Project 1
CSE 4/563, Knowledge Representation
Using SNARK for an Ontology of Grocery-Store Products
Due Monday, March 6

Professor Shapiro

February 10, 2006

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 the products available in a typical large grocery store, 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/snark`
3. Load `/projects/shapiro/CSE563/ask`
4. Change the Common Lisp listener's package to `snark-user`.
5. Load a knowledge base.
6. 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>).
;   Fast loading /projects/shapiro/CSE563/ask.fasl
; Running on #x8316cefd at 2006-02-22T10:16:01

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 the SNARK to implement and demonstrate a propositional logic version of an ontology of products available in a large grocery store such as Wegmans, Tops, or Martin’s.

2.2 Background

Grocery products can be categorized in a taxonomy (in computer science, also called a hierarchy or tree). For example, on their web page [2], Wegmans lists “Fresh Products”, and underneath lists “Baked Goods”, “Cheese”, “Meat”, “Produce”, and “Seafood”. (They also list “Food Safety” there, but that doesn’t really seem to belong.) You can click on “Produce”, and then on “vegetable encyclopedia”, and see a list of 89 vegetables. If you then click on “roots”, you get a list of several varieties of roots. It seems that the vegetable list could be better organized. For example, “Bibb Lettuce” and “Boston Lettuce” are both on the list, but link to the same page, where they are categorized, along with other items on the vegetable list, as “salad greens”.

An important property of hierarchies is that the subcategory relation is **transitive**. For example, since salad greens are a subcategory of vegetables, and vegetables are a subcategory of fresh products, then salad greens are a subcategory of fresh products. Since you will be using propositional logic for this project, you can represent the proposition that

salad greens are a subcategory of vegetables using implication, such as $SaladGreens \Rightarrow Vegetables$, and transitivity will be provided by the logic.

If b, c, and d are subcategories of a, we say that they **exhaust** a if any subcategory of a must be b, c, d, or some 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, $b_1, \ldots, b_n$ **partition** a category a, if: b_1, and $\ldots$, and b_n are subcategories of a; b_1, and $\ldots$, and b_n exhaust a; and b_1, and $\ldots$, and b_n are mutually disjoint. For example, baked goods, cheese, meat, produce, and seafood seem to partition the category of fresh products.

Besides the categorization of fresh products discussed above, some grocery items can be categorized as organic or non-organic, and as having an origin that is either local, New York State, domestic, or imported. For example, the category of locally grown, organic salad greens would include locally grown, organic bibb lettuce, and would be a subcategory of locally grown products, organic products, and salad greens. Of course these large grocery stores stock many items other than fresh products. For example, they also stock canned and boxed items, paper goods, pharmacy items, even furniture. You should consider all of these in your project.

2.3 The Task

You are to create a knowledge base of grocery products that includes at least 50 categories of products. You must include the main taxonomy of grocery-store products, plus at least two other categorizations such as the organic/non-organic and origin categorizations mentioned in the previous paragraph. There must be at least three places where a group of sibling categories partitions their supercategory. Wherever appropriate, your KB must indicate which sets of categories partition their supercategory, which exhaust their supercategory, and which siblings are mutually exclusive.

Your KB must do a reasonable job of spanning the kinds of products available at a large grocery store like Wegmans, Tops, or Martin's. You may collect your data yourself, or from any reference source. However, if you do use a reference source, you must cite it 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. $A \Rightarrow B \vee C$.

They must include questions that show that your KB is designed correctly for 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 KB.

2.4 Deliverables

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

> "hand in a conference-style 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.)" [3]

The paper is due at the start of class,
11:00 AM, Monday, March 6, 2006.

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, Monday, March 6, 2006 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 `ask`. 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.`*`ext`*, for some *`ext`*.

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 semantics of propositions
   (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 product taxonomy with a reasonable spread of products	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 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. Wegmans, “Welcome to Wegmans Food Markets Inc.,” `http://www.wegmans.com/`.

3. Stuart C. Shapiro, “UB CSE2/563,” `http://www.cse.buffalo.edu/~shapiro/Courses/CSE563/2006/`.