6.4110/16.420
Representation, Inference and Reasoning in AI

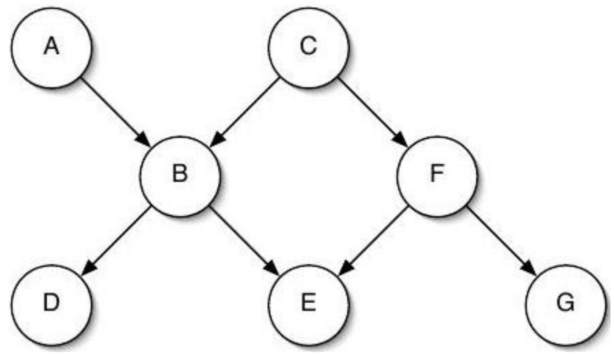
Extra practice problems

Danger—Warning—Not carefully debugged

Old homework and exam questions from the archive

1 Bayesian Network

Consider the following Bayesian network:



(a) Is it a polytree? **No**

(b) Is A independent of C? **Yes**

(c) Is C independent of E? **No**

(d) Is D independent of C? **No**

(e) Name a variable that, if it were an evidence variable, your answer to the question in part (b) would be different, or say that there is no such variable. (So, if your answer to (b) was that they are independent, then name a variable X for which A is not conditionally independent of C given X.) **B (makes A and C dependent)**

(f) Name a variable that, if it were an evidence variable, your answer to part (c) would be different, or say that there is none. **No such (single) variable (2 variables B,F)**

(g) Name a variable that, if it were an evidence variable, your answer to part (d) would be different, or say that there is none. **B**

(h) If all the nodes are binary, how many parameters would be required to specify all the CPTs in this network? (Remember that if p is specified then it is not necessary to specify $1-p$ as well.) **16**

(i) Give an expression for $\Pr(D \mid C)$ given probabilities that are stored in the CPTs. Don't include any unnecessary terms.

Solution:

$$\Pr(D \mid C) = \sum_b \sum_a \Pr(D \mid b) \times \Pr(b \mid a, C) \times \Pr(a)$$

(j) What factor is created if we eliminate B first in the course of using variable elimination to compute $\Pr(A \mid G)$?

Solution: $f\{A, C, D, E, F\}$ There are many correct answers to this problem because A is independent of G.

(k) What is the Markov blanket of B? A, C, D, E, F

(l) Imagine that you're doing likelihood weighting to compute $\Pr(E = e \mid A = a)$. What weight would you have to assign to sample $\langle a, b, c, d, f, g \rangle$? $P(A = a)$

2 Surveillance

You are performing surveillance, trying to decide which destination in a harbor a particular submarine is headed toward. At each time step, the situation can be characterized by the following variables:

destination Which destination the submarine is headed to.

location The submarine's current location.

observation Your noisy observation (via underwater sensing) of the submarine's location

action The submarine's actions (speed and steering)

In addition, there is a variable, **type**, which encodes the type of the submarine (which is useful to know, because it affects the sub's speed and maneuverability).

(a) Draw a dynamic Bayesian network diagram that describes this system. Only the **observation** variable is directly observable. Show how the values of the variables at the current time step depend on their values in the previous time step. Your model should be able to encode these relationships:

- Submarines tend to stick with the same destination, and don't frequently change which one they're aiming at.
- The choice of action depends on relative position of the submarine and its destination.

$$Parents(destination_{t+1}) = \{destination_t\}$$
$$Parents(location_{t+1}) = \{location_t, action_t\}$$
$$Parents(observation_{t+1}) = \{location_{t+1}\}$$
$$Parents(action_{t+1}) = \{destination_{t+1}, location_{t+1}\}$$

Solution:

(b) What would you change in your diagram in order to model the idea that some types of submarines have different ranges and that the distance to a location might affect its selection as a destination?

Solution: Add type as a parent of all the D_t variables. Note that we only need a single static type variable.

(c) If you have made three observations, O_1, O_2, and O_3, give an expression for the probability distribution over the destination at step 3:

$$\Pr(D_3 \mid O_1, O_2, O_3)$$

using only probabilities stored in the CPTs of the network (in part (a)).

Solution:

$$\Pr(D_3 \mid O_1, O_2, O_3) = \frac{\sum_{L_1,L_2,L_3,A_1,A_2,A_3,D_1,D_2} \Pr(D_3, O_1, O_2, O_3)}{\sum_{L_1,L_2,L_3,A_1,A_2,A_3,D_1,D_2,D_3} \Pr(O_1, O_2, O_3)}$$
$$= \frac{\sum \Pr(D_3, O_1, O_2, O_3)}{\sum \Pr(O_1, O_2, O_3)} = \frac{\sum \Pr(DBN)}{\sum \Pr(DBN)}$$

$$DBN = \left\{ \begin{array}{l} \Pr(D_3 \mid D_2) \times \Pr(D_2 \mid D_1) \times \Pr(D1) \times \\ \Pr(L_3 \mid L_2, A_2) \times \Pr(L_2 \mid L_1, A_1) \times \Pr(L1) \times \\ \Pr(O_3 \mid L_3) \times \Pr(O_2 \mid L_2) \times \Pr(O_1 \mid L_1) \times \\ \Pr(A_3 \mid D_3, L_3) \times \Pr(A_2 \mid D_2, L_2) \times \Pr(A_1 \mid D_1, L_1) \end{array} \right\}$$

3 Logic

1. **Entailment**

   Let S_1 and S_2 be sentences in propositional logic, let I_1 be the set of interpretations that make S_1 true, and let I_2 be the set of interpretations that make S_2 true. Assume that I_1 is a subset of I_2. Mark all of the statements which *must* be true in this case:

   - √ **S_1 entails S_2.**
   - ◯ S_2 entails S_1.
   - ◯ S_2 can be proven from S_1 with a sound proof system.
   - √ **S_2 can be proven from S_1 with a complete proof system.**
   - √ **The sentence $(S_1 \rightarrow S_2)$ is satisfiable.**
   - √ **The sentence $(S_1 \rightarrow S_2)$ is valid.**
   - ◯ The sentence $(S_2 \rightarrow S_1)$ is satisfiable.
   - ◯ The sentence $(S_2 \rightarrow S_1)$ is valid.

2. **Propositional Proof**

   Prove that the sentence

   $$(P \vee Q \vee R) \wedge (\neg P \vee Z) \wedge (\neg Q \vee W) \wedge \neg R \wedge \neg Z \wedge \neg W$$

   is unsatisfiable.

   1. Explain the method you are going to use.

      Solution: If it's unsatisfiable, a contradiction will follow from it directly. So, just convert it into clausal form and do resolution.

   2. Do the proof.

      Solution:

      (a) $P \vee Q \vee R$

      (b) $\neg P \vee Z$

      (c) $\neg Q \vee W$

      (d) $\neg R$

      (e) $\neg Z$

      (f) $\neg W$

      (g) $P \vee Q$ (1, 4)

      (h) $\neg Q$ (3, 6)

      (i) P (7, 8)

      (j) Z (9, 2)

      (k) **False** (5, 10)

4 Itinerary

We're all looking forward to when we can travel again! So, let's think about how to make a flight search engine... just like Travelocity only (much, much) smaller.

A flight is specified with a starting city and time and an ending city and time. We want to be able to answer queries that are specified with a starting city, starting time, final destination city and deadline time. The objective is to find a sequence of flights, starting in the starting city, leaving sometime at or after the starting time, and arriving at the final destination before (or at) the deadline.

Assume you have a database D consisting of tuples (c_o, t_o, c_d, t_d) specifying the city and time where the flight originates and the site and time where it arrives at its destination.

How should we formulate this as a path-search problem? Please specify:

1. State space

   Solution: current city (one of the cities in the DB) and current time

2. Action space

   Solution: all flights in the database

3. Successor function

   Solution: Given state (c, t) and action (co, to, cd, td), if c = co and t <= to, then next state is (cd, td) else it is (c, t).

4. Goal test

   Solution: If goal is (cg, tg) then state (c, t) satisfies goal if c = cg and t <= tg.

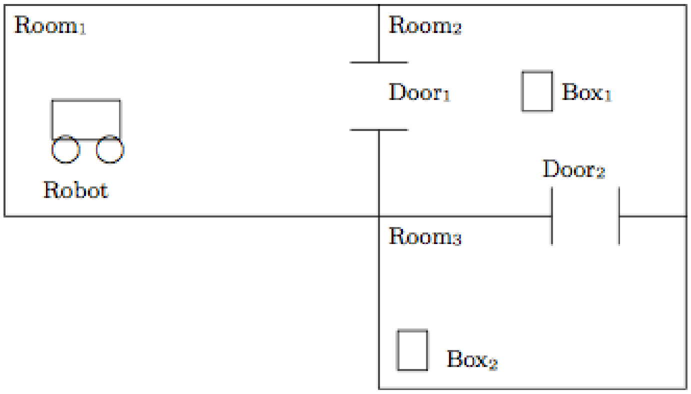
Pat wants to find a way to get from Rome at time 1 to Istanbul at the earliest time possible. They propose to repeatedly call a path search algorithm on the formulation above, with different deadlines, to accomplish this goal. Sketch a strategy that Pat could use. As an additional challenge, try to minimize the number of times your strategy calls the search procedure.

Solution: Let t range from the initial time to the last time a flight arrives at Istanbul in the database. Try to find a solution for every t, going from t+1 to the last flight to Istanbul.

Smarter would be to sort the arrival times of flights in Istanbul from earliest to latest and only try those (in order). Smarter still would be some sort of bisection search.

5 Shakey the Robot

Consider the following planning domain consisting of a robot pushing boxes between connected rooms.



We will represent this domain with the following symbols

- `b1, b2` the two boxes; `r1,r2,r3` the three rooms, `d1, d2` the two doors.
- `open(?X)` – door `?X` is open.
- `in(?X,?Y)` – box `?X` is in room `?Y`.
- `robin(?X)` – the robot is in room `?X`.
- `join(?X,?Y,?Z)` – door `?X` joins rooms `?Y` and `?Z`.

1. Specify the initial state shown in the figure. Assume that the doors start out closed.

Solution:

```
(robin r1)
(in b1 r2)
(in b2 r3)
(join d1 r1 r2)
(join d1 r2 r1)
(join d2 r2 r3)
(join d2 r3 r2)
```

2. Give a STRIPS representation of the following action. Use the above symbols to specify sensible preconditions and effects for the action `pushthru(?B,?R1,?R2,?D)`, meaning the robot pushes box `?B` from room `?R1` to room `?R2` via door `?D`.

Solution:

```
(:action pushthru
 :parameters (?D ?B ?Rx ?Ry)
 :precondition (and (open ?D) (in ?B ?Rx) (robin ?Rx) (join ?D ?Rx ?Ry))
 :effect (and (in ?B ?Ry) (not (in ?B ?Rx)) (robin ?Ry) (not (robin ?Rx))))
```

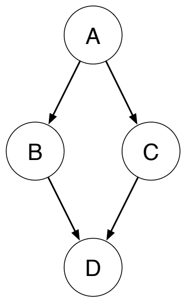
3. We want an action to open a door `?D`. This action should work when the robot is in either of the rooms that the door joins. Give a STRIPS representation for this action; make sure that you have all the necessary facts in your initial state.

Solution:

```
(:action open
 :parameters (?D ?Rx ?Ry)
 :precondition (and (join ?D ?Rx ?Ry) (robin ?Rx))
 :effect (open ?D))
```

6 The Deciding Factor

Consider the following directed graphical model:



Assume the variables are all binary and the CPTs are as follows:

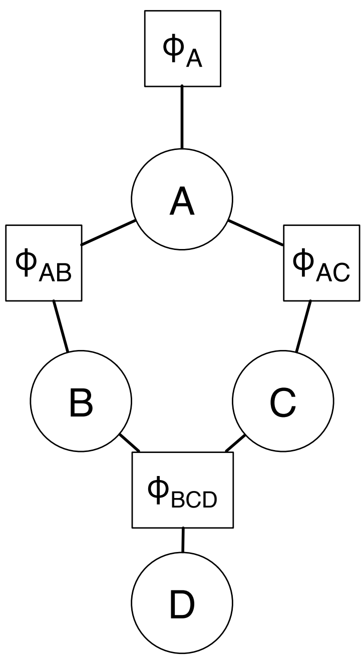
A	$P(B=1)$
0	0.3
1	0.6

A	$P(C=1)$
0	0.9
1	0.2

B	C	$P(D=1)$
0	0	0.9
0	1	0.1
1	0	0.1
1	1	0.9

$P(A=1)$
0.3

1. Draw its associated factor graph and specify the factors in terms of the CPTs given above.



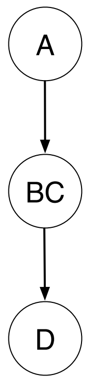
Solution:

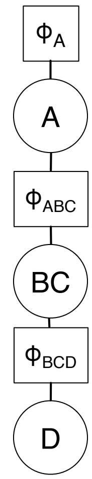
The factors are just the CPTs.

2. What algorithm is appropriate for exact inference on this model?

Solution: Variable elimination.

3. Jody suggests converting the original directed graph to the following one, where BC is a random variable that can take on four possible values: 00, 01, 10, 11 which correspond to joint assignments to variables B and C from the original model. We also show its associated factor graph.





How does this directed graph compare in expressive power to the original one?
√ **More** ◯ Less ◯ Same Briefly explain your answer.

Solution: B and C do not have to be conditionally independent given A any more.

4. Provide tables for any factors in the factor graph from part 3 that differ from the one in part 1.

Solution:
Factor ϕ_{ABC} is

A	B	C	
0	0	0	.07
0	0	1	.63
0	1	0	.03
0	1	1	.27
1	0	0	.32
1	0	1	.08
1	1	0	.48
1	1	1	.12

5. Show how to use belief propagation on the factor graph in part 3 to compute $P(A \mid D = 0)$, by supplying formulas for each of the messages that is computed. Your expressions may use factor values and values of any previously computed messages. You do not need to do numeric computation.

- Message $\mu_{D \to \phi_{BCD}}(d)$

Solution:

```
D
0  1
1  0
```

- Message $\mu_{\phi_{BCD} \to BC}(b, c)$

Solution: $\sum_D \phi_{BCD}(b, c, d)\mu_{D \to \phi_{BCD}}(d)$

- Message $\mu_{BC \to \phi_{ABC}}(b, c)$

Solution: $\mu_{\phi_{BCD} \to BC}(b, c)$

- Message $\mu_{\phi_{ABC} \to A}(a)$

Solution: $\sum_{b,c} \mu_{BC \to \phi_{ABC}}(b, c)\phi_{ABC}(a, b, c)$

- Message $\mu_{\phi_A \to A}(a)$

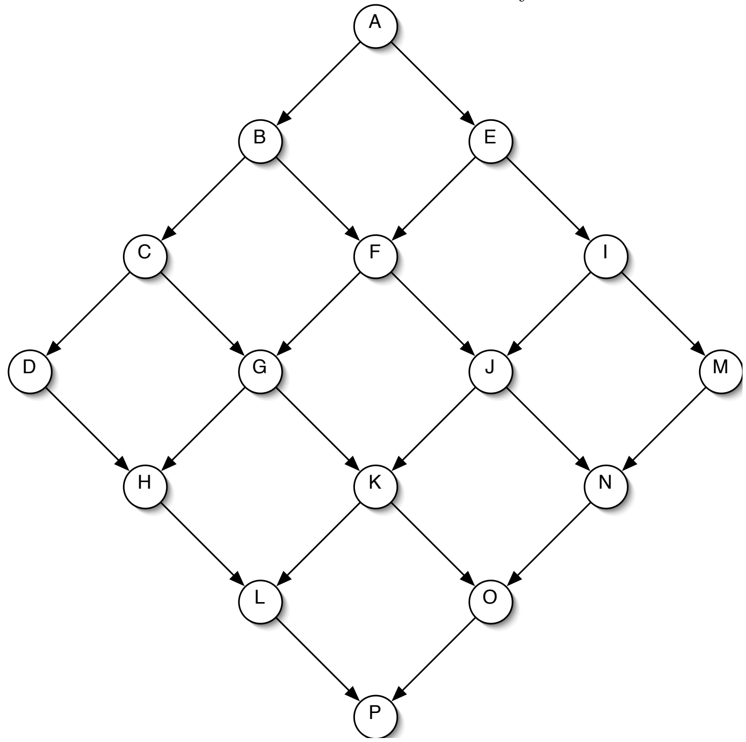
Solution: $\phi_A(a)$

- Final result: $\Pr(A = a \mid D = 0)$

Solution: $\mu_{\phi_A \to A}(a)\mu_{\phi_{ABC} \to A}(a)$

7 Bayesian network inference

Consider the Bayesian network below where all variables are binary.



1. What is the size of the largest CPT in this network? **4**
2. What nodes can be ignored while computing $\Pr(H|M)$? **JKLNOP**
3. What is the time complexity of the problem of finding the elimination order that generates the smallest-size largest factor? **Exponential in number of nodes**
4. If you were computing $\Pr(P|B = b)$ for a very unlikely value of b, would you prefer importance sampling or Gibbs sampling? Why?

   Solution: Gibbs, because the importance weights for all your samples will be low, which means the estimates will be inaccurate.

5. If you were computing $\Pr(B|P = p)$ for a very unlikely value of p, would you prefer importance sampling or Gibbs sampling? Why?

   Solution: Gibbs, because the importance weights for all your samples will be low, which means the estimates will be inaccurate.

8 Planning

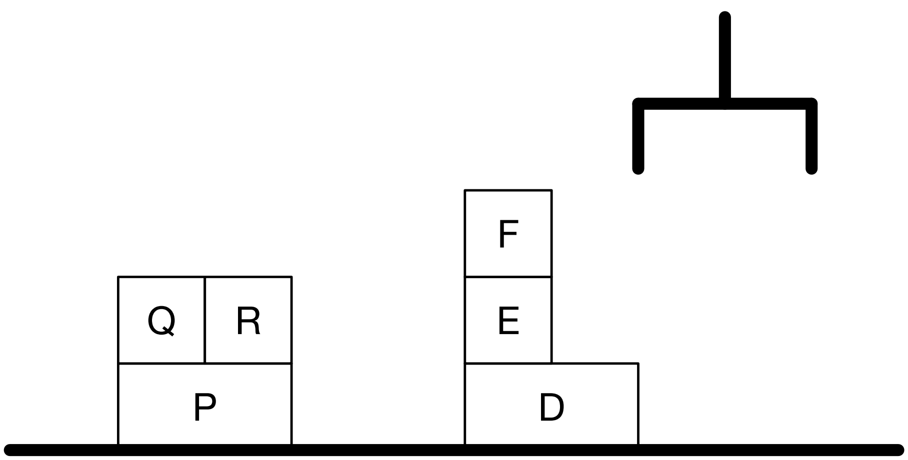
We have seen the standard "blocks world" domain in PDDL. It assumes that objects are all the same size and only one object can be on another object. In this problem, we want to extend the blocks world domain so that there are objects of two sizes: **big** and **small**. In this new domain, the stacking rules are as follows:

- One **big** object can be stacked on a **big** object, leaving no extra room.
- One **small** object can be stacked on another **small** object, leaving no extra room.
- No **big** object can be stacked on a **small** object.
- Two **small** objects can stacked (in separate **stack** operations) on a **big** object.

We will use the following new predicates:

- `(big ?x)`: object `?x` is **big**
- `(small ?x)`: object `?x` is **small**
- `(zero-on ?x)`: object `?x` has no other objects on top of it
- `(one-on ?x)`: object `?x` has exactly one object directly on top of it
- `(two-on ?x)`: object `?x` has exactly two objects directly on top of it

In this problem, we will ignore the `on-table` predicate.



1. Write a PDDL description of the state shown above, using predicates `on`, `arm-empty`, `holding`, `big`, `small`, `zero-on`, `one-on`, `two-on`.

Solution:

```
(on q p)
(on r p)
(two-on p)
(zero-on q)
(zero-on r)
(zero-on f)
```

```
(one-on e)
(one-on d)
(on e d)
(on f e)
(big p)
(big d)
(small q)
(small r)
(small f)
(small e)
(arm-empty)
```

2. Here is a PDDL operator description and a state description.

Operator description	State description

`(:action unstack-one-big`
`  :parameters (?ob ?underob)`
`  :precondition (and (arm-empty)`
`                     (two-on ?underob)`
`                     (on ?ob ?underob)`
`                     (zero-on ?ob))`
`  :effect (and (holding ?ob)`
`               (not (arm-empty))`
`               (one-on ?underob)`
`               (not (two-on ?underob))`
`               (not (on ?ob ?underob))))`
	`(and (on a b)`
	`     (on c b)`
	`     (on e d)`
	`     (small a)`
	`     (small c)`
	`     (small e)`
	`     (big b)`
	`     (big d)`
	`     (arm-empty)`
	`     (zero-on a)`
	`     (zero-on c)`
	`     (zero-on e)`
	`     (one-on d)`
	`     (two-on b))`

How many ways are there to apply this operator to this state? For each one, provide a binding of the parameters in the operator.

Solution: Two. In both `?underob` is `b`. In one, `?ob` is `a`. In the other, `?ob` is `c`.

3. For one of the applicable operator instances you identified in the previous question, what is the state that results from applying the operator?

Solution: For `(unstack-one-big a b)`

```
(and (on c b)
     (on e d)
     (small a)
     (small c)
     (small e)
     (big b)
     (big d)
     (holding a)
     (zero-on a)
     (zero-on c)
     (zero-on e)
     (one-on d)
     (one-on b))
```

4. Write a PDDL operator description that describes the operation of stacking a second block onto a big block that currently has a single small block directly placed on it.

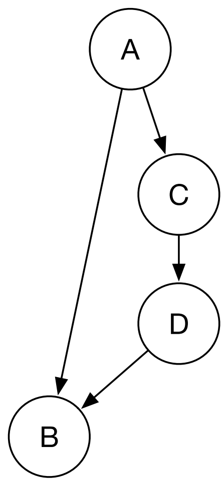
Solution:

```
(:action stack-second-on-big
```

```
:parameters (?ob ?underob)
:precondition (and (holding ?ob)
                   (one-on ?underob))
:effect (and (not (holding ?ob))
             (arm-empty)
             (not (one-on ?underob))
             (two-on ?underob)
             (on ?ob ?underob)))
```

9 Bayesian networks and probabilistic inference

9.1 Bayesian network definitions

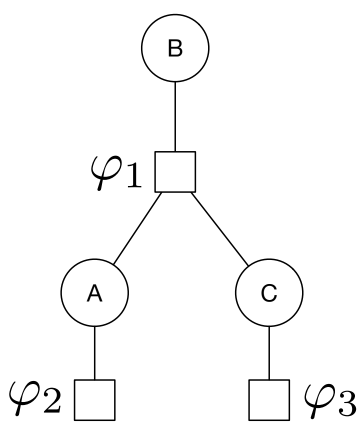


1. If each node is binary, how many parameters are needed to specify the distribution encoded by this network? (Assume we only need one parameter to define a distribution over two values). **9**

2. Write a simple expression for $P(A = 1, B = 0, C = 1, D = 0)$, using only quantities explicitly represented in the conditional probability tables of the network.

   $P(A = 1)P(C = 1|A = 1)P(D = 0|C = 1)P(B = 0|A = 1, D = 0)$

9.2 Factor graph



A	B	C	$\phi_1(A, B, C)$
0	0	0	.3
0	0	1	.2
0	1	0	.7
0	1	1	.8
1	0	0	.1
1	0	1	.3
1	1	0	.9
1	1	1	.7

A	$\phi_2(A)$
0	1
1	3

C	$\phi_3(C)$
0	3
1	3

1. Draw a Bayesian network graph that specifies the same distribution as this factor graph with the same number of parameters (but not necessarily the same parameter values).

   Solution: A —-> B <—— C

2. Specify parameters for the Bayesian network so that it specifies the same distribution as this factor graph.

Solution:

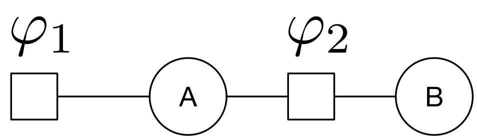
```
A | P(A)
0    .25
1    .75

C | P(C)
0    .5
1    .5

A B C | P(B | A, C)
(same as table given)
```

9.3 Computing messages

Consider performing belief propagation on the following simple factor graph:



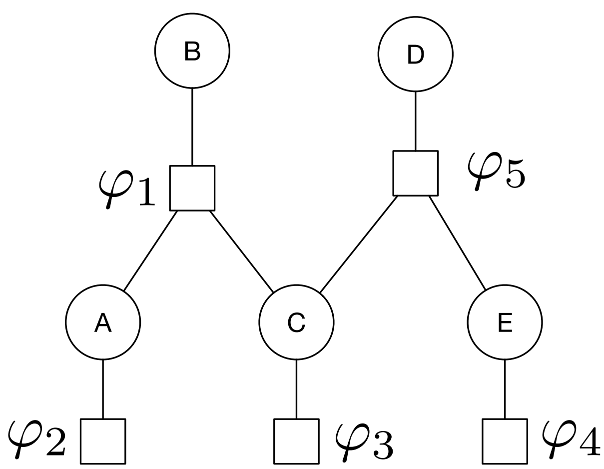
A	$\phi_1(A)$
0	.3
1	.7

A	B	$\phi_2(A, B)$
0	0	.1
0	1	.6
1	0	.1
1	1	.2

What are the values of the following messages? Specify the entire message, including all values for the relevant variable(s).

1. $\mu_{1\to A}$ $\{0:.3, 1:.7\}$
2. $\mu_{A\to 2}$ $\{0:.3, 1:.7\}$
3. $\mu_{2\to B}$ $\{0:.03+.07, 1:.18+.14\} = \{0:.1, 1:.32\}$

9.4 Message passing structure



Our goal is to compute $P(C)$ in this factor graph **using the belief propagation algorithm**. List a legal sequence of messages that would be generated by belief propagation, specifying which variable or factor originates the message and which receives it. You do not need to calculate the actual numerical values.

Solution: C is the root, so tree layers are:

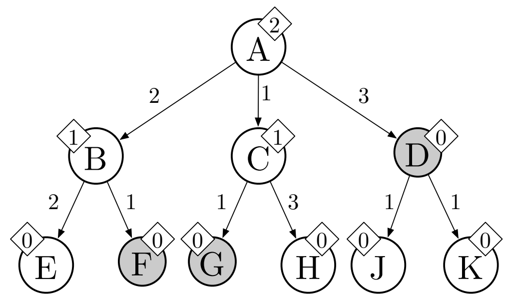
```
C
phi1, ph5, phi3
B, A, D, E
phi2, phi4
So messages go up the tree, like, for instance,
phi2 -> A    {A}
A -> phi1 {A}
B -> phi1 {B}   * will be degenerate, okay to omit *
phi1 -> C {C}
phi3 -> C {C}
phi4 -> E {E}
E -> phi5 {E}
D -> phi5 {D}    * will be degenerate, okay to omit *
phi5 -> C {C}
```

10 In search of...

The following assumptions apply to all the search problems

- Assume that the data structure implementations and successor state orderings are all such that *ties are broken alphabetically.* For example, a partial plan $S \to X \to A$ would be expanded before $S \to X \to B$; similarly, $S \to A \to Z$ would be expanded before $S \to B \to A$.
- All search algorithms are *graph* search (as opposed to tree search). In particular, when a node is expanded (popped from the agenda/frontier), if a node with this state has been previously expanded, then the node is pruned (discarded). Note that this is not the same as never *visiting* (adding to the agenda/frontier) a state twice.
- All the algorithms terminate when a node corresponding to a goal state is **expanded**, not when the node is added to the agenda/frontier.
- All edge costs are positive.
- When asked for a path, give your answers as a list of node names '$SADG$.'

Consider the state space search problem shown to the right. A is the start state and the shaded states are goals. Arrows encode possible state transitions, and numbers by the arrows represent action costs. Note that state transitions are directed; for example, $A \to B$ is a valid transition, but $B \to A$ is not. Numbers shown in diamonds are heuristic values that estimate the optimal (minimal) cost from that node to a goal.



For each of the following search strategies, give the path that would be returned, or write *none* if no path will be returned. Also, give all the states that are expanded in the order that they are expanded.

1. Depth-first graph search (ignores costs)

   Solution: Expanded: A, B, E, F Path: A, B, F

2. Breadth-first graph search (ignores costs)

   Solution: Expanded: A, B, C, D Path: A, D

3. Uniform cost graph search

   Solution: Expanded: A, C, B, G Path: A, C, G

4. Greedy graph search

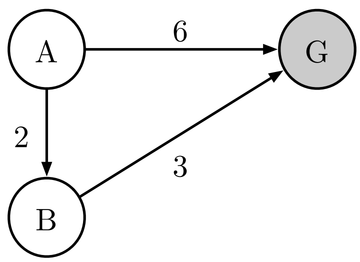
   Solution: Expanded: A, D Path: A, D

5. A* graph search

> **Solution:** Expanded: A, C, G Path: A, C, G

11 Admit it!

For the following questions, consider the search problem shown on the left. It has only three states, and three directed edges. A is the start node and G is the goal node. To the right, four different heuristic functions are defined, numbered I through IV.



	h(A)	h(B)	h(G)
I	4	1	0
II	5	4	0
III	4	3	0
IV	5	2	0

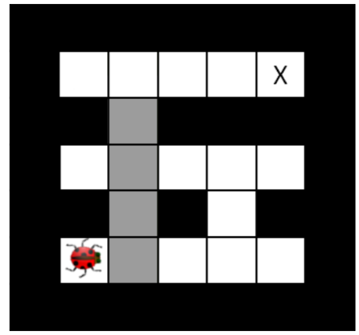
For each heuristic function, circle whether it is admissible for the search problem given above.

	Admissible?		Consistent?	
I	Yes	No	Yes	No
II	Yes	No	Yes	No
III	Yes	No	Yes	No
IV	Yes	No	Yes	No

> **Solution:** II is the only inadmissible heuristic, as it overestimates the cost from B: $h(B) = 4$, when the actual cost to G is 3.
>
> To check whether a heuristic is consistent, ensure that for all paths, $h(N) - h(L) \leq \text{path}(N \to L)$, where N and L stand in for the actual nodes. In this problem, $h(G)$ is always 0, so making sure that the direct paths to the goal ($A \to G$ and $B \to G$) are consistent is the same as making sure that the heuristic is admissible.
>
> The path from A to B is a different story.
>
> Heuristic I is not consistent: $h(A) - h(B) = 4 - 1 = 3 > \text{path}(A \to B) = 2$.
> Heuristic III is consistent: $h(A) - h(B) = 4 - 3 = 1 \leq 2$
> Heuristic IV is not consistent: $h(A) - h(B) = 5 - 2 = 3 > 2$

12 Hive mind

This hive of insects needs your help. You control an insect in a rectangular maze-like environment with dimensions $M \times N$, as shown to the right. At each time step, the insect can move into a free adjacent square or stay in its current location. All actions have cost 1.
In this particular case, the insect must pass through a series of partially flooded tunnels. Flooded squares are lightly shaded in the example map shown. The insect can hold its breath for A time steps in a row. Moving into a flooded square requires your insect to expend 1 unit of air, while moving into a free square refills its air supply.



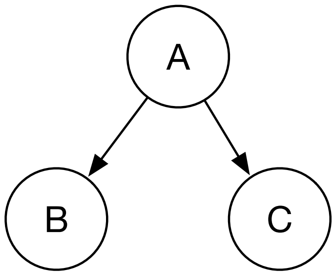
1. Give a minimal state space for this problem (i.e. do not include extra information). You should answer for a general instance of the problem, not the specific map shown.

   Solution: A tuple of location coordinates $m \in \{1, \ldots, M\}$ and $n \in \{1, \ldots, N\}$ and the remaining air supply $a \in \{1, \ldots, A\}$.

2. Give the maximum size of your state space, in the general case of an $M \times N$ grid. $\underline{\quad M \times N \times A \quad}$

13 Banana Nut

In the following Bayesian network



A	P(A)
0	.9
1	.1

B	A	P(B | A)
0	0	.7
1	0	.3
0	1	.6
1	1	.4

C	A	P(C | A)
0	0	.9
1	0	.1
0	1	.5
1	1	.5

1. What is $P(A = 1, B = 1)$

   Solution: $P(A = 1, B = 1) = P(B = 1|A = 1) * P(A = 1) = .4 * .1 = 0.04$

2. What is $P(A = 1, B = 1, C = 1)$

   Solution: $P(A = 1, B = 1, C = 1) = P(C = 1|A = 1) * P(B = 1|A = 1) * P(A = 1) = .5 * .4 * .1 = 0.02$

3. What is $P(B = 1 \mid A = 1)$

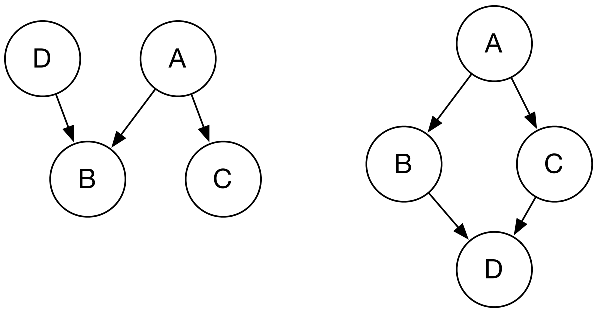
   Solution: $P(B = 1|A = 1) = 0.4$ can be directly read off the table

4. What is $P(B = 1 \mid C = 1)$

Solution:

$$\begin{aligned}
P(B=1|C=1) &= \frac{P(B=1,C=1)}{P(C=1)} \\
&= \frac{\sum_{A\in\{0,1\}} P(A,B=1,C=1)}{\sum_{A\in\{0,1\}} P(C=1,A)} \\
&= \frac{\sum_{A\in\{0,1\}} P(C=1|A) * P(B=1|A) * P(A)}{\sum_{A\in\{0,1\}} P(C=1|A) * P(A)} \\
&= \frac{.1 * .3 * .9 + .5 * .4 * .1}{.1 * .9 + .5 * 0.1} = \frac{0.047}{0.14}
\end{aligned}$$

14 The Reverend Thomas



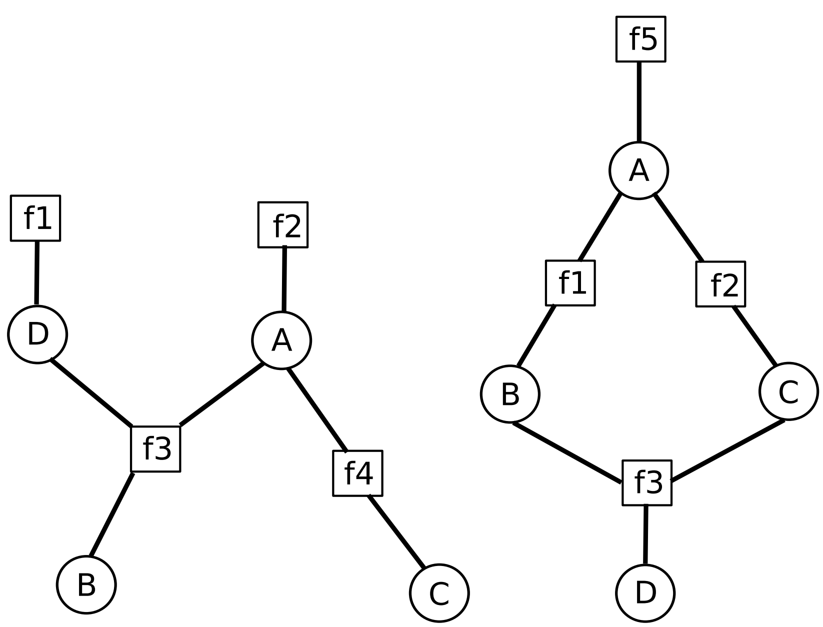
For each of the Bayesian networks above:

1. Write a simple expression for $P(A = 1, B = 0, C = 1, D = 0)$, using only quantities explicitly represented in the conditional probability tables of the network.

Solution:
$P(A = 1, B = 0, C = 1, D = 0) = P(D = 0)P(A = 1)P(B = 0|D = 0, A = 1)P(C = 1|A = 1)$
$P(A = 1, B = 0, C = 1, D = 0) = P(A = 1)P(B = 0|A = 1)P(C = 1|A = 1)P(D = 0|B = 0, C = 1)$

2. Draw the corresponding factor graph.

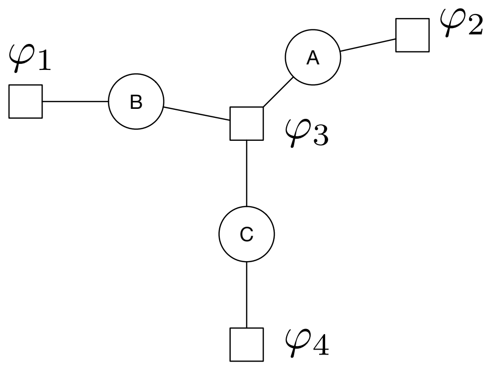
Solution:



3. Indicate whether belief propagation on each of these corresponding factor graphs will correctly compute marginal probabilities.

> **Solution:** *Yes for the first one. No for the second one since it is "loopy"*

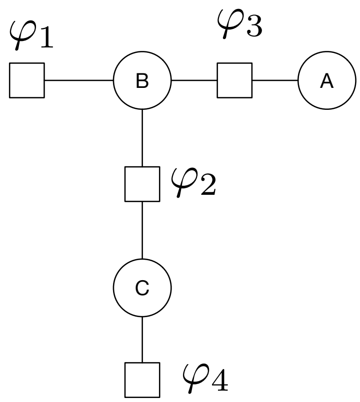
15 Factor it out



In the factor graph above, assuming the variables are all binary,

1. What are the variable involved in the potential φ_1? **B**
2. What are the variable involved in the potential φ_3? **A, B, C**
3. Mark all that are true:
   - ◯ The values in a factor have to sum to 1.
   - ◯ The values in a factor have to represent a conditional probability distribution.
   - ◯ The values in a factor have to be less than 1.
   - √ **The values in a factor have to be non-negative.**

16 Factor Fiction



1. Our goal is to compute $P(B)$ in this factor graph. List a legal sequence of message, specifying which variable or factor originates the message and which receives it. You do not need to calculate the actual numerical values (but it's good practice!).

Solution:

from	to	variables over which the messages is defined
ϕ_4	C	C
C	ϕ_2	C
ϕ_2	B	B (C is marginalized over)
ϕ_1	B	B
A	ϕ_3	A
ϕ_3	B	B (A is marginalized over)

2. Repeat for computation of $P(C)$.

Solution:

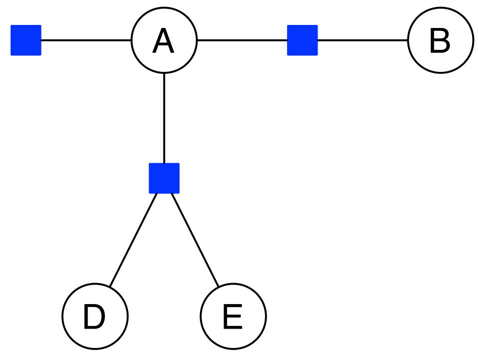
from	to	variables over which the messages is defined
ϕ_4	C	C
ϕ_1	B	B
A	ϕ_3	A
ϕ_3	B	B (A is marginalized over)
B	ϕ_2	B
ϕ_2	C	C (B is marginalized over)

3. Repeat for computation of $P(C|A = 1)$.

Solution: This runs the same way as computing $P(C)$, but now the message sent from variable A is $\{1 : 1.0\}$ instead of being the identity.

17 Belief propagation

Consider the sum-product algorithm on this factor graph:



We will refer to factors by the set of variables they are connected to.

1. $\phi_{ADE} =$

A	D	E	
0	0	0	1
0	0	1	2
0	1	0	1
0	1	1	3
1	0	0	2
1	0	1	1
1	1	0	4
1	1	1	3

$\phi_{AB} =$

A	B	
0	0	1
0	1	1
1	0	2
1	1	4

$\phi_A =$

A	
0	2
1	1

What message does A send to ϕ_{AB}? Your answer should be a table of numbers.

Solution:

$$\mu_{D \to \phi_{ADE}} = 1$$
$$\mu_{E \to \phi_{ADE}} = 1$$
$$\mu_{\phi_{ADE} \to A}(a) = \sum_d \sum_e \phi(a, d, e,)$$
$$\mu_{D \to \phi_{ADE}} \mu_{E \to \phi_{ADE}}$$
$$= [(1 + 2 + 1 + 3), (2 + 1 + 4 + 3)] = [7, 10]$$
$$\mu_{\phi_A \to A}(a) = [2, 1]$$
$$\mu_{A \to \phi_{AB}}(a) = [7, 10]. * [2, 1] = [14, 10]$$