Assignment 5

CSG120, Fall 2003
Due: Thursday, Nov. 20

Part I. Pencil-and-paper exercises

1. Do Exercise 14.2, parts a-d only (pp. 533-534).

2. For the Bayes net in Figure 14.11 (p. 510), use either the method of enumeration or the method of variable elimination to compute exactly the conditional probability $P(Rain = true | Sprinkler = true, WetGrass = true)$. You must show all your steps to receive credit. (You will get a chance to check the correctness of your final result when you do Problem 4 below.)

Part II. Computer exercises

Look in the `programs/bayes-net` subdirectory of the course web site. Examine the files `test-net.lisp` and `burglar-alarm-net.lisp` there to see how to represent a Bayes net with discrete-valued variables for use with a Lisp program provided for you as well as a Lisp program you'll write in Exercise 5.

3. Follow these examples to create a corresponding Lisp representation of the Bayes net shown in Figure 14.11 (the same one you've used in Problem 2 above). Turn in the Lisp file for this network.

4. Now load into Lisp the files `enumeration.lisp` and `bayes-net-utils.lisp`, together with the network you've created in Problem 3. Then use the function `query-by-enumeration` to compute the following three conditional probabilities:

- $P(Rain = true | Sprinkler = true, WetGrass = true)$
- $P(Rain = true | Sprinkler = false, WetGrass = true)$
- $P(Rain = true | WetGrass = true)$

Turn in dribble output showing the corresponding function calls and their results, and then indicate exactly what these 3 probabilities are. (The results will be 3 *distributions*, but you are to explicitly provide 3 *numbers*.) Do the relative values of these 3 probabilities make sense intuitively? Explain briefly.

5. Write a program that accepts a Bayes net, a query variable X, and a vector of evidence variables and their values $(\mathbf{E}, \mathbf{e})$, and performs stochastic simulation with likelihood weighting to compute the conditional distribution $\mathbf{P}(X | \mathbf{E} = \mathbf{e})$ approximately. It should also accept as a parameter the number of samples to generate. The Bayes net and evidence arguments should be encoded exactly as they are for the `query-by-enumeration` function in `enumerate.lisp`. For simplicity, you may assume that all variables take on only the values `true` and `false`. (The main way this simplifies the program is that it is a little more involved to generate discrete random values from a set of $n > 2$ possible values.)

Run your program to compute $P(Rain = true | Sprinkler = true, WetGrass = true)$. Perform some informal experiments with different numbers of samples and collect dribble output from these experiments. Comment on any variability you observe in these results and the extent to which these results differ from that obtained from exact calculation. Turn in this dribble file along with your source code for this program and your comments on the results.

Suggestion for writing this program: Have your code call functions from `bayes-net-utils.lisp` wherever possible and simply load this file with your file to run your program. Particular functions from that file you are strongly encouraged to take advantage of are `get-values`, `topological-sort-vars`, `normalize`, `prob-given-parents`, and `get-val`. You may also find it helpful to examine `enumerate.lisp` closely and borrow ideas from that code.