d'avoir une visualisation de l'ontologie avec la vue OntoGraf disponible à Window>Views>Class views>Ontograf.

#### Ajout d'individus

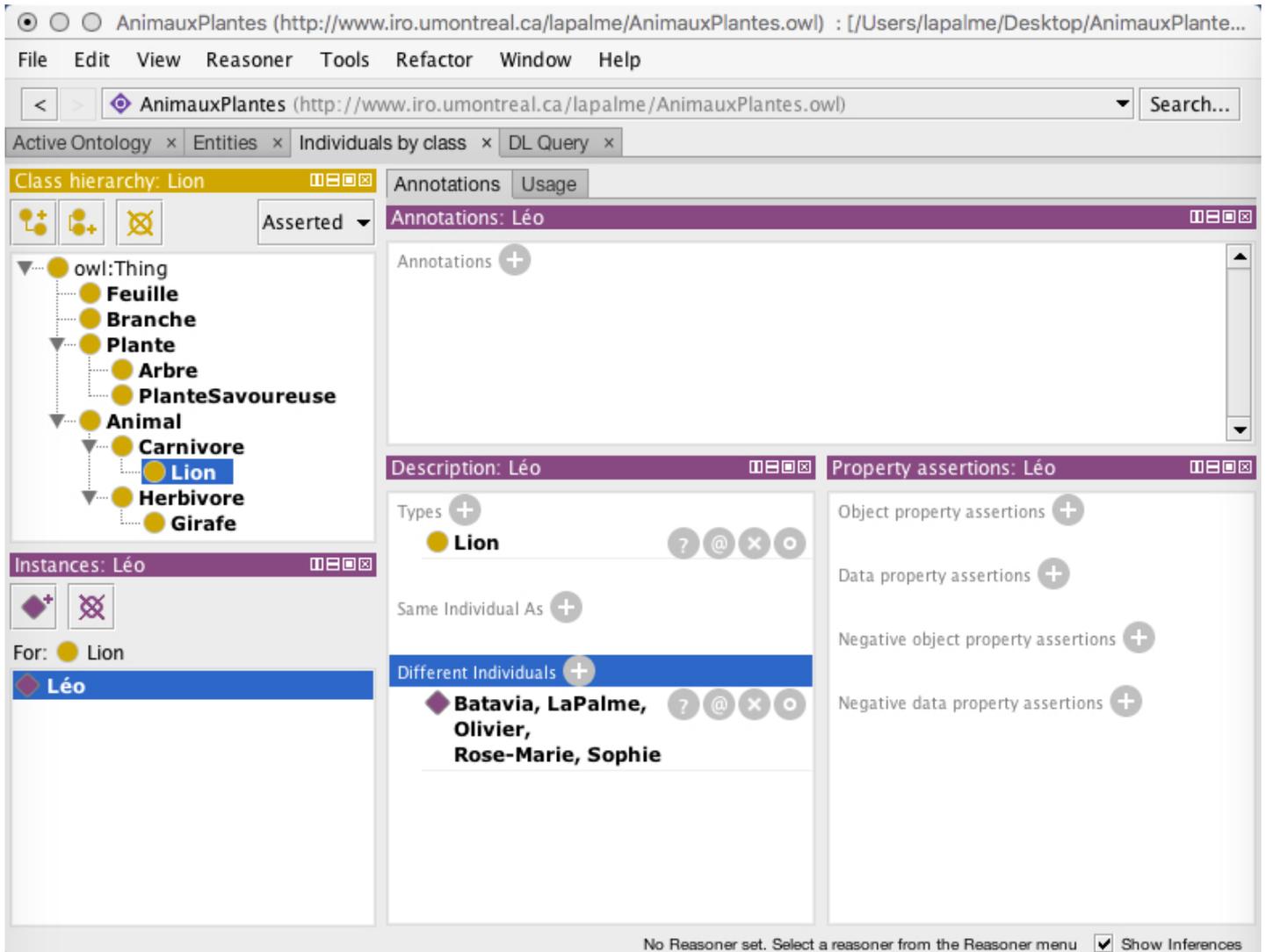
Ajouter des instances des classes dans l'onglet Individuals by class en sélectionnant une classe dans la vue en haut à gauche et en ajoutant un individu comme membre dans la vue Instances.

Ajouter les individus suivants:

- Léo le Lion
- Sophie la Girafe
- Olivier l'Arbre
- Rose-Marie la PlanteSavoureuse
- Batavia la Feuille
- LaPalme la Branche.

Pour indiquer que ces individus sont tous différents, il suffit d'en choisir un et d'appuyer sur le + à côté de Different individuals dans la vue Description et de sélectionner les autres individus dans la fenêtre qui apparaîtra. Tous les individus seront alors considérés comme distincts comme on peut le constater en les sélectionnant dans la vue Individuals.

Une figure équivalente à la suivante devrait alors apparaître:



### Vérification de la cohérence de l'ontologie

Le grand intérêt de l'utilisation de Protégé est la possibilité de vérifier si l'ontologie créée ne contient pas des définitions contradictoires. Avec une ontologie aussi simpliste que celle que nous avons construite dans ce document, il est facile de vérifier à la main la cohérence du tout mais ceci est pratiquement impossible pour des cas plus complexes. Nous allons illustrer le processus de vérification en créant une classe qui serait à la fois une plante et un animal.

Ajouter une nouvelle classe `AnimalPlante` sous `owl:Thing` et définir à `SubClass Of` l'expression `Animal and Plante`. La classe `AnimalPlante` sera alors placée à la fois sous `Animal` et `Plante` comme dans la figure suivante:

The screenshot shows the Protégé software interface for an ontology named 'AnimauxPlantes'. The main window displays the class hierarchy for 'AnimalPlante'. The hierarchy is as follows:

- owl:Thing
  - Feuille
  - Branche
  - Plante
    - AnimalPlante** (highlighted in blue)
    - Arbre
    - PlanteSavoureuse
  - Animal
    - AnimalPlante
    - Carnivore
      - Lion
    - Herbivore
      - Girafe

The right-hand pane shows the 'Description: AnimalPlante' section, which includes a 'SubClass Of' relationship with 'Animal and Plante'. The status bar at the bottom indicates 'No Reasoner set. Select a reasoner from the Reasoner menu' and 'Show Inferences' is checked.

Dans le menu Reasoner, sélectionner HerMiT, s'il n'y a pas déjà un crochet dans l'item de menu; sélectionner ensuite Start Reasoner. Dans la vue Class hierarchy en choisissant l'item de menu Inferred devraient apparaître les classes inconsistantes en rouge, comme dans cette figure:

The screenshot shows the Protégé interface for the ontology 'AnimauxPlantes'. The left pane displays the 'Class hierarchy: AnimalPlante' with a tree structure where 'owl:Nothing' is highlighted in red, indicating inconsistency. The right pane shows the 'Description: AnimalPlante' with the following details:

- Equivalent To:** owl:Nothing
- SubClass Of:** Animal and Plante
- General class axioms:** (empty)
- SubClass Of (Anonymous Ancestor):** (empty)
- Instances:** (empty)

At the bottom right, the status bar indicates 'Reasoner active' and 'Show Inferences' is checked.

La classe `AnimalPlante` apparaît comme inconsistante car nous avons déclaré ([à l'énoncé II](#)) que les classes `Animal` et `Plante` doivent être disjointes. Cette explication est d'ailleurs fournie par Protégé si on clique sur le point d'interrogation à côté de `owl:Nothing` dans la vue `Description`.

Pour revenir à une ontologie consistante, il suffit d'éliminer la définition des superclasses de `AnimalPlante`. Il faut choisir l'item de menu `Asserted`, choisir la classe à éliminer et cliquer sur le troisième bouton. Pour s'assurer que tout est maintenant cohérent, choisir à nouveau `Synchronize reasoner` dans le menu système `Reasoner`. Il ne devrait plus rien apparaître sous l'item `Nothing` en rouge dans la vue `Class hierarchy` lorsqu'on choisit l'item `Inferred`.  
Pour aller plus loin...

Ce petit exemple ne fait pas ressortir toutes les possibilités de Protégé, en particulier nous n'avons pas expliqué comment modifier les vues et leur position, ou la création de plusieurs classes en spécifiant leur hiérarchie, ni toutes les subtilités de la modélisation d'ontologie.

La prochaine étape devrait être maintenant de lire et d'expérimenter avec [A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools](#) qui développe en détails l'exemple du *domaine de la pizza* qui est aussi exploré [dans cet autre tutoriel](#). Ces tutoriels utilisent une version antérieure de l'interface de Protégé, mais les principes de base restent les mêmes.

