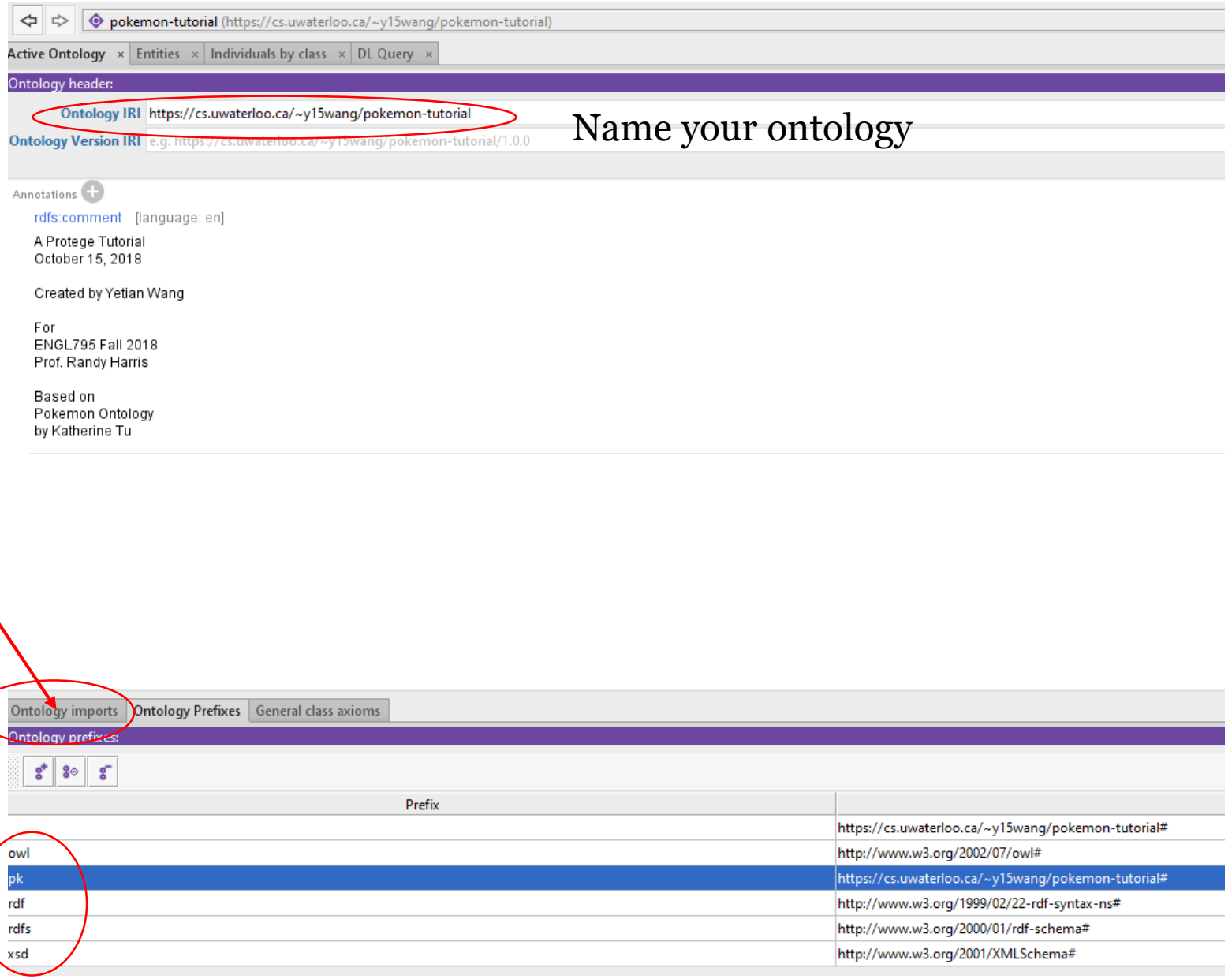
A Protégé Tutorial for ENGL 795

Yetian Wang

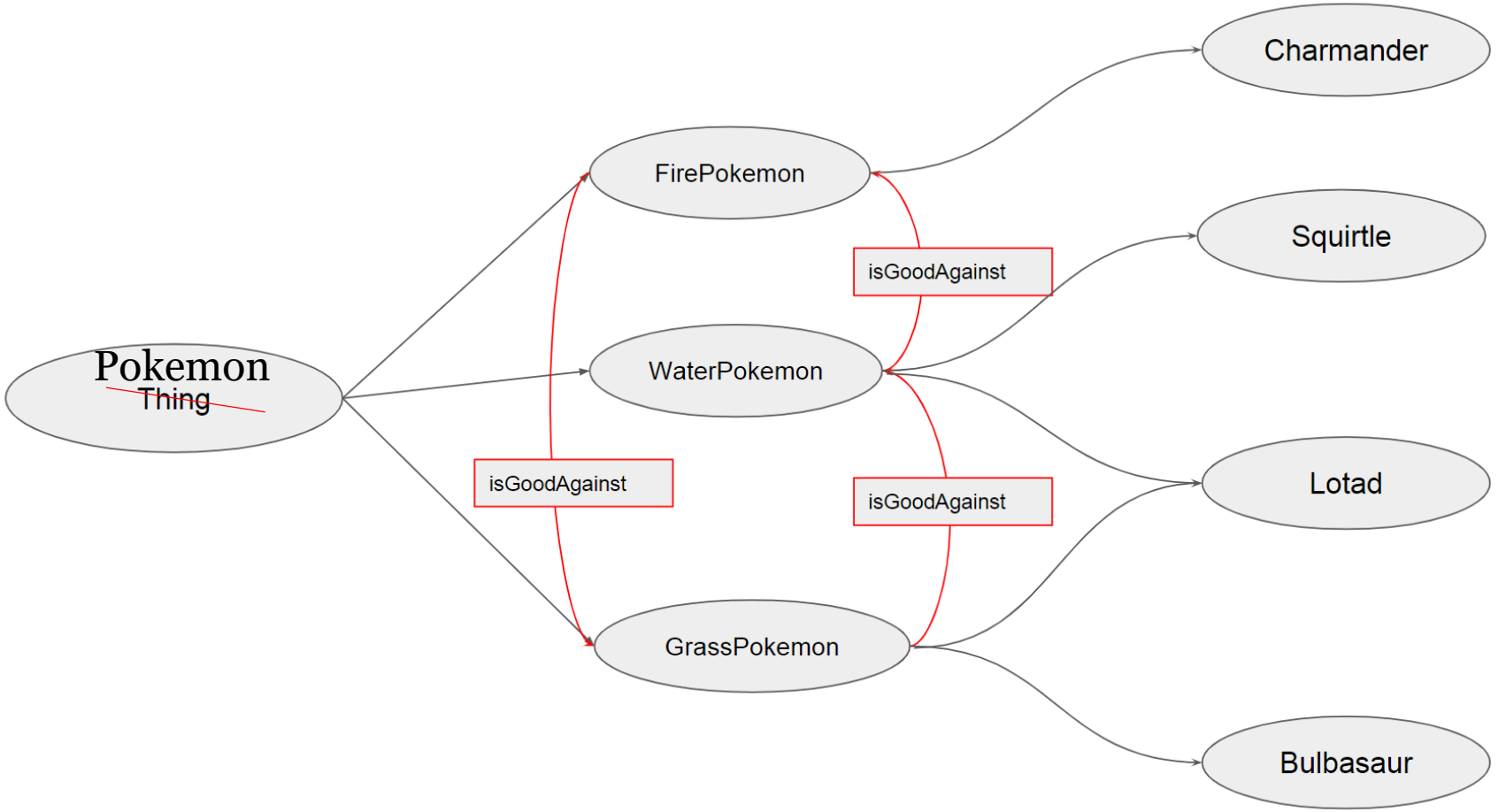
October 15, 2018

Active Ontology Tab

- IRI
- Annotation
- Import Existing Ontologies
- Prefix



Pokemon Ontology

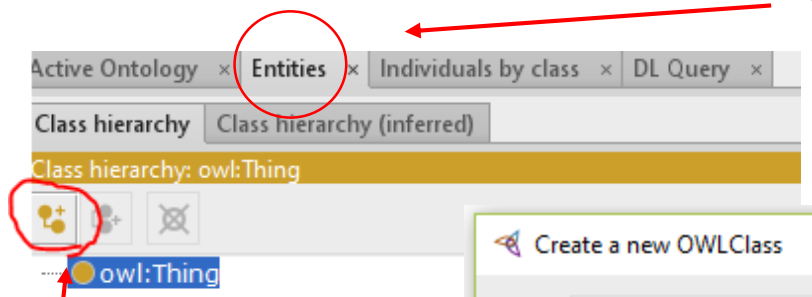


Created by Katherine Tu

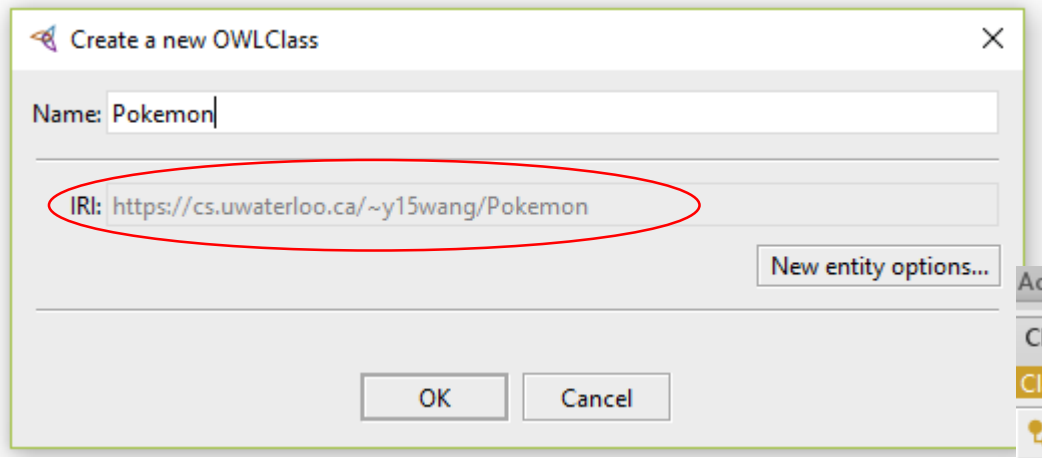
UNIVERSITY OF WATERLOO

Creating a class

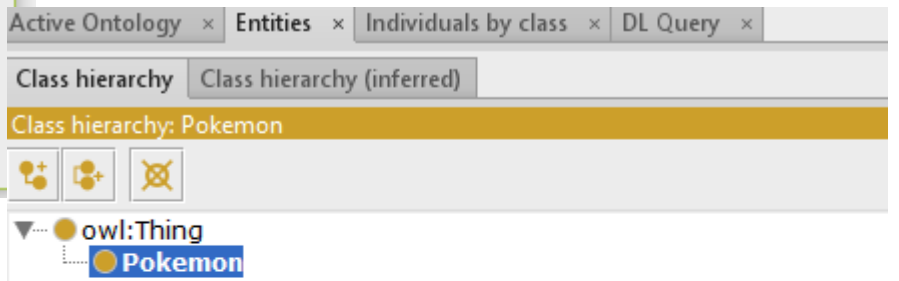
1 Go to Entities Tab



2 Click to add a class



3 Give a name. Note that the 'full name' of this class is the ontology IRI+classname

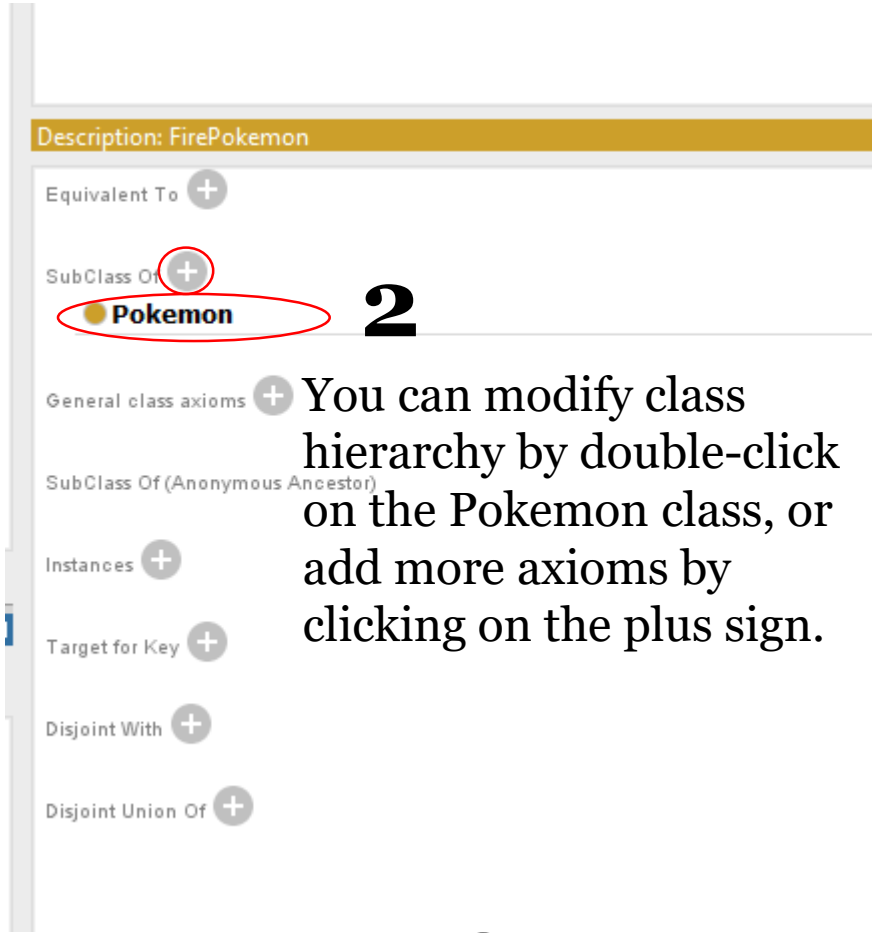


4 Class created

Creating a Subclass

1

Select Pokemon class then create a FirePokemon class

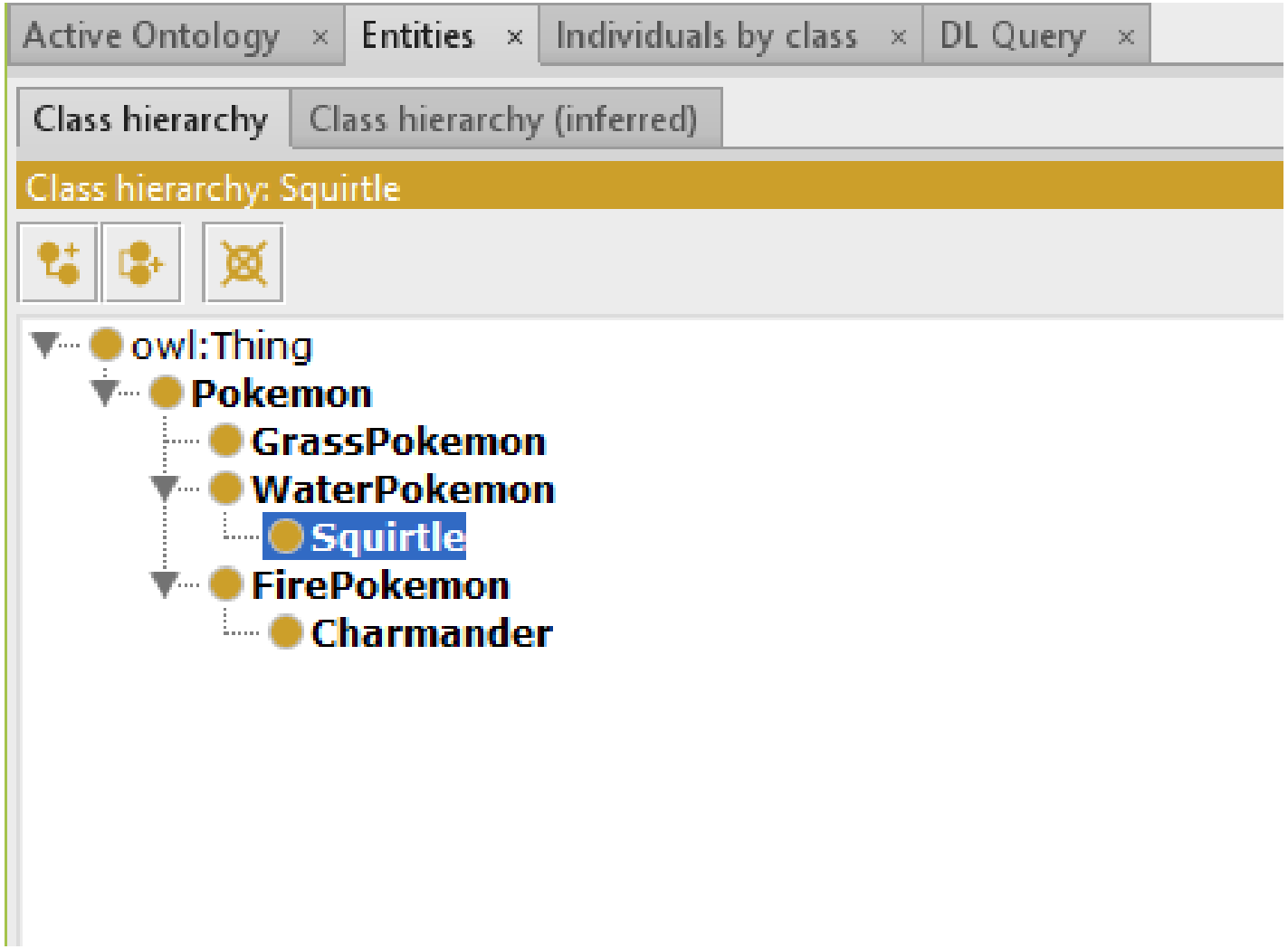


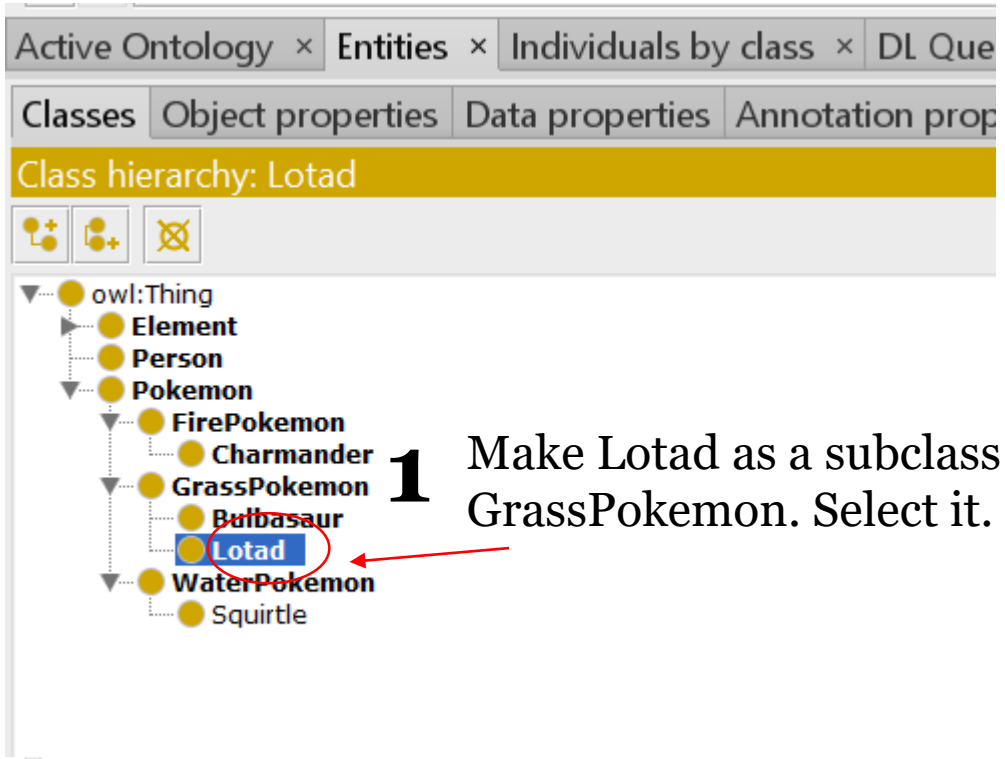
2

You can modify class hierarchy by double-click on the Pokemon class, or add more axioms by clicking on the plus sign.

UNIVERSITY OF WATERLOO

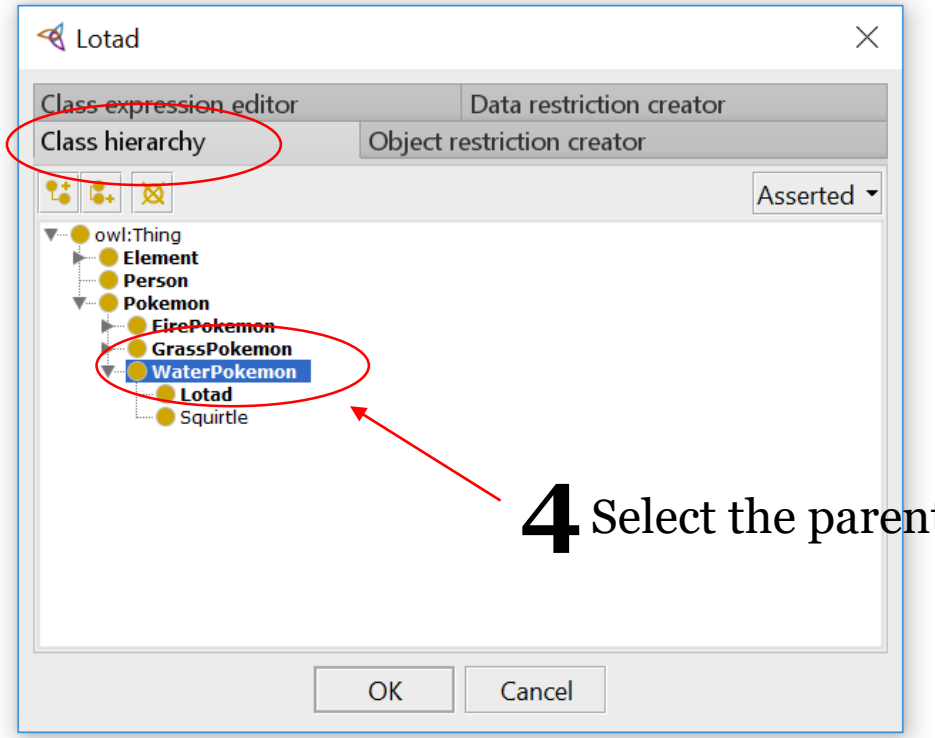
More Subclasses



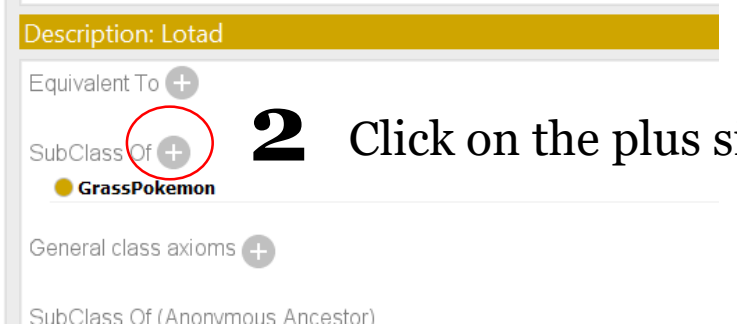


1 Make Lotad as a subclass of GrassPokemon. Select it.

3 Go to Class hierarchy tab



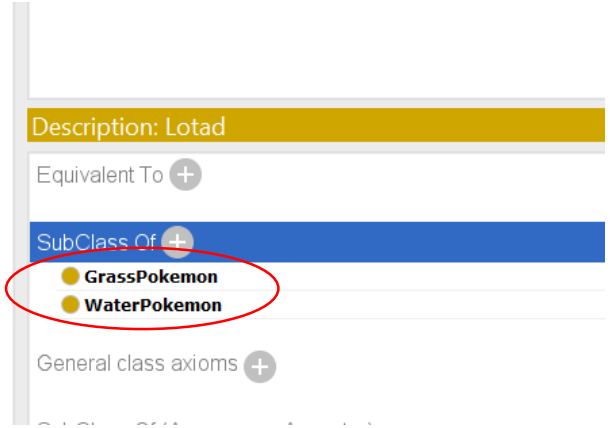
4 Select the parent class

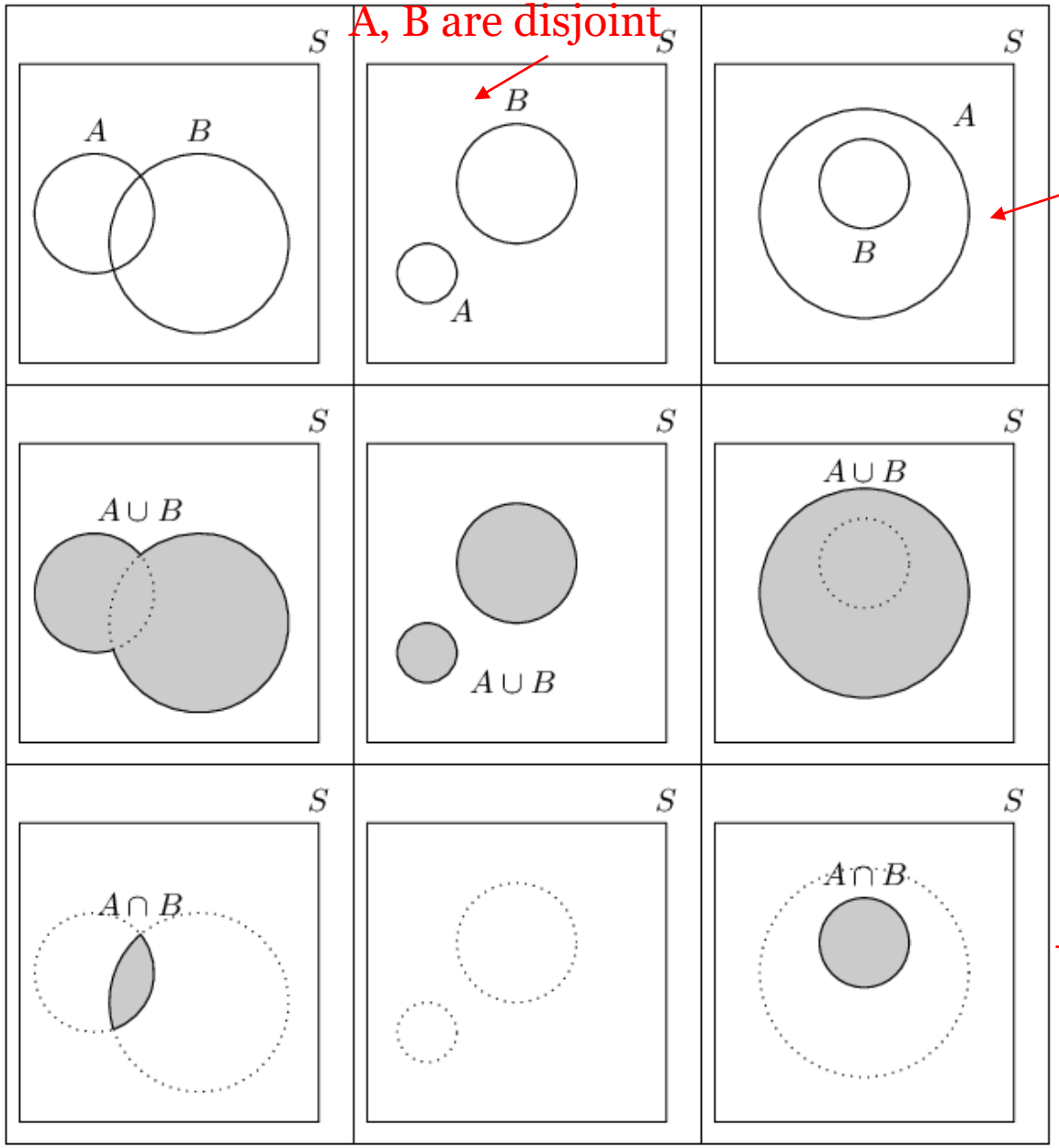


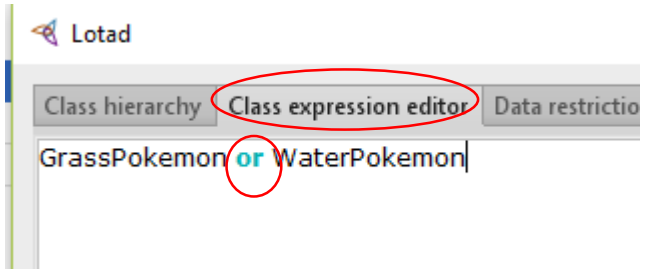
2 Click on the plus sign

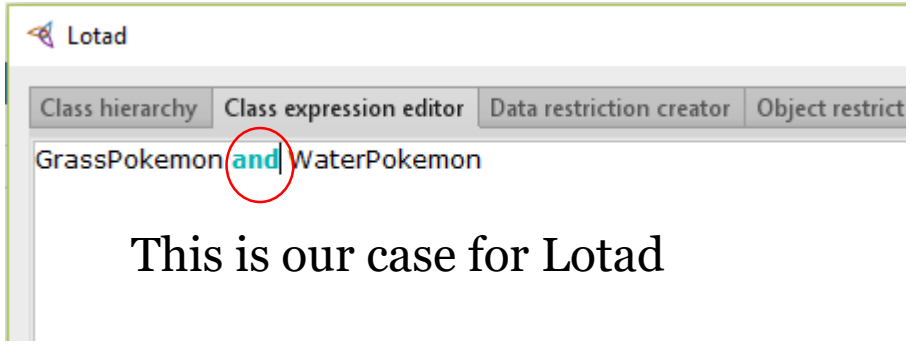
5 You should see this. This is equivalent to

'GrassPokemon **and** WaterPokemon'









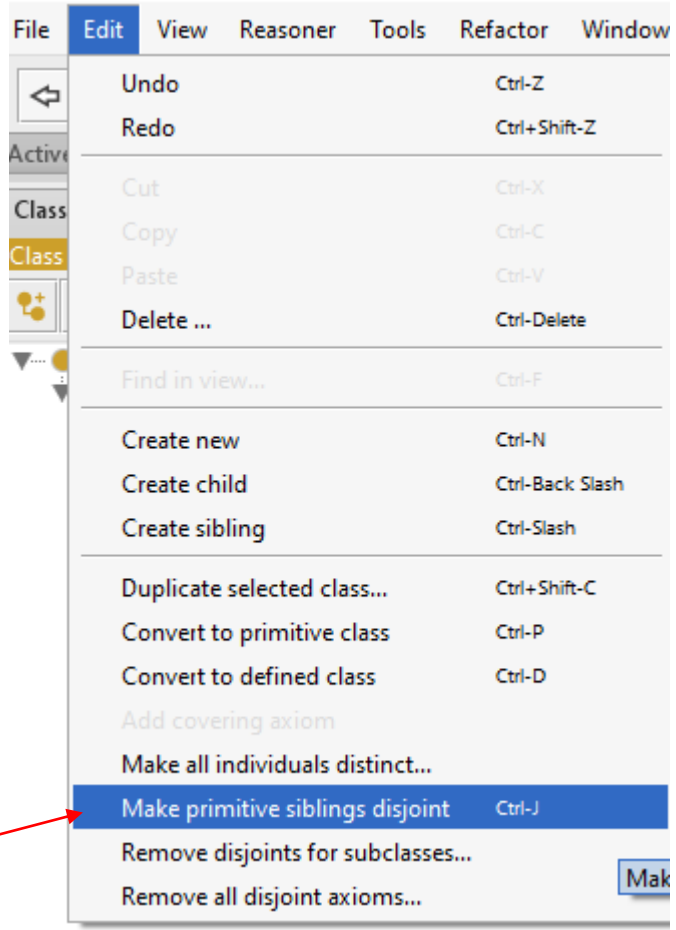
This is our case for Lotad

UNIVERSITY OF WATERLOO

Disjoint Classes

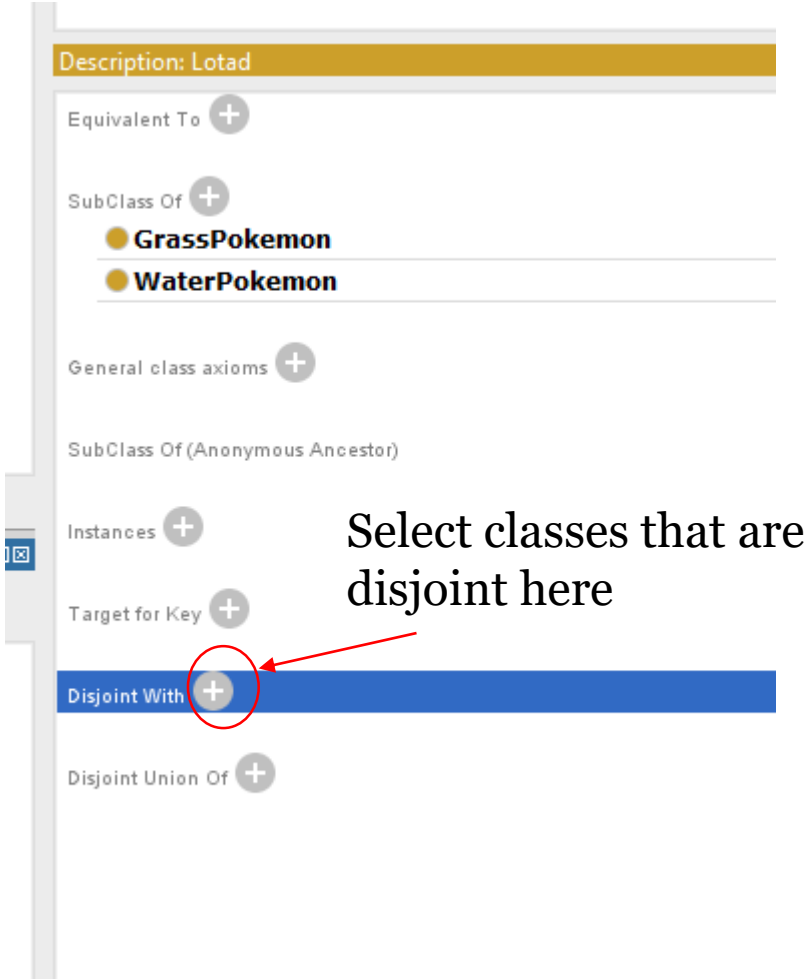
- Class names are different ≠ classes are different
- You need to specify this in your ontology
- But be careful with inconsistencies

Option 1 – may not appear in newer versions



Select a class. This option makes all sibling classes on that level disjoint

Option 2



Select classes that are disjoint here

UNIVERSITY OF WATERLOO

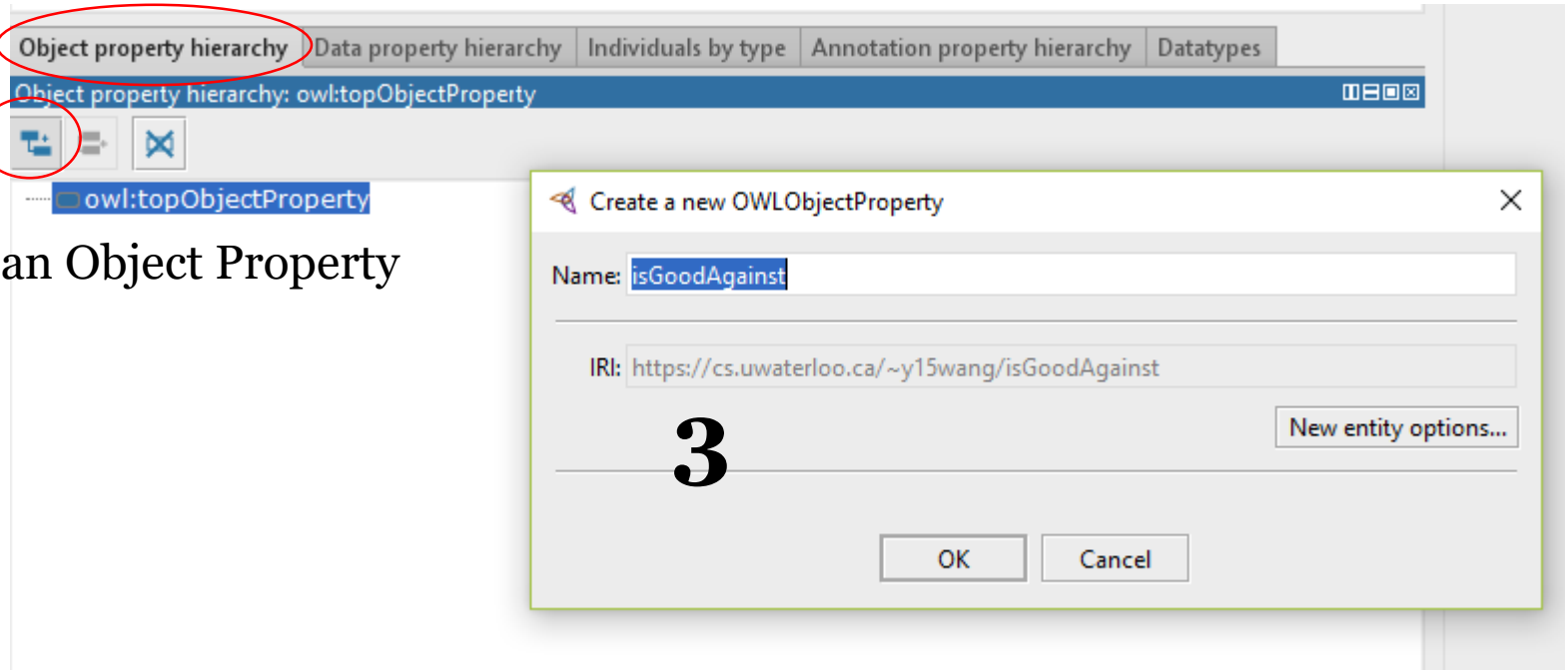
Properties

- Object Properties
  - Properties that relate instances of a **class** to instances another **class**
- Data Properties
  - Properties that relate instances of a **class** to **values** (e.g. string and numerical values)

Create a Property

1 Go to Object Properties Tab

2 Add an Object Property



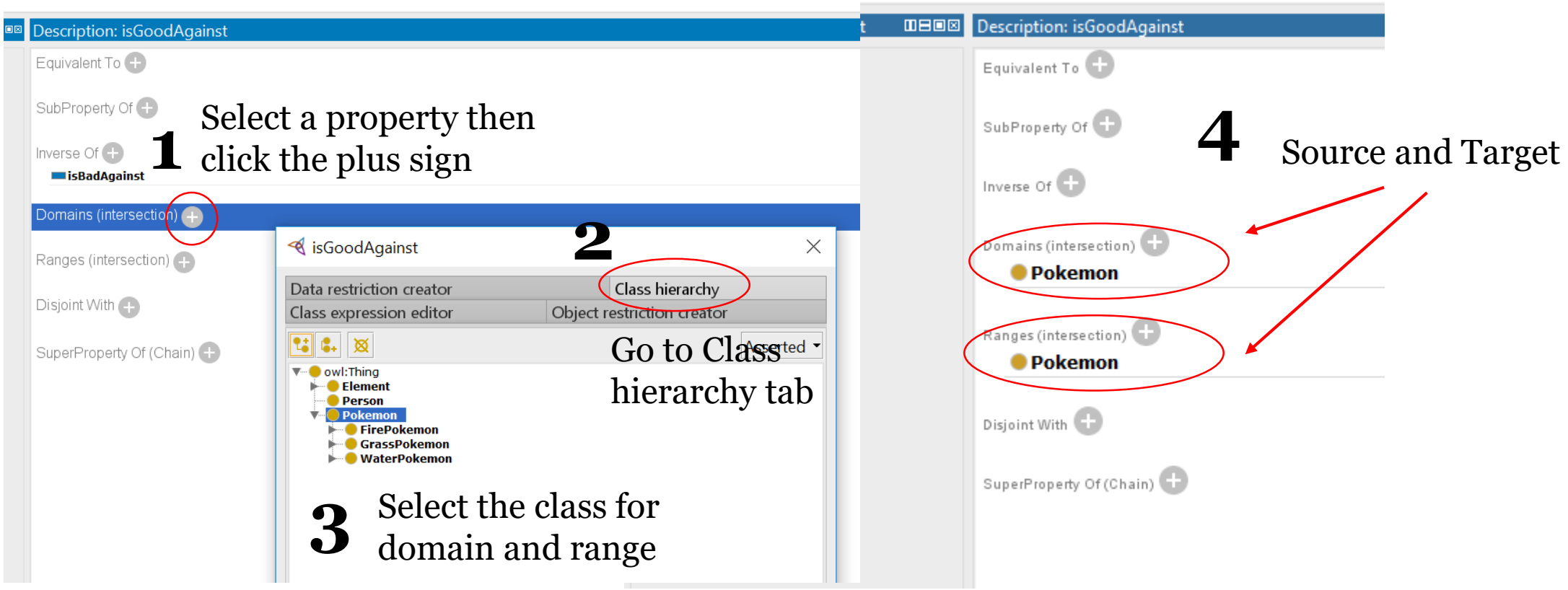
UNIVERSITY OF WATERLOO

Domain and Range

- Domain
  - the subjects of such property statements must belong to the class extension of the indicated class description [*]
- Range
  - that the objects/values of this property must belong to the class extension of the class description or to data values in the specified data range [*].
- Example
  - suppose we have a property '*takesCourse*' and an axiom (**Students** *takesCourse* **Course**)
  - The domain is the class **Student**
  - The range is the class **Course**

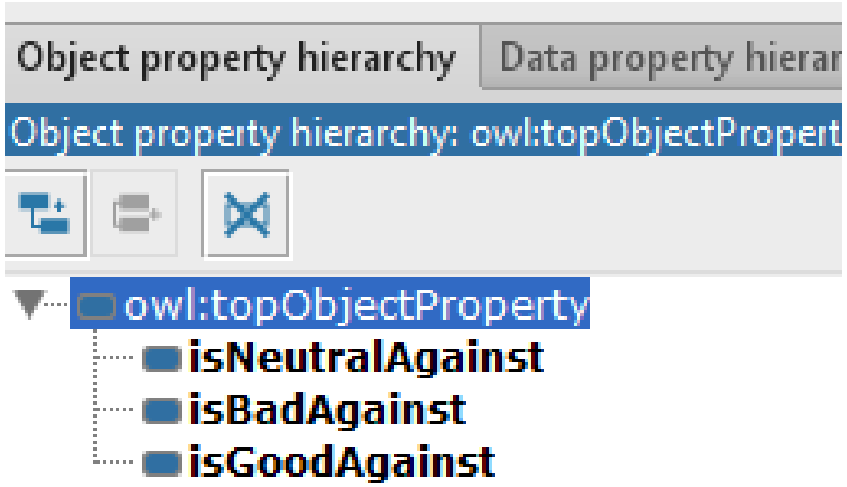
[*] https://www.w3.org/TR/owl-ref/#domain-def

Specify Domain and Range



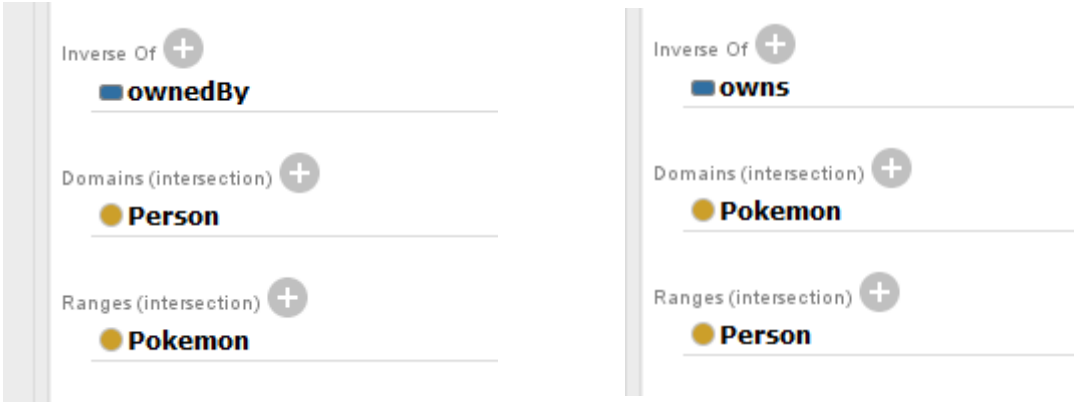
UNIVERSITY OF WATERLOO

Create the remaining properties and specify their domain and range



Sub-properties and Inverse Properties

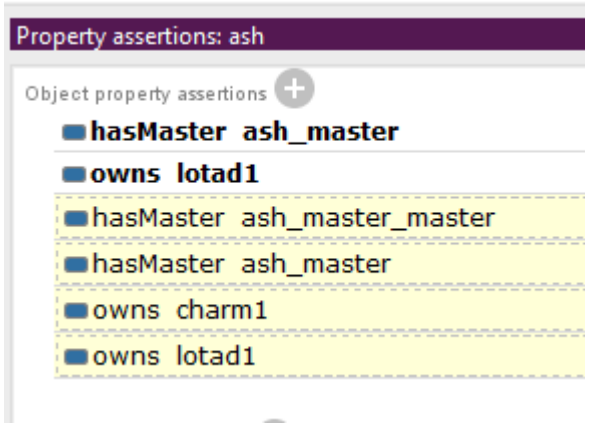
- Sub-properties
  - hasMother and hasFather are sub-properties of hasParent
- Inverse
  - If (A, p, B) then (B, inverse of p, A)
  - E.g. hasParent, hasChild

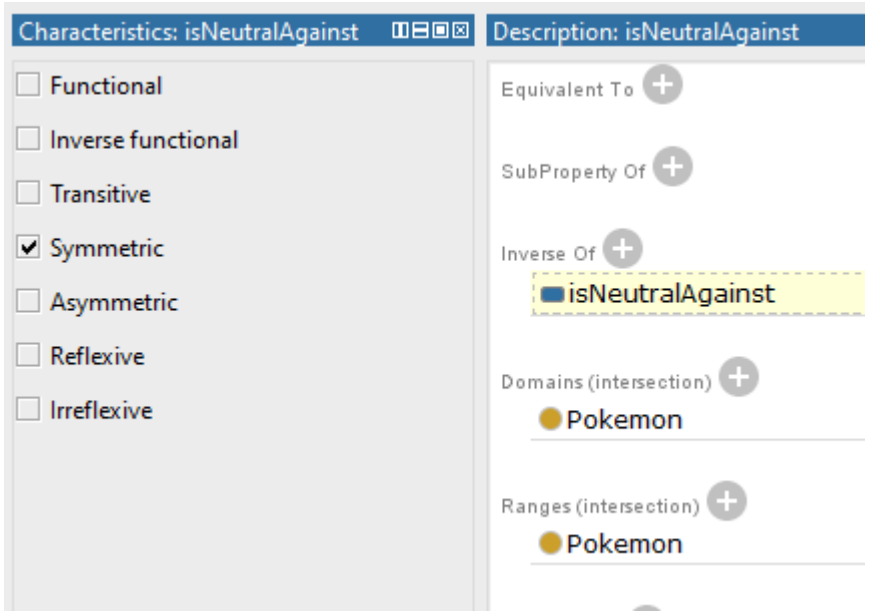


UNIVERSITY OF WATERLOO

Property Characteristics

- Functional
  - One individual only
  - E.g. hasHusband
- Transitive
  - If (A, p, B) and (B p C) then (A p C)
  - E.g. has Ancestor, hasMaster
- Symmetric
  - If (A, p, B) then (B, p, A)
  - E.g. marriedTo
- Reflexive
  - relates everything to itself





UNIVERSITY OF WATERLOO

Specify Class Axioms

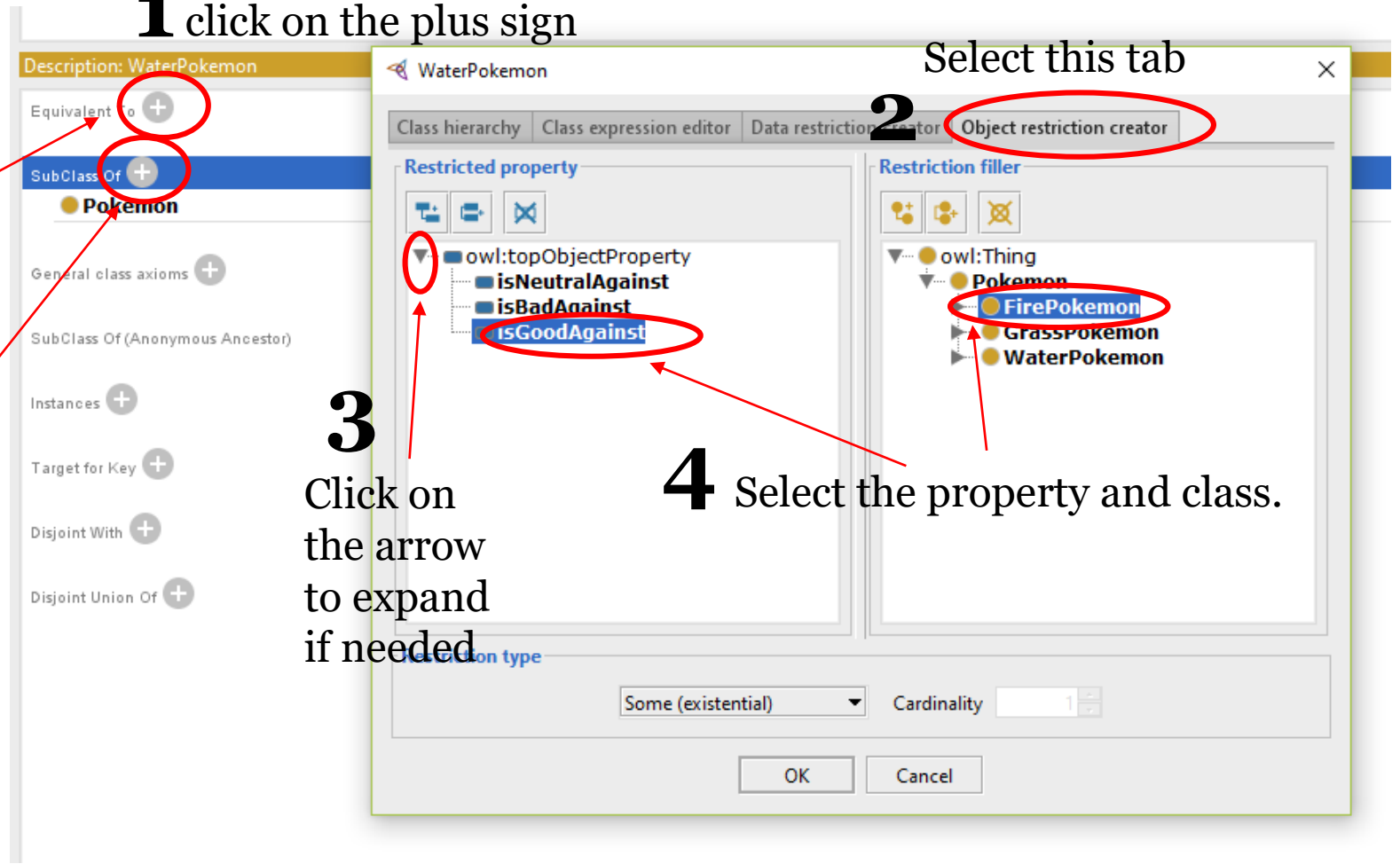
WaterPokenmon is good against FirePokemon

Necessary and Sufficient:
It's a WaterPokenmon iff it is good against FirePokemon'

e.g. a female who has at least 1 child is equivalent to a mother

Necessary:
WaterPokenmon is a subclass of 'things that are good against FirePokemon'

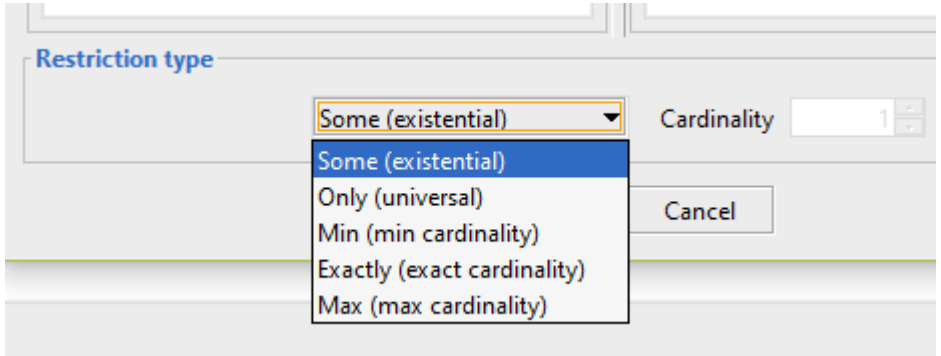
e.g. a female who has at least 1 child (i.e. mother) is a parent. But not all parents are females (e.g. fathers)



UNIVERSITY OF WATERLOO

Restriction Type

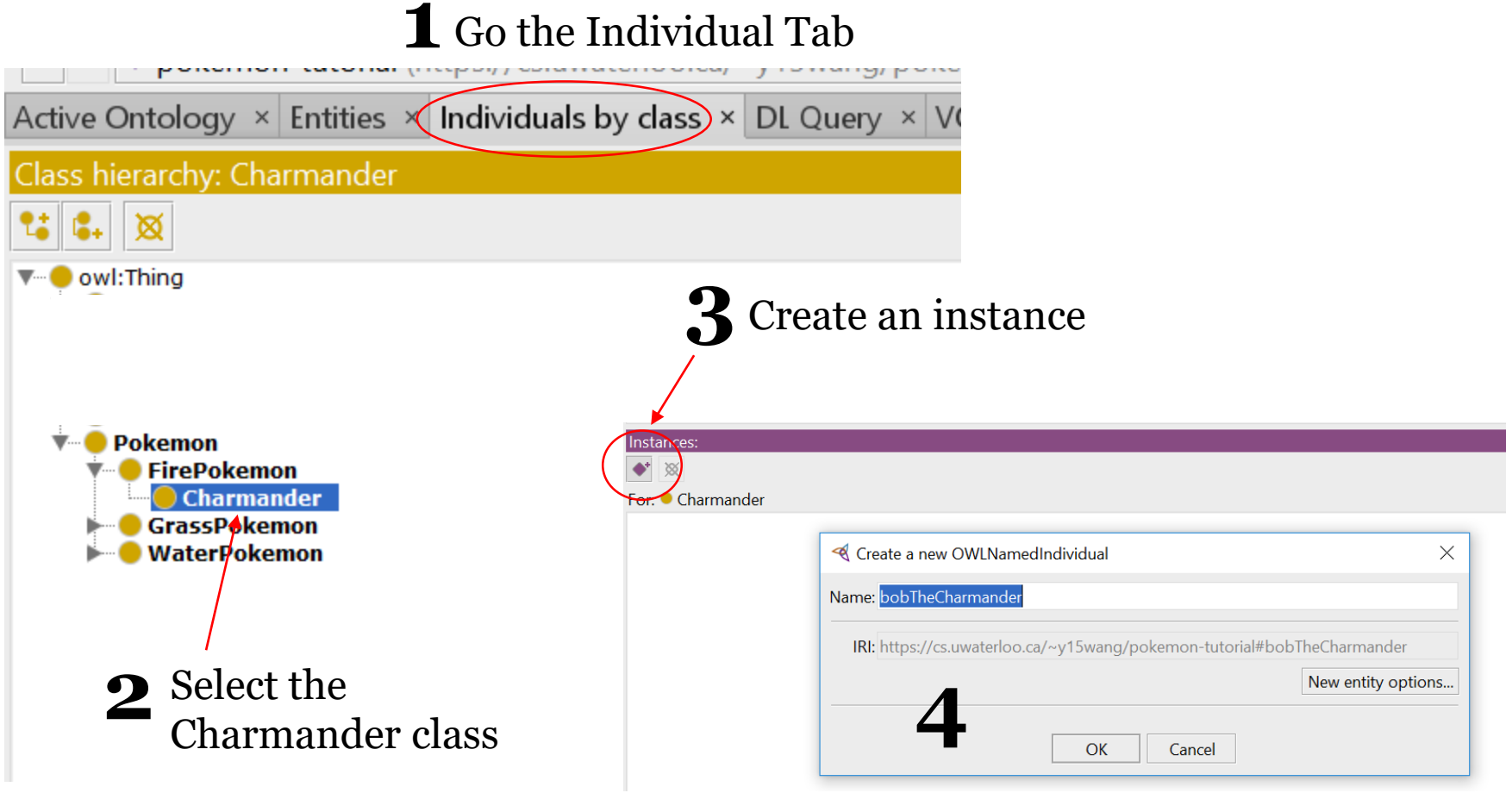
- Some
  - There exists a FirePokemon that WaterPokemon isGoodAgainst
- Only
  - WaterPokemon is only good against FirePokemon
- Cardinality
  - There is <min, exactly, max> ***n*** FirePokemon that a WaterPokemon isGoodAgainst



UNIVERSITY OF WATERLOO

Individual

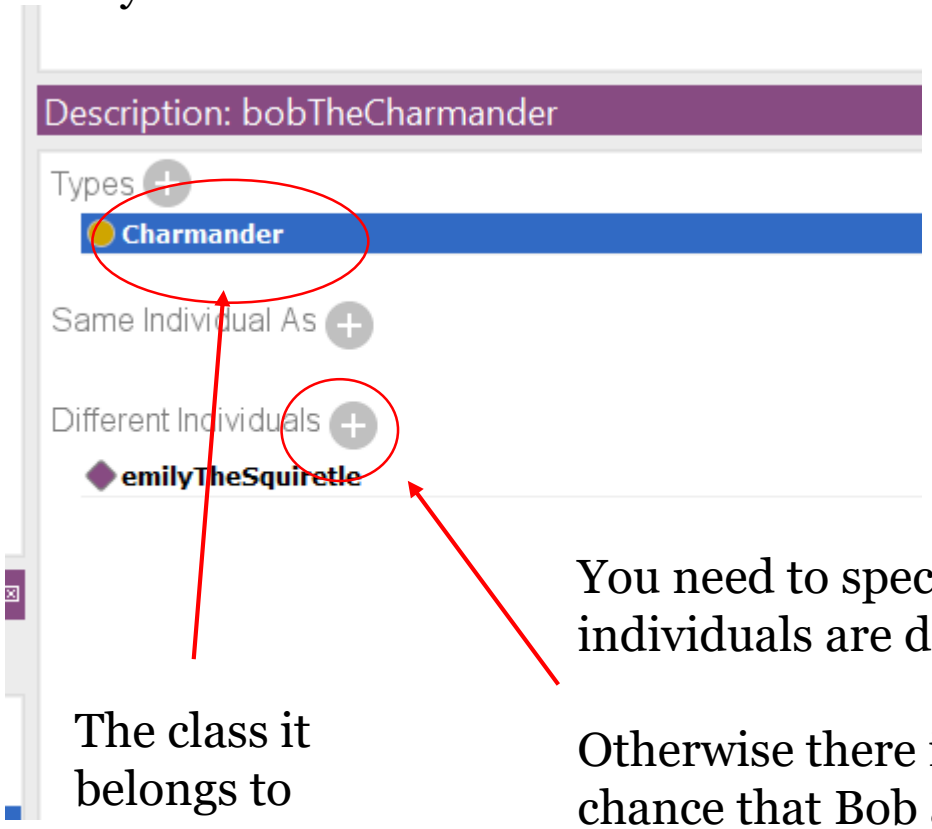
- Instances of classes.
- The Data
- E.g. *bobTheCharmander* is an instance of the class **Charmander**
- Then create *emilyTheSquirtle* as an instance of **Squirtle**



UNIVERSITY OF WATERLOO

Specify Individual Description

1 Select the instance *bobTheCharmander* then you should see this

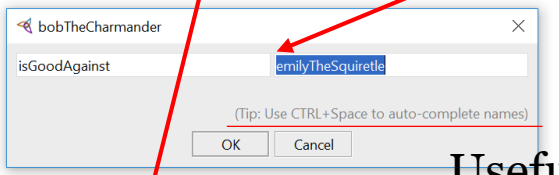


The class it belongs to

You need to specify if individuals are different.

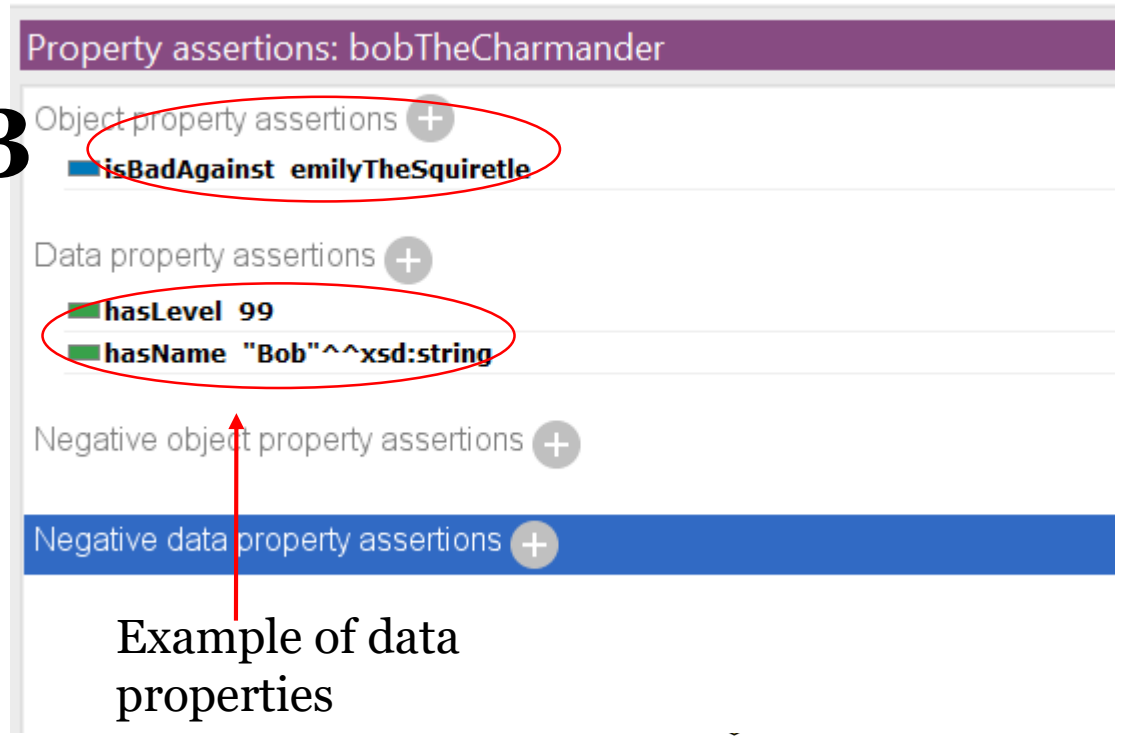
Otherwise there is a chance that Bob and Emily are the same Pokemon

2 Add an object property assertion then type the property and individual names



Useful tip

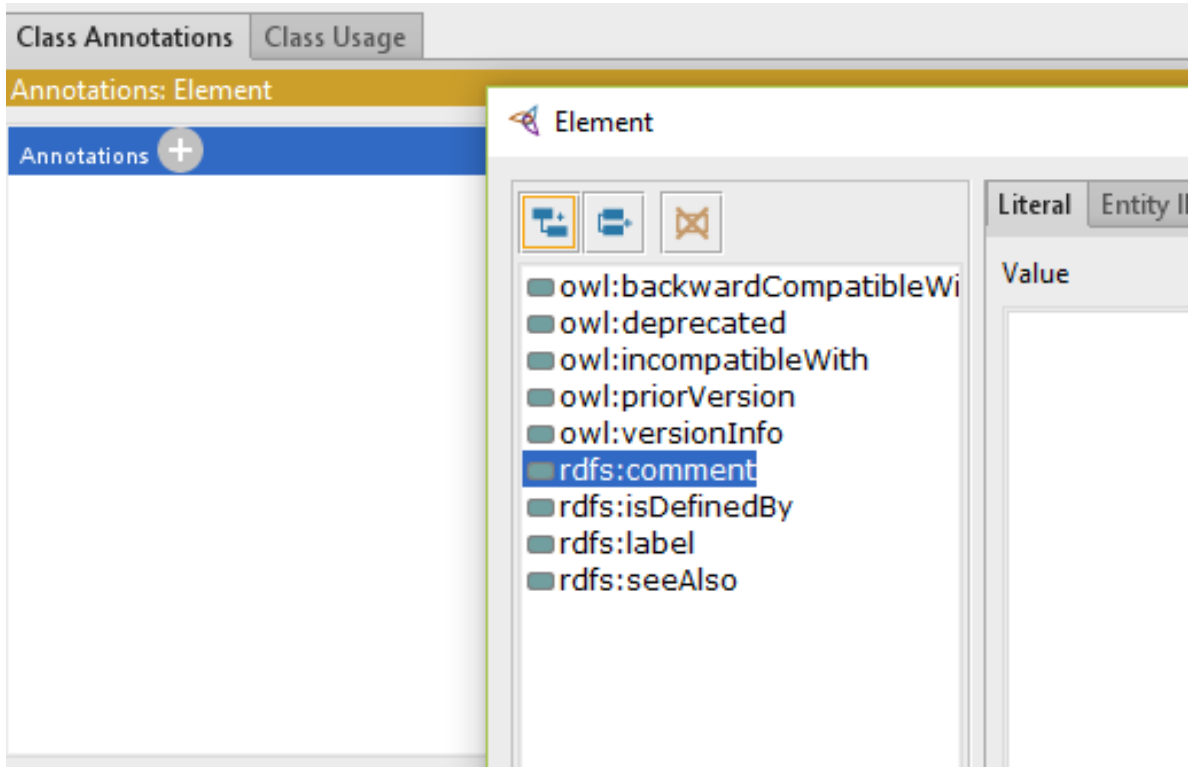
3



Example of data properties

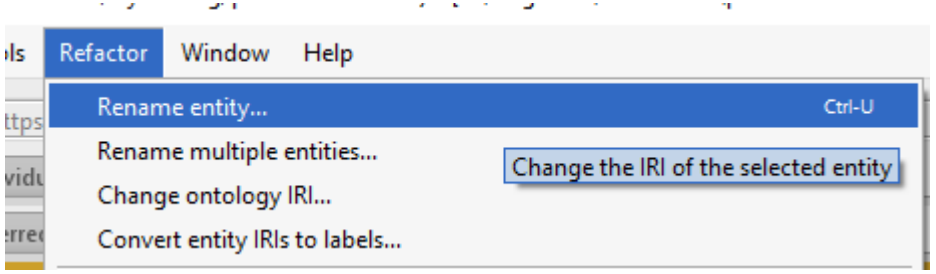
Annotation

It is important to describe what you are trying to do with each class, property, and individual in your ontology.

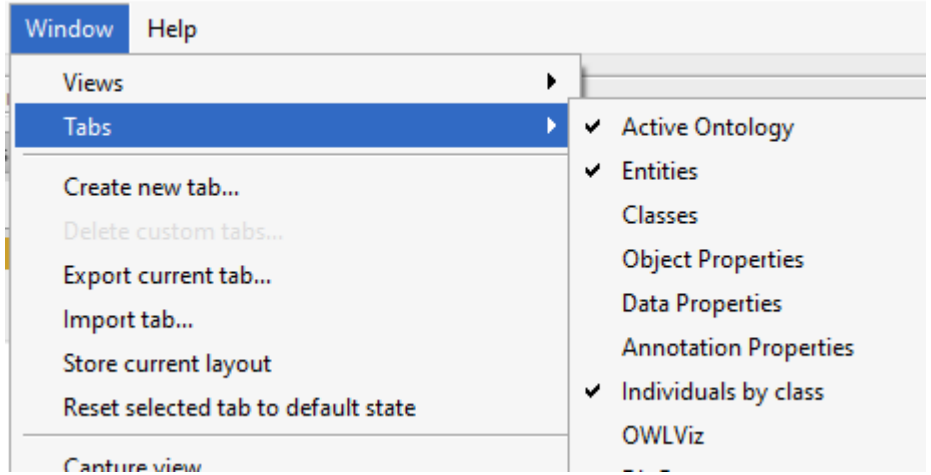


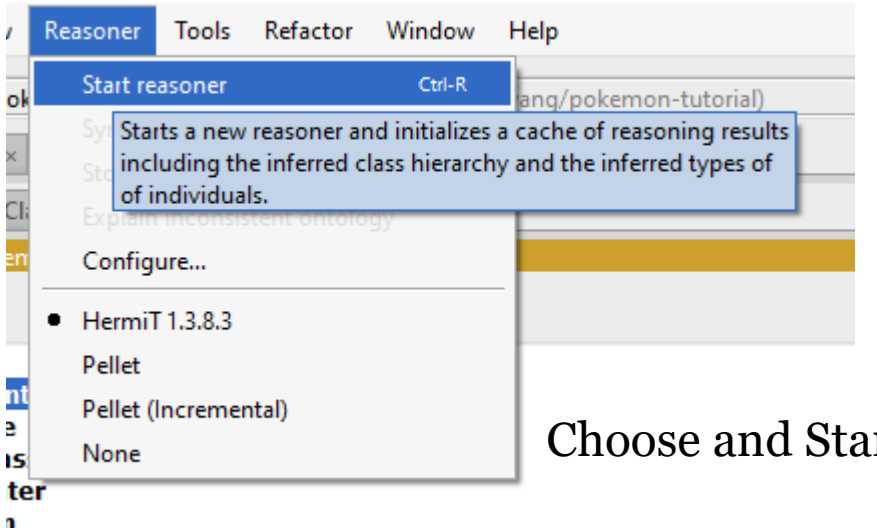
Other Features

Rename and change IRI



Customize View and Tabs

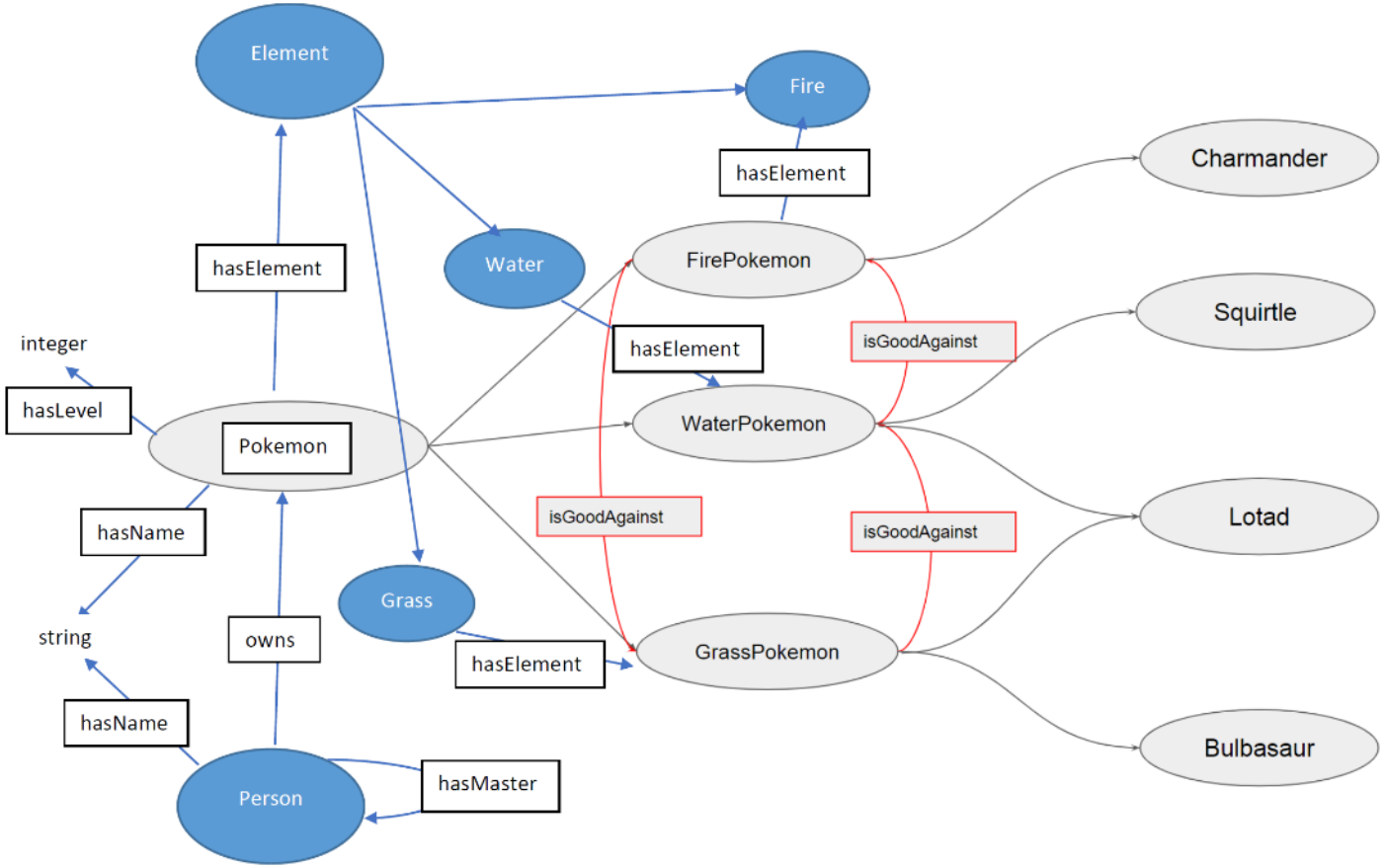




Choose and Start a Reasoner

UNIVERSITY OF WATERLOO

Extended Pokemon Ontology



UNIVERSITY OF WATERLOO

Thank You

- A Practical Guide To Building OWL Ontologies Using Protege 4 and CO-ODE Tools
  - http://mowl-power.cs.man.ac.uk/protegeowltutorial/resources/ProtegeOWLTutorialP4_v1_3.pdf
- Pizza Tutorial
  - https://cwi.unik.no/images/e/ef/UNIK4710-Protege_Presentation.pdf

- yetian.wang@uwaterloo.ca