

Project 3

The Pizza Ontology in SNePSLOG

CSE 4/563, Knowledge Representation

Due Thursday, December 9, 2010

Professor Shapiro

November 16, 2010

1 Project Statement

The ontology you created for Project 1 was just one part of an ontology: the class (category) hierarchy. Ontologies also typically include information about the properties of, and the relationships among, the classes and instances of the classes. However, that information cannot be formalized in Propositional Logic. For this project, you are to do a more complete formalization of an ontology.

The ontology you will formalize for this project is the Pizza Ontology, as presented in (Horridge, 2009), a tutorial on the use of the Protégé Ontology Editor. Protégé, and the logic that it uses, Description Logic (DL) (Baader et al., 2007; Brachman and Levesque, 2004), will be discussed briefly this semester, and are discussed in more detail in CSE 663. DL is less expressive than full first-order logic, but is popular for creating formal ontologies. One reason for this is that decision procedures exist for DLs, and several such reasoners are available.

For this project, you are to use SNePSLOG mode 3 to represent and reason about some of the information about the Pizza Ontology presented in (Horridge, 2009). SNePSLOG is more expressive than DLs, but also less efficient than several available DL reasoners. This project will concentrate on the ability in SNePSLOG to axiomatize rules about the relations among categories, and among the proposition-valued functions we normally think of as relations, and which DL calls “properties.”

This project differs from the previous projects by being more a directed set of exercises than they were. Like the homeworks, there is a series of problem exercises where the information you are to formalize will be provided to you. Intermixed with the assertions are questions to test your formalizations.

2 Deliverables

As it says on the CSE 4/563 web page, you are to

“hand in a paper, produced using a document formatting program such as LaTeX or Microsoft Word, and printed on 8.5 by 11 inch paper, stapled in the upper left-hand corner, with a title, your name, and other identifying information at the top of the first page (Do not use the header page automatically produced by the printer), plus a well-documented listing and run of your program. (Do not enclose your paper in a folder or cover.)” [5]

The paper is due at the start of class on the date shown at the beginning of this document.

You will be provided with a plain text version of the exercises in the required order. You are to edit that text into the body of your paper, with your formalizations inserted in the appropriate places. Immediately after the formalization of each question, paste in the answer provided by SNePSLOG.

In addition to the paper, you are to submit (using `submit_cse463` or `submit_cse563`) your program, so that it can be run and checked if the instructors choose. Name your file `pizza.snepslog`. You are to submit that file by one half-hour before the start of class on the date shown at the beginning of this document, so that you can get to class on time to hand in the paper.

Projects will be considered early or late according to the date and time that both the program is submitted and the paper is handed in.

2.1 The Paper

Your paper should have the following parts:

1. Descriptive title
2. Name(s) and user name(s) of authors
3. Introduction: general description of the project
4. Ontology Formalization and Test Questions
 - (a) Syntax and semantics of function symbols
 - (b) Ontology exercises with your formalizations and the results of your questions filled in
5. Acknowledgments as needed
6. References as needed

3 Grading

The project will be graded according to the following table:

	CSE 463	CSE 563
Introduction	4	9
Syntax & semantics of function symbols	9	15
Ontology items and test questions: 53 items \times 3 points each	159	159
Acknowledgments	3	5
Quotes, citations and references	5	12
Project Total	180	200

Your project grade will be your total earned points divided by the total possible points: 180 points for CSE463 students; 200 points for CSE563 students.

4 Ontology Formalization and Test Questions

4.1 Syntax and semantics of function symbols

List here the function symbols you use, showing their syntax and intensional semantics. For this purpose, you should list the `define-frame` commands from your program, and include the documentation strings. Also be sure to organize them in a reasonable manner. You needn't list the individual constants you use. You are to use the individual constants shown in the next section, and, where provided, the function symbols.

4.2 Ontology and Test Questions

All the following information specific to the Pizza Ontology is taken from (Horridge, 2009). Page numbers are shown for some pieces of information. Where page numbers are not shown, the information, if specific to the Pizza Ontology, is taken from (Horridge, 2009) after the preceding page number, and before the subsequent page number.

There are times when a question won't be answered unless you put `clear-infer` before it. However, using `clear-infer` when it isn't needed will just make it take longer for the question to be answered. Also, there are times when all the answers aren't printed unless you ask the question twice. All you uses of these two techniques should be shown in your answer file as well as in your program.

1. Express these three rules as path-based inference rules
 - (a) If a has b as a superclass, and b has c as a superclass, then a has c as a superclass; if a does not have c as a superclass, but b has c as a superclass, then a does not have b as a superclass.
 - (b) If a has b as a subclass, and b has c as a subclass, then a has c as a subclass; if a does not have c as a subclass, but b has c as a subclass, then a does not have b as a subclass.
 - (c) If x is an instance of class c, then it is an instance of any superclass of c; if x is not an instance of class c, then it is not an instance of any subclass of c.
2. Page 17: Pizza, PizzaTopping, and PizzaBase are subclasses of Thing.
3. Page 18: Pizza, PizzaTopping, and PizzaBase are disjoint classes.
4. If two classes are disjoint, neither is a subclass of the other, a subclass of one is not a subclass of the other, and an instance of one is not an instance of the other.
5. Page 20: ThinAndCrispy and DeepPan are disjoint subclasses of PizzaBase
6. What are the superclasses of ThinAndCrispy? [PizzaBase and Thing, but not DeepPan, PizzaTopping, nor Pizza]
7. Is ThinAndCrispy a subclass of DeepPan? [No]
8. Page 22: CheeseTopping, MeatTopping, SeafoodTopping, and VegetableTopping are disjoint subclasses of PizzaTopping.
9. MozzarellaTopping and ParmesanTopping are disjoint subclasses of CheeseTopping.
10. HamTopping, PepperoniTopping, SalamiTopping, and SpicyBeefTopping are disjoint subclasses of MeatTopping.
11. AnchovyTopping, PrawnTopping, and TunaTopping are disjoint subclasses of SeafoodTopping.
12. CaperTopping, MushroomTopping, OliveTopping, OnionTopping, PepperTopping, and TomatoTopping are disjoint subclasses of VegetableTopping.
13. RedPepperTopping, GreenPepperTopping, and JalapenoPepperTopping are disjoint subclasses of PepperTopping.
14. Page 27: hasTopping and hasBase are subproperties of hasIngredient
15. If p1 is a subproperty of p2, and p1(x,y), then p2(x,y). [For example, isMotherOf is a subproperty of isParentOf.]
16. Pizza1 has topping1 as its topping, and base1 as its base.
17. Pizza2 has topping2 as its topping, and base2 as its base.
18. What are the ingredients of pizza1? [topping1 and base1]

19. Page 28: isIngredientOf is the inverse property of hasIngredient
20. isToppingOf is the inverse property of hasTopping
21. isBaseOf is the inverse property of hasBase
22. If p_1 and p_2 are inverse properties of each other and $p_1(x,y)$ then $p_2(y,x)$.
23. What is topping1 a topping of? [pizza1]
24. If p_1 and p_2 are inverse properties, p_1 is a subproperty of q_1 , and q_1 and q_2 are inverse properties, then p_2 is a subproperty of q_2 .
25. What is a subproperty of isIngredientOf? [isBaseOf and isToppingOf]
26. Page 29: A property p being functional means that if $p(x,y)$, then $p(x,z)$ iff z is coreferential with y .
27. The hasBase property is functional.
28. Base1 and base2 are not coreferential.
29. Does pizza1 have base2 as its base? [No]
30. Pizza2 has tastyBase2 as its base.
31. Are base2 and tastyBase2 coreferential? [Yes]
32. Does pizza1 have topping2 as its topping? [I don't know]
33. A property p being inverse functional means that if $p(x,y)$, then $p(z,y)$ iff z is coreferential with x .
34. hasTopping is inverse functional.
35. Pizza1 is not coreferential with pizza2.
36. What has topping1 as its topping? [pizza1, but not pizza2]
37. If properties p_1 and p_2 are inverses of each other, then p_1 is functional iff p_2 is inverse functional.
38. What is topping2 the topping of? [pizza2, but not pizza1]
39. Page 33: If the domain of the property p is the class c , then, for any x and y , if $p(x,y)$ then x is an instance of c .
40. If the range of the property p is the class c , then, for any x and y , if $p(x,y)$ then y is an instance of c .
41. If p_1 and p_2 are inverse properties of each other, then the domain of one is the range of the other.
42. Page 34: The domain of hasTopping is Pizza, and its range is PizzaTopping.
43. Pizza1 is an instance of what categories? [Pizza and Thing, but not a lot of other categories]
44. Topping2 is an instance of what categories? [PizzaTopping, but neither Pizza, nor PizzaBase]
45. Page 39: An existential restriction on a class, c_1 , a property, p_1 , and another class, c_2 says that every instance of c_1 has the property p_1 to some instance of c_2 . In other words, every c_1 has the relation p_1 to some c_2 .
46. Page 41: Every pizza must have a pizza base.
47. Page 45: NamedPizza is a subclass of Pizza.
48. MargheritaPizza is a subclass of NamedPizza.

49. Page 45, 46: Every MargheritaPizza must have a topping which is a MozzarellaTopping and a topping which is a TomatoTopping.
50. Pizza3 is a MargheritaPizza.
51. What categories are the ingredients of pizza3 instances of? (Use askwh.) [MozzarellaTopping, TomatoTopping, CheeseTopping, VegetableTopping, PizzaTopping, PizzaBase, and Thing]

References

- Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., and Patel-Schneider, P. F., editors (2007). *The Description Logic Handbook*. Cambridge University Press, Cambridge, UK, second edition.
- Brachman, R. J. and Levesque, H. J. (2004). *Knowledge Representation and Reasoning*. Morgan Kaufmann, San Francisco, CA.
- Horridge, M. (2009). *A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools: Edition 1.2*. The University of Manchester. Available at http://owl.cs.manchester.ac.uk/tutorials/protegeowltutorial/resources/ProtegeOWLTutorialP4_v1_2.pdf.