CMSC 671 (Introduction to AI) – Fall 2016

Homework 3: CSPs, Games, and Probabilities (70 points)
Due: 10/17 at 11:59pm.
Turnin: Blackboard.

Please submit Parts I–IV together as a **single PDF file** named *lastname*_hw3.pdf, with parts clearly marked and delineated, and Part IV as a zipfile named *lastname*_hw3.zip containing python code.

All files must start with your last name and have your full name(s) in the file, at/near the top.

You are **encouraged** to work on Parts I-IV of this homework assignment in groups of up to three students. If you work in a group, you only need to turn in one shared solution, with everyone's name on the assignment (files should be named after the person who submits). Students in the group will receive the same grade on the assignment. If you work in a group, you must actually work on the problems *as a group* and produce group solutions. Don't have each member work independently on one part of the assignment, then submit the collection of independent solutions.

PART I. CONSTRAINT SATISFACTION EXERCISE (10 POINTS)

Assignment: provide a fully worked solution to Russell & Norvig, Exercise 6.5. *You must show and explain clearly each step of the solution in order to receive full credit.*

PART II. GAME PLAYING: FLIPPING A COIN (10 POINTS)

Consider a two-player coin-flipping game where two players alternate flipping a two-sided coin. The rules are as follows:

- If the coin lands heads up, the player who flipped gains two points.
- If the coin lands tails up, the opponent gains one point.
- If a player exceeds four points, they automatically lose all of their points, and the game ends.
- On their turn, a player can choose to stop the game, in which case both players keep their current scores.
- The goal is to beat the other player **by as many points as possible**.

Assignment: Answer the following questions about playing this game.

1. Draw the 4-ply (two moves for each player) expectiminimax tree for this problem.
2. Using the static evaluation function (score(player1) - score(player2)), back up the leaf values to the root of the tree.
3. What is the best action for the first player to take? (Play or stop?)
4. If player 1 flips tails, what should player 2 do? Why? (1-3 sentences)
5. Would you describe this game as fair? Why or why not? (1-3 sentences)

PART III. BAYES' NETS AND PROBABILITY (30 POINTS)

An ecological engineering research team has noticed that private school buses (B) consume more gas (G), but are involved in fewer accidents (A) than the national average. They have constructed this Bayes' net (Fig. 1):

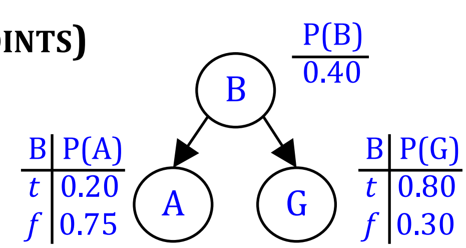

Figure 1. Bayesian network for private school bus, accidents, and gas usage.

Part A: Compute the following, showing all steps.

1. Compute P(a, ¬b, g) using the chain rule.

2. Compute P(a) using inference by enumeration.
3. Using conditional independence, compute P(¬g, a | b) and P(¬g, a | ¬b).
4. Then, use Bayes' rule to compute P(b | ¬g, a).

After further study, the team determines that two types of schools use private buses: very small public schools (S) and charter schools (C). After collecting some statistics, they construct the following Bayesian network (Fig. 2):

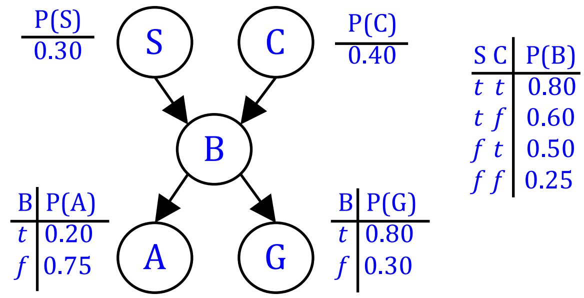

Figure 2. Bayes' net for private buses, accidents, and gas usage for charter and small schools.

Part B: Compute the following, showing all steps.

5. Compute P(a, ¬b, g) using the chain rule.
6. Using the chain rule, compute the probability P(¬g, a, b, s, ¬c).
7. Write, in summation form, the *formula* for computing P(a, ¬c) using inference by enumeration. (You do not need to actually compute the probability.)
8. Using the rules for determining when two variables are (conditionally) independent of each other in a Bayes' net, answer the following (true or false) for the BN given in Figure 2:
9. $(S \perp\!\!\!\perp G) =$
10. $(C \perp\!\!\!\perp A \mid B) =$
11. $(S \perp\!\!\!\perp C) =$
12. $(S \perp\!\!\!\perp C \mid B) =$
13. $(S \perp\!\!\!\perp C \mid A) =$

Part IV. Calculating Probabilities (20 points)

Assignment: For the Bayesian network in Figure 14.2 in the textbook, implement a program that computes and prints out the probability of any conjunction of events given any other conjunction of events. This executable should take six arguments, in the order A, B, E, J, M, with values as follows:

0	given false	1	given true		
2	query false	3	query true	4	unspecified

In this formulation, 0 and 1 mean a variable is known to be true or false; 2 and 3 mean we are looking for the probability of a variable being true or false; and 4 means none of the above.

Some examples:

(A,B,E,J,M)	Calculate	Meaning
(2,4,3,4,4)	P(¬A ∧ E)	P(Alarm=*f* ∧ Earthquake=*t*)
(3,4,2,1,4)	P(A ∧ ¬E | J)	P(Alarm=*t* ∧ Earthquake=*f* | JohnCalls=*t*)
(2,1,0,3,4)	P(J ∧ ¬A | B ∧ ¬E)	P(JohnCalls=*t* ∧ Alarm=*f* | Burglary=*t* ∧ Earthquake=*f*)
(2,3,4,4,0)	P(B ∧ ¬A | ¬M)	P (Burglary=*t* ∧ Alarm=*f* | MaryCalls=*f*)

Your code should use the probability values in the tables in 14.2, and appropriate formulas to evaluate the probability of the specified event(s). It is OK to hardcode values from the tables in your code, but not values for all possible command arguments, or probability values for all possible atomic events.