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

Final Exam

Solutions

December 15, 2021

Answer the questions in the spaces provided on the question sheets. Please keep the exam packet together (don't un-staple).

You are permitted to use two sheets 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: ____

Kerberos: ______________________________________

Question	Points	Score
1	11	
2	11	
3	12	
4	10	
5	7	
6	4	
7	14	
8	16	
9	15	
Total:	100	

6.s058/16.420
Final Exam, December 15, 2021

1 Fireless

1. (11 points) Imagine a robot with sensors that can only observe certain aspects of the state of the world, but can do so without error. It can move among squares in a grid world.

   We are given predicates

   - $at(loc, t)$ meaning the robot is at location loc at time t
   - $detectSmoke(t)$ meaning the robot detects smoke at time t
   - $withinDetectionRange(l_1, l_2)$ meaning that the robot standing at location l_1 could detect smoke at location l_2
   - $fire(loc, t)$ meaning there is fire at location loc at time t
   - $did(a, t)$ meaning the robot executed action a at time t; a can be in the set of constant symbols $\{MoveNorth, MoveSouth, \ldots\}$
   - function $northOf(l)$ denoting the location that is directly to the north of location l; same for $southOf$, etc.

   Also, you can write $t + 1$ to stand for the function $nexttime(t)$, stands for the time step that is the successor to t.

   (a) (2 points) Write a logical sentence that encodes the following rule in the transition dynamics: if the robot attempts to move north from some location, then it will either move one or two steps north (you don't need to worry about encoding the fact that the robot can't be in two places at the same time).

   Solution:

   $$\forall t, l. did(MoveNorth, t) \wedge at(l, t) \rightarrow at(northOf(l), t + 1) \vee at(northOf(northOf(l)), t + 1)$$

   (b) (2 points) Our robot has a longer-range smoke sensor that allows it to detect smoke at a distance. Write a logical sentence to express that if the robot detects smoke then there is fire somewhere within the detection range.

   Solution:

   $$\forall t, l_1. at(l_1, t) \rightarrow (detectSmoke(t) \rightarrow \exists l_2. withinDetectionRange(l_1, l_2) \wedge fire(l_2, t))$$

(c) (4 points) Now, it is inference time! Here is a set of clauses. You don't have to worry about their intended meaning.

$$\neg h(y) \vee v(y) \tag{1}$$
$$h(\mathrm{g}(x)) \tag{2}$$
$$\neg f(x) \vee \neg v(\mathrm{g}(x)) \vee s() \tag{3}$$
$$\neg s() \tag{4}$$
$$f(\mathrm{C}) \tag{5}$$

Clause 5 represents the negation of a sentence that is entailed by the first 4 clauses. Please prove that this is so, using resolution refutation.

Solution: Here is one proof:

$$\neg f(x) \vee \neg v(\mathrm{g}(x)) \vee s() \tag{6}$$
$$\neg v(\mathrm{g}(\mathrm{C})) \tag{7}$$
$$\neg h(\mathrm{g}(\mathrm{C})) \tag{8}$$
$$\bot \tag{9}$$

(d) (3 points) For cases where we know a finite set of possible locations, describe an alternative approach to a syntactic proof like the one we just did, that would use a SAT solver. What is a potential downside of that approach?

Solution: An alternative to a syntactic proof like the one we did above is to enumerate the domain of locations for some specific problem or problems, convert the quantifiers to disjunction or conjunction, and then test for satisfiability. You could get really big formulas if you have a lot of locations.

2 What are the odds?

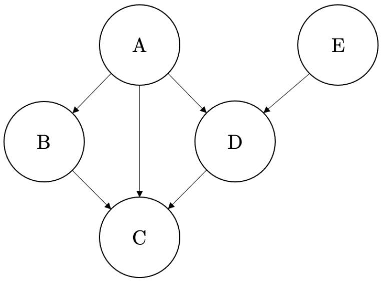
2. (11 points) (a) (2 points) Imagine that X_1 is a binary random variable with two parent variables X_2 and X_3 that are not necessarily binary. Imagine that the first parent X_2 can assume three different values, and that the second parent X_3 can assume two values. How many parameters are needed to represent the conditional probability table $P(X_1|X_2, X_3)$?

Solution: $3 \times 2 \times 2 = 12$ (or 6)

(b) (2 points) The sum-product algorithm over factor graphs relies on the ability to multiply messages and then marginalise out variables. Let us consider a factor ϕ with k neighbouring variables. If all the variables are binary, what is the time complexity of multiplying together the messages inbound into ϕ?

Solution: $O(2^k)$

(c) (3 points) Consider this Bayes net:



Which of the following statements are true?

1. $E \perp\!\!\!\perp B|D$
2. $D \perp\!\!\!\perp B|A, C$
3. $D \perp\!\!\!\perp B|A$
4. $D \perp\!\!\!\perp B|C$
5. D is in the Markov blanket of B
6. E is not in the Markov blanket of B

Solution:
5. D is in the Markov blanket of B
6. E is not in the Markov blanket of B

(d) (4 points) Consider an HMM estimating the state of a propulsion system using IMU measurements. The propulsion system is either working (W) or not working ($\neg W$). The IMU measurements are accelerating (A) or not accelerating ($\neg A$). We can specify our HMM by the following model:

Prior $p(s_0)$:

s_t	$p(s_t)$
W	.5
$\neg W$	.5

$P(s_{t+1}|s_t)$:

s_t	s_{t+1}	$P(s_{t+1}|s_t)$
W	W	.75
W	$\neg W$	.25
$\neg W$	W	0
$\neg W$	$\neg W$	1

$P(o_t|s_t)$:

s_t	o_t	$P(o_t|s_t)$
W	A	.8
W	$\neg A$	.2
$\neg W$	A	.1
$\neg W$	$\neg A$	.9

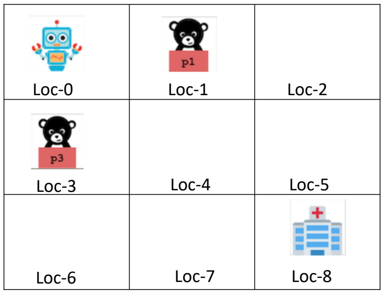
If there is no observation at time $t = 0$, and the observation at time $t = 1$ is $o_1 = A$, please compute $p(s_1|o_1 = A)$. You can express the probabilities as fractions if that is helpful.

Solution:

$$\begin{aligned} p(s_1|o_1 = A) &= \alpha p(o_1 = A|s_1) \odot \sum p(s_1|s_0)p(s_0) \\ &= \alpha \begin{bmatrix} .8 \\ .1 \end{bmatrix} \odot \begin{bmatrix} .75 \times .5 + 0 \times .5 \\ 1 \times .5 + .25 \times .5 \end{bmatrix} \\ &= \alpha \begin{bmatrix} .8 \\ .1 \end{bmatrix} \odot \begin{bmatrix} .375 \\ .625 \end{bmatrix} \\ &= \alpha \begin{bmatrix} .3 \\ .0625 \end{bmatrix} \\ &= \begin{bmatrix} .828 \\ .172 \end{bmatrix} \end{aligned}$$

3 Pikup Andropov

3. (12 points) Let's consider a very simple instance of Search and Rescue in a 3x3 grid with no obstacles and two "persons" to rescue:



The grid squares are connected vertically and horizontally.

The initial state is:

```
    (robot-at loc-0)
    (handsfree)
    (person-at p1 loc-1)
    (person-at p3 loc-3)
    (conn loc-0 loc-1)
    (conn loc-1 loc-0)
    (conn loc-0 loc-3)
    (conn loc-3 loc-0)
    etc.
```

The goal is `(and (person-at p1 loc-8) (person-at p3 loc-8))`

The domain has the following predicates:

```
  (:predicates
    (conn ?v0 - location ?v1 - location)
    (robot-at ?v0 - location)
    (person-at ?v0 - person ?v1 - location)
    (carrying ?v0 - person)
    (handsfree))
```

We have an action to move the robot, defined as follows:

```
  (:action move-robot
    :parameters (?from - location ?to - location)
    :precondition (and
      (conn ?from ?to)
      (robot-at ?from))
    :effect (and
      (not (robot-at ?from))
      (robot-at ?to)))
```

(a) (1 point) The following action, to pick up a person, is missing its effects:

```
(:action pickup-person
:parameters (?person - person ?loc - location)
:precondition (and
  (robot-at ?loc)
  (person-at ?person ?loc)
  (handsfree))
:effect [NEED EFFECTS] )
```

What are the effects of `pickup-person`?

Solution:

```
(and (not (person-at ?person ?loc))
      (not (handsfree))
      (carrying ?person))
```

(b) (1 point) The following action, to drop off a person, is missing its preconditions:

```
  (:action dropoff-person
:parameters (?person - person ?loc - location)
:precondition [NEED PRECONDITIONS]
:effect (and
  (person-at ?person ?loc)
  (handsfree)
  (not (carrying ?person))))
```

What are the preconditions of `dropoff-person`?

Solution: (and (carrying ?person) (robot-at ?loc))

(c) (2 points) Given the complete domain above, what is the length (number of actions) of the shortest valid plan?

Solution: 14. M[l1] - P[P1] - M[L2] - M[L5] - M[L8] - D[P1] - M[L7] - M[L6] - M[L3] - P[P3] - M[L4] - M[L5] - M[L8] - D[P3]

(d) (2 points) What is the value of $h^{max}(s_0)$, where s_0 is the initial state. Explain your answer.

Solution: 5. M[l1] - P[P1] - M[L2] - M[L5] - M[L8] - D[P1]

(e) (2 points) What is the value of $h^{add}(s_0)$? Explain your answer.

Solution: 12. Both goals appear at the same depth in the RPG.

(f) (2 points) What is the value of $h^{ff}(s_0)$? Explain your answer.

Solution: 9. M[L1], M[L3], P[P1], P[P3], M[L2], M[L5], M[L8], D[P1], D[P3]. We need achievers for each sub-goal, but some of the achievers do double duty here.

(g) (2 points) Consider an alternative initial state where the robot starts out at the hospital (`loc-8`) and there is a single person to rescue at `loc-0`. What is the value of h^{max}? for that initial state? Explain your answer.

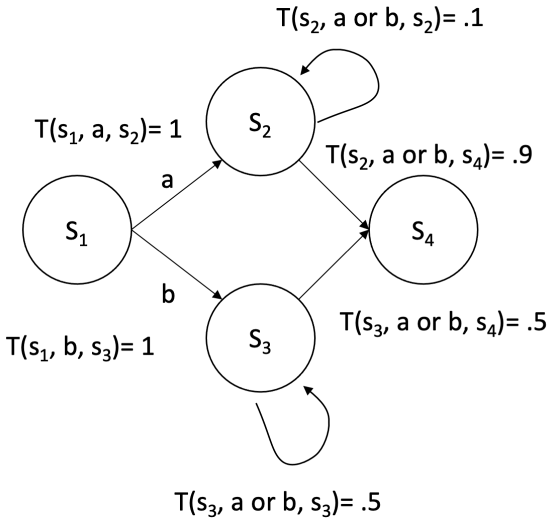
Solution: 6. M[l5]- M[L2] - M[L1] - M[L0] - P[P1] - D[P1]. Note that once a location is visited on the way out, it remains visited.

4 The best laid plans

4. (10 points) A *stochastic shortest paths* problem is a specific type of MDP in which

- $\mathcal{S}$ and $\mathcal{A}$ are discrete sets of states and actions, as in a standard MDP.
- There is a *goal set* $G \subset S$.
- The transition function $T(s, a, s') = P(S_{t+1} = s' \mid S_t = s, A_t = a)$, is almost as usual, except that all states in G are absorbing; that is, for all $s \in G$, and all $a \in A$, $T(s, a, s) = 1$.
- The reward function is $R(s, a, s') = 0$ for all $s \in G$ and $R(s, a, s') = -1$ otherwise (it can really be any negative value, but we will restrict our attention to this case).
- The discount factor $\gamma = 1$.

(a) (2 points) Here is a simple SSP.



- $\mathcal{S} = \{s_1, s_2, s_3, s_4\}$
- $\mathcal{A} = \{a, b\}$
- $\mathcal{G} = \{s_4\}$
- $T(s_1, a, s_2) = 1$, $T(s_1, b, s_3) = 1$,
  $T(s_2, a \text{ or } b, s_2) = .1$, $T(s_2, a \text{ or } b, s_4) = .9$, $T(s_3, a \text{ or } b, s_3) = .5$, $T(s_3, a \text{ or } b, s_4) = .5$

What is the optimal action to take in s_1?

Solution: a

(b) (2 points) What is the optimal Q function for the following state-action pairs? You can write an unevaluated numerical expression, but it may help to be reminded that $\sum_{i=1}^{\infty} ar^i = (ar)/(1-r)$.

$Q(s_1, a)$ **-2.11**

$Q(s_1, b)$ **-3**

$Q(s_2, a)$ **-1.11**

$Q(s_3, a)$ **-2**

(c) (2 points) Now consider the same SSP, but where we change just the transition function for s_3:

$$T(s_3, a \text{ or } b, s_3) = 1.0, \quad T(s_3, a \text{ or } b, s_4) = 0.0$$

What is the optimal Q function for the following state-action pairs?

$Q(s_1, b)$ $-\infty$

$Q(s_3, a)$ $-\infty$

(d) (2 points) Will value iteration converge on either or both of these SSPs? Explain.

Solution: It will converge on the first but not the second, for which, on every iteration, the value function will change by 1.

(e) (2 points) For each of the SSPs, is there a finite number of iterations after which you could terminate value iteration and extract an optimal policy? Explain your answer. (You don't have to provide a precise number).

Solution: There is a point at which the estimated value of action b is worse than the value of action a and after that, if we stop, the greedy policy will in fact be optimal.

5 All up in my grid

5. (7 points) Consider a grid-world stochastic shortest-paths problem (SSP) with the following properties:

- When the agent tries to move in one of the cardinal directions (N, S, E, W), it goes to the intended square with probability 0.8, and to each of the four neighbors of the intended with probability 0.05.
- As in the previous question, the goal states are absorbing: for all $s \in \mathcal{G}$, and all $a \in \mathcal{A}$, $T(s, a, s) = 1$.
- The reward function is $R(s, a, s') = 0$ for all $s \in \mathcal{G}$ and $R(s, a, s') = -1$ otherwise.
- The discount factor $\gamma = 1$.

In the following we will compare some aspects of the optimal strategy with an alternative approximation. The approximation is variation on the all-outcomes determinization, in which we transform the SSP into a determinstic path-search problem. In the transformed problem, we allow a transition from s to any s' such that there exists an a so that $T(s, a, s') > 0$. We will ignore the original cost function and instead assign the total cost associated with this transition to be $-\log T(s, a, s')$, so that although we are allowed to include unlikely transitions in our solution, they will cost a lot more. The minimum cost path in this problem corresponds to the single action sequence that is most likely to reach a goal state. We will consider using uniform-cost search (UCS) to find such a sequence.

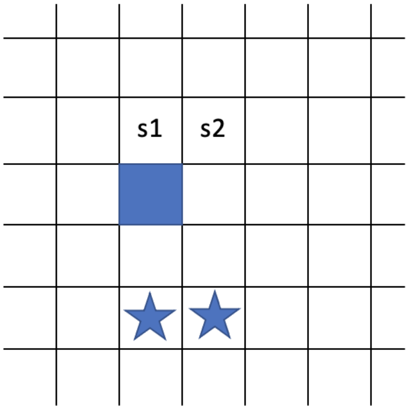
(a) (2 points) What would be a good (admissible, non-zero) heuristic function for A* in the approximate deterministic model?

It may help to know the following values:

log(.8)	log (.2)	log(.1)	log(.05)
-.223	-1.609	-2.303	-2.996

Solution: The cost of the moves is $-log.8 \approx .223$, so the Manhattan distance to the closest goal state times .2 would be admissible.

(b) (2 points) Now we add some hazards to the grid. Once you enter those states, all actions stay there with probability 1.0. In the figure below, the solid square is a hazard and the squares with stars are goal states.



In the determinized model, what action will be taken in

- state s_1 **east**

- state s_2 **south**

(c) (1 point) What is the optimal action in the underlying SSP in

- state s_2 **east**

(d) (2 points) Xerxes suggests addressing all problems of this class (with any combination of hazard squares) by removing any action from consideration that has a non-zero chance of landing in a hazard state and then running A* on the all-outcomes determinization with negative log costs. It is possible that they will obtain a path that has a risk of entering a hazard state?

Solution: The answer is yes. There are two acceptable explanations.

If we treat the generated path as sequence of actions and just execute them in sequence (without considering the actual states each action yields), it is possible we ends up hitting hazards. Meanwhile, there is possibility that we can't even find a path to any goal state from the initial state.

If we do replaning at each step, then it is possible that we can't find a path to any goal from the initial state, or from a state we land in during execution.

6 Simply unobservable

6. (4 points) (a) (2 points) Consider a POMDP in which the same observation is received, with probability 1, for every action in every state. In a problem with m states, n actions, and horizon h, characterize the size of an individual policy tree.

Solution: It's a sequence of actions of length h.

(b) (2 points) In the class of POMDPs described above, the finite-horizon value function, as a function of the belief b, is still piecewise-linear and convex. In a problem with m states, n actions, and horizon h, what is the maximum number of pieces it could have?

Solution: n^h

7 Charge!

7. (14 points) You may recall the example POMDP from homework 9. If not, here's all the info about it.

Our robot has lost its charger! The robot lives in a world with $K = 3$ locations but it needs to have the charger in location $L0$ (where the power outlet is) in order to plug it in and charge. The robot doesn't have any uncertainty about its own location (and we won't model that) but it does have uncertainty about the location of the charger. When the robot looks in location `loc`, by executing a `look(loc)` action, it will receive an observation in $\{0, 1\}$.

- If the charger is in the location `loc`, it will get observation 1 with probability 0.9.
- If the charger is not in location `loc`, it will get observation 1 with probability 0.4.

The robot can also try to execute an action `move(loc1, loc2)` (which moves the charger):

- If the charger is actually in `loc1`, then with probability 0.8, the charger will be moved to `loc2`, otherwise, it will stay in `loc1`.
- If the charger is not in `loc1`, then nothing will happen.

Because of the constraints of the robot's arm, we can't do all possible move actions. Specifically, the only valid movements are `move(0, 1)`, `move(1, 2)`, and `move(2, 0)`. The robot has two more actions, `charge` and `nop`:

- If it executes the `charge` action when the charger is in location 0, then it gets a reward of 10 and the game terminates.
- If it executes the `charge` action in any other state, it gets reward -100 and the game terminates.
- The `nop` action has reward 0, does not supply a useful observation, and does not affect the world state.

It gets reward -0.1 for every look action and -0.5 for every move action in all other states.
We are not discounting: $\gamma = 1$.
For actions that don't yield useful information about the environment (`move`, `charge`, and `nop`), we assume that they get observation 0 with probability 1, independent of the state.

In some belief states, the optimal horizon-2 action was to look in location 0 and in others it was to look in location 1. We are going to explore this asymmetry.

(a) (2 points) Starting with belief $b = (.9, .1, 0)$ over the charger being in locations `(loc0, loc1, loc2)`, what is the likelihood of seeing the object (getting observation 1) if you look in location 0?

Solution: .85

(b) (3 points) What is the posterior belief (distribution over all three possible charger locations) if you look in location 0 and see the object?

Solution: (.953, .047, 0)

(c) (3 points) What is the posterior belief (distribution over all three charger locations) if you look in location 0 and don't see the object?

Solution: (.6, .4, 0)

In addition to the values you computed above, we have computed some additional ones:

- The likelihood of seeing the object if you look in location 1 is .45.
- The posterior if you look in location 1 and see the object is $(.8, .2, 0)$.
- The posterior if you look in location 1 and don't see the object is $(.98, .02, 0)$

(d) (3 points) At horizon 1, if $b($`loc0`$) >\approx .91$, then it is worthwhile to charge, otherwise not. What is the optimal horizon-2 policy tree if you start with belief $b = (.9, .1, 0)$?

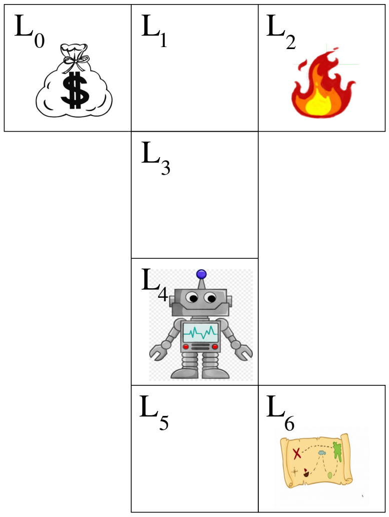
Solution: look(1); if obs = 0 then charge else nop

(e) (3 points) What is this policy tree's expected value in $b = (.9, .1, 0)$?

Solution: -0.1 + 0.55 * (0.98*10+0.02*(-100))

8 Fortune or ruin?

8. (16 points) Recall the Fortune-or-Ruin example from class.



- The agent is in the maze above.
- At the top of the T junction, there is a reward of +100 to one side (Fortune), and a reward of −1000 to the other side (Ruin), but the agent does not initially know whether the positive reward is on the left or the right — the agent thinks it's 50% probability that the positive reward is on either side (equal probability). Notice that in this case, the rewards are not symmetric — the negative penalty is much larger than the positive reward.
- The agent's motions are deterministic as it moves from cell to cell.
- When the agent receives either the positive or negative reward, the game is over. The locations L_0 and L_2 are absorbing, terminal locations — once the agent enters either of those locations, it can never leave, and receives no additional rewards.
- All actions prior to entering the terminal states have reward -1.
- The agent's observations are of its own location. Additionally, when the agent reaches the cell at the bottom of the L maze, L_6, there is a map which provides an observation that describes which side of the T junction has which reward. The agent receives this observation as soon as it enters location L_6. All observations, including the agent's position and the map (when available) are perfectly accurate.
- The agent starts at location L_4 as shown.
- The discount factor is 1.0

(a) (1 point) How many states are there in this problem?

Solution: 14 states. 7 agent locations × 2 reward positions.

(b) (1 point) What is the initial belief state?

Solution: $(s_4, p(\text{reward\_left}) = 0.5)$

(c) (2 points) How many belief states are reachable from the initial belief state, under any possible choice of action sequence, but assuming the belief update is always performed correctly, and what are they? (There are a finite and small number of belief states.)

Solution: This is a little complicated. It's 20 states.

7 agent positions × $[p(\text{reward\_left}) = 1, p(\text{reward\_left}) = 0, p(\text{reward\_left}) = 0.5]$. except that the agent can't be in belief state $(s_6, p(\text{reward\_left}) = 0.5)$, so one fewer.

(d) (2 points) Let us consider the maximum-likelihood state approximation strategy, where the agent assumes that the most likely state is the true state, and executes action given by the MDP policy for that state. What is the value of the initial belief state computed under this strategy, $V_{MLS}(b_0)$?

Solution: $+100 - 3 = 97$

(e) (2 points) If we were to run this strategy 100 times, what is (approximately) the mean reward collected by the agent under this strategy?

Solution: $0.5(+100 - 3) + 0.5(-1000 - 3) = -453$

(f) (2 points) Let us consider the QMDP approximation strategy. Recall that the QMDP policy is

$$\pi_{\text{QMDP}}(b) = \arg\max_a \sum_s b(s) Q^{\text{MDP}}(s, a),$$

where Q^{MDP} is the Q-function of the corresponding MDP that assumes the state is fully observable. What is the value of the initial belief state computed under this strategy, $V_{QMDP}(b_0)$?

Solution: $0.5(+100 - 3) + 0.5(+100 - 3) = 97$

(g) (2 points) Will the QMDP strategy ever enter the terminal state? Explain.

Solution: No. It has no reason to go and look at the map and assumes it can safely claim the reward. But when it reaches state s_1, both left and right actions have value $0.5(+100 - 1) + 0.5(-1000 - 1) = -451$, which is much worse than moving back down to state s_3. So the agent will oscillate between s_1 and s_3 forever.

(h) (2 points) Let us consider the maximum-likelihood observation approximation strategy, where the agent uses the best action for a policy tree that assumes that only the most likely observation will be received. What is the value of the initial belief state computed under this strategy, $V_{MLO}(b_0)$?

Solution: $0.5(+100 - 7) + 0.5(+100 - 7) = 93$

(i) (2 points) If we were to run this strategy 100 times, what is (approximately) the mean reward collected by the agent under this strategy?

Solution: $0.5(+100 - 7) + 0.5(+100 - 7) = 93$

9 Expectoration

9. (15 points) Consider a family of discrete MDPs, each with 1000 states. (In the following, use modular arithmetic, so state -1 is the same as state 999, and state 1000 is the same as state 0.)

- There are two actions, A and B.
- The reward for entering entering state 0 is $+99$.
- The reward for entering state 999 is -99.
- All other rewards are 0.
- The discount factor is 1.0.

MDPs in this family are parameterized by an integer $k \in \{0, \ldots, 100\}$, which governs the transition model as follows:

- Action A: State i transitions with probability $1/(2k+1)$ to each state in $\{i-k+1, \ldots, i+k+1\}$.
- Action B: State i transitions with probability $1/(2k+1)$ to each state in $\{i-k-1, \ldots, i+k-1\}$.

So, action A produces an uniform distribution centered at $i+1$ and action B produces an uniform distribution centered at $i-1$, both of width $2k+1$. For example,

- When $k=0$, action A moves deterministically one step "up" and action B moves deterministically one step "down".
- When $k=1$, action A moves with equal probability "up" 2, "up" 1, or stays in the original state; action B moves with equal probability "down" 2, "down" 1, or stays in the original state.
- When $k=2$, action A moves with equal probability "up" 3, "up" 2, "up" 1, stays in the original state, or moves "down" 1; action B moves with equal probability "down" 3, "down" 2, "down" 1, stays in the original state, or moves "up" 1.

(a) (2 points) How many leaves are there in the expectimax tree for horizon 3 in the MDP with $k=1$? (It is fine to write an unevaluated expression).

Solution: $6^3 = 216$

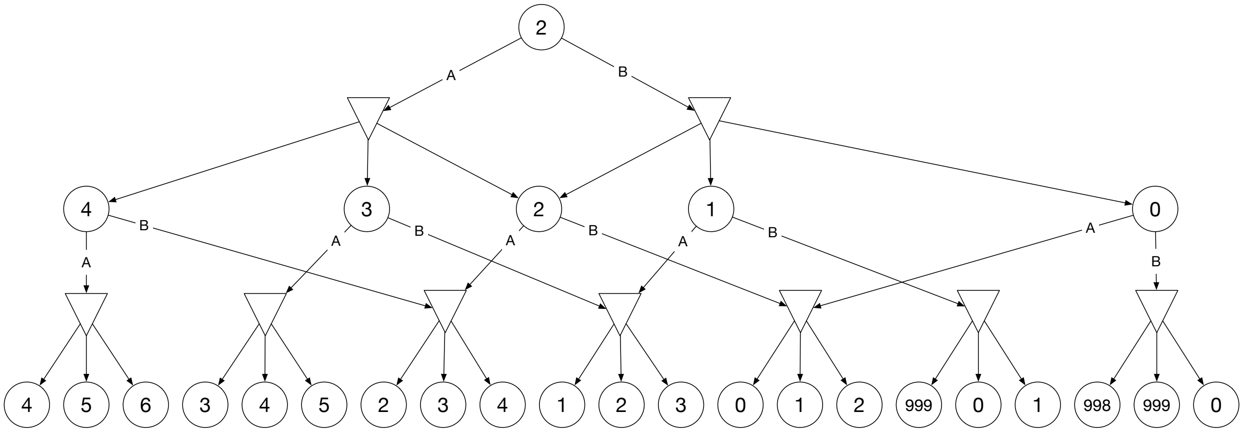
(b) (2 points) How many leaves are there in the expectimax tree for horizon 3 in the MDP with $k=100$? (It is fine to write an unevaluated expression)

Solution: 402^3

(c) (2 points) How many distinct leaves are there in the expectimax AODAG (that is, graph in which states that are reachable in the same level of the tree are represented as a single node) for horizon h in the MDP? (Provide your answer in terms of h and k).

> **Solution:** $min(2h(k+1)+1, 1000)$

(d) (3 points) The following figure illustrates the horizon-2 expectimax AODAG starting at state 2 for $k = 1$. (Note that we did not coalesce the leaf nodes in this figure, because it becomes very difficult to read.)



Provide the horizon-2 Q values $Q(2, A)$ and $Q(2, B)$.

> **Solution:** $Q(2, A) = 11$ and $Q(2, B) = 55$

(e) (2 points) What is the optimal horizon 2 policy for the $k = 1$ MDP, starting at state 2?

> **Solution:**

```
S2: Take action B (move left)
Then:
  If S0: Take action A (move right)
  If S1: Doesn’t matter
  If S2: Take action B (move left)
```

(f) (2 points) Consider the case of $k = 100$ and horizon 10. We might prefer to use a sampling based method (sparse sampling or MCTS). Explain why.

Solution: Even the AODAG will be too huge to evaluate all of and so sampling will be the only plausible strategy.

(g) (2 points) Treesa wants to use sparse sampling with 50 samples at each node. In the case when $k = 100$, horizon is just 1 and the state is 101, what difficulty might Treesa's method face?

Solution: The optimal action in that case is B, but to see that, you have to "hit" a single outcome out of 100, and that is unlikely to happen with only 50 samples.