CSE 4/563 Knowledge Representation
Professor Shapiro
Homework 5
Maximum Points: 12
Due: 10:30 AM, Wednesday, February 28, 2007

February 21, 2007

Put your answers in a file named hw5.*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 hw5.`*`ext`*

or

`submit_cse563 hw5.`*`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.

Whenever you run a programmed reasoner include a transcript of the run in your answer file, and append a clear English statement saying what you conclude from your run, and why.

1. (3) Use `predicate-wang` to show that the domain rules given in the lecture notes for the finite-model, **single-car** CarPool World logically entail the wff $Driver(Betty) => Passenger(Tom)$.

2. Using the syntax of `ubdecreasoner`, complete the formalization of the 2-Car CarPool World in finite-model predicate logic, that was started in the lecture notes.

   - Assume that: all four people, Betty, Tom, Jane, and John are commuting; they are using two cars; there is one driver and one passenger in each car.
   - Remember that `decreasoner`'s syntax for $\forall x P(x)$ is `[x]P(x)`, and `decreasoner's` syntax for $\exists x P(x)$ is {x}`P(x)`.
   - `Decreasoner` has equality (and inequality) available as infixed predicate symbols without the need for you to declare their types. For example, the atomic formulae `(commuter1 = commuter2)` and `(commuter1 != commuter2)` are syntactically correct, and may be used.
   - `Decreasoner` makes the **unique names assumption**. That is, for any two different individual constants, c_1 and c_2, `decreasoner` assumes that `(`c_1 `!=` c_2`)`. You do **not** need to include domain rules like `Tom = Tom` and `Betty != Jane` in your program.
   - You are to use **only** the predicate symbols: `Drives`, `Driver`, `Passenger`, `=`, and `!=`.
   - Notice that there should be 12 models. This may be calculated as follows:
     (a) Think about one of the commuters, say Tom. There are 3 other commuters Tom might be in the car with.
     (b) For each of these 3 possibilities, Tom might be the driver or the passenger. That makes 6 possibilities.
     (c) For each of these 6 possibilities, there are two commuters left for the other car, and either of them might be driving. That makes 12 possibilities.

   (a) (6) List your input file for `ubdecreasoner`. Be sure that every declaration of a predicate is preceded by a comment that gives the intensional semantics of the predicate, and that every domain rule is preceded by a comment that gives an English interpretation of the rule.

   (b) (3) Show your run of `ubdecreasoner` here: