Project 3
The Pizza Ontology in SNePSLOG
CSE 4/563, Knowledge Representation
REVISED
Due Monday, April 27, 2009

Professor Shapiro

April 8, 2009

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 et al., 2007), a tutorial on the use of the Protégé Ontology Editor. Protégé, and the logic that it uses, Description Logic (DL) (Baader et al., 0007; Brachman and Levesque, 2004), are discussed 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 will use SNePSLOG to represent and reason about some of the information about the Pizza Ontology presented in (Horridge et al., 2007). SNePSLOG is more expressive than DLs, but also less efficient than several available DL reasoners.

This project will differ from the previous projects by being more a directed set of exercises than they were. These will come in three groups: first, a section where you are to collect all the function and predicate symbols you use for the ontology, stating the syntax and intensional semantics of them; second, a series of problem exercises where the information you are to formalize will be provided to you; third a series of questions you are to pose to the SNePS reasoner. You should use SNePSLOG mode 3 for this project.

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.

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 your KB 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.

2.1 The Paper

Your paper should have the following parts:

1. Descriptive title
2. Author identification
3. Introduction: general description of the project
4. Ontology Formalization and Test Questions
   (a) Syntax and semantics of function and predicate symbols
   (b) Ontology exercises with your solutions filled in
   (c) English versions of the test questions provided to you with your formalizations and your program's results 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
Syntax & semantics of predicate and function symbols	9	14
Ontology items: 61 items $\times$ 3 points each	183	183
Test Questions: 18 items $\times$ 3 points each	54	54
Project Total	**270**	**275**

Your project grade will be your total earned points divided by the total possible points: 270 points for CSE463 students; 275 points for CSE563 students.

4 Ontology Formalization and Test Questions

4.1 Syntax and semantics of function and predicate symbols

List here the function and predicate symbols you use, showing their syntax and intensional semantics. For this purpose, you may just list the `define-frame` commands from your program, but be sure to include the documentation strings, and 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 predicate symbols.

4.2 Ontology

All the following information specific to the Pizza Ontology is taken from (Horridge et al., 2007). Page numbers are shown for some pieces of information. Where page numers are not shown, the information, if specific to the Pizza Ontology, is taken from (Horridge et al., 2007) after the preceding page number, and before the subsequent page number.

1. If a is a subclass of b, and b is a subclass of c, then a is a subclass of c. If a is not a subclass of c, and b is a subclass of c, then a is not a subclass of b.
2. If x is an instance of a, and a is a subclass of b, then x is an instance of b. If x is not an instance of b, and a is a subclass of b, then x is not an instance of a.
3. If p1 is a subproperty of p2, and p2 is a subproperty of p3, then p1 is a subproperty of p3. If p1 is not a subproperty of p3, and p2 is a subproperty of p3, then p1 is not a subproperty of p2.
4. Pizza, PizzaTopping, and PizzaBase are subclasses of Thing [p. 17].
5. Pizza, PizzaTopping, and PizzaBase are disjoint classes [p. 19].
6. If two classes are disjoint: neither is a subclass of the other; and a subclass of one is not a subclass of the other.
7. If two classes are disjoint, an instance of one is not an instance of the other.
8. ThinAndCrispy and DeepPan are disjoint subclasses of PizzaBase [p. 21].
9. CheeseTopping, MeatTopping, SeafoodTopping, and VegetableTopping are disjoint subclasses of PizzaTopping [p. 22].
10. MozzarellaTopping and ParmesanTopping are disjoint subclasses of CheeseTopping.
11. HamTopping, PepperoniTopping, SalamiTopping, and SpicyBeefTopping are disjoint subclasses of MeatTopping.
12. AnchovyTopping, PrawnTopping, and TunaTopping are disjoint subclasses of SeafoodTopping.
13. CaperTopping, MushroomTopping, OliveTopping, OnionTopping, PepperTopping, and TomatoTopping are disjoint subclasses of VegetableTopping.
14. RedPepperTopping, GreenPepperTopping, and JalapenoPepperTopping are disjoint subclasses of PepperTopping (Horridge et al., 2007, p. 22).
15. hasTopping and hasBase are subproperties of hasIngredient.
16. If p1 is a subproperty of p2, and p1(x,y), then p2(x,y). [For example, isMotherOf is a subproperty of isParentOf.]
17. isIngredientOf is the inverse property of hasIngredient.
18. If p1 and p2 are inverse properties of each other, then p1(x,y) iff p2(y,x).
19. If p1 and p2 are inverse properties, p1 is a subproperty of q1, and q1 and q2 are inverse properties, then p2 is a subproperty of q2.
20. isBaseOf is the inverse property of hasBase.
21. isToppingOf is the inverse property of hasTopping.
22. A property p being functional means that if p(x,y), then p(x,z) iff z is coreferential with y. [p. 29]
23. A property p being inverse functional means that if p(x,y), then p(z,y) iff z is coreferential with x.

24. If properties p1 and p2 are inverses of each other, then p1 is functional iff p2 is inverse functional.
25. A property p being transitive means that whenever p(x,y) and p(y,z), then p(x,z). [p. 31]
26. If properties p1 and p2 are inverses of each other, then one is transitive iff the other is.
27. hasIngredient is transitive.
28. If a property is transitive, then it is not functional. [p. 33]
29. The hasBase property is functional.
30. 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. [p. 35]
31. 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.
32. if p1 and p2 are inverse properties of each other, then the domain of one is the range of the other, and vice versa.
33. The range of hasTopping is PizzaTopping. [p. 36]
34. The domain of hasTopping is Pizza. [p. 37]
35. The domain of hasBase is Pizza, and its range is PizzaBase. [p. 38]
36. A necessary some restriction on a class, c1, a property, p1, and another class, c2 says that every instance of c1 has the property p1 to some instance of c2. In other words, every c1 has the relation p1 to some c2. [p. 42]
37. Every Pizza has a base that is a PizzaBase.
38. NamedPizza is a subclass of Pizza. [p. 46]
39. MargheritaPizza is a subclass of NamedPizza.
40. Every MargheritaPizza has a topping which is a MozzarellaTopping.
41. Every MargheritaPizza has a topping which is a TomatoTopping. [p. 47]
42. AmericanaPizza is a subclass of NamedPizza. [p. 48]
43. Every AmericanaPizza has a topping which is a MozzarellaTopping.
44. Every AmericanaPizza has a topping which is a TomatoTopping.
45. Every AmericanaPizza has a topping which is a PepperoniTopping.
46. An AmericanHotPizza is just like an AmericanaPizza, but, in addition, has a JalapenoPepperTopping. [p. 49]
47. A SohoPizza is just like a MargheritaPizza, but, in addition, has OliveTopping and ParmesanTopping.
48. MargheritaPizza, AmericanaPizza, AmericanHotPizza, and SohoPizza are disjoint classes. [p. 50]
49. ProbeInconsistentTopping is a subclass of both CheeseTopping and VegetableTopping. [p. 52]
50. A sufficient some restriction on a class, c1, a second class, c2, a property, p1, and a third class, c3 says that every instance of c2 that has the property p1 to some instance of c3 is an instance of c1. [p. 56]
51. A necessary and sufficient some restriction on a class, c1, a second class,c2, a property, p1, and a third class, c3 is both a necessary some restriction and a sufficient some restriction, and says that c1 is a subclass of c2.
52. A necessary and sufficient restriction on CheesyPizza is that every instance of CheesyPizza is a Pizza with a topping which is a CheeseTopping.

53. If a class, cs1, has a sufficient some restriction that says that every instance of c2 that has the property p to some instance of cs3 is an instance of cs1, then every class cb1 that is a subclass of c2 and that has a necessary some restriction saying that every instance of cb1 has a property p to some instance of some subclass of cs3 is itself a subclass of cs1.

54. VegetarianTopping and MeatyTopping are disjoint subclasses of PizzaTopping. [p. 64]

55. CheeseTopping and VegetableTopping are disjoint subclasses of VegetarianTopping.

56. MeatTopping and SeafoodTopping are disjoint subclasses of MeatyTopping

57. A necessary all restriction on a class, c1, a property, p, and another class, c2 says that for every instance, x, of c1, whenever the property p holds to some y, y is an instance of class c2.

58. A necessary all restriction on a class implies the same necessary some restriction.

59. VegetarianPizza and MeatyPizza are disjoint subclasses of Pizza.

60. Every topping of every VegetarianPizza is a VegetarianTopping.

61. A necessary and sufficient condition for being a MeatyPizza is that it has a topping which is a MeatyTopping.

4.3 Test Questions

Start each test question item with a `clear-infer` command. The statements are to be formalized as SNePSLOG assertions. The questions are to be formalized as SNePSLOG queries. The correct answers are shown in brackets.

1. Item1 is a pizza.
   Is item1 a pizza base? [No]

2. Is red pepper topping a subclass of pizza topping? [Yes]
   Is red pepper topping a subclass of cheese topping? [No]

3. Item2 is a mozzarella topping.
   Is item2 a pizza topping? [Yes]
   Is item2 a ham topping? [No]

4. What are the subproperties of isIngredientOf? [isBaseOf and isToppingOf]

5. Item3 has topping3 as its topping, and base3 as its base.
   What are the ingredients of item3? [topping3, base3 and another base that should be expressed by a Skolem functional term]

6. Is isIngredientOf transitive? [Yes]
   Is isIngredientOf functional? [No]

7. Base4 is a PizzaBase, but it is not the same thing as base3.
   Is base4 the base of item3? [No]

8. Are the two PizzaBases returned in the answer for test question (5) the same? [Yes]

9. What is the domain of isToppingOf? [PizzaTopping]
   What is the range of isToppingOf? [Pizza]

10. Item4 is a pizza.
    Does item4 have a base, and if so, what kind of thing is its base? [Yes, a PizzaBase and a Thing]

11. Item5 is a MargheritaPizza.
    What are the ingredients of item5, and in what classes are they? [one ingredient is a Thing and a PizzaBase; another ingredient is a PizzaTopping and a TomatoTopping; a third ingredient is a PizzaTopping and a MozzarellaTopping]

12. What are the superclasses of ProbeInconsistentTopping? [This should cause SNeBR to complain. Respond by discarding the assertion that ProbeInconsistentTopping is a subclass of both CheeseTopping and VegetableTopping.]

13. Item6 is a CheesyPizza.
    Is item6 a pizza? [Yes]
    Does item6 have a topping that is a cheese topping? [Yes]

14. The topping of item1 is topping1.
    Topping1 is a ParmesanTopping.
    Is item1 a cheesy pizza? [Yes]

15. Is MargheritaPizza a subclass of CheesyPizza? [Yes]

16. Is item5 a CheesyPizza? [Yes]

17. Item7 is a VegetarianPizza.
    Topping7 is a topping of item7.
    What kind of thing is topping7? [VegetarianTopping]

18. What NamedPizzas are MeatyPizzas? [AmericanaPizza and AmericanHotPizza]

References

Baader, F., Calvanese, D., McGuiness, D. L., Nardi, D., and Patel-Schneider, P. F., editors (20007). *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., Jupp, S., Moulton, G., Rector, A., Stevens, R., and Wroe, C. (2007). *A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools: Edition 1.1*. The University of Manchester. Available at `http://www.co-ode.org/resources/tutorials/ProtegeOWLTutorial-p4.0.pdf`.