(PRINT) Name Jason Sun Student No 20387090

Signature Jason Sun 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).

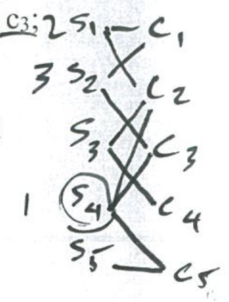
This examination has 8 pages. Check that you have a complete paper.

1	15 / 22
2	20 / 24
3	24 / 24
4	24 / 30
Total	83 / 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.

t_1 t_2 t_3

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 t_1, t_2, t_3

domain c_1, c_2, c_3, c_4, c_5

constraints $s_i \in c_j \in t_k \; != \; s_i \in c_l \in t_k$

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.

s_4 is most constrained. 3 edges out: c_2, c_3, c_5
then s_1, s_2, s_3 then s_5.
c_4 is least constraing value, only: s_3.
then c_1, c_3, c_5 then c_2.

start empty {∅} $\xrightarrow{\text{assign } s_4}$ $t_1 = \{c_2\}$, $t_2 = \{c_3\}$, $t_3 = \{c_5\}$ $\xrightarrow{\text{assign } s_2}$ $\{c_2\}$ $\{c_3, c_1\}$ $\xrightarrow{\text{assign } s_2}$ bad state. c_1, c_3..

assign {c_5}

assign s_1 → $\{c_2\}$ $\{c_3\}$ $\{c_5, c_1\}$ $\xrightarrow{s_2}$ same $\xrightarrow{s_3}$ $\{c_3\}$ $\{c_5, c_1\}$

bad stat $\{c_2, c_4\}$

s_3 ↓ $\{c_2\}$ $\{c_3, c_4\}$ $\{c_5, c_1\}$ $\xrightarrow{s_5}$ same

end stat

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.

(1)

Breadth first search on assignments;
min_seen = ∅ //
loop

return assignment ~~stop~~ if conflicts = 0
min_seen ← min (min_seen, current_node.conflicts)
else Keep searching

return min_seen assignment

not accounting for conflicts

-5

(2)

~~Iterative Depth First Deepening A*~~

~~define h(n) = most constrained and least constraining~~
~~g(n) =~~

~~Itp~~ Iterative Depth First Search.
min_seen = (root)
Search(n) until all assigned or error state
min_seen = min (min_seen, ~~current~~ n)
assign n_to (n)

search (n)

return min_seen

incomplete explanations

-2

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)	~~Exp~~ Linear	Exp	Linear ~~Exp~~	Linear	Exp	Exp
Considers Cost (Yes/No)	No	No	~~Yes~~ No	Yes	Yes	Yes
Guarantees a solution (Yes/No)	No	Yes	~~Yes~~ No	No	No ~~Yes~~	Yes
Guarantees shortest/lowest cost path (Yes/No)	No	Yes	Yes	No	No	Yes

20

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

a) Value of information may be negative.

~~No~~ False

b) Backtracking performs a kind of iterative deepening search.

~~No~~ F

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

~~Yes~~ T

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

T

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

F

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

F

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

T

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

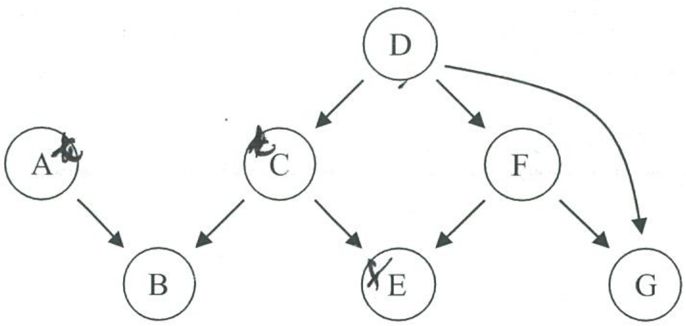
T

✓

24

Bravo!

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.



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

i) ~~$f_8(B,C)$~~ $f_9(E)$
ii) $f_{10}(C)$
iii) ~~$f_{11}(F)$~~ $f_{12}(G,F)$
iv) ~~$f_{13}(G)$~~
v) 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:* $f_8(B,C)$ given A $f_9(E)$ given E

deleted factors: $f_1(A)$ given A, $f_2(B,A,C)$ restrict A, $f_5(E,C,F)$ restrict E

5/9

ii) Eliminate B: *new factors:* $f_{10}(C)$

deleted factors: $f_8(B,C)$

iii) Eliminate D: *new factors:* $f_{11}(F)$ $f_{12}(G,F)$

deleted factors: $f_6(F,D)$ $f_7(G,D,F)$ $f(D)$

iv) Eliminate F: *new factors:* $f_{13}(G)$

deleted factors: $f_{11}(F)$ $f_{12}(G,F)$

v) Eliminate G: *new factors:*

deleted factors: $f_{13}(G)$

answer: Pr(C|A=true,E=false) = $\alpha f_1(A)\ f_9(E)\ f_{10}(C)$

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?

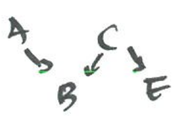
B-G-F-D ordering by eliminating leafs first

4/6

running time ~~linear~~ because tree pruning off ~~states~~ factors instead of generating more (from query of parents first)

smaller intermediate factors

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.



Status of A not ~~bee~~ inferrable from A.
i.e. A is useless.

6/6

status of E has C as parent, outcome of E is dependent on C, can infer A better.
i.e. E is useful.

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, it gives us probability to inference query.

9/9

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

No. normalizing utility isn't meaningful, and can cause wrong answers. i.e. $U(s) = \{0\}$'s. or negative utilities

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

No. Value, when normalized, loses meaning. I.e. it is no longer a value-of-information query result.

Although it can provide a decision still, when a binary decision variable.