The Description Logic Project - 2016 KR course

Group A

Shuai Wang + Aashish Venkatesh

The Description Logic Project - 2016 KR course

Contents

1. TASK 1: Consistency

Find which classes are inconsistent, find ways to resolve the inconsistency, implement them in the Protege editor, and save the result as your own consistent-pizza.owl

1.1. Inconsistency 1: IceCream

The IceCream Class is equivalent to Nothing (from inferred). That is not what we want. To fix it, we remove the hasTopping some FruitTopping. This is because FruitTopping is only for pizza, not for ice cream.

1.20 .Inconsistency 2: IceCream

CheeseVegetableTopping is also leading to inconsistency. The reason is that it is a sub-class of CheeseTopping and VegetableTopping, which are disjoint. To fix this, we can remove the relation with CheeseTopping. This way, it is only a subclass of VegetableTopping.

2.Task 2: Redundancy

Find if there are places where the pizza.owl ontology contains redundant statements. If you find any, report them and explain why they are redundant.

2.1. SpicyPizzaEquivalent

Since there is already a SpicyPizza class, this class can be removed.

2.2. VegetarianPizzaEquivalent 1 and 2

Both of the VegetarianPizzaEquivalent1 and VegetarianPizzaEquivalent2 are with redundent definition as VegetarianPizza. We removed them.

2.Task 3: Add Redundancy

Extend the ontology with some obvious redundant statements, such as redundant number restrictions, redundant subclass statements, redundant class-intersections, and many others. And save as redundant-pizza.owl.

2.1. PopularPizza[redundant number restriction, redundant class]

We introduce PopularPizza as NamedPizza with at least one base, which is redundent since every pizza has at least one base.

2.2. AmericanNotHot [redundant class, redundant class-intersection]

We introduce AmericanNotHot equivalent to American (as opposite to AmericanHot) and disjoint with NamedPizza. This is a redundant definition since we already defined American and American is itself a NamedPizza.

2.3. Dessert [redundant class]

We introduce the dessert class as **Food and not Pizza and not PizzaBase and not PizzaTopping.** This is equivalent as IceCream

2.4. Soho [redundant subclass statement, redundant disjoint statement]

We add an additional constraint for Soho to make it a subclass of **Food** and in disjoint with **Hot**. It is redundant since its superclass is already a subclass of Food. And it is not a value partition so of course it is disjoint with Hot.