6.s058/16.420
Representation, Inference and Reasoning in AI

Midterm Exam

Solutions

November 1, 2021

Answer the questions in the spaces provided on the question sheets. If you run out of room for an answer, continue on the back of the page.

You are permitted to use a single sheet of paper with notes on (both sides), and a calculator and a timer. If you use your phone for the calculator and timer, please restrict yourself to these functions.

Name: ______________________________

Student ID: ______________________________

Question	Points	Score
1	20	
2	15	
3	20	
4	10	
5	5	
6	10	
7	8	
8	12	
Total:	100	

1 Oration

1. (20 points) Your professor asks you to organize a seminar series with 4 talks on different dates (1, 2, 3, and 4).

   - The following constraints must be satisfied

     C1. No speaker can talk on more than one date.

     C2. Each speaker can speak on a particular subset of the topics.

     C3. No two consecutive seminars can be on the same topic.

     C4. No two consecutive speakers can be from the same university.

     C5. Each date has exactly one speaker.

     C6. Each date has exactly one topic.

   - The three topics are: theory, modeling, and implementation (T, M, I).
   - The speakers are: Antonius, Bibulus, Claudius, Dracontius, and Emporius (A, B, C, D, E).
     - Antonius can speak about theory or modeling.
     - Bibulus can speak about theory, modeling, or implementation.
     - Claudius can speak about modeling or implementation.
     - Dracontius can speak about theory or implementation.
     - Emporius can speak about implementation.
     - Antonius and Emporius are from Romulus University.
     - Bibulus, Claudius, and Dracontius are from Remus University.

   (a) Consider a formulation with variables S_d indicating the speaker for each date d, and variables T_d indicating the topic for each date d.

      i. What is the value domain for the variables, taking any unary constraints into account?

      Solution: For S_d it's (A, B, C, D, E). For T_d it's (T, M, I).

      ii. Which of the constraints are automatically satisfied because of the formulation? Mark all that apply. ○ C1 ○ C2 ○ C3 ○ C4 √ **C5** √ **C6**

   (b) Consider a formulation with variables D_s indicating a date for each speaker and T_s indicating a topic for each speaker.

      i. What are the value domains for the variables, taking any unary constraints into account?

      Solution: For D it's $(1, 2, 3, 4, None)$. For T_s it's different for each speaker:
      $(T, M), (T, M, I), (M, I), (T, I), (I)$.

      ii. Which of the constraints are automatically satisfied because of the formulation? Mark all that apply. √ **C1** √ **C2** ○ C3 ○ C4 ○ C5 ○ C6

(c) Consider a formulation with variables ST_d indicating both the speaker and topic for each date.

i. What is the value domain for the variables, taking any unary constraints into account?

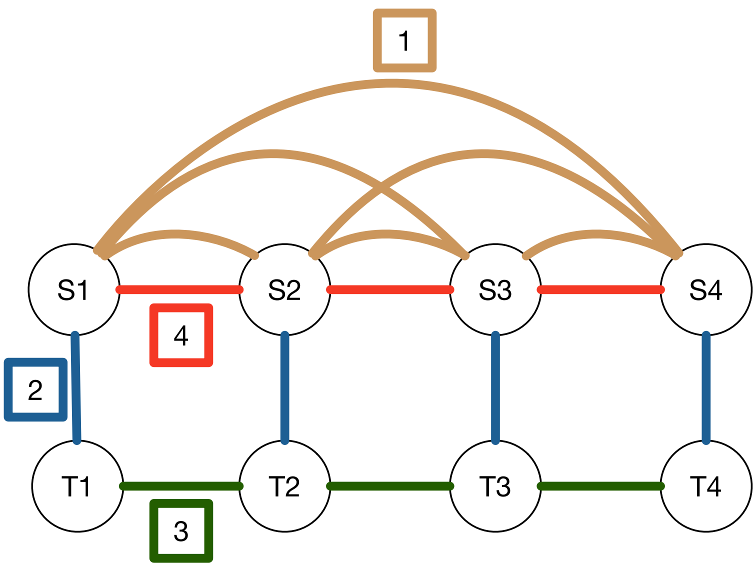
> **Solution:** $\{(A,T),(A,M),(B,T),(B,M),(B,I),(C,M),(C,I),(D,T),(D,I),(E,I)\}$

ii. Which of the constraints are automatically satisfied because of the formulation? Mark all that apply. ◯ C1 √ **C2** ◯ C3 ◯ C4 √ **C5** √ **C6**

(d) Which of the three previous formulations do you think would be most effective for a standard CSP solver, and why?

> **Solution:** The last one because it has the smallest total set of assignments and the most constraints built in. Also accepted: the first one because the constraints are already binary.

(e) In the formulation from part (a), all the constraints are binary. Here is a constraint diagram. Label each type of arc in the diagram with the constraint (1, 2, 3, 4, 5, 6) it represents.



(f) Assume we are using the formulation from part (a). We have already chosen variable S_1 and assigned it to E. After doing forward checking, which variable will be chosen, using the most-constrained-variable heuristic, and what is its domain after forward checking?

> **Solution:** Variable T_1, which has unit domain $\{I\}$.

2 Logic and proof

2. (15 points) (a) Farmer O'Dell is planting their fields, has dug several holes, and is now putting the seeds in the holes.

   i. The farmer's goal is for there to be at least one seed in each hole. Express this goal as a sentence in first-order logic with predicates *in* (arity 2), *seed* (arity 1), *hole* (arity 1).

   Solution: $\forall h.hole(h) \rightarrow \exists s.seed(s) \wedge in(s, h)$

   ii. The sentence holds in any interpretation where no objects satisfy the interpretation of *hole*.
   √ **True** ◯ False

   iii. The sentence holds in any interpretation where no objects satisfy the interpretation of *seed*.
   ◯ True √ **False**

   iv. The sentence holds in an interpretation $\mathcal{I}$, where the universe is the real numbers, $\mathcal{I}(in)$ is our standard $>$ relation, and $\mathcal{I}(seed)$ and $\mathcal{I}(hole)$ are both the set of integers.
   √ **True** ◯ False

   v. The farmer's sentence entails "If there are no holes then there are no seeds".
   ◯ True √ **False**

   vi. The farmer's sentence entails "If there are no seeds then there are no holes".
   √ **True** ◯ False

(b) Jan has a Python implementation of a resolution refutation proof procedure, designed to determine whether some sentence α entails another sentence β. Inside their implementation, which takes α and β as inputs, they convert α and $\neg\beta$ into clausal form and repeatedly apply the resolution inference rule with a 10-minute run-time cut-off.

- If resolution has derived a contradiction, it returns 1.
- If resolution has terminated without deriving a contradiction, it returns 2.
- If resolution has not terminated, it returns 3.

Jan has an additional function `f` that takes in the return code from the prover and outputs `True` or `False`.

Jan would like to implement an `f` so that the output of the combination of the prover and `f` is `True` if and only if α entails β.

i. Provide a definition for `f` that makes Jan's whole implementation **sound**, or argue that one does not exist.

Solution:

```
def f(return_code):
  return return_code == 1
```

ii. Provide a definition for `f` that makes Jan's whole implementation both **sound** and **complete**, or argue that one does not exist.

Solution: There is no sound and complete proof procedure for FOL.

3 Catching the bus

3. (20 points) Busta Rhydes needs to catch a bus in an infinite 2-d grid (from $-\infty$ to $+\infty$ in both directions). They know the bus's schedule, and want to plan a path such that they end up in the same grid location at the same time as the bus.

   Assume that the search starts at time 0 with Busta in location (0,0). On each step, time increases by 1, and Busta can move to any of the **eight** neighboring locations, or remain in the same location.

   The position of the bus is specified by a function B which takes a time as input and returns the bus's location at that time as a tuple `(r,c)`.

   (a) For the following implementations of the bus's schedule, what is one sequence of locations (representing Busta's location at time 0, 1, 2...) that might result from running a breadth-first search in this domain? Enter `None` if no path is returned, either because BFS runs indefinitely, or because it returns without having found a path. Note that there may be more than one correct answer, but you need only find one.

   i. $B_1(t) = (1, t-3)$

   Solution: Busta Path (t,r,c) = (0,0,0),(1,0,0),(2,1,-1) (Other solutions are possible)
   Bus Path (t,r,c) = (0,1,-3),(1,1,-2),(2,1,-1)

   ii. $B_2(t)$ defines a counterclockwise loop, shaped like a square of size 10 with its lower-left corner at (0,0). Bus starts at (10,0) at time 0 and advances by **ten** grid squares each time step, i.e. (10,0), (10,10), (0, 10), ...

   Solution: Busta Path (t,r,c) = (0,0,0),(1,0,0),(2,0,0),(3,0,0) (Other solutions are possible)
   Bus Path (t,r,c) = (0,10,0),(1,10,10),(2,0,10),(3,0,0)

   (b) To formalize this type of problem for a heuristic search method, for any $B(t)$:

   i. What would be a minimal definition of the state?

   Solution: $s = (t, r, c)$ where t is time and r, c is Busta's location. We can infer bus position as $br, bc = B(t)$.
   Note that the bus position is not necessary and, also, the bus position at time t does not determine the bus position at time $t+1$ and so you need to remember the time in the state.

   ii. Specify a goal test $g(s)$ using your state representation that works for any $B(t)$.

   Solution: $t, r, c = s$. If $br(t), bc(t)$ is bus location $B(t)$, then goal test is $br(t) = r$ and $bc(t) = c$.

(c) What is a useful admissible heuristic function for this problem, given you know the bus schedule is (arbitrary) $B(t)$ and that the bus moves at most K grid squares per time step? For this question, we'll say the heuristic is *useful* if it is 0 only at the goal state and is greater than 1 at at least one state. Explain your answer.

Solution: Assuming state $s = (t, r, c)$ encodes Busta's location and $br(t), bc(t) = B(t)$ is the bus location, then $h(s) = \max(|br(t) - r|/(K+2), |bc(t) - c|/(K+2))$ is an admissible heuristic. The $K + 2$ in the denominator is the maximum Manhattan displacement per step between the bus and Busta. We gave credit for different forms that had similar idea.

4 SDDL

4. (10 points) This is the Stata Definition Language! We want to be able to find our way around Stata, so we need a planner. Note: upper case symbols are **constants** and upper case predicates denote **static** facts (they are always true).

   We can represent floor plans as graphs between spaces, such as lobbies, hallways and offices. Assume we have an arbitrary **directed** graph encoded by `(CONN ?x ?y)` facts and a single `(move ?person ?x ?y)` PDDL action which updates a `(loc ?person ?x)` fluent.

```
(:action move
    :parameters (?person ?x ?y)
    :precondition (and (loc ?person ?x) (CONN ?x ?y))
    :effect (and (loc ?person ?y) (not (loc ?person ?x))))
```

   An example goal might be to go from initial state `(loc ME S)` to goal `(loc ME G)`.

   (a) How do the values of $h^{\max}(s)$, $h^{\text{add}}(s)$, and $h^{\text{ff}}(s)$ compare to the optimal path-length in the graph (as found by BFS). Indicate whether the values for each of the heuristics would in general be larger, smaller or equal to the optimal value. Also, indicate whether any of the heuristic values would be equal to each other.

   - $h^{\max}(s)$ has what relation to the optimal value? ○ larger √ **equal** ○ smaller
   - $h^{\text{add}}(s)$ has what relation to the optimal value? ○ larger √ **equal** ○ smaller
   - $h^{\text{ff}}(s)$ has what relation to the optimal value? ○ larger √ **equal** ○ smaller
   - $h^{\max}(s) = h^{\text{ff}}(s)$ √ **True** ○ False
   - $h^{\max}(s) = h^{\text{add}}(s)$ √ **True** ○ False
   - $h^{\text{add}}(s) = h^{\text{ff}}(s)$ √ **True** ○ False

   Justify your answers.

   Solution:
   $h^{\max}(s) = h^{\text{add}}(s) = h^{\text{ff}}(s)$ and all are equal to the optimal path length. The RPG essentially carries out a BFS, a fact (location) appears at level i if it takes i actions to reach there, and since there is only one goal fact, all these heuristics are equal.

Moving in and out of Stata is not so easy these days, different connections have different requirements. To go from the STATA-LOBBY to the CSAIL hallway one needs an ATTESTATION and a PHONE, while to get into an office one needs a KEY-111 for office 111. We will extend our representation from the first part of this problem to indicate how many requirements there are on a given connection by adding one more argument to `CONN`: `(CONN ?x ?y ?nr)` where `?nr` is 0, 1 or 2, indicating how many requirements there are.

```
(CONN STATA-HALLWAY STATA-111 1)
(CONN STATA-LOBBY CSAIL-HALLWAY 2)
```

We can then specify the requirements with some additional static facts:

- `(REQ1 ?x ?y ?r)`: represents one requirement for going from location `?x` to location `?y`;
- `(REQ2 ?x ?y ?r)`: represents another requirement, when there is more than one; and
- `(CONN ?x ?y ?r)`: represents a (directed) arc in the map and indicates how many requirements are needed to make the transition.

For example:

```
(REQ1 STATA-HALLWAY STATA-111 KEY-111)
(REQ1 STATA-LOBBY STATA-HALLWAY ATTESTATION)
(REQ2 STATA-LOBBY STATA-HALLWAY PHONE)
```

We also introduce a predicate `(has ?person ?r)` which indicates that the person satisfies a requirement, e.g. `(has Alexa PHONE)`.

(b) Write an operator for `?person` moving from `?loc1` to `?loc2` when the transition has two requirements `?r1` and `?r2`.

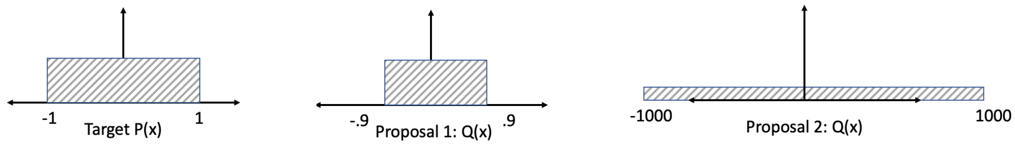
```
:action move2
    :parameters (?person ?loc1 ?loc2 ?r1 ?r2)
    ...)
```

Solution:

```
(:action move2
    :parameters (?person ?loc1 ?loc2 ?r1 ?r2)
    :precondition (and (loc ?person ?loc1) (CONN ?loc1 ?loc2 2)
                       (REQ1 ?loc1 ?loc2 ?r1) (REQ2 ?loc1 ?loc2 ?r2)
                       (has ?person ?r1) (has ?person ?r2))
    :effect (and (loc ?person ?loc2) (not (loc ?person ?loc1)))
```

5 Sampling for probabilistic inference

5. (5 points) (a) Consider the target distribution, and two proposal distributions shown below. (All three distributions are uniform over different ranges.) Which proposal distribution should you use for rejection sampling, and why?



Solution: Should use proposal 2, because it is wider than the target. Proposal 1 is slightly narrower than the target, but will not sample parts of the target, leading to a possibly biased estimate. Proposal 2 might be very inefficient in that it will generate many samples that will be rejected because it is a lot wider than the target, but the samples that it keeps will be from the target. That isn't true of Proposal 1.

(b) Is it possible to tell if you need more samples during sample-importance-resampling?
$\surd$ **Yes** ◯ No

(c) Let us assume that we are using particle filtering to estimate the $[x, y, \theta]$ pose of a robot. The prior estimate as well as the noise applied to the motion model and the sensor model are all Gaussian distributions. Will the posterior estimate of the robot pose always be Gaussian? If yes, say why. If not, say why not.

Solution: No. The motion and sensor models also have to be linear for the posterior to always be Gaussian.

6 Independence in graphical models

6. (10 points) For the following set of independence relations, draw a Bayesian network graph that encodes the relations, or show that no such graph exists.

(a) For the following set of independence relations, draw a Bayesian network graph that encodes the relations, or show that no such graph exists.

- A is always independent of B
- A is independent of C given B
- A is not independent of C

Solution: No graph exists. If A $\perp\!\!\!\perp$ C given B, then there must be a subgraph A $\rightarrow$ B $\rightarrow$ C or A $\leftarrow$ B $\rightarrow$ C. But in neither case is A $\perp\!\!\!\perp$ B.

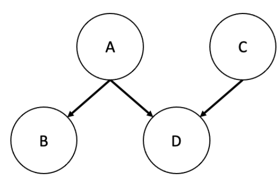
(b) For the following set of independence relations, draw a Bayesian network graph that encodes the relations, or show that no such graph exists.

- D is independent of B given A
- B is independent of C given no other information
- B is not independent of D given no other information
- B is not independent of C given D

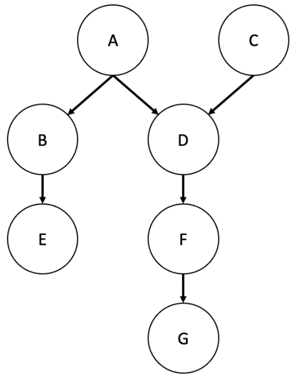
Solution:

- The second and fourth facts mean that B and C are parents of D, B $\rightarrow$ D $\leftarrow$ C.
- The first fact means that A must be between B and D, such that either B $\rightarrow$ A $\rightarrow$ D or B $\leftarrow$ A $\rightarrow$ D.

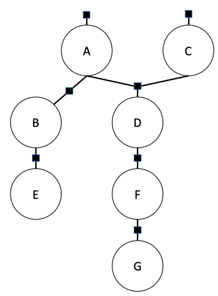
One of the solution:



(c) Consider the following Bayes net:

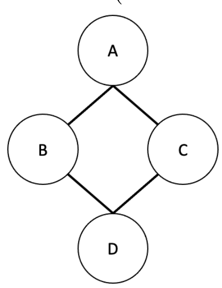


Please draw the corresponding factor graph:



Solution:

(d) Consider the following Markov random field (undirected graphical model):

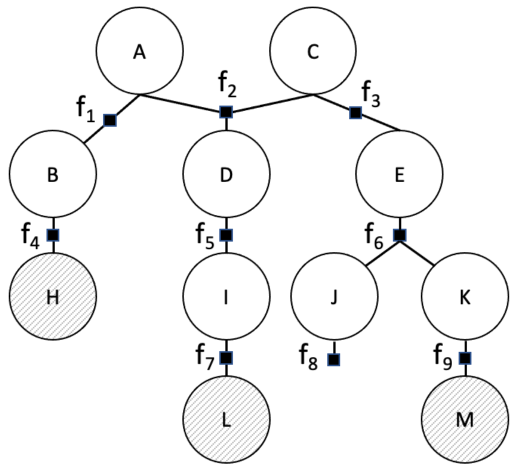


Please draw the corresponding Bayes net that preserves the same independence relations if one exists. If no Bayes net exists that can preserve the same independence relations, please explain why by specifying the smallest set of independence relations encoded by this undirected graph that together cannot be expressed by a 4-node Bayes net.

Solution: No Bayes net exists because there is no way to draw the arrows that make *both* A and D independent given B and C *and* B and C independent given A and D.

7 Message passing

7. (8 points) Consider the following factor graph, where we have shaded in the evidence nodes (variables whose values we know):



(a) If we were to run the recursive SUM-PRODUCT on this polytree, starting at node A, what is the first message that is computed and passed? Please write your answer as "From foo to bar." and remember that the evidence is multiplied into the relevant potentials before message passing starts. When expanding the child nodes, use alphabetical ordering for variables and numerical ordering for factors.

> **Solution: Post-exam note: when the exams were printed, the shading of the nodes in the graph did not render, so it was announced during the exam that no evidence should be assumed. With this assumption, the answer is:** From H to f_4.

(b) If we were to run the recursive SUM-PRODUCT on this polytree, starting at node A, please give an expression for the message received by variable E during the collect phase of SUM-PRODUCT. You can write it in terms of messages, or in terms of potentials and factors.

> **Solution:** $\mu_{f_6 \to E} = \sum_{J,K} (\phi_{f_6} \cdot \mu_{J \to f_6} \cdot \mu_{K \to f_6})$.

(c) Please give an expression for the marginal over A. You can write it in terms of messages.

> **Solution:** $P(A) = \left(\sum_{B,C,D} \mu_{f_1 \to A} \mu_{f_2 \to A}\right) / \left(|| \sum_{B,C,D} \mu_{f_1 \to A} \mu_{f_2 \to A} ||\right)$, in which case we treat $\mu_{f_1 \to A}$ as a function of A and B.
> Another acceptable solution: $P(A) = (\mu_{f_1 \to A} \mu_{f_2 \to A}) / (||\mu_{f_1 \to A} \mu_{f_2 \to A}||)$, in which case we treat $\mu_{f_1 \to A}$ as a function of just A.

(d) Once the collect phase of SUM-PRODUCT completes with messages returned to the query variable, we run a second phase of distributing the messages back out from the query variable. What quantities does the second phase allow us to compute?

> **Solution:** The collect phase computes the marginal distribution at the query variable. The distribute phase finishes computing the marginals at all the other variables.

8 HMM

8. (12 points) Consider the following Markov chain (an HMM without observations – the state is observed after every transition):

The conditional probability tables are:

X_1	$p(X_1)$
T	.25
F	.75

For X_2, X_3, X_4:

X_i	X_{i+1}	$p(X_{i+1}\|X_i)$
T	T	.25
T	F	.75
F	T	.75
F	F	.25

(a) Please compute $p(X_2 = T, X_3 = T)$. (We're looking for a specific probability value.)

Solution:

$$\begin{aligned}
p(X_2 = T, X_3 = T) &= \sum_{X_1} p(X_1)p(X_2 = T, X_3 = T|X_1) \\
&= \sum_{X_1} p(X_1)p(X_2 = T|X_1)p(X_3 = T|X_2 = T, X_1) \\
&= \sum_{X_1} p(X_1)p(X_2 = T|X_1)p(X_3 = T|X_2) \\
&= (.25 \times .25 \times .25) + (.75 \times .75 \times .25) \\
&= .15626
\end{aligned}$$

Now add observations Y_1, Y_2, Y_3, Y_4, such that

X_i	Y_i	$p(Y_i \mid X_i)$
T	T	.8
T	F	.2
F	T	0
F	F	1

(b) Infer the most likely state sequence given the observation sequence $Y_1 = T$, $Y_2 = F$, $Y_3 = F$,$Y_4 = T$. You are welcome to work out the numerical values of the messages in Viterbi! But as a hint, you can use the properties of the observation probabilities to take shortcuts on needing to compute all the messages.

Solution:

$$\delta_1 = \begin{bmatrix} .25 \times .8 \\ .75 \times 0 \end{bmatrix} = \begin{bmatrix} .2 \\ 0 \end{bmatrix}$$

$$\delta_2 = \begin{bmatrix} \max(\mathbf{.2} \times \mathbf{.25}, 0 \times .75) \times .2 \\ \max(\mathbf{.2} \times \mathbf{.75}, 0 \times .25) \times 1 \end{bmatrix} = \begin{bmatrix} .001 \\ .15 \end{bmatrix}$$

$$\delta_3 = \begin{bmatrix} \max(.001 \times .25, \mathbf{.15} \times \mathbf{.75}) \times .2 \\ \max(.001 \times .25, \mathbf{.15} \times \mathbf{.75}) \times 1 \end{bmatrix} = \begin{bmatrix} .0225 \\ .1125 \end{bmatrix}$$

$$\delta_4 = \begin{bmatrix} \max(.0225 \times .25, \mathbf{.1125} \times \mathbf{.75}) \times .8 \\ \max(.0225 \times .75, \mathbf{.1125} \times \mathbf{.25}) \times 0 \end{bmatrix} = \begin{bmatrix} .0675 \\ 0 \end{bmatrix}$$

$\psi_4 = T$

$\psi_3 = F$

$\psi_2 = F$

$\psi_1 = T$