

Name – Dhanraj Babasaheb  
Bhosale

ASU ID - 1225506620

Q1.

Ans -

Yes, the MeatyVegetable topping class came out as inconsistent.

This is so that this class can inherit from the VegetableTopping and MeatTopping classes. Then, however, we separated the classes for MeatTopping and VegetableTopping. It appeared inconsistent since MeatyVegetableTopping connected the two of them.

We may delete the MeatyVegetableTopping as a child of either the VegetableTopping class or the MeatTopping class to fix this discrepancy. There wouldn't be any contradiction in such scenario.

Q2.

Ans.

I added an Indian pizza named ChickenTikkaPizza which has the following toppings :

1. ChickenTopping
2. MozzarellaTopping
3. OreganoTopping
4. TomatoTopping

I also added a SausagePizza which originates in Italy. The toppings for these pizza were :

1. SausageTopping
2. MozzarellaTopping
3. TomatoTopping

Q3.

Ans.

I added two pizza bases namely :

1. WholeWheatBase
2. MixedGrainBase

Q4.

Ans

I added multiple topping classes.They are :

1. OreganoTopping
2. PineappleTopping
3. SausageTopping
4. CottageCheeseTopping
5. BlackGarlicTopping

Q5.

Ans

I added four more properties to the ones showed in the tutorial. They are :

1. hasCheeseAmount - This shows how much cheese is in a pizza
2. hasShape - Pizzas come in different shapes like rectangular and round. This property helps distinguish between round and rectangular
3. hasNationality - This property lets us know what nation did the pizza originate from.
4. hasCrust - This tells us what type of crust does the pizza have. Mainly there are two types of crust , thin and thick.

Q6.

Ans

For instance, a person who like tomatoes a lot would prefer to eat pizza that contains them. In order to achieve so, I ran a query that locates all the pizzas with tomatoes. As a result, the inquiry contains some tomato topping. This search determines which classes have the hasTopping attribute with the value TomatoTopping. A limitation of the hasValue type is hasTopping. The link between the PizzaTopping and NamedPizza subclasses is reflected in the hasTopping method. When this query is executed, the reasoner uses the hasTopping limitation to identify classes that are connected to the TomatoTopping class.

The screenshot displays a software interface for a DL query. On the left, a 'Class hierarchy' tree shows the ontology structure, with 'SausagePizza' highlighted. The main panel shows a 'DL query' window with the expression 'hasTopping some TomatoTopping'. Below the query, there are buttons for 'Execute' and 'Add to ontology'. The 'Query results' section is divided into several categories: 'Equivalent classes (0 of 0)', 'Superclasses (0 of 1)', 'Direct superclasses (0 of 1)', 'Direct subclasses (6 of 6)', 'Subclasses (6 of 7)', and 'Instances (1 of 1)'. The 'Direct subclasses' and 'Subclasses' lists include AmericanPizza, ChickenTikkaPizza, MargheritaPizza, PaneerPizza, SausagePizza, and SohoPizza. The 'Instances' list includes 'ThisPizza'. On the right side, there are 'Query for' and 'Result filters' sections. The 'Query for' section has checkboxes for 'Direct superclasses', 'Superclasses', 'Equivalent classes', 'Direct subclasses', 'Subclasses', and 'Instances'. The 'Result filters' section has a 'Name contains' field and two checkboxes: 'Display owl:Thing (in superclass results)' and 'Display owl:Nothing (in subclass results)'.

Q7

Ans. The second query I ran was to find all the pizzas which have Italy as nationality. This query uses hasNationality property of the Pizza class. The query is

hasNationality some Italy.

The reasoner tries to find all the classes where the Pizza is a subclass of “hasNationality some Italy” class. All the pizzas which are connected to the hasNationality property through the value Italy are selected.

The screenshot displays a software interface with two main panels. The left panel, titled "Class hierarchy: SausagePizza", shows a tree structure of classes. The root is "owl:Thing", which branches into "PizzaDomainConcept" and "PizzaBase". "PizzaDomainConcept" further branches into "CheeseAmount", "PizzaShape", "PizzaCrust", and "Nationality". "Nationality" branches into "America", "India", and "Italy". "Italy" branches into "Pizza". "Pizza" branches into "MeatyPizza" and "NamedPizza". "NamedPizza" branches into "AmericanPizza", "ChickenTikkaPizza", "HawaiianPizza", "MargheritaPizza", "PaneerPizza", "SausagePizza", and "SohoPizza". "SausagePizza" is highlighted with a blue box. The right panel, titled "DL query:", contains a text input field with the query "hasNationality some Italy". Below the input field are two buttons: "Execute" and "Add to ontology". Below the buttons is a section titled "Query results" which lists several categories of results: "Equivalent classes (0 of 0)", "Superclasses (0 of 1)", "Direct superclasses (0 of 1)", "Direct subclasses (2 of 2)", "Subclasses (2 of 3)", and "Instances (0 of 0)". The "Direct subclasses" and "Subclasses" sections list "MargheritaPizza" and "SausagePizza". To the right of the "Query results" section is a "Query for" section with checkboxes for "Direct superclasses", "Superclasses", "Equivalent classes", "Direct subclasses", "Subclasses", and "Instances". Below this is a "Result filters" section with a text input field and two checkboxes: "Display owl:Thing (in superclass results)" and "Display owl:Nothing (in subclass results)".

Class hierarchy: SausagePizza

DL query:

Query (class expression)

hasNationality some Italy

Execute Add to ontology

Query results

Equivalent classes (0 of 0)

Superclasses (0 of 1)

Direct superclasses (0 of 1)

Direct subclasses (2 of 2)

Subclasses (2 of 3)

Instances (0 of 0)

Query for

Result filters

Name contains

Display owl:Thing (in superclass results)

Display owl:Nothing (in subclass results)