(PRINT) Name ____Student No__________

Signature ____Total Mark_____/100

University of Waterloo

Computer Science 486/686 – Introduction to Artificial Intelligence

Midterm Test
2015 June 19
Time: 4:35 pm – 5:50 pm

Time: 75 minutes
Total marks: 100

Answer all questions on this paper. No aids are permitted (i.e., no book, no notes, no calculator, no computer).

This examination has 8 pages. Check that you have a complete paper.

1	**/ 22**
2	**/ 24**
3	**/ 24**
4	**/ 30**
Total	**/ 100**

1) [22 pts] Every term, the university must design a schedule for the final exams. Ideally the schedule should be conflict free, meaning that students should not have to write two exams simultaneously.

a) **[6 pts]** Consider 5 students (s_1, s_2, s_3, s_4 and s_5) and 5 courses (c_1, c_2, c_3, c_4 and c_5) such that s_1 and s_2 are taking c_1; s_1, s_3 and s_4 are taking c_2; s_2 and s_4 are taking c_3; s_3 is taking c_4; and s_4 and s_5 are taking c_5. Suppose that the 5 courses must be scheduled in 3 time slots t_1, t_2 and t_3. Describe how you would encode this scheduling problem as a constraint satisfaction problem. List the variables and their domain as well as the constraints.

variables: c_1, c_2, c_3, c_4, c_5

domain: $c_i \in \{t_1, t_2, t_3\}$

constraints : $c_1 \neq c_2$, $c_1 \neq c_3$, $c_2 \neq c_4$, $c_2 \neq c_3$, $c_2 \neq c_5$, $c_3 \neq c_5$

b) **[6 pts]** Suppose that you use backtracking search with the most constrained variable and least constraining value heuristics. Show the search tree expanded by backtracking search until a satisfying assignment is found for the CSP in a). Indicate in which order the nodes are expanded in the search tree.

c1 = t1 (all variables are equally constrained so start with any variable)
|
c2 = t2 (c2 and c3 can only take two values)
|
c3 = t3 (c3 can only take one value)
|
c5 = t1 (c5 can only take one value
|
c4 = t1 (c4 is the only variable left)

no backtracking necessary

c) [10 pts] In some cases, there is no way to avoid all conflicts so the goal is to find the schedule with the smallest number of conflicts. Describe **two** algorithms to find a schedule with a minimum number of conflicts.

Several algorithms are possible. Here are two possible algorithms:

1) Perform IDA* search. Nodes are partial assignments. Costs are the number of students that have a conflict. Goal nodes are complete assignments. Instead of using a heuristic function that estimates the future cost, simply assign the least constraining value to the most constrained variable at each step.

2) Perform WalkSat. Start with a random complete assignment. If there is a conflict, change the time slot assigned to a course as long as the number of conflicts is reduced. Otherwise, with probability p, change time slot of the course that yields the smallest number of conflicts and with probability 1-p, randomly change the time slot of a course. Repeat until time is up. Return the schedule with the smallest number of conflicts found.

2) [24 pts] Complete the following table, stating advantages and disadvantages of various search algorithms. Indicate whether the search space is exponential or linear in the length of the path. Indicate which algorithms make use of arc costs, which algorithms are guaranteed to find a solution (when one exists) and which algorithms are guaranteed to find the shortest or least-cost path. Assume that arc costs are strictly positive and that the branching factor is finite. Assume that each algorithm is executed *without* cycle checking and *without* multiple path checking, but do not assume that heuristics are admissible or consistent.

	Depth-First	**Breadth-First**	**Iterative Depth-First**	**Greedy Best-first**	**A***	**IDA***
Space (Exp/Linear)	**L**	**E**	**L**	**E**	**E**	**L**
Considers Cost (Yes/No)	**N**	**N**	**N**	**Y**	**Y**	**Y**
Guarantees a solution (Yes/No)	**N**	**Y**	**Y**	**N**	**N**	**Y**
Guarantees shortest/lowest cost path (Yes/No)	**N**	**Y**	**Y**	**N**	**N**	**N**

3) [24 points] Are the following statements true or false? No justification required.

a) Value of information may be negative.

False, more information never hurts hence its value is never negative

b) Backtracking performs a kind of iterative deepening search.

False, there is no depth limit that we iterate over

c) Value iteration for MDPs and the forward-backward algorithm for HMMs are instances of variable elimination.

True, they are instances of variable elimination with specific elimination orderings

d) A heuristic is admissible when it underestimates the true cost.

True, admissible heuristics always underestimate the true cost

e) Multiplying utilities by a positive or negative constant does not change the decision problem.

False, multiplying by a negative constant changes the decision

f) In a Bayesian network, two variables are independent when there is no arc between them.

False, variables are dependent is there is an open path

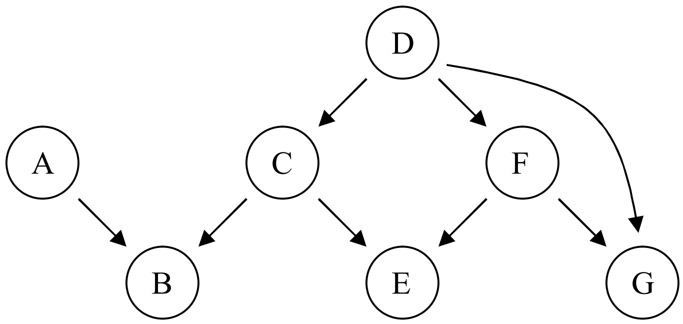
g) A hidden Markov Model is a dynamic Bayesian network.

True, an HMM is a DBN with a single state variable and a single observation variable

h) In a decision network, the utility node cannot be the parent of any other node.

True, utility nodes cannot be the parent of any other node

4) [30 pts] You have just been hired as a consultant for a car manufacturer. The company would like to improve their maintenance service at its dealerships by assisting the mechanics with an automated fault diagnosis tool. After talking with several experts, you've built the following Bayesian network. Each node is a Boolean variable that corresponds to the status (e.g., working or not) of a car component and each edge indicates a probabilistic dependency for failure.



$f_1(A) = Pr(A)$
$f_2(B,A,C) = Pr(B|A,C)$
$f_3(C,D) = Pr(C|D)$
$f_4(D) = Pr(D)$
$f_5(E,C,F) = Pr(E|C,F)$
$f_6(F,D) = Pr(F|D)$
$f_7(G,D,F) = Pr(G|D,F)$

Mechanics will typically query the network by asking for the probability of failure of some component given the status of other components. You've decided to implement the variable elimination algorithm to compute those probabilities.

a) **[9 pts]** Suppose a mechanics would like to know Pr(C|A=true,E=false) and your variable elimination algorithm eliminates variables in the order B-D-F-G, show the factors created and removed at each step. *Do not ignore irrelevant variables: eliminate each variable.*

i) Restrict evidence variables: *new factors:* $\mathbf{f_8()=f_1(A=true), f_9(B,C)=f_2(B,A=true,C), f_{10}(C,F)=f_5(E=false,C,F)}$
deleted factors: $\mathbf{f_1, f_2, f_5}$

ii) Eliminate B: *new factors:* $\mathbf{f_{11}(C) = \Sigma_B\, f_9(B,C)}$

deleted factors: $\mathbf{f_9}$

iii) Eliminate D: *new factors:* $\mathbf{f_{12}(C,F,G) = \Sigma_D\, f_3(C,D)\, f_4(D)\, f_6(F,D)\, f_7(G,D,F)}$

deleted factors: $\mathbf{f_3, f_4, f_6, f_7}$

iv) Eliminate F: *new factors:* $\mathbf{f_{13}(C,G) = \Sigma_F\, f_{10}(C,F)\, f_{12}(C,F,G)}$

deleted factors: $\mathbf{f_{10}, f_{12}}$

v) Eliminate G: *new factors:* $\mathbf{f_{14}(C) = \Sigma_G\, f_{13}(C,G)}$

deleted factors: $\mathbf{f_{13}}$

answer: Pr(C|A=true,E=false) = $\boldsymbol{\alpha}\, \mathbf{f_8()\, f_{11}(C)\, f_{14}(C)}$

b) [6 pts] Since mechanics want to know the answers to their queries in real-time, the implementation of variable elimination should be as efficient as possible. Can you come up with a better elimination ordering than B-D-F-G for the query Pr(C|A=true,E=false)? If yes, how will the running time be improved by your new ordering?

Answer: yes
Possible new ordering: G, F, D, B
Running time: all intermediate factors depend on at most 2 variables (in contrast to 3 variables in part a), which decreases the running time accordingly.

c) [6 pts] In a car, it is often the case that some components have no influence on the failure of others. Suppose the mechanics is trying to determine the status of component C. Since he cannot directly observe C, he is considering to run some tests that will determine the status of A and E in order to better infer the status of C. Does the status of C depend on the status of A and E? Justify your answer.

C depends on E since there is a link between C and E
C does not depend on A since the path between C and A is blocked (B is not observed)

d) [9 pts] At the end of variable elimination, is it ok to normalize the answer when computing

i. an inference query? Explain briefly.

yes, since probabilities can always be normalized, this is ok.

ii. an expected utility query to find the optimal policy of a decision node? Explain briefly.

no, since utilities may be negative, so normalizing may change the optimal decision

iii. an expected utility query to find the value of some information gathering action? Explain briefly.

no, value of information should not be normalized since it does not generally sum to 1.