

# CSE 4/563 Knowledge Representation

Professor Shapiro

## Homework 7

Maximum Points: 32

Due: 10:30 AM, Wednesday, March 25, 2009

March 18, 2009

Put your answers in a file named `hw7.ext`, for an appropriate value of `ext`. **Include your name at the top of the file.** Submit that file by executing the Unix command

`submit_cse463 hw7.ext`

or

`submit_cse563 hw7.ext`

whichever is appropriate for you. The file can be a text file, or produced by some word processing software, but it must be formatted so it is easy to read.

You are also to submit a file as instructed in question (3), below.

1. (16) Formalize the following domain in Standard Full First-Order Predicate Logic using the syntax of SNARK wffs. You must have one and only one sentence of FOL (SNARK assertion) for each sentence. Make sure that SNARK issues no warnings when you load your assertions.

- (a) The way someone warns someone on some day is by telling them on that day that there's danger on that day.
- (b) If someone tells someone something on some day, then it's true if and only if they are not a liar.
- (c) If someone  $x$  tells someone  $y$  something  $p$  on some day  $d$ , then  $y$  believes  $p$  if and only if  $y$  doesn't believe that  $x$  is a liar.
- (d) Bob takes evasive action on any day if and only if he believes that there's danger on that day.
- (e) If there's danger on some day, a person gets injured on that day if and only if they do not take evasive action then.
- (f) Neither Larry nor Teri are liars.
- (g) Bob believes that Larry is a liar, but doesn't believe that Teri is a liar.
- (h) Teri warns Bob on Bob's 25th birthday, and Larry warns Bob on the following day.

Note:

- You will need a representation of days, and you will need to decide which predicates and which functions take days as arguments. The technical term for a function or predicate that takes a time as one of its argument is a **fluent**. Such a predicate is called a **propositional fluent**, and other such functions are called **functional fluents**. It is traditional to make the time the last argument of a fluent.

- In this homework, you will need to represent time at the **granularity** of days, but you will not need any finer nor any larger granularity of time. That is, you will not need to distinguish hours, minutes, seconds, nanoseconds, etc. You will also not need to represent different weeks, months, years, centuries, etc.
  - Although one can believe a proposition on one day, and not believe it on some other day, for simplicity in doing this homework, let's assume that all beliefings are independent of time.
  - You only need to represent categories of objects when it will make a difference in the reasoning tasks you will perform (or ask the program to perform). In this homework, you do not need to represent any categories. Specifically, you do not need to represent the category of days nor of persons.
2. (10) Give the syntax and intensional semantics of all the atomic symbols you used in your formalization in the following categories of symbols. (You needn't mention symbols that are built into SNARK.)

### **Individual Constants**

### **Functional Fluents**

### **Non-Fluent Functions**

### **Propositional Fluents**

### **Non-Fluent Predicate Symbols**

3. (6) Use the `query` function of the `ask` program, in the form

```
(query <comment>
      <query>
      :answer <pattern>
      :printProof t)
```

to answer the following questions based on the knowledge base consisting of your formalization of the above sentences.

- When does Bob not get injured?
- When does Bob get injured?

Include in your answer file a transcript of the output produced by your two calls to `query`.

Add a comment in the form

```
; From sentence (x)
```

after each “Row *n*” that labels a row from an assertion, showing the sentence (a–h) that that row comes from. If you have formalized the sentences correctly, there should be at least one assertion from each of the 8 sentences.

**Continued on next page.**

**Also** submit the file of your SNARK assertions and your calls to `query`. Name this file `hw7kb.cl`. It should have the following format:

```
;;; <Name and other identifying information>

(load "/projects/shapiro/CSE563/ask" )

(in-package :snark-user)
(initialize)

<Commented assertions for the sentences in question (1)>

(query "When does Bob not get injured?"
       <query>
       :answer <pattern>
       :printProof t)

(query "When does Bob get injured?"
       <query>
       :answer <pattern>
       :printProof t)
```