Assignment 4: due Wed, Oct 15, 2008

Reminder: The assignment is due at 11:59pm on Wednesday, October 15. To hand in the assignment, upload it to the course directory using the `provide` online submission system. For programming questions, use your favorite programming language but comment extensively and make sure your code compiles and runs without major errors. Include a makefile or a readme file with compiling instructions. If you don't manage to finish the programming part of the assignment, and so don't have running code, hand in what you have for partial credit.
Please use one file per question, unless noted otherwise.

Question 1 – Programming a theorem prover – 50 pts

This part of the assignment is a reinforcement of the class first-order logic concepts, and programming practice. You will write a backward-chaining theorem prover for first order knowledge bases in rule (definite clause, Horn clause) form.

Try to do this on your own, but by all means consult the AIMA code base for implementation ideas if you become stuck.

1. Your theorem prover needs to have a basic data structure to represent a rule (definite clause). The rule should have antecedents (premises, a body) and a consequent (conclusion, head). You must be able to represent facts using the same data structure. We will call the data structure a `Rule`. Another data structure called a `KnowledgeBase` consists of any number of `Rule`s. Write code for these data structures.

2. You now need methods to convert input text of a particular form into a `KnowledgeBase` of `Rule`s and to convert a `Rule` into output text. You can assume that text comes in a form of one string per rule, and the string is in some reasonable form (e.g., "IF Blah(x) THEN Something(x)" or "Blah(x)→ Something(x)", or a similar syntax, pick one). Write input/output methods for your `KnowledgeBase`.

3. Now you can write a main function or method that takes as input a file from which to read the knowledge base, and a string that represents a clause you'd like to prove, and returns `true` or `false`. It should return `true` if the clause is entailed by the knowledge base, and `false` otherwise[1]. It should print a log of its steps as it executes. It will use the backward chaining algorithm (AIMA p.288) to try to prove the intended conclusion.

4. The backward chaining function above calls two other methods that need to be implemented. The first one, `StandardizeApart`, makes sure that there are no variable name conflicts in a `Rule`, and renames variables as necessary. The second one, `Unify` returns a substitution that is a most general unifier of two `Rule`s (AIMA p.278), if one exists, and fails otherwise.

At this pint, you should have a fully functioning backward-chaining theorem prover. To test it, write a small KB using well-formed clauses of the kind your program can parse, then ASK it with two different queries that you expect to be true. Please hand in the test KB and the output of your theorem prover in addition to your code.

Continued on next page.

[1]Of course, first-order entailment is semidecidable, so it may never stop. There isn't much we can do about that.

Question 2 – Probability revision – 50 pts

This part of the assignment is a probability refresher. We will use probability extensively in the second part of the course, so it's good to remember the basics. In class, I will assume that you understand this material.

Read Sections 13.2-3 (probability notations and axioms) and 13.6 (Bayes' rule). Other good sources: http://www-math.bgsu.edu/ albert/m115/probability/outline.html and http://betterexplained.com/articles/an-intuitive-and-short-explanation-of-bayes-theorem/.

1. (10 pts) A fair coin has been tossed n times. What's the probability of k heads coming up, where $0 \leq k \leq n$?

2. (10 pts) The Chevalier de Mere bets he can get a "6" in four rolls of a fair die. If he get a "6" in four throws, you give him a dollar. If he does not, he gives you a dollar. Do you want to play? Explain. If you play this game 1,000 times, how much money can you expect to win or lose?

3. (10 pts) Alice, a cognitive science student, has conducted an experiment measuring her friends' response times to a visual stimulus. Here's a table of her raw data:

person	1	2	3	4	5	6	7	8	9	10
time (msec)	420	360	445	462	423	405	398	490	474	448

   What's the sample mean and variance of this distribution?

4. (10 pts) Two contestants are playing the following game, very famous in probability theory. There are three closed doors. Behind one there is a big prize, behind two others there is nothing. The contestant's objective is to guess which door leads to the prize. The game host always follows the same procedure: he asks the contestant to make a guess, then he opens another door to reveal nothing behind it and asks the contestant if he or she wants to change their guess. Contestant Bob refuses to change his guess. What's the probability of him winning the prize, given all the information so far? Contestant Mary changes her guess. What's the probability that she wins the prize? Assume contestants' guesses are independent. Assume also that the prized is placed behind each door with equal probability. If the host has a choice of which door to open, he opens them with equal probability.

5. (10 pts) Wumpusphobia is a rare condition in which the patient suffers from delusions of being pursued by a horrible smelly monster. One in a hundred persons has wumpusphobia. 90% of people with wumpusphobia have shaky hands. Only 5% of people without wumpusphobia have shaky hands. What's the probability that a person selected uniformly at random has wumpusphobia, given that she has shaky hands?