

Project 1

Using SNARK for an Ontology of Businesses

CSE 4/563, Knowledge Representation

Due Friday, March 9, 2007

Professor Stuart C. Shapiro

February 9, 2007

1 Introduction

1.1 Ontologies

In philosophy, “ontology” is the study of what there is. In AI, there have been many definitions of “ontology”, but simply put, an “ontology” is a taxonomy of the terms in some domain of knowledge, possibly along with other domain rules relating the terms. In this project you will create an ontology of businesses, represent the ontology in the language of SNARK [1], and use SNARK to answer some questions about the ontology.

1.2 SNARK

SNARK (SRI’s New Automated Reasoning Kit) [1] may be used as a KRR system by performing the following steps.

1. Run Common Lisp.
2. Load `/projects/shapiro/CSE563/ask`
3. Change the Common Lisp listener’s package to `snark-user`.
4. Load a knowledge base.
5. Ask the system questions.

The first Lisp form in the KB must be `(initialize)`. This must be followed by assertions in the format `(assert wfp)`, where *wfp* is a quoted form in the syntax required by SNARK. (See [1], Chapter 2.) Then `prove` may be called, or as an alternative, the `query` and/or `ask` front end from `/projects/shapiro/CSE563/ask.cl` may be used. Note that if you use `prove` you must evaluate `(new-row-context)` between successive calls to `prove`. For an example KB file, see `/projects/shapiro/CSE563/CarPoolWorld.cl`.

An example run of a propositional CarPoolWorld system using SNARK and the query front end follows. Most of the output produced by the loading of `snark` has been deleted.

```
cl-user(1): :ld /projects/shapiro/CSE563/CarPoolWorld
...
To start using SNARK, change the package to snark-user.
Then use (initialize), (assert <wff>), and (prove <wff>).
; Running SNARK from ...

Ask if ``Tom is the driver or Betty is the driver``
  is implied by the KB. (It is.)
  (ask '(or TomIsTheDriver BettyIsTheDriver)) = True

Ask if ``Betty is the driver
  implies that Betty is not the passenger``
  is implied. (It is.)
  (ask '(implies BettyIsTheDriver (not BettyIsThePassenger))) = True

Ask if ``Tom drives Betty and Tom is the passenger`` is implied.
  (It's false according to the domain rules.)
  (ask '(and TomDrivesBetty TomIsThePassenger)) = False

Ask if ``Betty drives Tom`` is implied by the KB.
  (Neither it nor its negation is implied.)
  (ask '(BettyDrivesTom)) = Unknown
```

Notice that `(ask wff)` can return `True`, `False`, or `Unknown`. It returns `True` if `wff` can be proved from the KB, `False` if `(not wff)` can be proved from the KB, and `Unknown` if there are models that satisfy $KB \wedge wff$ as well as models that satisfy $KB \wedge \neg wff$. We know that `ask` will terminate and return one of these answers because resolution refutation is a decision procedure for propositional logic.

2 The Project

2.1 Project Statement

You are to use SNARK to implement and demonstrate a propositional logic version of an ontology of businesses. You are to include supermarkets such as Wegmans and Tops, and restaurants such as Burger King and Burrito Bay. You should include some other types of businesses as well.

2.2 Background

2.2.1 Category Hierarchies

Businesses can be categorized in a taxonomy (in computer science, also called a hierarchy or tree). For example, on the web site <http://www.bizbuysell.com/>, under “Business Categories” is listed 12 different categories including “RETAILING BUSINESSES”, “INTERNET BUSINESSES”, and “CONSTRUCTION BUSINESSES”, and under “RETAILING BUSINESSES” there are 18 subcategories, including “Restaurants” and “Supermarkets.” (There are other web sites that list categories of businesses, as does the Yellow Pages.)

We will treat the subcategory relation as reflexive. That means that every category is a subcategory of itself. For example, the category of restaurants is a subcategory of the category of restaurants. If we want to talk about a subcategory of A other than A itself, we will call it a **proper subcategory** of A . For example, the category of restaurants is a proper subcategory of the category of retail businesses.

An important property of hierarchies is that the subcategory relation is **transitive**. For example, since Mexican Restaurant is a subcategory of Restaurant, and Restaurant is a subcategory of Retail Business, then Mexican Restaurant

is a subcategory of Retail Business. (It is traditional in KR to use the singular rather than the plural to name a category, that is “Restaurant” rather than “Restaurants.”)

2.2.2 An Issue of Semantics

You will use propositional logic for this project. But this raises an issue of semantics. What is the intensional semantics of an atom like *Restaurant*? Above, we spoke about the category of restaurants. So is [*Restaurant*] the category of restaurants? In propositional logic, the intensional semantics of an atomic proposition is a sentence, something that can be True or False. The category of restaurants is not a sentence, and can't be True or False. So, if we're to retain the usual flavor of propositional logic, we'd have to imagine that we're talking about a particular thing, and [*Restaurant*] would be “*It's a restaurant*”. Then $MexicanRestaurant \Rightarrow Restaurant$ makes sense—*If it's a Mexican restaurant then it's a restaurant*”. The transitivity of the subcategory relation is captured by the transitivity of \Rightarrow . That is, $[(A \Rightarrow B) \wedge (B \Rightarrow C)] \Rightarrow (A \Rightarrow C)$. Similarly, the reflexivity of the subcategory relation is captured by the reflexivity of \Rightarrow . That is, $A \Rightarrow A$.

2.2.3 Instances

In our commonsense understanding of the world of businesses, we not only have categories of businesses, we also have particular, individual businesses. For example, we not only have the category of Burger King restaurants, we also have the particular, individual Burger King restaurant in the UB Commons. We say that that Burger King restaurant is **an instance** of the category of Burger King restaurants. Things can work out if we are careful in our intensional semantics. For example, we can declare the semantics of some atomic proposition, say *UBCommonsBK*, be “*It is the Burger King restaurant in the UB Commons*”. Then, if [*BurgerKing*] is “*It is a Burger King restaurant*”, the wfp $UBCommonsBK \Rightarrow BurgerKing$ makes sense, and accords with our understanding of the world.

2.2.4 Partitions and Other Category Groupings

If *B*, *C*, and *D* are proper subcategories of *A*, we say that they **exhaust** *A* if every proper subcategory of *A* must be a subcategory of *B*, *C* or *D*.

We say that a set of categories are **mutually exclusive** or **mutually disjoint** if any subcategory of one is not a subcategory of the others.

We say that a set of categories, B_1, \dots, B_n **partition** a category *A*, if: B_1 , and \dots , and B_n are proper subcategories of *A*; B_1 , and \dots , and B_n exhaust *A*; and B_1 , and \dots , and B_n are mutually disjoint. For example, on <http://www.bizbuysell.com/>, the three categories of Building Construction, Heavy Construction, and Special Trades Construction, seem to partition the category of CONSTRUCTION BUSINESSES.

2.2.5 Multiple Inheritance

Although hierarchies usually form a tree (every category has at most one immediate supercategory), there may be multiple ways to categorize a domain, so that one category winds up being an immediate subcategory of several supercategories, one in each way of categorizing the domain. For example, besides the 18 subcategories of retail businesses listed on <http://www.bizbuysell.com/>, we can also divide retail businesses into national chains, regional chains, and single businesses. Taco Bell is a national chain Mexican restaurant, Mighty Taco is a regional chain Mexican restaurant, and Burrito Bay is (as far as I know) a single business. Businesses can also be categorized as either corporations, partnerships, or individually owned businesses. One can imagine a partnership-owned, regional chain Mexican restaurant. You are to include some such multiple categorizations in your project.

2.3 The Task

You are to create a knowledge base of an ontology of businesses that includes at least 50 atomic propositions. You must include the main taxonomy of businesses, plus at least two other categorizations such as the chain/single and ownership categorizations mentioned in the previous paragraph. There must be at least three places where a group of sibling categories partitions their supercategory, and at least one place where a group of sibling categories are mutually exclusive without exhausting their supercategory. For each group of sibling subcategories, you must indicate

in a comment whether the group is a partitioning, is mutually exclusive, and/or exhausts its supercategory, and you must have domain rules that enforce those properties.

Your KB must do a reasonable job of spanning the kinds of businesses there are. You may collect your data from any reference source, and you must cite your source properly in your paper.

You are to include a demonstration of your system that shows the `ask` function answering questions that you devise. The questions you demonstrate should be carefully chosen to show off your KB and the abilities of `ask` and `SNARK`. They must include:

1. at least one question correctly answered `True`;
2. at least one question correctly answered `False`;
3. and at least one question correctly answered `Unknown`.

If A , B , and C are wfps, not necessarily atomic, your questions must include at least one of each of the following forms for which the answer is `True`:

4. $A \Rightarrow B$;
5. $A \Rightarrow \neg B$;
6. $A \wedge \neg B \Rightarrow C$;
7. $A \Rightarrow B \vee C$.

Do not use the same three categories for all of these.

Your questions must also include questions that show that your KB is designed correctly for proper subcategories, exhaustive subcategories, and mutually exclusive categories, and that you can make proper use of combinations of the several categorizations. As much as possible, your questions should require your system to use the transitivity property of the subcategory relation.

2.4 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 Microsoft Word or LaTeX, 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.)” [2]

The paper is due at the start of class, on the date specified in the title of this document.

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. You are to submit your program by 10:30 AM on the date due, so that you can get to class by 11:00 AM to hand in the paper.

Note that “your program” is the KB file and the sequence of calls to `query`. In fact, for ease of testing and debugging, your KB file can end with the sequence of calls to `query`, as does `/projects/shapiro/CSE563/CarPoolWorld.cl`. Name your file `proj1KB.cl`.

2.4.1 The Paper

Your paper should have the following parts:

1. Descriptive title
2. Author identification
3. Introduction: general description of the project

4. Domain: English description of the domain your program works in
5. Formalism
 - (a) Syntax and intensional semantics of atomic propositions, ordered alphabetically
 - (b) English and formal presentation of information in the KB
6. Demonstration run, properly commented and formatted
7. Acknowledgments as needed
8. References as needed

2.5 Grading

Project grading will be according to the following table.

	CSE 463	CSE 563
KB		
KB has at least 50 categories	6	2
KB has the main business taxonomy with a reasonable spread of businesses	6	3
KB has at least 2 other categorizations	6	2
KB has at least 3 partitions	6	4
KB has some mutually exclusive, non-exhaustive categories	5	4
Correct representation of subcategory	2	3
Correct representation of exhaustive subcategories	2	3
Correct representation of mutually exclusive subcategories	2	3
KB Subtotal	35	24
Questions		
At least one False question	3	2
At least one Unknown question	3	2
True question of form $A \Rightarrow B$	3	3
True question of form $A \Rightarrow \neg B$	3	3
True question of form $A \wedge \neg B \Rightarrow C$	3	3
True question of form $A \Rightarrow B \vee C$	3	3
Questions show subcategories	3	4
Questions show exhaustive subcategories	4	4
Questions show mutually exclusive categories	4	4
Questions show combinations of categorizations	3	4
A substantial number of questions requires the use of transitivity	3	4
Questions Subtotal	35	36
Paper		
Paper format	2	3
General project description	4	6
English description of domain	5	6
Syntax and semantics of atomic propositions	6	7
Description of formalization of KB	6	7
Presentation of demonstration run	5	7
Acknowledgments and References	2	4
Paper Subtotal	30	40
Project Total	100	100

Acknowledgments

Prof. Shapiro appreciates the advice of Richard Waldinger, SRI International, on the SNARK function (`new-row-context`) that flushes clauses generated after the assertions.

References

1. Mark E. Stickel, Richard J. Waldinger, & Vinay K. Chaudhri, "A Guide to SNARK", <http://www.ai.sri.com/snark/tutorial/tutorial.html>.
2. Stuart C. Shapiro, "UB CSE2/563," <http://www.cse.buffalo.edu/~shapiro/Courses/CSE563/2007/>.