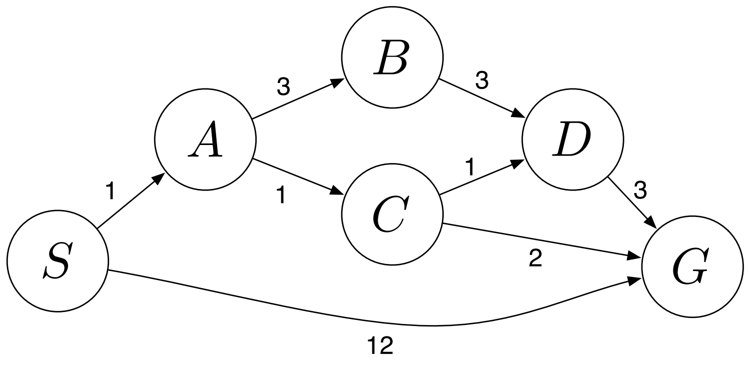
1. (12 points) Search



Answer the following questions about the search problem shown above. Break any ties alphabetically. For the questions that ask for a path, please give your answers in the form '$S - A - D - G$.'

(a) (2 pt) What path would breadth-first graph search return for this search problem?

$S - G$

(b) (2 pt) What path would uniform cost graph search return for this search problem?

$S - A - C - G$

(c) (2 pt) What path would depth-first graph search return for this search problem?

$S - A - B - D - G$

(d) (2 pt) What path would A* graph search, using a consistent heuristic, return for this search problem?

$S - A - C - G$

(e) (4 pt) Consider the heuristics for this problem shown in the table below.

State	h_1	h_2
S	5	4
A	3	2
B	6	6
C	2	1
D	3	3
G	0	0

i. (1 pt) Is h_1 admissible? **Yes** [✓ **No**]

ii. (1 pt) Is h_1 consistent? **Yes** [✓ **No**]

iii. (1 pt) Is h_2 admissible? [✓ **Yes**] **No**

iv. (1 pt) Is h_2 consistent? **Yes** [✓ **No**]

An admissible heuristic must underestimate or be equal to the true cost.
A consistent heuristic must satisfy $h(N) - h(L) \leq \text{path}(N \rightarrow L)$ for all paths and nodes N and L.

$h1$ overestimates the cost $S \rightarrow G$ as 5 when it is 4, so it is inadmissible.
$h1$ is not consistent because $h(S) - h(A) \leq \text{path}(S \rightarrow A)$ is violated as $5 - 3 \leq 1$.
$h2$ does not overestimate costs and is admissible.
$h2$ is not consistent because $h(S) - h(A) \leq \text{path}(S \rightarrow A)$ is violated as $4 - 2 \leq 1$.

Q2. [17 pts] CSPs: Midterm 1 Staff Assignments

CS188 Midterm I is coming up, and the CS188 staff has yet to write the test. There are a total of 6 questions on the exam and each question will cover a topic. Here is the format of the exam:

- q1. Search
- q2. Games
- q3. CSPs
- q4. MDPs
- q5. True/False
- q6. Short Answer

There are 7 people on the course staff: Brad, Donahue, Ferguson, Judy, Kyle, Michael, and Nick. Each of them is responsible to work with Prof. Abbeel on one question. (But a question could end up having more than one staff person, or potentially zero staff assigned to it.) However, the staff are pretty quirky and want the following constraints to be satisfied:

(i) Donahue (D) will not work on a question together with Judy (J).

(ii) Kyle (K) must work on either Search, Games or CSPs

(iii) Michael (M) is very odd, so he can only contribute to an odd-numbered question.

(iv) Nick (N) must work on a question that's before Michael (M)'s question.

(v) Kyle (K) must work on a question that's before Donahue (D)'s question

(vi) Brad (B) does not like grading exams, so he must work on True/False.

(vii) Judy (J) must work on a question that's after Nick (N)'s question.

(viii) If Brad (B) is to work with someone, it cannot be with Nick (N).

(ix) Nick (N) cannot work on question 6.

(x) Ferguson (F) cannot work on questions 4, 5, or 6

(xi) Donahue (D) cannot work on question 5.

(xii) Donahue (D) must work on a question before Ferguson (F)'s question.

(a) [2 pts] We will model this problem as a constraint satisfaction problem (CSP). Our variables correspond to each of the staff members, J, F, N, D, M, B, K, and the domains are the questions 1, 2, 3, 4, 5, 6. After applying the unary constraints, what are the resulting domains of each variable? (The second grid with variables and domains is provided as a back-up in case you mess up on the first one.)

B					5	
D	1	2	3	4		6
F	1	2	3			
J	1	2	3	4	5	6
K	1	2	3			
N	1	2	3	4	5	
M	1		3		5	

(b) [2 pts] If we apply the Minimum Remaining Value (MRV) heuristic, which variable should be assigned first?

Brad – because he has the least values left in his domain.

(c) [3 pts] Normally we would now proceed with the variable you found in (b), but to decouple this question from the previous one (and prevent potential errors from propagating), let's proceed with assigning Michael first. For value ordering we use the Least Constraining Value (LCV) heuristic, where we use *Forward Checking* to compute the number of remaining values in other variables domains. What ordering of values is prescribed by the LCV heuristic? Include your work—i.e., include the resulting filtered domains that are different for the different values.

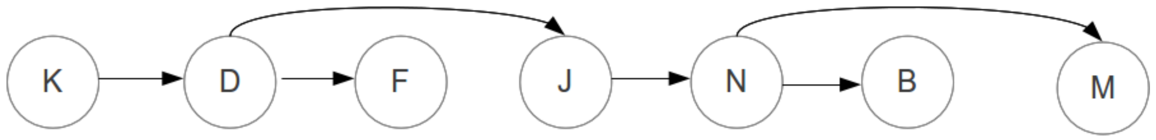
Michael's value will be assigned as 5, 3, 1, in that order.

Why these variables? They are the only feasible variables for Michael. Why this order? This is the increasing order of the number of constraints on each variable.

The only binary constraint incolving Michael is "Nick (N) must work on a question that's before Michael (M)'s question." So, only Nick's domain is affected by forward checking on these assignments, and it will change from {1, 2, 3, 4, 5} to {1, 2, 3, 4}, {1, 2}, and { } for the assignments 5, 3, 1, respectively.

(d) Realizing this is a tree-structured CSP, we decide not to run backtracking search, and instead use the efficient two-pass algorithm to solve tree-structured CSPs. We will run this two-pass algorithm after applying the unary constraints from part (a). Below is the linearized version of the tree-structured CSP graph for you to work with.

(i) [6 pts] **First Pass: Domain Pruning.** Pass from *right to left* to perform Domain Pruning. Write the values that remain in each domain below each node in the figure above.



K	D	F	J	N	B	M
1	1	1	1	1	1	1
2	2	2	2	2	2	2
3	3	3	3	3	3	3
4	4	4	4	4	4	4
5	5	5	5	5	5	5
6	6	6	6	6	6	6

Remaining values in each domain after the domain pruning right-to-left pass:
Kyle: 1
Donahue: 1,2
Ferguson: 1,2,3
Judy: 2,3,4,5,6
Nick: 1,2,3,4
Brad: 5
Michael: 1,3,5

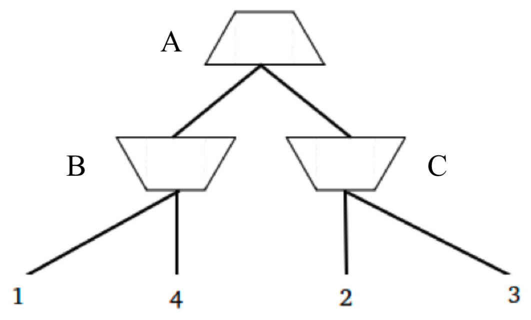
(ii) [4 pts] **Second Pass: Find Solution.** Pass from *left to right*, assigning values for the solution. If there is more than one possible assignment, choose the highest value.

Assigned Values after the left-to-right pass:

Kyle: 1
Donahue: 2
Ferguson: 3
Judy: 6
Nick: 4
Brad: 5
Michael: 5

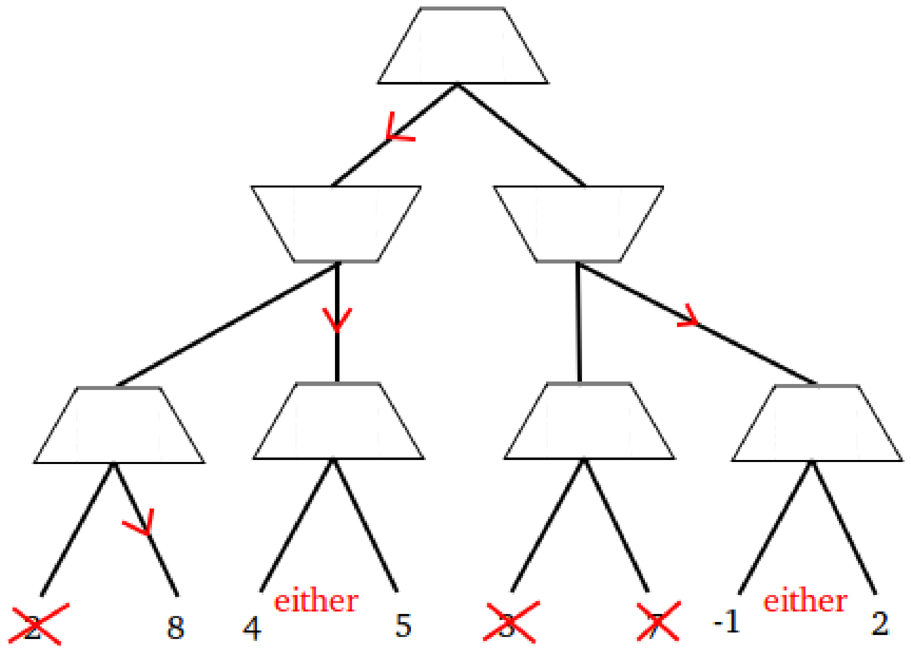
Q6. [10 pts] Pruning and Child Expansion Ordering

The number of nodes pruned using alpha-beta pruning depends on the order in which the nodes are expanded. For example, consider the following minimax tree.



In this tree, if the children of each node are expanded from left to right for each of the three nodes then no pruning is possible. However, if the expansion ordering were to be first Right then Left for node A, first Right then Left for node C, and first Left then Right for node B, then the leaf containing the value 4 can be pruned. (Similarly for first Right then Left for node A, first Left then Right for node C, and first Left then Right for node B.)

For the following tree, give an ordering of expansion for each of the nodes that will maximize the number of leaf nodes that are never visited due the search (thanks to pruning). ***For each node, draw an arrow indicating which child will be visited first. Cross out every leaf node that never gets visited.***
Hint: Your solution should have three leaf nodes crossed out and indicate the child ordering for 6 of the 7 internal nodes.



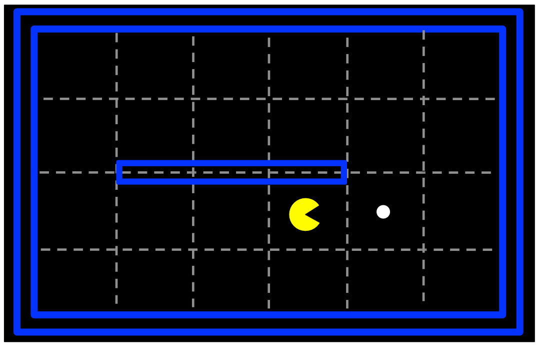
The thing to understand here is how pruning works conceptually. A node is pruned from under a max node if it "knows" that the min node above it has a better – smaller – value to pick than the value that the max node just found. Similarly, a node is pruned from under a min node if it knows that the max node above it has a better – larger – value to pick than the value that the min node just found.

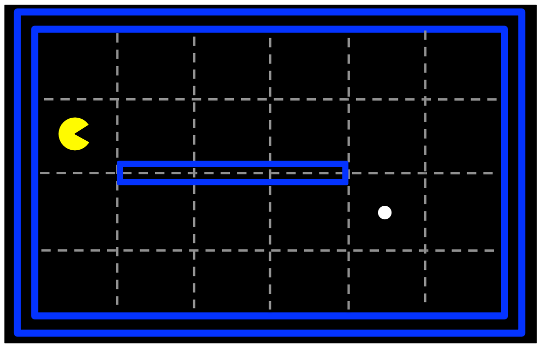
Q3. [11 pts] Solving Search Problems with MDPs

The following parts consider a Pacman agent in a deterministic environment. A goal state is reached when there are no remaining food pellets on the board. Pacman's available actions are $\{N, S, E, W\}$, but Pacman **can not** move into a wall. Whenever Pacman eats a food pellet he receives a reward of $+1$.

Assume that pacman eats a food pellet as soon as he occupies the location of the food pellet—i.e., the reward is received for the transition into the square with the food pellet.

Consider the particular Pacman board states shown below. Throughout this problem assume that $V_0(s) = 0$ for all states, s. Let the discount factor, $\gamma = 1$.


State A


State B

(a) [2 pts] What is the optimal value of state A, $V^*(A)$?
1

(b) [2 pts] What is the optimal value of state B, $V^*(B)$?
1
The reason the answers are the same for both (b) and (a) is that there is no penalty for existing. With a discount factor of 1, eating the food at any future step is just as valuable as eating it on the next step. An optimal policy will definitely find the food, so the optimal value of any state is always 1.

(c) [2 pts] At what iteration, k, will $V_k(B)$ first be non-zero?

5
The value function at iteration k is equivalent to the maximum reward possible within k steps of the state in question, B. Since the food pellet is exactly 5 steps away from Pacman in state B, $V_5(B) = 1$ and $V_{K<5}(B) = 0$.

(d) [2 pts] How do the optimal q-state values of moving W and E from state A compare? (*choose one*)

○ $Q^*(A, W) > Q^*(A, E)$ ○ $Q^*(A, W) < Q^*(A, E)$ [✓ $Q^*(A, W) = Q^*(A, E)$]

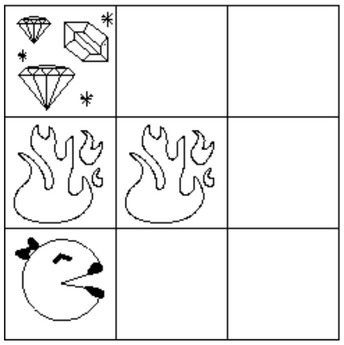
Once again, since $\gamma = 1$, the optimal value of every state is the same, since the optimal policy will eventually eat the food.

(e) [3 pts] If we use this MDP formulation, is the policy found guaranteed to produce the shortest path from pacman's starting position to the food pellet? If not, how could you modify the MDP formulation to guarantee that the optimal policy found will produce the shortest path from pacman's starting position to the food pellet?

No. The Q-values for going *West* and *East* from state A are equal so there is no preference given to the shortest path to the goal state. Adding a negative living reward (example: -1 for every time step) will help differentiate between two paths of different lengths. Setting $\gamma < 1$ will make rewards seen in the future worth less than those seen right now, incentivizing Pacman to arrive at the goal as early as possible.

5. (13 points) MDPs: Treasure Hunting

While Pacman is out collecting all the dots from `mediumClassic`, Ms. Pacman takes some time to go treasure hunting in the Gridworld island. Ever prepared, she has a map that shows where all the hazards are, and where the treasure is. From any unmarked square, Ms. Pacman can take the standard actions (N, S, E, W), but she is surefooted enough that her actions always succeed (i.e. there is no movement noise). If she lands in a hazard (H) square or a treasure (T) square, her only action is to call for an airlift (X), which takes her to the terminal 'Done' state, receiving a reward of -64 if she's escaping a hazard, but +128 if she's running off with the treasure. There is no "living reward."



(a) (2 pt) What are the optimal values, V^* of each state in the above grid if $\gamma = 0.5$?
Solution:

128	*64*	*32*
-64	*-64*	*16*
2	*4*	*8*

(b) (1 pt) What's the optimal policy?
Solution:

X	W	W
X	X	N
E	E	N

Call this policy π_0. Ms. Pacman realizes that her map might be out of date, so she decides to do some Q-learning to see what the island is really like. Because she thinks π_0 is close to correct, she decides to Q-learn while following an ϵ-random policy based on (b). Specifically, with probability ϵ she chooses amongst the available actions uniformly at random. Otherwise, she does what π_0 recommends. Call this policy π_ϵ.

An ϵ-random policy like π_ϵ is an example of a *stochastic* policy, which assigns probabilities to actions rather than recommending a single one. A stochastic policy can be written as $\pi(s, a)$, the probability of taking action a when the agent is in state s.

(c) **(2 pt)** Write out a modified Bellman equation for policy evaluation when the policy $\pi(s,a)$ is stochastic.
Solution:

$$V^{\pi}(s) = \sum_{a} \pi(s,a) \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma V^{\pi}(s')\right]$$

We also accepted answers specifically for π_ϵ, written in terms of ϵ, A_s, the set of legal actions from s, and $Succ(s,a)$, the state reached by taking action a from state s:

$$\begin{aligned} V^{\pi_\epsilon}(s) = \quad & (1-\epsilon)\left[R(s,\pi_0(s),Succ(s,\pi_0(s))) + \gamma V^{\pi_\epsilon}(Succ(s,\pi_0(s)))\right] + \\ & \frac{\epsilon}{|A_s|}\sum_{a}\left[R(s,a,Succ(s,a)) + \gamma V^{\pi_\epsilon}(Succ(s,a))\right] \end{aligned}$$

(d) **(1 pt)** If Ms. Pacman's map is correct what relationship will hold for all states?

i. $V^{\pi_0} \geq V^{\pi_\epsilon}$
ii. $V^{\pi_0} = V^{\pi_\epsilon}$
iii. $V^{\pi_0} \leq V^{\pi_\epsilon}$

Solution:
(i) will hold because π_0 is optimal.

It turns out that Ms. Pacman's map is mostly correct, but some of the fire pits may have fizzled out and become regular squares! Thus, when she starts Q-learning, she observes the following episodes:
[(0, 0), N, 0, (0, 1), N, 0, (0, 2), X, 128, Done]
[(0, 0), N, 0, (0, 1), N, 0, (0, 2), X, 128, Done]
[(0, 0), N, 0, (0, 1), E, 0, (1, 1), X, -64, Done]

(e) **(2 pt)** What are Ms. Pacman's Q-values after observing these episodes? Assume that she initialized her Q-values all to 0 (you only have to write the Q-values that aren't 0) and used a learning rate of 1.0.

Solution:
$Q((0,2),X) = 128$
$Q((0,1),N) = 64$
$Q((0,0),N) = 32$
$Q((1,1),X) = -64$
$Q((0,1),E) = 0$

(f) **(2 pt)** In most cases, a learning rate of 1.0 will result in a failure to converge. Why is it safe for Ms. Pacman to use a learning rate of 1.0?

Solution:
Ms. Pacman's actions are deterministic, so she does not need to average over possible outcomes of a particular action.

(g) **(1 pt)** Based on your knowledge about the structure of the maze and the episodes Ms. Pacman observed, what are the *true* optimal values of each state?
Solution:

128	*64*	*32*
64	*-64*	*16*
32	*16*	*8*

Q5. [12 pts] Independence in Hidden Markov Models

Below is a full derivation of the forward algorithm updates for Hidden Markov Models. As seen in lecture, we used $e_{1:t}$ to denote all the evidence variables $e_1, e_2, \ldots, e_t$. Similarly, $e_{1:t-1}$ denotes $e_1, e_2, \ldots, e_{t-1}$. For reference, the Bayes net corresponding to the usual Hidden Markov Model is shown on the right side of the derivation below.

$$\begin{aligned}
P(x_t|e_{1:t}) &\propto P(x_t, e_{1:t}) && (1)\\
&= \sum_{x_{t-1}} P(x_{t-1}, x_t, e_{1:t}) && (2)\\
&= \sum_{x_{t-1}} P(x_{t-1}, x_t, e_{1:t-1}, e_t) && (3)\\
&= \sum_{x_{t-1}} P(e_t \mid x_{t-1}, x_t, e_{1:t-1}) P(x_{t-1}, x_t, e_{1:t-1}) && (4)\\
&= \sum_{x_{t-1}} P(e_t \mid x_t) P(x_{t-1}, x_t, e_{1:t-1}) && (5)\\
&= \sum_{x_{t-1}} P(e_t \mid x_t) P(x_t \mid x_{t-1}, e_{1:t-1}) P(x_{t-1}, e_{1:t-1}) && (6)\\
&= \sum_{x_{t-1}} P(e_t \mid x_t) P(x_t \mid x_{t-1}) P(x_{t-1}, e_{1:t-1}) && (7)\\
&= P(e_t \mid x_t) \sum_{x_{t-1}} P(x_t \mid x_{t-1}) P(x_{t-1}, e_{1:t-1}) && (8)
\end{aligned}$$

$$\begin{tikzcd}
\cdots \arrow[r] & X_{t-1} \arrow[r] \arrow[d] & X_t \arrow[r] \arrow[d] & \cdots \\
\cdots & E_{t-1} & E_t & \cdots
\end{tikzcd}$$

(a) [2 pts] The following assumption(s) are needed to justify going from step (4) to step (5):
(select all that apply) also correct to select first on left and first on right

[✓ $E_t \perp\!\!\!\perp X_{t-1} \mid X_t$]	○ $E_t \perp\!\!\!\perp E_k \mid X_{t-1}$ for all $1 \le k \le t-1$
[✓ $E_t \perp\!\!\!\perp E_k \mid X_t$ for all $1 \le k \le t-1$]	○ $X_t \perp\!\!\!\perp E_{t+1} \mid X_{t+1}$
○ $E_t \perp\!\!\!\perp E_k$ for all $1 \le k \le t-1$	○ $X_t \perp\!\!\!\perp E_k \mid X_{t-1}$ for all $1 \le k \le t-1$
○ $E_t \perp\!\!\!\perp E_{t+1} \mid X_t$	○ none

(b) [2 pts] The following assumption(s) are needed to justify going from step (5) to step (6):
(select all that apply)

○ $E_t \perp\!\!\!\perp X_{t-1} \mid X_t$	○ $E_t \perp\!\!\!\perp E_k \mid X_{t-1}$ for all $1 \le k \le t-1$
○ $E_t \perp\!\!\!\perp E_k \mid X_t$ for all $1 \le k \le t-1$	○ $X_t \perp\!\!\!\perp E_{t+1} \mid X_{t+1}$
○ $E_t \perp\!\!\!\perp E_k$ for all $1 \le k \le t-1$	○ $X_t \perp\!\!\!\perp E_k \mid X_{t-1}$ for all $1 \le k \le t-1$
○ $E_t \perp\!\!\!\perp E_{t+1} \mid X_t$	[✓ none]

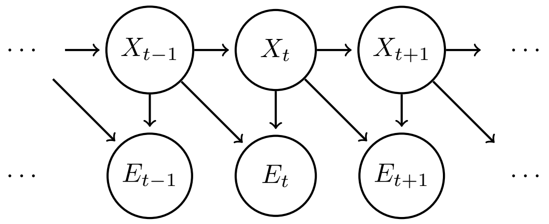
(c) [2 pts] The following assumption(s) are needed to justify going from step (6) to step (7):
(select all that apply)

○ $E_t \perp\!\!\!\perp X_{t-1} \mid X_t$	○ $E_t \perp\!\!\!\perp E_k \mid X_{t-1}$ for all $1 \le k \le t-1$
○ $E_t \perp\!\!\!\perp E_k \mid X_t$ for all $1 \le k \le t-1$	○ $X_t \perp\!\!\!\perp E_{t+1} \mid X_{t+1}$
○ $E_t \perp\!\!\!\perp E_k$ for all $1 \le k \le t-1$	[✓ $X_t \perp\!\!\!\perp E_k \mid X_{t-1}$ for all $1 \le k \le t-1$]
○ $E_t \perp\!\!\!\perp E_{t+1} \mid X_t$	○ none

Hidden Markov Models can be extended in a number of ways to incorporate additional relations. Since the independence assumptions are different in these extended Hidden Markov Models, the forward algorithm updates will also be different.

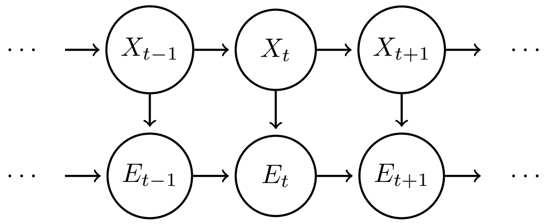
Complete the forward algorithm updates for the extended Hidden Markov Models specified by the following Bayes nets:

(d) [2 pts] $P(x_t|e_{1:t}) \propto \sum_{x_{t-1}} P(x_{t-1}, e_{1:t-1}) \cdot \underline{P(e_t \mid x_t, x_{t-1})P(x_t \mid x_{t-1})}$



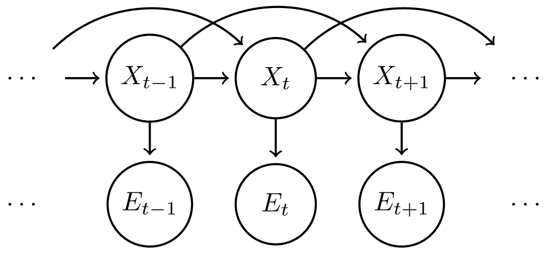
$$\begin{aligned}
P(x_t|e_{1:t}) \propto P(x_t, e_{1:t}) &= \sum_{x_{t-1}} P(x_{t-1}, x_t, e_{1:t-1}, e_t) \\
&= \sum_{x_{t-1}} P(e_t \mid x_{t-1}, x_t, e_{1:t-1}) P(x_{t-1}, x_t, e_{1:t-1}) \\
&= \sum_{x_{t-1}} P(e_t \mid x_t, x_{t-1}) P(x_{t-1}, x_t, e_{1:t-1}) \\
&= \sum_{x_{t-1}} P(e_t \mid x_t, x_{t-1}) P(x_t \mid x_{t-1}, e_{1:t-1}) P(x_{t-1}, e_{1:t-1}) \\
&= \sum_{x_{t-1}} P(e_t \mid x_t, x_{t-1}) P(x_t \mid x_{t-1}) P(x_{t-1}, e_{1:t-1})
\end{aligned}$$

(e) [2 pts] $P(x_t|e_{1:t}) \propto \sum_{x_{t-1}} P(x_{t-1}, e_{1:t-1}) \cdot \underline{P(e_t \mid x_t, e_{t-1})P(x_t \mid x_{t-1})}$



$$\begin{aligned}
P(x_t|e_{1:t}) \propto P(x_t, e_{1:t}) &= \sum_{x_{t-1}} P(x_{t-1}, x_t, e_{1:t-1}, e_t) \\
&= \sum_{x_{t-1}} P(e_t \mid x_{t-1}, x_t, e_{1:t-1}) P(x_{t-1}, x_t, e_{1:t-1}) \\
&= \sum_{x_{t-1}} P(e_t \mid x_t, e_{t-1}) P(x_{t-1}, x_t, e_{1:t-1}) \\
&= \sum_{x_{t-1}} P(e_t \mid x_t, e_{t-1}) P(x_t \mid x_{t-1}, e_{1:t-1}) P(x_{t-1}, e_{1:t-1}) \\
&= \sum_{x_{t-1}} P(e_t \mid x_t, e_{t-1}) P(x_t \mid x_{t-1}) P(x_{t-1}, e_{1:t-1})
\end{aligned}$$

(f) [2 pts] $P(x_t, x_{t+1}|e_{1:t}) \propto \sum_{x_{t-1}} P(x_{t-1}, x_t, e_{1:t-1}) \cdot \underline{P(e_t \mid x_t) P(x_{t+1} \mid x_{t-1}, x_t)}$



$$\begin{aligned}
P(x_t, x_{t+1}|e_{1:t}) \propto P(x_t, x_{t+1}, e_{1:t}) &= \sum_{x_{t-1}} P(x_{t-1}, x_t, x_{t+1}, e_{1:t-1}, e_t) \\
&= \sum_{x_{t-1}} P(e_t \mid x_{t-1}, x_t, x_{t+1}, e_{1:t-1}) P(x_{t-1}, x_t, x_{t+1}, e_{1:t-1}) \\
&= \sum_{x_{t-1}} P(e_t \mid x_t) P(x_{t-1}, x_t, x_{t+1}, e_{1:t-1}) \\
&= \sum_{x_{t-1}} P(e_t \mid x_t) P(x_{t+1} \mid x_t, x_{t-1}, e_{1:t-1}) P(x_t, x_{t-1}, e_{1:t-1}) \\
&= \sum_{x_{t-1}} P(e_t \mid x_t) P(x_{t+1} \mid x_t, x_{t-1}) P(x_t, x_{t-1}, e_{1:t-1})
\end{aligned}$$