CMSC 671 – HW 2, Fall 2022

Turnin: Blackboard.
Please submit Parts I, II, and IV as a **single PDF file** named *yourlastname*_hw2.pdf.
Please submit Part III as a **.py** file, named *yourlastname*_hw2.py.

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

PART I. CONSTRAINT SATISFACTION

For these questions, you will be solving a map-coloring problem on the map shown here. Each of the regions A–E must be green, blue, or red, and no adjacent regions may be the same color.

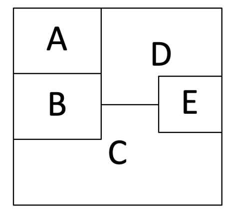

Figure 1. Color this map blue, green and red, with no adjacent areas sharing a color.

Backtracking search

You will use backtracking search to find a solution to this constraint satisfaction problem. The **variable ordering heuristic** is to make assignments to variables in **minimum remaining value** order, breaking ties **alphabetically by variable name**; the **value ordering heuristic** is to consider the values in the order blue, green, red. For each node, indicate all variables with, e.g., *A=r*, *B=g*, etc. Uninstantiated variables can be shown as, e.g, A=. (See Figure 2.) At each layer of the search tree, show only nodes for the variable you are assigning a value to—you do not have to draw all assignments of values to the remaining variables.

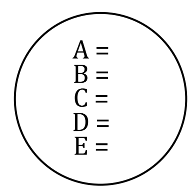

Figure 2. Start node.

Assignment: (10 pts)

1) Show the search tree created by following this algorithm, and circle the solution node.

Scheduling

You have been asked to schedule seven computer science professors to give talks in your department. Unfortunately, you discover that there are only four available dates, meaning that some of the talks will necessarily overlap. To make things more complicated, your advisors insist on being able to see certain talks, which means you have to figure out how to schedule the talks such that nobody is disappointed. The following are all the constraints you find when trying to create the schedule:

The list of guest lecturers consists of Drs. Brian, Chris, Linda, Nina, Peter, Sally, and Tom. The available dates are November 1st, 2nd, 3rd, and 4th.

- Tom can only speak on November 1st.
- Linda and Chris want to see each others' talks and so cannot overlap.
- Your advisor wants to see all of Linda, Peter, and Nina, so they cannot overlap.
- Brian, Chris, and Nina have been competing to be first to prove that P=NP and all dislike each other intensely. They must be scheduled for different days so they don't run into each other.
- The department chair wants to be able to see all of Tom, Linda, and Nina.

- Sally and Peter cannot be scheduled for the same day, as a particular professor wants to take them each to lunch.
- Chris and Peter cannot travel on the same days, so they cannot overlap.

Assignment: (20 pts)

2) Give a CSP formulation for this problem:

  a) What are the *random variables*? (There are multiple options; choose carefully.)

  b) What is the *domain of each variable*?

  c) Express the constraints using the domain for all variables. Spell out constraints—do not just state the constraints in English sentences.

3) Draw a constraint network that captures all the constraints. Include labels on nodes and edges.

4) Give a full legal instantiation for this problem, or explain why none exists.

5) Explain how you found the solution. What search approach did you use? How did you order variables and values? Did you do constraint propagation? Forward checking? Give details.

PART II. BAYES' NETS AND PROBABILITY

Consider the Bayesian belief network shown in Figure 3, with the following conditional probabilities:[1]

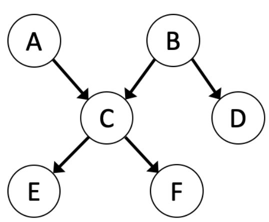

Figure 3. A six-variable Belief Network with associated CPTs.

$P(a) = 0.9$

$P(b) = 0.2$

$P(c \mid a,b) = 0.1$
$P(c \mid a,\neg b) = 0.8$
$P(c \mid \neg a,b) = 0.7$
$P(c \mid \neg a,\neg b) = 0.4$

$P(d \mid b) = 0.1$
$P(d \mid \neg b) = 0.8$

$P(e \mid c) = 0.7$
$P(e \mid \neg c) = 0.2$

$P(f \mid c) = 0.2$
$P(f \mid \neg c) = 0.9$

Assignment: (25 pts)

6) Compute P(e) using variable elimination. You should first prune irrelevant variables. Show the factors that are created for a given elimination ordering.

---

[1] Question drawn from Mackworth & Poole, Artificial Intelligence: Foundations of Computational Agents, 2nd Edition.

7) 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 3:

a) (B ⫫ E) =

b) (E ⫫ F | C) =

c) (A ⫫ B) =

d) (A ⫫ B | C) =

e) (F ⫫ D) =

PART IV. CALCULATING PROBABILITIES PROGRAMMATICALLY

Assignment: (25 pts)

For the Bayesian network shown in Figure 3, implement a program in Python 3 that computes and returns the probability of any **conjunction** of events given any other conjunction of events. Your function should take five arguments, in the order A, B, C, D, E, F, 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 observed to be false or true; 2 or 3 means we are querying the probability of assigning false or true to that variable; and 4 means the variable doesn't appear in the probability statement we are calculating.

Some examples:

(A,B,C,D,E,F)	Calculate	Meaning
(3,4,4,4,4,4)	P(A)	P(A=*t*)
(2,4,3,4,4,4)	P(¬ A ∧ C)	P(A=*f* ∧ C=*t*)
(3,4,2,1,4,4)	P(A ∧ ¬C | D)	P(A=*t* ∧ C=*f* | D=*t*)
(2,1,0,3,4,4)	P(D ∧¬A | B∧¬C)	P(D=*t* ∧ A=*f* | B=*t* ∧ C=*f*)
(2,3,4,4,0,4)	P(B ∧ ¬A | ¬E)	P(B=*t* ∧ A=*f* | E=*f*)
(4,4,1,4,3,2)	P(E ∧ ¬F | C)	P(E=*t* ∧ F=*f* | C=*f*)

Your code should use the probability values in the tables shown, 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 possible arguments or probability values for all possible atomic events.

There are possible combinations of values that don't resolve to meaningful probability queries; be sure to test for these as error cases and handle them gracefully.

PART III. GAME PLAYING: FLIPPING A COIN

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: (15 points)

8) Given the expectiminimax algorithm:

a) Draw the 4-ply (two moves for each player) expectiminimax tree for this problem.

b) Using the static evaluation function (score(player1) - score(player2)), back up the leaf values to the root of the tree.

9) Consider the actions taken by each player:

a) What is the best first action for the first player to take? (Play or stop?)

b) If player 1 flips tails, what should player 2 do? Why? (1-3 sentences)

c) Would you describe this game as fair? Why or why not? (1-3 sentences)