

Task 1:

Task 1.1:

Rename the Ontology IRI. Since the instruction doesn't declare rename this IRI to which name, so I just rename it to Jieni_Yan_hw05.

Ontology header:

Ontology IRI	http://www.semanticweb.org/yanjieni/ontologies/2022/9/Jieni_Yan_hw05
Ontology Version IRI	e.g. http://www.semanticweb.org/yanjieni/ontologies/2022/9/Jieni_Yan_hw05

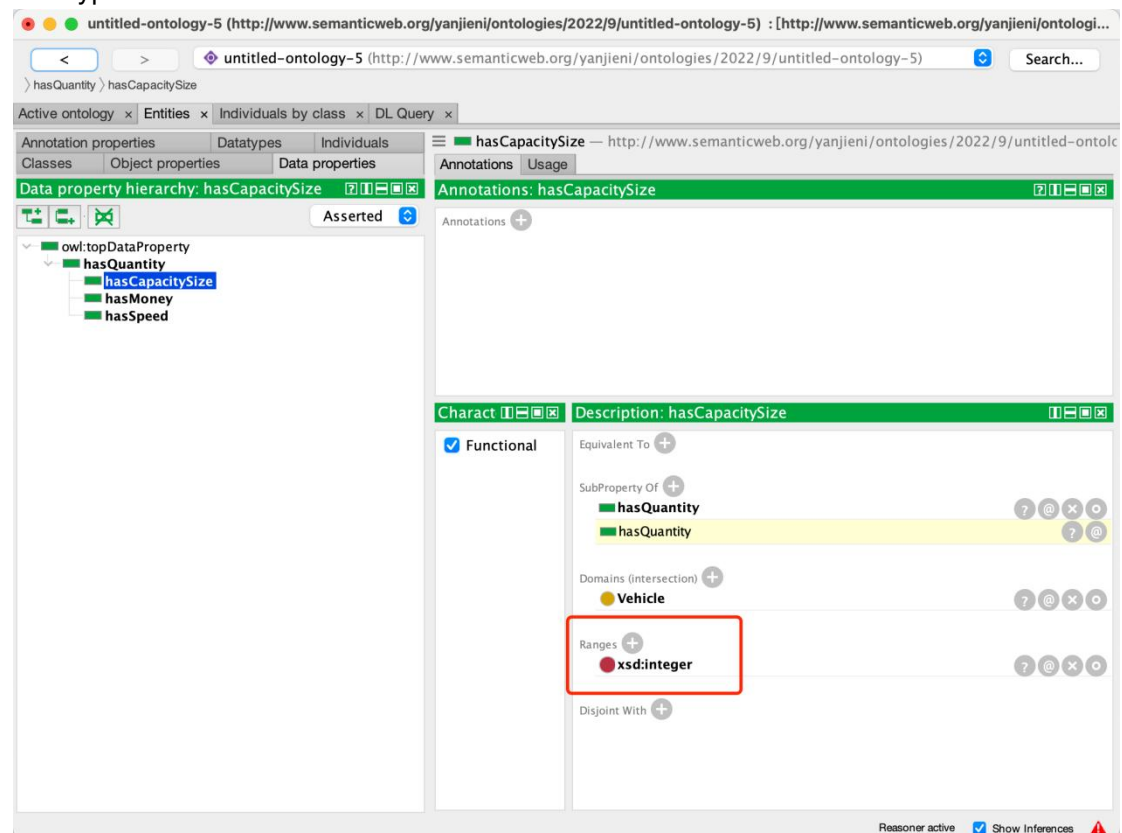
In Classes tab, using “create class hierarchy” to create Vehicle, Brand, Part, and the sub-classes under them. In Vehicle's description, click + button near “Disjoint with”, and select Brand and Part as disjoint relation with Vehicle. After clicking synchronize reasoner, in Brand and Part's Description, the “Disjoint with” part is set automatically according to Vehicle.

The screenshot shows the Protégé ontology editor interface. The top bar indicates the active ontology is 'untitled-ontology-5'. The left pane shows the 'Class hierarchy' for 'Vehicle', which includes subclasses like 'ExpensiveCar', 'Accord', 'Camry', 'Civic', 'Corolla', 'CR-V', 'Part', 'GPS', 'Speedometer', 'Brand', 'BMW', 'Benz', 'Toyota', 'Audi', and 'Honda'. The right pane shows the 'Description: Vehicle' tab, where the 'Disjoint With' section is highlighted with a red box, showing 'Part, Brand' as disjoint classes. The 'Reasoner active' checkbox is checked at the bottom right.

In Object properties tab, using “create object properties hierarchy” to create hasBand and hasPart properties.

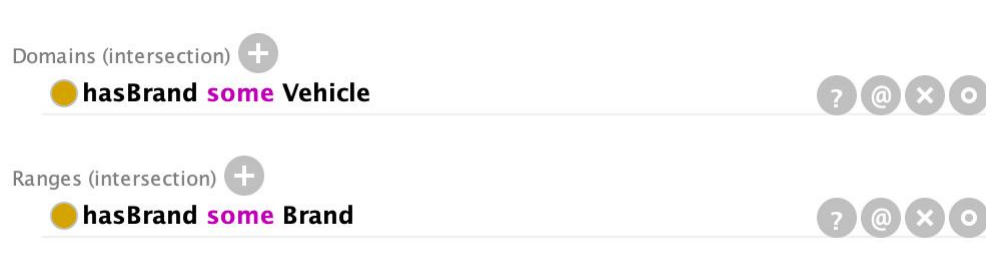
The screenshot shows the 'Object properties hierarchy' for 'hasBrand'. The left pane shows the hierarchy starting from 'owl:topObjectProperty', with 'hasBrand' and 'hasPart' listed as object properties. The right pane shows the 'Object properties hierarchy: hasBrand' tab, which is currently empty.

In Data properties tab, using “create data properties hierarchy” to create hasQuantity property and hasCapacitySize, hasMoney, hasSpeed under hasQuantity. For each property under hasQuantity, click + button near “Ranges” and add xsd:integer in build in datatype.



Task 1.2:

For hasBrand, click + button near “Domains” to add hasBrand some Vehicle. And click + button near “Ranges” to add hasBrand some Brand. Do similar thing for hasPart property. Although the instruction doesn't ask to set hasPart, I'd like to set it for consistent.



Then, set the Domains for hasSpeed properties as the instruction ask. Also for consistent, I set the Domains for hasCapacitySize and hasMoney.



To add a restriction, in the class tab, click “SubClass of” in description in Vehicle. Add “hasBrand some Brand” and “hasPart Some Part”.

SubClass Of

hasBrand some Brand

hasPart some Part

?

@

x

o

Task 1.3:

To add object property for each vehicle, I click the + button near the “SubClass of”. For Civic, Accord and CR-V, I add “hasBrand some Honda” to make sure it’s brand is Honda. For Camry and Corolla, I add “hasBrand some Toyota” to make sure it’s brand is Toyota.

SubClass Of

hasBrand some Honda

Vehicle

?

@

x

o

To add data instance and data properties for each vehicle, I use the individuals tab. In this tab, I add Civic2022, Accord2022, Camry2022, Corolla2022, and CR-V2022. For each individual, in “Data property assertions”, I set the “hasCapacitySize”, “hasSpeed”, and “hasMoney” properties to the number as instruction ask.

Annotation properties
Datatypes
Individuals
Classes
Object properties
Data properties

Accord2022

Camry2022

Civic2022

Corolla2022

CR-V2022

Annotations: Accord2022

Annotations

Description: Accord2022

Property assertions: Accord2022

Types

Accord

ExpensiveCar

Same Individual As

Different Individuals

Data property assertions

hasCapacitySize 5

hasSpeed 145

hasMoney 40000

Negative object property assertions

Negative data property assertions

Then go back to class tab, for each vehicle, I click the + button near “Instances” and add the correct individual.

Instances

Accord2022

?

@

x

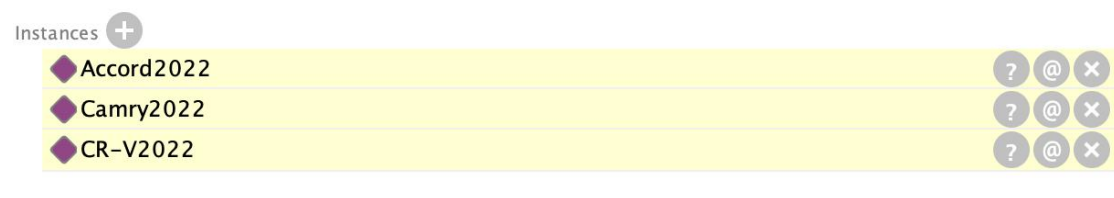
Task 1.4:

I create an ExpensiveCar class under Vehicle. Then I click the + button near “Equivalent

to” and add “Vehicle and (hasMoney some xsd:integer[>=40000])” in “class expression editor” tab.



Then, I click synchronize reasoner, three qualified instances automatically show up under “Instances”.



Task 2.3:

F1 score = 0.94