**AI – Theo Assignment 2**

Pan Eyal – 208722058

**Question 1**

Begin with the initial plan containing the dummy steps:  
:

Preconditional:   
 effects:

:

precondition:   
effects:

Constraints order:

This state is not a solution, so we pick the unsatisfied precondition in .  
This precondition cannot be satisfied by existing step, so add:

:

Preconditional:   
 effects:

Constraints order:

Now, precondition of can be satisfied by effect of so this link is added.  
Now, precondition of can be satisfied by effect of so this link is added.  
Now, precondition cannot be satisfied by existing step, so add:

:

Preconditional:   
 effects:

Constraints order:

Add the link from effect with of to the same precondition of .  
Now, precondition of can be satisfied by effect of so this link is added.  
Now, precondition cannot be satisfied by existing step, so add:

:

Preconditional:   
 effects:

Constraints order:

Add the link from effect with of to the same precondition of .  
Now, precondition of can be satisfied by effect of so this link is added.  
Now, precondition of can be satisfied by effect of so this link is added.

However, clobbers the precondition of , so add:

:

Preconditional:   
 effects:

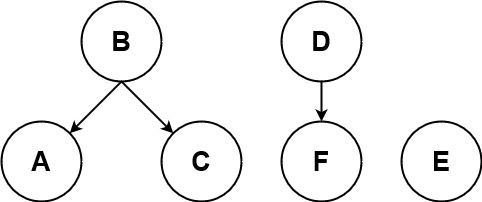
Constraints order:

Now, precondition of can be satisfied by effect of so this link is added.  
Now, precondition of can be satisfied by effect of so this link is added.  
Now, precondition of can be satisfied by effect of so this link is added.

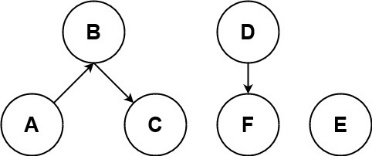
**No more clobbering remaining, and all preconditions are met.  
We can declare on the final plan with above steps and constraints:**

**Question 2**

1. An example for a possible Bayes network:



1. No, it is not unique. You can also create the following networks:

**, A picture containing text, clipart

Description automatically generated,**

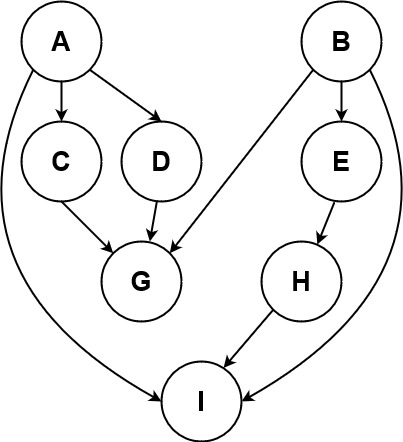
**A picture containing text, clipart

Description automatically generated, A picture containing text, clipart

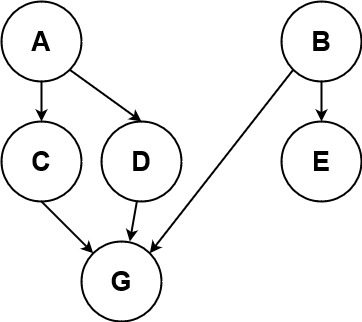
Description automatically generated, A picture containing text, clipart

Description automatically generated**

**Question 3**



1. No, it is not a poly-tree. The underlying undirected graph contains a cycle path:
2. No, it is not singly connected. We can go to from in two paths: and
   1. Yes, every path from to is blocked by or .
   2. Yes, every path from to is blocked by or .
   3. Yes, every path from to is blocked by or .
   4. No, and are connected.
   5. Yes, every path from to is blocked by or .
   6. No, path isn’t blocked now.
3. For computing we can remove unnecessary nodes and get the following graph:



Now, we can see that is independent from because B blocks all paths from to them. Therefore: .

And we know that   
So, as well.

**Question 4**

* 1. For undiscounted rewards:

Without filing a request is guaranteed to gain 1000 MEMU.

So,

By filing a request with no facilitator, we lose 200 MEMU, and now can calculate the utility value:

By filing a request with facilitator, we lose 600 MEMU (200 for the request and 400 for the officials to “see reason”) and gain 2500 MEMU for sure, and now can calculate the utility value:

Therefore, the preferred option is to go with **option** **A.**

* 1. For discounted rewards:

The only changed value will be of option C:

Now, the preferred option is to go with **option** **C.**

* 1. For undiscounted rewards & taking legal advice:

Without filing a request is guaranteed to gain 1000 MEMU and lose 100 MEMU for the attorney.

So,

By filing a request with no facilitator, we lose 200 MEMU, lose 100 MEMU for the attorney. Now, can calculate the utility value:

By filing a request with facilitator, we lose 100 MEMU for the attorney. There are now two options based on his answer: F1 and F2

F1: Filing a request with facilitator, gain 2500 MEMU and if charges will be brought, they won’t stick.

F2: Filing a request with facilitator, gain 2500 MEMU and if charges will be brought, they will stick.

Now, being convicted based on charges can happen in 50% chances. Meaning that the information from receiving the layer advice will also be split accordingly. We can calculate the utility value for the taking legal advice G:

Therefore, the preferred option is to go with legal advice**.**

* 1. For discounted rewards:

The only changed value will be of option F1 and F2:

F1: Filing a request with facilitator, gain 2500 MEMU and if charges will be brought, they won’t stick.

F2: Filing a request with facilitator, gain 2500 MEMU and if charges will be brought, they will stick.

We can calculate the utility value for the taking legal advice G:

So, the preferred option is still to go with the legal advice.

**Question 5**

**Diagram

Description automatically generated**Our graph:

b1)

* Belief state is composed of the following state variables:

// The current location in state

for every // T if blocked, if not, if unknown. And are never blocked.

// The accumulating cost until now

* Transition probabilities are stochastic when the resulting location is adjacent to one or more edges with state , in which case its state becomes or , depending on the blockage probability of its vertex.
* Our reward will be given only when arriving to .  
  So, for all states, except when .  
  Which for it: . Where ““ can receive any of the following values .

b2) Let’s starts with the quickest path .  
For state we have two traversals to vertex :

For

For

From we have 2 traversals to vertex :

For

For

From we have 1 traversal to vertex :

For

From we have 2 traversals to vertex :

For

For

In state we know the values of all variables in this state. So, we can easily see that the optimal path will be through . And therefore, we receive that . The cost along the route .

Now, if is blocked, we can’t go from to and we go to instead. In state we know the values of all variables in this state. We can easily see that the optimal path will be returning from where we came from. And therefore, we receive that . The cost along the route .

That means that

Now, if V3 is blocked, we can’t go from to , and we go to instead We can easily see that the optimal path will be returning from where we came from. For that let’s continue building possible states on the route

From we have 1 traversal to vertex :

For

From we have 1 traversal to vertex :

For

From we have 2 traversals to vertex :

For

For

In state we know the values of all variables in this state. So, we can easily see that the optimal path will be through . And therefore, we receive that . The cost along the route .

Now, if is blocked, we can’t go from to and we go to instead. In state we know the values of all variables in this state. We can easily see that the optimal path will be returning from where we came from. And therefore, we receive that . The cost along the route .

That means that

And that means that

Now, if V2 is blocked, we can’t go from to , and we go to instead We can easily see that the optimal path will be returning from where we came from. For that let’s continue building possible states on the route

From we have 1 traversal to vertex :

For

From we have 4 traversals to vertex :

For

For

For

For

In state and we know the values of all variables in this state.  
So, we can easily see that the optimal path will be through .  
And therefore, we receive that .  
The cost along the route .

Now, if is blocked, we can’t go from to or and we go to or instead.  
In state and we know the values of all variables in this state.  
We can easily see that the optimal path in both cases will be returning from where we came from. And therefore, we receive that .  
The cost along the route .

That means that

**Now**, we can start working on a different path from : For state we have 4 traversals to vertex :

For

For

For

For

In state and we can easily see that the optimal path will be through .  
And therefore, we receive that .  
The cost along the route .

Now, if is blocked, we can’t go from to or and we can go to instead.  
Let’s try to go for the optimal new path: .

For state we have 2 traversals to vertex :

For

For

In state we know the values of all variables in this state.  
So, we can easily see that the optimal path will be through .  
And therefore, we receive that .  
The cost along the route .

Now, if is blocked, we can’t go from to and we can go to instead.  
Let’s try to go for the optimal new path: .

In state we know the values of all variables in this state.  
So, we can easily see that the optimal path will be through .  
And therefore, we receive that .  
The cost along the route .

And that means that, however, just getting back to from will get us a cost of . And therefore

Now, if is blocked, we can’t go from to and we can go to instead.  
In this state we can easily see that the optimal path will be through .  
And therefore, we receive that .  
The cost along the route .

Now, when we finished calculating the optimal policy for the given start vertex, we know that by going to we will get that the utility is:

By going to ( or ) we will get that the utility is:

**So, we can see that our optimal policy will be to go to from the initial state.  
Then if is not blocked, going through it will be the quickest. If it is blocked, it will be best to just go back as well to and don’t try our luck with .**

**We’ve reached our destination.**

b3) The new information SENSING can give us is only about the state of . Therefore, the decision to do the sensing or not to do it can happen as the first thing.  
For state we have two traversals to make a SENSING action:

For

For

Let’s starts with the quickest path .

For state we have 1 traversal to vertex :

For

For state we have 2 traversals to vertex :

For

For

For state we have 1 traversal to vertex :

For

For state we have 2 traversals to vertex :

For

For

In state we know the values of all variables in this state.  
So, we can easily see that the optimal path will be through .  
And therefore, we receive that .  
The cost along the route .

Now, if is blocked, we can’t go from to and we go to instead.  
In state we know the values of all variables in this state.  
We can easily see that the optimal path will be returning from where we came from.  
And therefore, we receive that .  
The cost along the route .

That means that

Now, if V3 is blocked, we can’t go from to , and we go to instead We can easily see that the optimal path will be returning from where we came from. For that let’s continue building possible states on the route

From we have 1 traversal to vertex :

For

From we have 1 traversal to vertex :

For

From we have 2 traversals to vertex :

For

For

In state we know the values of all variables in this state.  
So, we can easily see that the optimal path will be through .  
And therefore, we receive that .  
The cost along the route .

Now, if is blocked, we can’t go from to and we go to instead.  
In state we know the values of all variables in this state.  
We can easily see that the optimal path will be returning from where we came from.  
And therefore, we receive that .  
The cost along the route .

That means that

And that means that

Another option is to go to from state .  
From we have 4 traversals to vertex :

For

For

For

For

In state and we can easily see that the optimal path will be through .  
And therefore, we receive that .  
The cost along the route .

Now, if is blocked, we can’t go from to or . We can go to instead.  
In state we know the values of all variables in this state.  
We can easily see that the optimal path will be returning from .  
And therefore, we receive that .  
The cost along the route .

Now, if is blocked, we can’t go from to either. We can go to instead.  
In state we know the values of all variables in this state.  
We can easily see that the optimal path will be back through .  
And therefore, we receive that .  
The cost along the route .

That means that we can now update :

**Now**, if V2 is blocked, we know from the SENSING action that we better try going to V4.  
Let’s starts with the path . From we will go to instead.

From we have 4 traversals to vertex :

For

For

For

For

In state and we can easily see that the optimal path will be through .  
And therefore, we receive that .  
The cost along the route .

Now, if is blocked, we can’t go from to or . Because we know that is blocked, we shouldn’t go to , and the quickest path from here is through . So, for both we can easily see that the optimal path will be back through .  
And therefore, we receive that .  
The cost along the route .

That means that

Now:

**So, we can see that doing the SENSING action did help us and got our reward a bit higher.  
Our optimal policy now will be doing the SENSING action and then go to instead of risking our ways with the passage in more blocking vertices (such as ). If is not blocked we will go through there, but if it is, since we know blockage, we can choose to go through if it is not blocked or go through in the case that it is.**

**Question 6**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **decision** |
| 2 | F | L | 0 | N |
| 2 | F | H | 1 | Y |
| 2 | T | L | 0 | N |
| 3 | F | H | 0 | N |
| 3 | T | H | 2 | Y |
| 3 | T | L | 0 | Y |
| 3 | T | H | 1 | Y |

Need to check all 4 attributes as candidate for root. We get:

With A as root:

A=2: 2N, 1Y   
 A=3: 1N, 3Y

With B as root:

B=F: 2N, 1Y   
 B=T: 1N, 3Y

With C as root:

C=L: 2N, 1Y   
 C=H: 1N, 3Y

With D as root:

D=0: 3N, 1Y   
D=1: 0N, 2Y   
D=2: 0N, 1Y

We easily see that has the best separation in features. For or we can make a final decision and wouldn’t need to make another branching. Let’s continue for .

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **decision** |
| 2 | F | L | 0 | N |
| 2 | T | L | 0 | N |
| 3 | F | H | 0 | N |
| 3 | T | L | 0 | Y |

With A as root:

A=2: 2N, 0Y   
 A=3: 1N, 1Y

With B as root:

B=F: 2N, 0Y   
 B=T: 1N, 1Y

With C as root:

C=L: 2N, 1Y   
 C=H: 1N, 0Y

We can choose either or as they have the same entropy. For we can make a final decision and wouldn’t need to make another branching. Let’s continue for .

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **decision** |
| 3 | F | H | 0 | N |
| 3 | T | L | 0 | Y |

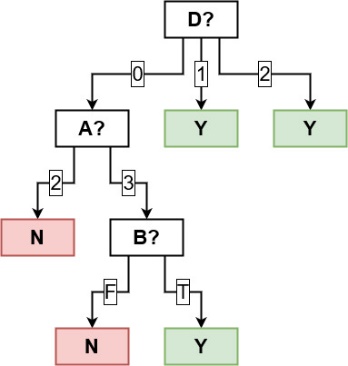
With B as root:

B=F: 1N, 0Y   
 B=T: 0N, 1Y

With C as root:

C=L: 0N, 1Y   
 C=H: 1N, 0Y

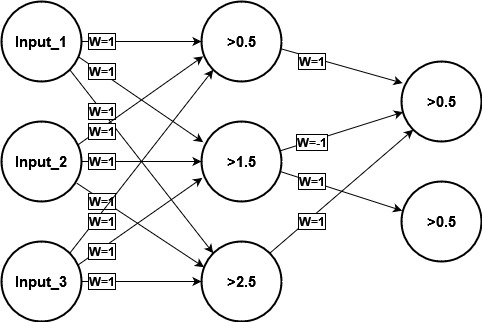
We can choose either or as they have the same entropy. Let’s choose and get the following decision tree:



1. Less than 3 nodes are not possible. If we choose a different node than as our root, then a concise decision still cannot be made (since the entropy wasn’t 0 in for the other cases), and we would have gotten 3 internals for sure. If we choose as our root but a different node for the case where , then we would still have another branch with incomplete classification. Therefore, a more compact decision tree is not possible for the above table.

**Question 7**

1. No, in case where we will have a 0 in some input bit of the input layer, we will encounter a scenario that we could not solve. The network needs to determine if the other input bits are 0 and 1 or 1 and 0 to decide if the number of the lightened bits is odd.
2. Let’s look at the following neural network:



Let’s create a hidden unit to understand the number of overall lighted bits.  
The fully connected first layer with weights 1 for each edge will tell us:

If passed the threshold: at least 1 bit is lightened.  
If passed the threshold: at least 2 bit are lightened.  
If passed the threshold: at least 3 bit are lightened.

Now, if , and thresholds does not passed in the hidden layer, we can deduce that all bits are not lightened up and the number of inputs with value 1 is even.  
And there aren’t at least 2 inputs with value 1.  
So overall, for output 1 we get that:  
and for output 2 we get that:

So, the output will be as needed.

Now, if threshold pass but and does not passed in the hidden layer, we can deduce that only 1 bit is lightened up and the number of inputs with value 1 is odd.  
But there aren’t at least 2 inputs with value 1.  
So overall, for output 1 we get that:  
and for output 2 we get that:

So, the output will be as needed.

Now, if and thresholds pass but does not passed in the hidden layer, we can deduce that only 2 bits are lightened up and the number of inputs with value 1 is even.  
And there are at least 2 inputs with value 1.  
So overall, for output 1 we get that:  
and for output 2 we get that:

So, the output will be as needed.

Now, if , and thresholds pass in the hidden layer, we can deduce that all 3 bits are lightened up and the number of inputs with value 1 is odd.  
And there are at least 2 inputs with value 1.  
So overall, for output 1 we get that:  
and for output 2 we get that:

So, the output will be as needed.