Problem 1: (25 points)

Pretend you were transported back in time before the advent of money and instead everything is done with bartering. In the local area, the following trades are what people are willing to do (first item is what you give, second is what you receive):

1. 10 bushels of wheat for 1 horse
2. 1 horse for 10 bushels of wheat
3. 1 cow for 1 pig
4. 3 cows for 1 horse
5. 1 horse for 3 goats
6. 1 pig for 1 goat
7. 2 goats for 1 cow
8. 1 goat for 2 bushels of wheat

(1.1) Write these "willing" trades as preferences.

$$\begin{aligned}
\text{1 horse} &\sim \text{10 bushels of wheat}\\
\text{1 cow} &\succ \text{1 pig}\\
\text{3 cows} &\succ \text{1 horse}\\
\text{1 horse} &\succ \text{3 goats}\\
\text{1 pig} &\succ \text{1 goat}\\
\text{2 goats} &\succ \text{1 cow}\\
\text{1 goat} &\succ \text{2 bushels of wheat}
\end{aligned}$$

(1.2) Convert these preferences into a mathematical representation for the utility of the items.

$$\begin{aligned}
U(\text{10 bushels of wheat}) &= U(\text{1 horse})\\
U(\text{1 cow}) &> U(\text{1 pig})\\
U(\text{3 cows}) &> U(\text{1 horse})\\
U(\text{1 horse}) &> U(\text{3 goats})\\
U(\text{1 pig}) &> U(\text{1 goat})\\
U(\text{2 goats}) &> U(\text{1 cow})\\
U(\text{1 goat}) &> U(\text{2 bushels of wheat})
\end{aligned}$$

As such,

$$U(\text{2 bushels of wheat}) < U(\text{1 goat}) < U(\text{1 pig}) < U(\text{1 cow}) < U(\text{2 goats})$$

and

$$U(\text{3 goats}) < U(\text{1 horse}) = U(\text{10 bushels of wheat}) < U(\text{3 cows})$$

If we assume that the utility of three goats is greater than the utility of 2 goats (i.e. that the agesnts exhibit a monotonic preference for more of some good), then the following preference ranking on all of the states can be determined:

$$U(\text{2 bushels of wheat}) < U(\text{1 goat}) < U(\text{1 pig}) < U(\text{1 cow}) < U(\text{2 goats}) < U(\text{3 goats}) < U(\text{1 horse}) = U(\text{10 bushels of wheat}) < U(\text{3 cows})$$

(1.3) Find valid utility values for each of the items.

Given the nature of trading, we know that the trade environment is deterministic (ex. a hypothetical agent acting in this environment will be able to trade 1 cow for 1 pig, 100% of the time as 1 cow is preffered over 1 pig). This finding, when considered in conjunction with the preference ranking on all states found in problem (1.2), means that any utility values that maintain the following preferential ordering, will be valid.

$$U(2 \text{ bushels of wheat}) < U(1 \text{ goat}) < U(1 \text{ pig}) < U(1 \text{ cow}) < U(2 \text{ goats}) < U(3 \text{ goats}) < U(1 \text{ horse}) = U(10 \text{ bushels of wheat}) < U(3 \text{ cows})$$

One such set of utility values is now presented:

$$\begin{aligned}
U(2 \text{ bushels of wheat}) &= 2 \\
U(1 \text{ goat}) &= 3 \\
U(1 \text{ pig}) &= 4 \\
U(1 \text{ cow}) &= 5 \\
U(2 \text{ goats}) &= 6 \\
U(3 \text{ goats}) &= 9 \\
U(1 \text{ horse}) &= 10 \\
U(10 \text{ bushels of wheat}) &= 10 \\
U(3 \text{ cows}) &= 15
\end{aligned}$$

(1.4) Can you do any affine transformation on these utilities? Why or why not?

As mentioned in problem (1.3), the conditions of a trade are deterministic. That is, the result of a trade is know 100% of the time. For example, people are always willing to trade 1 horse for 10 bushels of wheat. Furthermore, although we assummed that the agents in this problem exhbit a monotonic preference for more of some good, we did not go so far as to assume that the utility of two goods is necessarily equal to twice the utility of one good (even though the set of utilities presented in problem (1.3) does possess this quality). As such, any set of utilities that agrees with the preference ranking on all of the states, is valid. This means that we can do any affine transformation $U'(S) = aU(S) + b$ on these utilities so long as the ordinal relationship of each utility is not affected.

Consider the value of a in the above affine transformation. If $a = 0$, all of the utilities will be mapped to the same value b. This clearly does not maintain the preference ranking for all of the states, since the new utilities represent an indifference between all states, which is not what was found in problem (1.1). Similarily, if $a < 0$, the ordering of the preferences will be reversed, as larger utilities will map to lower (more negative) values. Again, this new ordering does not reflect the preferences found in problem (1.1). If $a > 0$, however, there will be no change to the ordering of the preferences.

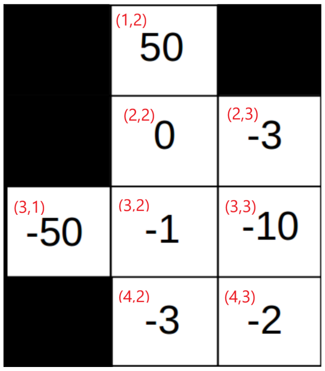
In other words, as long as $a > 0$, applying an affine transformation of the form $U'(S) = aU(S) + b$ on a valid utility function $U(S)$ will result in another valid utility function $U'(S)$.

Problem 2: (25 points)

Assume that when moving there is a 70% chance to end up where you want to go and a 15% chance to end up 90 degrees left/right of where you want to go. So for example, if you intend to go up: there is a 70% chance you go up, 15% chance you go right and 15% chance you go left.

You may assume that when you hit the 50 or -50, that you cannot move anymore and just get that reward then stop the game.

For all parts of this problem use the MDP above, and assume $\gamma = 0.8$



(2.1) Run value iteration until convergence and report the utilities for every state.

Let states be represented by their coordinates (row, column) as indicated in red on the MDP figure above. The code used to generate the following utilities can be found in ValueIteration.py (submitted to HW4-code on Canvas):

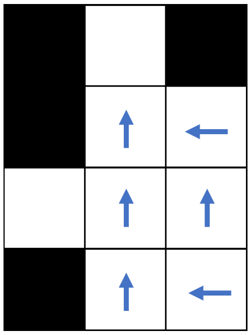
$$
\begin{gathered}
U(1,2) = 50 \\
U(2,2) = 34.37934744300792 \\
U(2,3) = 18.781881248724755 \\
U(3,1) = -50 \\
U(3,2) = 12.528055498877784 \\
U(3,3) = 2.2968410899445413 \\
U(4,2) = 4.704327292904657 \\
U(4,3) = 1.0341411532044917
\end{gathered}
$$

(2.2) If your initial guesses for the utilities are all zero, what is the least amount of iterations to find the best policy?

Only 4 iterations are needed to find the best policy if the initial guesses for the utilities are all zero. The code used to determine this result, and generate the utilities below can be found in ValueIteration.py (submitted to HW4-code on Canvas):

Here are the utilities for each state after 4 iterations:

$$
\begin{gathered}
U(1,2) = 50 \\
U(2,2) = 33.689152 \\
U(2,3) = 16.72736 \\
U(3,1) = -50 \\
U(3,2) = 11.004736000000001 \\
U(3,3) = -1.3843200000000007 \\
U(4,2) = 1.4773760000000005 \\
U(4,3) = -2.972992
\end{gathered}
$$



(2.3) On the iteration you found in part (2), what is the largest difference the estimated utility vs. actual utility (i.e. between parts (1) and (2)). How does this compare to the theoretical bound for the utility?

First, determine the largest difference between the estimate utilities on the 4th iteration, and the actual utilities:

$$\begin{aligned}
|U(1,2) - U_4(1,2)| &= 50 - 50 = 0 \\
|U(2,2) - U_4(2,2)| &= 34.37934744300792 - 33.689152 = 0.690195443 \\
|U(2,3) - U_4(2,3)| &= 18.781881248724755 - 16.72736 = 2.05452124872 \\
|U(3,1) - U_4(3,1)| &= -50 - -50 = 0 \\
|U(3,2) - U_4(3,2)| &= 12.528055498877784 - 11.004736000000001 = 1.52331949888 \\
|U(3,3) - U_4(3,3)| &= 2.2968410899445413 - -1.3843200000000007 = 3.68116108994 \\
|U(4,2) - U_4(4,2)| &= 4.704327292904657 - 1.4773760000000005 = 3.2269512929 \\
|U(4,3) - U_4(4,3)| &= 1.0341411532044917 - -2.972992 = 4.0071331532
\end{aligned}$$

$$\max(0, 0.690195443, 2.05452124872, 0, 1.52331949888, 3.68116108994, 3.2269512929, 4.0071331532) = 4.0071331532$$

Therefore, the largest difference between the estimated utilities on the 4th iteration, and the acutal utilities is 4.0071331532.

Next, find the theoretical bound for the utility. Recall (see Norvig p. 655) that the difference/error between any of the estimated utilities (U_i) on the Nth iteration and the actual utilily (U^*) is bounded by the expression:

$$\gamma^n \cdot 2R_{max}/(1-\gamma) \geq ||U_i - U^*||$$

Substituting in the given information for this problem, along with N=4, and simplyfing results in:

$$\begin{aligned}
& (0.8)^{(4)} \cdot 2(50)/(1-(0.8)) \geq ||U_i - U^*|| \\
\iff \quad & 204.8 \geq ||U_i - U^*||
\end{aligned}$$

Thus, the largest difference that was observed in part (2) of 4.0071331532 is significantly less that the theoretical bound of 204.8.

Problem 3: (25 points)

Use policy iteration to solve the MDP shown above. Start by assuming all actions are "Up" and $\gamma = 0.8$

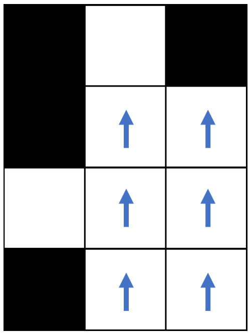
Let the utility of state 's' at iteration 'i+1' be recursively defined as

$$U_{i+1}(s) = R(s) + \gamma \sum_{s'} P(s'|s, \pi_i(s))U_i(s') \tag{1}$$

Aditionally, let the best policy for state 's' at iteration 'i' be defined as

$$\pi_i(s) = \max_{a \in A(s)} \sum_{s'} P(s'|s, a)U_i(s') \tag{2}$$

Note that $\pi_0(s) = "Up"$ for all states since all initial actions are "Up"



Using Eq. 1 then,

$$\begin{aligned}
&U_1(1,2) = 50\\
&U_1(2,2) = (0) + (0.8)[(0.7)U_1(1,2) + (0.15)U_1(2,3) + (0.15)U_1(2,2)]\\
&U_1(2,3) = (-3) + (0.8)[(0.7)U_1(2,3) + (0.15)U_1(2,3) + (0.15)U_1(2,2)]\\
&U_1(3,1) = -50\\
&U_1(3,2) = (-1) + (0.8)[(0.7)U_1(2,2) + (0.15)U_1(3,3) + (0.15)U_1(3,1)]\\
&U_1(3,3) = (-10) + (0.8)[(0.7)U_1(2,3) + (0.15)U_1(3,3) + (0.15)U_1(3,2)]\\
&U_1(4,2) = (-3) + (0.8)[(0.7)U_1(3,2) + (0.15)U_1(4,3) + (0.15)U_1(4,2)]\\
&U_1(4,3) = (-2) + (0.8)[(0.7)U_1(3,3) + (0.15)U_1(4,3) + (0.15)U_1(4,2)]
\end{aligned}$$

Solving this system of linear equations results in:

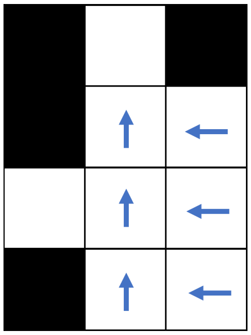
$$\begin{aligned}
&U_1(1,2) = 50\\
&U_1(2,2) = 32.1856\\
&U_1(2,3) = 2.69461\\
&U_1(3,1) = -50\\
&U_1(3,2) = 10.0302\\
&U_1(3,3) = -8.28113\\
&U_1(4,2) = 1.9821\\
&U_1(4,3) = -7.27225
\end{aligned}$$

Using these utilities, the best action for each state can be found using Eq. 2:

$$\begin{aligned}
&\pi_1(2,2) = \max[40.23203, 10.89076, 12.25317, 31.53445] = 40.23203 \rightarrow "Up"\\
&\pi_1(2,3) = \max[7.118259, 1.048249, -0.56476, 21.69194] = 21.69194 \rightarrow "Left"\\
&\pi_1(3,2) = \max[13.78775, -0.67164, -7.3547, -29.8748] = 13.78775 \rightarrow "Up"\\
&\pi_1(3,3) = \max[2.148588, -6.48344, -4.82821, 6.334494] = 6.334494 \rightarrow "Left"\\
&\pi_1(4,2) = \max[6.227618, -3.28873, 0.593948, 3.189315] = 6.227618 \rightarrow "Up"\\
&\pi_1(4,3) = \max[-6.59031, -7.42358, -5.8841, -0.94554] = -0.94554 \rightarrow "Left"
\end{aligned}$$

Continued on next page ⟶

Now find $U_2(s)$ for all states using Eq. 1:



$$\begin{aligned}
&U_2(1,2) = 50\\
&U_2(2,2) = (0) + (0.8)[(0.7)U_2(1,2) + (0.15)U_2(2,3) + (0.15)U_2(2,2)]\\
&U_2(2,3) = (-3) + (0.8)[(0.15)U_2(2,3) + (0.15)U_2(3,3) + (0.7)U_2(2,2)]\\
&U_2(3,1) = -50\\
&U_2(3,2) = (-1) + (0.8)[(0.7)U_2(2,2) + (0.15)U_2(3,3) + (0.15)U_2(3,1)]\\
&U_2(3,3) = (-10) + (0.8)[(0.15)U_2(2,3) + (0.15)U_2(4,3) + (0.7)U_2(3,2)]\\
&U_2(4,2) = (-3) + (0.8)[(0.7)U_2(3,2) + (0.15)U_2(4,3) + (0.15)U_2(4,2)]\\
&U_2(4,3) = (-2) + (0.8)[(0.15)U_2(3,3) + (0.15)U_2(4,3) + (0.7)U_2(4,2)]
\end{aligned}$$

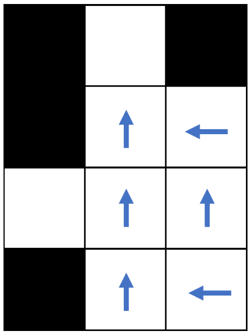
Solving this system of linear equations results in:

$$\begin{aligned}
&U_2(1,2) = 50\\
&U_2(2,2) = 34.3124\\
&U_2(2,3) = 18.2913\\
&U_2(3,1) = -50\\
&U_2(3,2) = 12.0963\\
&U_2(3,3) = -0.988872\\
&U_2(4,2) = 4.33657\\
&U_2(4,3) = 0.352059
\end{aligned}$$

Using these utilities, the best action for each state can be found using Eq. 2:

$\pi_2(2,2) = \max[42.890555, 22.118355, 16.357965, 33.333125] = 42.890555 \rightarrow$ ”Up”
$\pi_2(2,3) = \max[20.694465, 15.3992742, 7.1983446, 26.6140442] = 26.6140442 \rightarrow$ ”$Left$”
$\pi_2(3,2) = \max[16.3703492, 5.1051351, -4.6127318, -29.2026545] = 16.3703492 \rightarrow$ ”Up”
$\pi_2(3,3) = \max[14.4700242, 2.10429345, 1.9125555, 11.26391385] = 14.4700242 \rightarrow$ ”Up”
$\pi_2(4,2) = \max[9.17070435, 2.7113718, 3.73889335, 5.5005295] = 9.17070435 \rightarrow$ ”Up”
$\pi_2(4,3) = \max[0.01108395, 0.15091935, 0.94973565, 2.94007705] = 2.94007705 \rightarrow$ ”$Left$”

Now find $U_3(s)$ for all states using Eq. 1:



$$\begin{aligned}
&U_3(1,2) = 50\\
&U_3(2,2) = (0) + (0.8)[(0.7)U_3(1,2) + (0.15)U_3(2,3) + (0.15)U_3(2,2)]\\
&U_3(2,3) = (-3) + (0.8)[(0.15)U_3(2,3) + (0.15)U_3(3,3) + (0.7)U_3(2,2)]\\
&U_3(3,1) = -50\\
&U_3(3,2) = (-1) + (0.8)[(0.7)U_3(2,2) + (0.15)U_3(3,3) + (0.15)U_3(3,1)]\\
&U_3(3,3) = (-10) + (0.8)[(0.7)U_3(2,3) + (0.15)U_3(3,2) + (0.15)U_3(3,3)]\\
&U_3(4,2) = (-3) + (0.8)[(0.7)U_3(3,2) + (0.15)U_3(4,3) + (0.15)U_3(4,2)]\\
&U_3(4,3) = (-2) + (0.8)[(0.15)U_3(3,3) + (0.15)U_3(4,3) + (0.7)U_3(4,2)]
\end{aligned}$$

Continued on next page ⟶

Solving this system of linear equations results in:

$$\begin{aligned}
U_3(1,2) &= 50\\
U_3(2,2) &= 34.3793\\
U_3(2,3) &= 18.7819\\
U_3(3,1) &= -50\\
U_3(3,2) &= 12.5281\\
U_3(3,3) &= 2.29684\\
U_3(4,2) &= 4.70433\\
U_3(4,3) &= 1.03414
\end{aligned}$$

Using these utilities, the best action for each state can be found using Eq. 2:

$$\begin{aligned}
&\pi_3(2,2) = \max[42.97418, 22.526545, 16.74385, 33.444725] = 42.97418 \rightarrow "Up"\\
&\pi_3(2,3) = \max[21.12151, 16.309141, 9.581968, 27.227321] = 27.227321 \rightarrow "Left"\\
&\pi_3(3,2) = \max[16.910036, 7.4703325, -3.862443, -29.1374555] = 16.910036 \rightarrow "Up"\\
&\pi_3(3,3) = \max[15.371071, 4.580194, 2.947639, 11.742076] = 15.371071 \rightarrow "Up"\\
&\pi_3(4,2) = \max[9.6304405, 3.3087625, 4.1538015, 5.8778955] = 9.6304405 \rightarrow "Up"\\
&\pi_3(4,3) = \max[2.4685585, 1.223545, 1.5846685, 3.792678] = 3.792678 \rightarrow "Left"
\end{aligned}$$

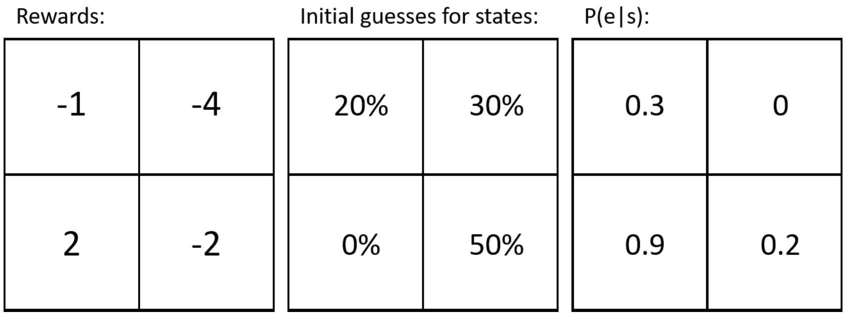
Notice that for each of the states, the best action has not changed from the previous iteration. As such, policy iteration has terminated, allowing us to conclude that:

$$\begin{aligned}
U(1,2) &= U_3(1,2) = 50\\
U(2,2) &= U_3(2,2) = 34.3793\\
U(2,3) &= U_3(2,3) = 18.7819\\
U(3,1) &= U_3(3,1) = -50\\
U(3,2) &= U_3(3,2) = 12.5281\\
U(3,3) &= U_3(3,3) = 2.29684\\
U(4,2) &= U_3(4,2) = 4.70433\\
U(4,3) &= U_3(4,3) = 1.03414
\end{aligned}$$

Problem 4: (25 points)

Assume you have the following POMDP with rewards as shown below. Assume your initial guess of where you are in this POMDP is 20% in the top-left, 30% in the top-right and 50% in the bottom-right (also shown below). Assume the movement is the same as in Problems 2 & 3, but the only actions are moving left or down. There is a boolean evidence variable, e, and $P(e|s)$ is shown for all possible states in the third picture.

Find all possible belief states that result from taking two actions and their associated likelihoods. Which sequence of actions is best if you only take two actions?



Rewards:

-1	-4
2	-2

Initial guesses for states:

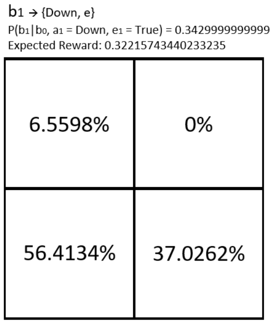
20%	30%
0%	50%

P(e|s):

0.3	0
0.9	0.2

First, notice that from the given belief state, there are two possible actions, move left or move down, and for each of those actions there are two possible evidence values that can be observed. Thus, after taking a single action from the initial belief state, there will be 4 possible belief states that can result. After taking yet another action–this time from the 4 possible belief states that resulted from the first action–each belief state will once again result in 4 possible belief states. Therefore, we expect there to be a total of 16 possible belief states that result from taking two actions from the initial belief state.

The 16 possible belief states that result from taking two actions from the initial belief state (b_0), along with the 4 possible belief states that result from taking a single action from the initial belief state, are all presented on the following three pages. The code used to generate the values used in these belief states can be found in POMDP.py (submitted to HW4-code on Canvas).

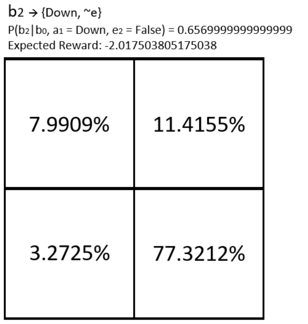


b1 → {Down, e}

$P(b_1|b_0, a_1 = \text{Down}, e_1 = \text{True}) = 0.34299999999999997$

Expected Reward: 0.32215743440233235

6.5598%	0%
56.4134%	37.0262%

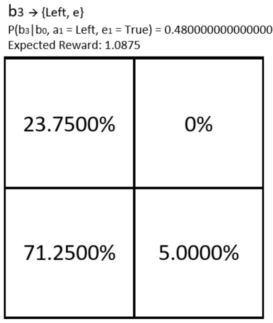


b2 → {Down, ~e}

$P(b_2|b_0, a_1 = \text{Down}, e_2 = \text{False}) = 0.6569999999999999$

Expected Reward: -2.017503805175038

7.9909%	11.4155%
3.2725%	77.3212%

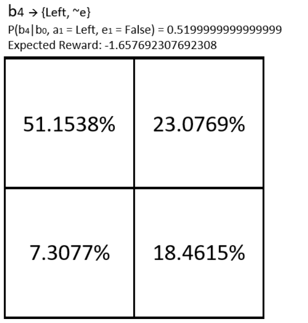


b3 → {Left, e}

$P(b_3|b_0, a_1 = \text{Left}, e_1 = \text{True}) = 0.48000000000000004$

Expected Reward: 1.0875

23.7500%	0%
71.2500%	5.0000%

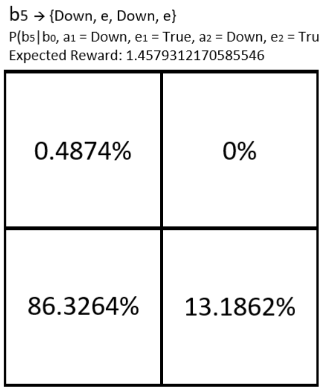


b4 → {Left, ~e}

$P(b_4|b_0, a_1 = \text{Left}, e_1 = \text{False}) = 0.5199999999999999$

Expected Reward: -1.657692307692308

51.1538%	23.0769%
7.3077%	18.4615%

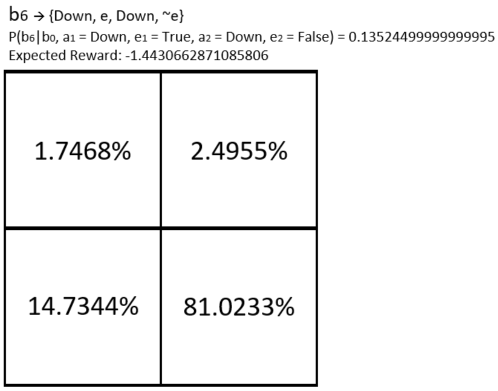


b_5 → {Down, e, Down, e}

P(b_5|b_0, a_1 = Down, e_1 = True, a_2 = Down, e_2 = True) = 0.20775500000000002

Expected Reward: 1.4579312170585546

0.4874%	0%
86.3264%	13.1862%

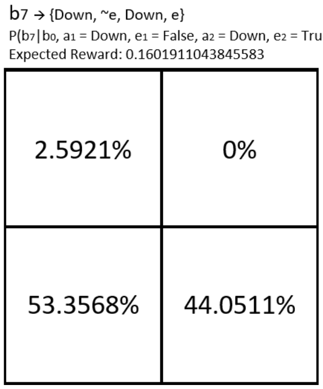


b_6 → {Down, e, Down, ~e}

P(b_6|b_0, a_1 = Down, e_1 = True, a_2 = Down, e_2 = False) = 0.13524499999999995

Expected Reward: -1.4430662871085806

1.7468%	2.4955%
14.7344%	81.0233%

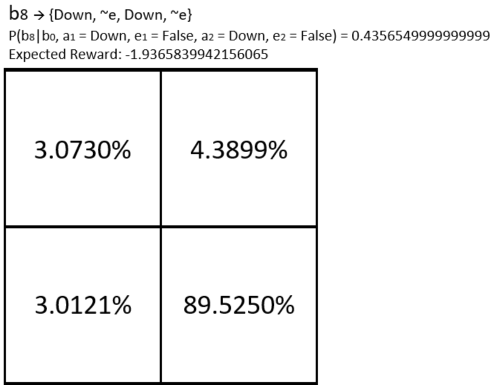


b_7 → {Down, ~e, Down, e}

P(b_7|b_0, a_1 = Down, e_1 = False, a_2 = Down, e_2 = True) = 0.22134499999999996

Expected Reward: 0.1601911043845583

2.5921%	0%
53.3568%	44.0511%

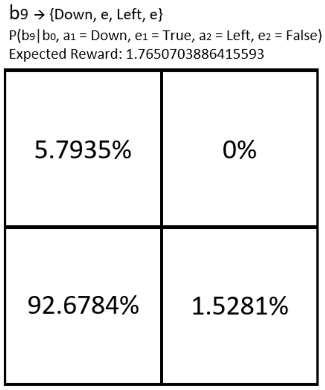


b_8 → {Down, ~e, Down, ~e}

P(b_8|b_0, a_1 = Down, e_1 = False, a_2 = Down, e_2 = False) = 0.4356549999999999

Expected Reward: -1.9365839942156065

3.0730%	4.3899%
3.0121%	89.5250%

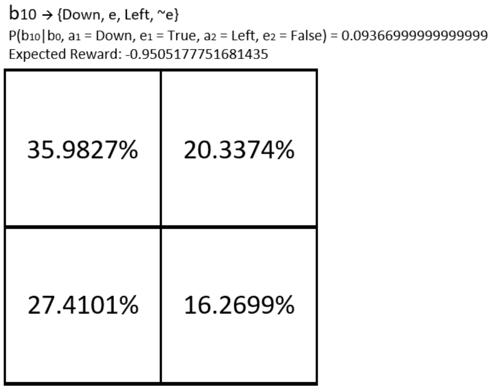


b_9 → {Down, e, Left, e}

P(b_9|b_0, a_1 = Down, e_1 = True, a_2 = Left, e_2 = False) = 0.24932999999999994

Expected Reward: 1.7650703886415593

5.7935%	0%
92.6784%	1.5281%

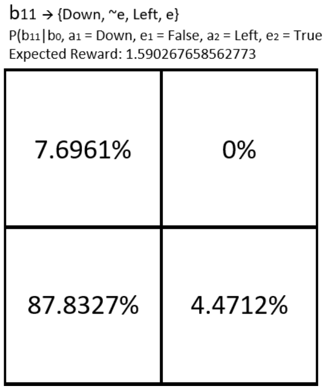


b_{10} → {Down, e, Left, ~e}

P(b_{10}|b_0, a_1 = Down, e_1 = True, a_2 = Left, e_2 = False) = 0.09366999999999999

Expected Reward: -0.9505177751681435

35.9827%	20.3374%
27.4101%	16.2699%

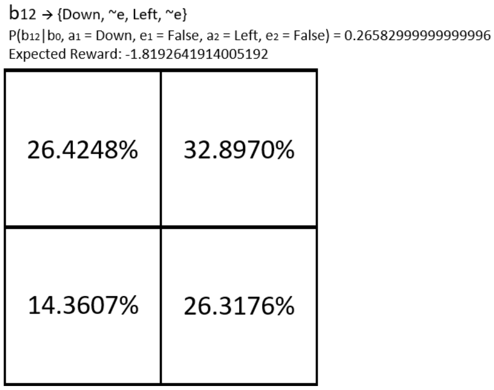


b_{11} → {Down, ~e, Left, e}

P(b_{11}|b_0, a_1 = Down, e_1 = False, a_2 = Left, e_2 = True) = 0.39116999999999996

Expected Reward: 1.590267658562773

7.6961%	0%
87.8327%	4.4712%



b_{12} → {Down, ~e, Left, ~e}

P(b_{12}|b_0, a_1 = Down, e_1 = False, a_2 = Left, e_2 = False) = 0.26582999999999996

Expected Reward: -1.8192641914005192

26.4248%	32.8970%
14.3607%	26.3176%

Continued on next page ⟶

b_{13} → {Left, e, Down, e}
P(b_{13}|b_0, a_1 = Left, e_1 = True, a_2 = Down, e_2 = True) = 0.3561599999999999
Expected Reward: 1.7957378706199463

1.4404%	0%
94.5334%	4.0263%

b_{14} → {Left, e, Down, ~e}
P(b_{14}|b_0, a_1 = Left, e_1 = True, a_2 = Down, e_2 = False) = 0.12383999999999999
Expected Reward: -0.9711724806201554

9.6657%	13.8081%
30.2083%	46.3178%

b_{15} → {Left, ~e, Down, e}
P(b_{15}|b_0, a_1 = Left, e_1 = False, a_2 = Down, e_2 = True) = 0.26123999999999997
Expected Reward: 1.2759531465319245

6.6491%	0%
80.2367%	13.1144%

b_{16} → {Left, ~e, Down, ~e}
P(b_{16}|b_0, a_1 = Left, e_1 = False, a_2 = Down, e_2 = False) = 0.25875999999999993
Expected Reward: -1.9308625753594066

15.6632%	22.3759%
9.0006%	52.9603%

- -

b_{17} → {Left, e, Left, e}
P(b_{17}|b_0, a_1 = Left, e_1 = True, a_2 = Left, e_2 = True) = 0.33731999999999995
Expected Reward: 1.5960512273212382

13.1804%	0%
86.6062%	0.2134%

b_{18} → {Left, e, Left, ~e}
P(b_{18}|b_0, a_1 = Left, e_1 = True, a_2 = Left, e_2 = False) = 0.14268
Expected Reward: -0.4133725820016822

72.7082%	2.5231%
22.7502%	2.0185%

b_{19} → {Left, ~e, Left, e}
P(b_{19}|b_0, a_1 = Left, e_1 = False, a_2 = Left, e_2 = True) = 0.22667999999999996
Expected Reward: 0.6318157755426154

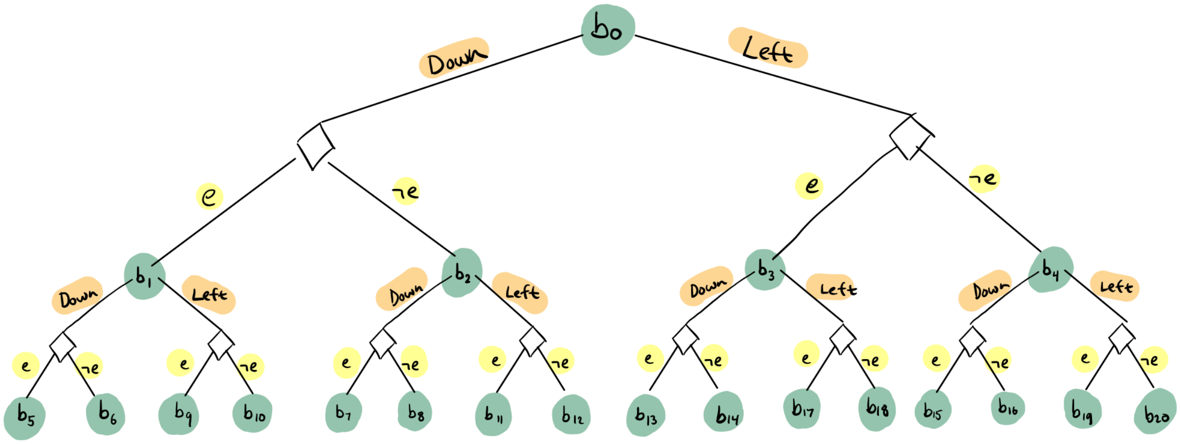
41.7946%	0%
55.3467%	2.8587%

b_{20} → {Left, ~e, Left, ~e}
P(b_{20}|b_0, a_1 = Left, e_1 = False, a_2 = Down, e_2 = False) = 0.2933199999999999
Expected Reward: -1.2771716896222556

75.3648%	11.0460%
4.7525%	8.8368%

Continued on next page ⟶

Using the preceding 21 belief states (including the initial belief state b_0), along with the search tree below that corresponds to this problem's 2 action state-space, the expected reward for each sequence of two actions can be calculated. Actions are highlighted in orange and evidence is highlighted in yellow, with the resulting belief states highlighted in green.



Working backwards from the leaves of the search tree, with $\gamma = 0.8$ as in problems 2 and 3, the expected reward for each sequence of actions is determined below:

$$\begin{aligned}
[Down, Down] \rightarrow ... + \big[... + \gamma\big(... + \gamma\big(\rho(b_5)P(b_5|a_1 = Down, e_1, a_2 = Down, e_2) + \rho(b_6)P(b_6|a_1 = Down, e_1, a_2 = Down, \neg e_2)\big)\big)\big] \\
&= ... + \big[... + \gamma\big(... + (0.8)\big((1.457931217)(0.207755) + (-1.44306628710858)(0.135244999999999)\big)\big)\big] \\
&= ... + \big[... + \gamma\big(... + 0.08618\big)\big] \\
&= ... + \big[... + \gamma\big(\rho(b_1)P(b_1|a_1 = Down, e_1) + 0.08618\big)\big] \\
&= ... + \big[... + (0.8)\big((0.322157434402332)(0.342999999999999) + 0.08618\big)\big] \\
&= ... + \big[... + 0.157344\big] \\
&= ... + \big[\gamma\big(... + \gamma\big[\rho(b_7)P(b_7|a_1 = Down, \neg e_1, a_2 = Down, e_2) + \rho(b_8)P(b_8|a_1 = Down, \neg e_1, a_2 = Down, \neg e_2)\big]\big) + 0.157344\big] \\
&= ... + \big[\gamma\big(... + (0.8)\big[(0.160191104384558)(0.221344999999999) + (-1.9365839942156)(0.435654999999999)\big]\big) + 0.157344\big] \\
&= ... + \big[\gamma\big(... + (-0.64658)\big) + 0.157344\big] \\
&= ... + \big[\gamma\big(\rho(b_2)P(b_2|a_1 = Down, \neg e_1) + (-0.64658)\big) + 0.157344\big] \\
&= ... + \big[(0.8)\big((-2.01750380517503)(0.656999999999999) + (-0.64658)\big) + 0.157344\big] \\
&= ... + \big[(0.8)\big((-1.3255) + (-0.64658)\big) + 0.157344\big] \\
&= ... + \big[(-1.577664) + 0.157344\big] \\
&= \rho(b_0)P(b_0) + \big[(-1.577664) + 0.157344\big] \\
&= (-2.4)(1) + \big[(-1.577664) + 0.157344\big] \\
&= -3.82032
\end{aligned}$$

Continued on next page ⟶

$$\begin{aligned}
[Down, Left] \rightarrow\ & ... + \big[... + \gamma\big(... + \gamma\big(\rho(b_9)P(b_9|a_1 = Down, e_1, a_2 = Left, e_2) + \rho(b_{10})P(b_{10}|a_1 = Down, e_1, a_2 = Left, \neg e_2)\big)\big)\big] \\
&= ... + \big[... + \gamma\big(... + (0.8)\big((1.76507038864155)(0.249329999999999) + (-0.950517775168143)(0.0936699999999999)\big)\big)\big] \\
&= ... + \big[... + \gamma\big(... + 0.28084\big)\big] \\
&= ... + \big[... + \gamma\big(\rho(b_1)P(b_1|a_1 = Down, e_1) + 0.28084\big)\big] \\
&= ... + \big[... + (0.8)\big((0.322157434402332)(0.342999999999999) + 0.28084\big)\big] \\
&= ... + \big[... + 0.313072\big] \\
&= ... + \big[\gamma\big(... + \gamma\big[\rho(b_{11})P(b_{11}|a_1 = Down, \neg e_1, a_2 = Left, e_2) + \rho(b_{12})P(b_{12}|a_1 = Down, \neg e_1, a_2 = Left, \neg e_2)\big]\big) + 0.313072\big] \\
&= ... + \big[\gamma\big(... + (0.8)\big[(1.59026765856277)(0.391169999999999) + (-1.81926419140051)(0.265829999999999)\big]\big) + 0.313072\big] \\
&= ... + \big[\gamma\big(... + (0.11076)\big) + 0.313072\big] \\
&= ... + \big[\gamma\big(\rho(b_2)P(b_2|a_1 = Down, \neg e_1) + (0.11076)\big) + 0.313072\big] \\
&= ... + \big[(0.8)\big((-2.01750380517503)(0.656999999999999) + (0.11076)\big) + 0.313072\big] \\
&= ... + \big[(0.8)\big((-1.3255) + (0.11076)\big) + 0.313072\big] \\
&= ... + \big[(-0.971792) + 0.313072\big] \\
&= \rho(b_0)P(b_0) + \big[(-0.971792) + 0.313072\big] \\
&= (-2.4)(1) + \big[(-0.971792) + 0.313072\big] \\
&= -3.05872
\end{aligned}$$

$$\begin{aligned}
[Left, Left] \rightarrow\ & ... + \big[... + \gamma\big(... + \gamma\big(\rho(b_{17})P(b_{17}|a_1 = Left, e_1, a_2 = Left, e_2) + \rho(b_{18})P(b_{18}|a_1 = Left, e_1, a_2 = Left, \neg e_2)\big)\big)\big] \\
&= ... + \big[... + \gamma\big(... + (0.8)\big((1.59605122732123)(0.337319999999999) + (-0.413372582001682)(0.14268)\big)\big)\big] \\
&= ... + \big[... + \gamma\big(... + 0.38352\big)\big] \\
&= ... + \big[... + \gamma\big(\rho(b_3)P(b_3|a_1 = Left, e_1) + 0.38352\big)\big] \\
&= ... + \big[... + (0.8)\big((1.0875)(0.48) + 0.38352\big)\big] \\
&= ... + \big[... + 0.724416\big] \\
&= ... + \big[\gamma\big(... + \gamma\big[\rho(b_{19})P(b_{19}|a_1 = Left, \neg e_1, a_2 = Left, e_2) + \rho(b_{20})P(b_{20}|a_1 = Left, \neg e_1, a_2 = Left, \neg e_2)\big]\big) + 0.724416\big] \\
&= ... + \big[\gamma\big(... + (0.8)\big[(0.631815775542615)(0.226679999999999) + (-1.27717168962225)(0.293319999999999)\big]\big) + 0.724416\big] \\
&= ... + \big[\gamma\big(... + (-0.18512)\big) + 0.724416\big] \\
&= ... + \big[\gamma\big(\rho(b_4)P(b_4|a_1 = Left, \neg e_1) + (-0.18512)\big) + 0.724416\big] \\
&= ... + \big[(0.8)\big((-1.6576923076923)(0.519999999999999) + (-0.18512)\big) + 0.724416\big] \\
&= ... + \big[(0.8)\big((-0.862) + (-0.18512)\big) + 0.724416\big] \\
&= ... + \big[(-0.837696) + 0.724416\big] \\
&= \rho(b_0)P(b_0) + \big[(-0.837696) + 0.724416\big] \\
&= (-2.4)(1) + \big[(-0.837696) + 0.724416\big] \\
&= -2.51328
\end{aligned}$$

Continued on next page ⟶

$$\begin{aligned}
[Left, Down] \rightarrow\ & ... + \big[... + \gamma\big(... + \gamma\big(\rho(b_{13})P(b_{13}|a_1 = Left, e_1, a_2 = Down, e_2) + \rho(b_{14})P(b_{14}|a_1 = Left, e_1, a_2 = Down, \neg e_2)\big)\big)\big] \\
&= ... + \big[... + \gamma\big(... + (0.8)\big((1.79573787061994)(0.356159999999999) + (-0.971172480620155)(0.123839999999999)\big)\big)\big] \\
&= ... + \big[... + \gamma\big(... + 0.41544\big)\big] \\
&= ... + \big[... + \gamma\big(\rho(b_3)P(b_3|a_1 = Left, e_1) + 0.41544\big)\big] \\
&= ... + \big[... + (0.8)\big((1.0875)(0.48) + 0.41544\big)\big] \\
&= ... + \big[... + 0.749952\big] \\
&= ... + \big[\gamma\big(... + \gamma\big[\rho(b_{15})P(b_{15}|a_1 = Left, \neg e_1, a_2 = Down, e_2) + \rho(b_{16})P(b_{16}|a_1 = Left, \neg e_1, a_2 = Down, \neg e_2)\big]\big) + 0.749952\big] \\
&= ... + \big[\gamma\big(... + (0.8)\big[(1.27595314653192)(0.261239999999999) + (-1.9308625753594)(0.258759999999999)\big]\big) + 0.749952\big] \\
&= ... + \big[\gamma\big(... + (-0.13304)\big) + 0.749952\big] \\
&= ... + \big[\gamma\big(\rho(b_4)P(b_4|a_1 = Left, \neg e_1) + (-0.13304)\big) + 0.749952\big] \\
&= ... + \big[(0.8)\big((-1.6576923076923)(0.519999999999999) + (-0.13304)\big) + 0.749952\big] \\
&= ... + \big[(0.8)\big((-0.862) + (-0.13304)\big) + 0.749952\big] \\
&= ... + \big[(-0.796032) + 0.749952\big] \\
&= \rho(b_0)P(b_0) + \big[(-0.796032) + 0.749952\big] \\
&= (-2.4)(1) + \big[(-0.796032) + 0.749952\big] \\
&= -2.44608
\end{aligned}$$

Thus the best sequence of two actions is $[Left, Down]$ as this sequence has the greatest expected reward (-2.44608) compared to the other possible sequences ($[Down, Down] \rightarrow -3.82032, [Down, Left] \rightarrow -3.05872, [Left, Left] \rightarrow -2.51428$).