

Project 2
John's Shopping Trip
CSE 4/563, Knowledge Representation
Due Tuesday, November 9, 2010

Professor Shapiro

October 14, 2009

1 Project Statement

This project is adapted from Russell & Norvig's AI texts [1,2]. You are to specify and formalize a domain sufficient to answer a series of questions about the information in the sentence (which I will call sentence S):

Yesterday, John went to the Tops Maple Road supermarket and bought five pounds of potatoes and a pound and a half of ground beef.

You are then to load the knowledge base (KB) and your formalization of S into SNARK [3, 4], and have it answer the following questions. Each question is followed by the correct answer in square brackets.

1. *Is John a child or an adult?* [Adult]
2. *Does John now have at least two potatoes?* [True]
3. *Did John buy any meat?* [True]
4. *If Mary was buying potatoes at the same time as John, did he see her?* [True]
5. *Are the potatoes made in the supermarket?* [No]
6. *What is John going to do with the potatoes?* [Eat them]
7. *Does Tops sell deodorant?* [True]
8. *Did John bring any money to the supermarket?* [True]
9. *Does John have less money after going to the supermarket?* [True] [1,2]

2 Discussion

You need to formulate your KB so that the questions can be answered correctly, but you must abide by the following rules:

1. Your KB must not contain any fact that is just an instance of one of the questions.
2. Each question must require at least two rule firings (resolution steps).
3. Background knowledge must be in terms of general categories which contain the individuals mentioned in S .

4. Your background knowledge need not be strictly true, but generally reasonable and appropriate for answering the questions.

For example, do not put into your KB the fact that John is an adult. Instead think about why you might conclude from S that John is an adult, and put that information in your KB. Similarly, don't put into your KB the fact that Tops sells deodorant, but categorize deodorant into some class of goods that supermarkets sell, and don't put into your KB the fact that potatoes are not made in the supermarket, but include some categories of goods that are made in supermarkets, as well as some that are not. Also, don't change any question to make it closer to S . For example, question 7 asks about "Tops", not about "the Tops Maple Road supermarket". Think about the relation between "the Tops Maple Road supermarket" and "Tops", and the relation between "the Tops Maple Road supermarket" and supermarkets.

You should not ask the literal question, "Is John a child or an adult?" (If you did, the answer should just be "True".) You also should not do this in two questions, "Is John a child?" and "Is John an adult?". Instead, you might ask what age bracket John is in.

Notice that time is involved in this project, for example,

"Yesterday..." [S]
"Mary was buying ... at the same time as ..." [Question 4]
"What is John going to do ..." [Question 6]
"Did John bring any money ..."[Question 8]
"... have less money after going ..." [Question 9]

You should take time seriously in every statement, question, and KB rule for which it's at all relevant. One exception to strict adherence to time is Question 6. Since John bought the potatoes yesterday, we actually don't know that he hasn't eaten them already. So interpret "is going to do" as future relative to the time he bought the potatoes, rather than future relative to today.

Notice that to answer question 2, you will probably need to weigh a sample potato, and have SNARK do some numerical calculation. Here's an example of SNARK doing arithmetic, using the `query` front-end:

```
snark-user(270): (query
                  "What is the product of 2 and 3? [6]"
                  '(= ?x ($$product 2 3))
                  :answer '(= ?x ($$product 2 3)))
```

```
What is the product of 2 and 3? [6]
  (ask '(= ?x ($$product 2 3)) = (= 6 ($$product 2 3)))
```

Some useful SNARK numerical functions are: `$$sum`, `$$difference`, `$$product`, `$$quotient`, `$$less`, and `$$greater`. You can find a complete list in the comments in the file `/projects/shapiro/CSE563/snark-20080805r009/src/code-for-numbers2.lisp`.

Bear in mind:

1. SNARK is a resolution refutation reasoner;
2. resolution refutation is only a semi-decision procedure for FOL;
3. `query` uses `ask`;
4. `ask` is given a conjecture (`query`), and perform these steps:
 - (a) it asks SNARK to prove the conjecture;
 - (b) if SNARK does prove the conjecture, `ask` returns `True`;
 - (c) if SNARK does not prove the conjecture, `ask` asks SNARK to prove the negation of the conjecture;
 - (d) if SNARK does prove the negation of the conjecture, `ask` returns `False`;

- (e) if SNARK does not prove the negation of the conjecture, ask returns Unknown;
- 5. since SNARK is only a semi-decision procedure, if the conjecture is not provable, step 4a may never terminate, and if it does terminate with failure, but the negation of the conjecture is not provable, step 4c may not terminate.

3 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(s), user name(s), 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.

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 `proj2KB.cl`. 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.

Your KB file is to include English and SNARK versions of the formalization of your domain, followed by a series of calls to `query`. Each `query` comment should include an English version of the query, and an English summary of the reasoning your program goes through to answer the query. An example of a properly formatted file, though in Propositional Logic and without the English descriptions of the reasoning steps, is in `/projects/shapiro/CSE563/Examples/SNARK/CarPoolWorld.cl`. A copy of that file with the reasoning steps properly described is here:

```
;;; Domain Rules for Propositional CarPool World
;;; for Loading into SNARK
;;; Stuart C. Shapiro
;;; February 2, 2004
;;; Modified for new query front end, defined in ask.cl 2/22/06
;;; Changed "iff" to "<=>" and "implies" to "=>"
;;;     because either set is allowed by SNARK,
;;;     and the latter symbols are what is used in the CSE 4/563 lecture notes
;;;     as "computer-readable" syntax.]
;;;     Stuart C. Shapiro February 13, 2008

;;; Atomic Propositions
;;; =====
;;; [BettyDrivesTom] = Betty drives Tom.
;;; [BettyIsTheDriver] = Betty is the driver.
;;; [BettyIsThePassenger] = Betty is the passenger.
;;; [TomDrivesBetty] = Tom drives Betty.
;;; [TomIsTheDriver] = Tom is the driver.
;;; [TomIsThePassenger] = Tom is the passenger.

(unless (find-package :snark-user)
  (load "/projects/shapiro/CSE563/ask"))
(in-package :snark-user)
(initialize)
```

```

(mapc #'assert
      '(
;;; Betty is the driver if and only if Betty is not the passenger.
      (<=> BettyIsTheDriver
         (not BettyIsThePassenger))

;;; Tom is the driver if and only if Tom is not the passenger.
      (<=> TomIsTheDriver
         (not TomIsThePassenger))

;;; If Betty drives Tom
;;;   then Betty is the driver and Tom is the passenger.
      (=> BettyDrivesTom
         (and BettyIsTheDriver TomIsThePassenger))

;;; If Tom drives Betty
;;;   then Tom is the driver and Betty is the passenger.
      (=> TomDrivesBetty
         (and TomIsTheDriver BettyIsThePassenger))

;;; Tom drives Betty or Betty drives Tom.
      (or TomDrivesBetty BettyDrivesTom)
      ))

;;; Sample Questions
(query
 "Ask if ``Tom is the driver or Betty is the driver``
  is logically entailed by the KB. (True.)
 Tom drives Betty or Betty drives Tom.
 If Tom drives Betty, then Tom is the driver,
 and if Betty drives Tom, then Betty is the driver.
 Therefore Tom is the driver or Betty is the driver."
 '(or TomIsTheDriver BettyIsTheDriver))

(query
 "Ask if ``Betty is the driver
  implies that Betty is not the passenger``
  is logically entailed. (True.)
 Betty is the driver iff Betty is not the passenger.
 Therefore, Betty is the driver implies that Betty is not the passenger."
 '(=> BettyIsTheDriver (not BettyIsThePassenger)))

(query
 "Ask if ``Tom drives Betty and Tom is the passenger``
  is logically entailed. (False.)
 If Tom drives Betty, then Tom is the driver.
 Tom is the driver iff Tom is not the passenger.
 Therefore Tom is not the passenger,
 which contradicts the proposition that Tom is the passenger."
 '(and TomDrivesBetty TomIsThePassenger))

```

```
(query
"Ask if ``Betty drives Tom`` is logically entailed by the KB. (Unknown.)
  These domain rules are satisfied by both
  the model in which Betty drives Tom,
  and the model in which Tom drives Betty.
  Therefore neither `Betty drives Tom` nor `Betty doesn't drive Tom`
  is satisfied by both models."
' BettyDrivesTom)
```

3.1 The Paper

Your paper should have the following parts:

1. Descriptive title
2. Author identification
3. Introduction: general description of the project
4. Domain and Knowledge Base
 - (a) Syntax and semantics of atomic symbols
 - (b) English and formal presentation of information in \mathcal{S} and in the KB
5. Demonstration run, properly commented and formatted
6. Acknowledgments as needed
7. References as needed

4 Grading

Project grading will be according to the following table.

	CSE 463	CSE 563
KB		
The portion of the KB used for each question follows the guidelines of Sect. 2 (9×4)	36	36
Statement \mathcal{S} is reasonably formalized	9	10
KB Subtotal	45	46
Questions		
Each question is reasonably formalized (9×3)	27	27
Each question is correctly answered ($9 \times (2 \text{ or } 1)$)	18	9
Questions Subtotal	45	36
Paper		
Paper format	1	2
General project description	1	2
English description of KB	2	3
Presentation of syntax and semantics of atomic symbols	3	5
Formatting & presentation of demonstration run	1	2
Correct use of quotation and citations	1	2
Acknowledgments and References	1	2
Paper Subtotal	10	18
Project Total	100	100

Acknowledgments

Prof. Shapiro appreciates the advice of Richard Waldinger and Mark Stickel, SRI International, on the SNARK arithmetic functions.

References

1. Stuart Russell & Peter Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall, Upper Saddle River, NJ, 1995, pp 263–264.
2. Stuart Russell & Peter Norvig, *Artificial Intelligence: A Modern Approach, Second Edition*, Prentice Hall, Upper Saddle River, NJ, 2003, pp 372–373.
3. Mark E. Stickel, SNARK - SRI's New Automated Reasoning Kit, <http://www.ai.sri.com/~stickel/snark.html>.
4. Mark E. Stickel, Richard J. Waldinger, & Vinay K. Chaudhri A Guide to SNARK, <http://www.ai.sri.com/snark/tutorial/tutorial.html>.
5. Stuart C. Shapiro, "UB CSE2/563," <http://www.cse.buffalo.edu/~shapiro/Courses/CSE563/2009/>.