CSE643 – Artificial Intelligence

Monsoon 2021 session

Mid-sem exam

25-Sep-2021   **Time: 10:00 AM to 1:00PM**

Max marks: 50 (will be normalized to 25 marks)

Deadline to submit your answer sheets: 25-Sep-21 - 1:00 PM

INSTRUCTIONS: You will have to create a PDF file with your answers, name the file as AI-MidSem-<Name>-<RollNo> and upload it on the classroom page by 1:00 pm. In the answer sheet write or type your name and roll number. In case you choose to have hand-written answers then those pages can be scanned and uploaded (make sure that it is clearly readable). Make sure it has the name and roll number on it.

Q1: There are three parts to this question   (5 marks)

i)   Express the following as FOPL sentences.

Pahul was a man and he was a gambler. All men are people. Viput Singh was a don. All gamblers were either loyal to Viput Singh or hated him. Everyone is loyal to someone. People only try to assassinate dons they are not loyal to. Pahul tried to assassinate Viput Singh.

ii)   Represent the FOPL sentences in CNF showing all steps.
iii)   Using resolution refutation (and also draw the graph) determine whether Pahul hated Viput Singh. <Note: in the graph do not use abbreviations for the CNF>

Answers for Q1

Predicates used

i)   man(x) /* x is a man */
ii)   gambler(x)   /*x is a gambler */
iii) people(x)   /* x is of type people */
iv) don(x)   /* x is a don */

v) loyal(x, y)   /* x is loyal to y */
vi) hates(x, y)   /* x hates y */
vii) assassinate(x,y)   /* x tries to assassinate y */

(i) Representing given sentences in FOPL

1) man(Pahul) ∧ gambler(Pahul)
2) ∀(x) man(x) → people(x)
3) don(Viput Singh)
4) ∀(x) gambler(x) → loyal(x, Viput Singh) v hates(x, Viput Singh)
5) ∀(x) people(x) → ∃(y) loyal(x, y)
6) ∀(x) ∀(y) people(x) ^ don(y) ^ assassinate(x, y) → ¬loyal(x, y)
7) assassinate(Pahul, Viput Singh)

(ii) Representing FOPL sentences in CNF

8) man(Pahul)
9) gambler(Pahul)
10) ¬man(x) v people(x)   /* rewriting (2) in clausal form
11) don(Viput Singh)
12) ¬gambler(x) v loyal(x, Viput Singh) v hates(x, Viput Singh)   /* rewriting (4) in clausal form
13) ∀(x) people(x) → loyal(x, g(x))   /* introducing a skolem function g(x)
14) ¬people(x) v loyal(x, g(x)) ·˙   /* rewriting the above in clausal form
15) ¬people(x) v ¬don(y) v ¬assassinate(x, y) v ¬loyal(x, y)   /* rewriting (6) in clausal form
16) ¬people(x) v ¬don(g(x)) v ¬assassinate(x, g(x)) v ¬loyal(x, g(x)) /* using skolem function for y
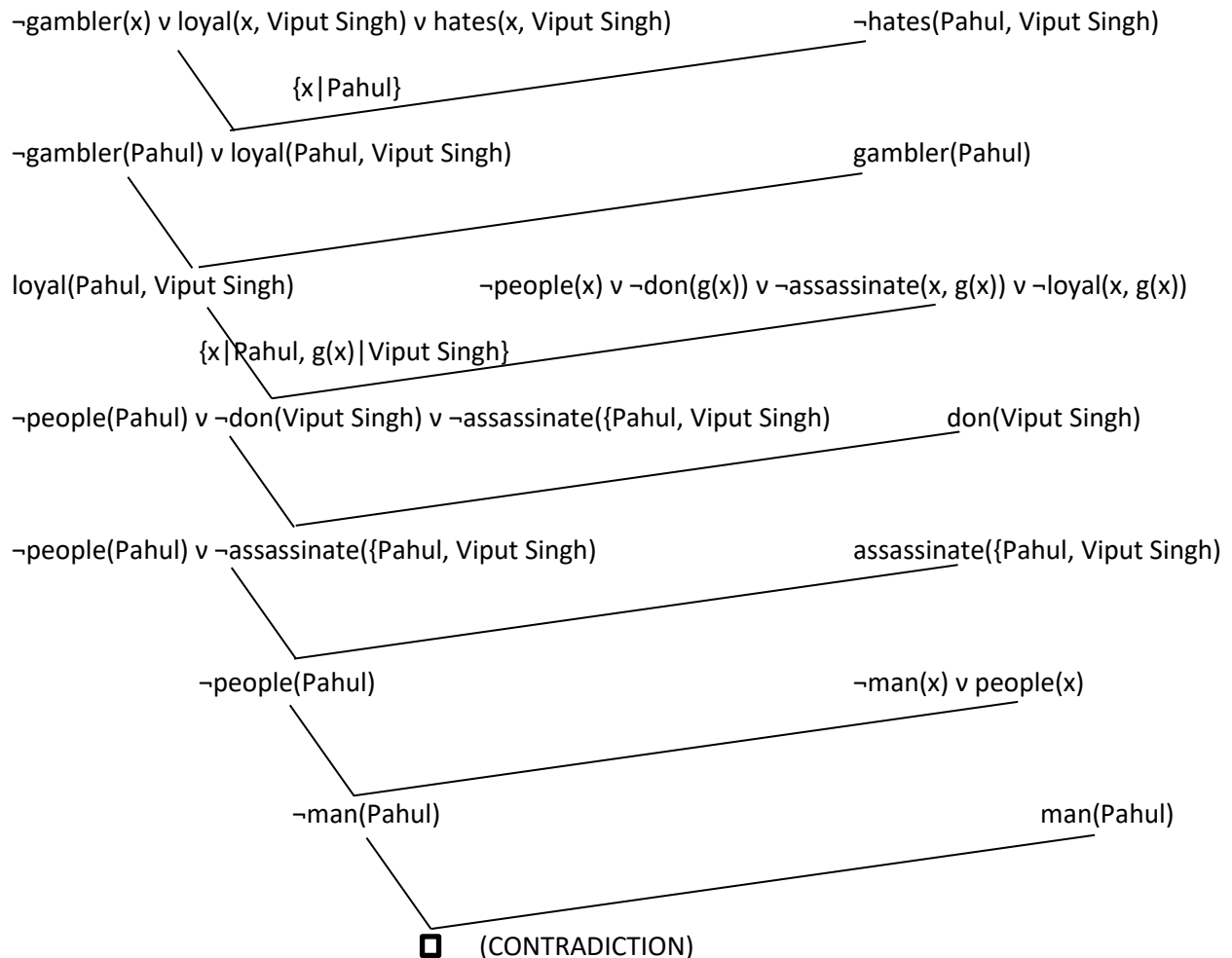17) assassinate(Pahul, Viput Singh)

(iii) Resolution refutation

To determine the truth value of the clause   hates(Pahul, Viput Singh).

18) Assume negation of the clause   ¬hates(Pahul, Viput Singh) and add it to the set of clauses
19) Unifying clause (12) with (18) with substitution {x|Pahul} and resolution of hates predicate we get ¬gambler(Pahul) v loyal(Pahul, Viput Singh)
20) Clause (9) resolves with (19) with resolution of gambler predicate to give loyal(Pahul, Viput Singh)
21) Clause (16) and clause (20) unify with the substitution {x|Pahul, g(x)|Viput Singh} and with the resolution of loyal predicate we get ¬people(Pahul) v ¬don(Viput Singh) v ¬assassinate({Pahul, Viput Singh)
22) Clause (21) resolves with (11) of don predicate and then resolves with (17) of assassinate predicate to give ¬people(Pahul)
23) Clause (22) unifies with (10) with substitution {x|Pahul} and resolves with people predicate to give ¬man(Pahul).

24) Clause (22) and (8) unify with substitution {x|Pahul} and resolve with man predicate to give a CONTRADICTION. Thus, the negation of our hypothesis ¬hates(Pahul, Viput Singh) is FALSE which implies that hates(Pahul, Viput Singh) is TRUE.

Resolution graph

¬gambler(x) v loyal(x, Viput Singh) v hates(x, Viput Singh)   ¬hates(Pahul, Viput Singh)

{x|Pahul}

¬gambler(Pahul) v loyal(Pahul, Viput Singh)   gambler(Pahul)

loyal(Pahul, Viput Singh)   ¬people(x) v ¬don(g(x)) v ¬assassinate(x, g(x)) v ¬loyal(x, g(x))

{x|Pahul, g(x)|Viput Singh}

¬people(Pahul) v ¬don(Viput Singh) v ¬assassinate({Pahul, Viput Singh)   don(Viput Singh)

¬people(Pahul) v ¬assassinate({Pahul, Viput Singh)   assassinate({Pahul, Viput Singh)

¬people(Pahul)   ¬man(x) v people(x)

¬man(Pahul)   man(Pahul)

☐ (CONTRADICTION)

Q2: Answer the following precisely and to the point: (5 marks)

(i)   Explain Branch-and-Bound search. What is bound in the search? How does it help?
(ii)   Explain Best-first search method. How is it useful?
(iii)   Explain A* search method. What is the difference between A* search and Best-first search? Which one is better and why?
(iv)   When Dijkstra's search algorithm is available for discovering the shortest path in a graph, why do we need any other search method? Justify.

Answers for Q2

(i)

    a. Branch-and-bound search is a refinement over depth-first and breadth-first search where the nodes in the OPEN list for expansion are sorted in terms of the cost of reaching that node from the start node. The cheapest one is picked up for expansion and the search algorithm ignores all nodes that are more expensive than the current cheapest node.

    b. The ==node expansion is bound by the current cheapest cost of reaching a node. This helps by reducing the nodes to consider for expansio==n.

(ii)

    a. Best-first search is a method that evaluates the estimated cost of reaching the goal node from the current child nodes and selects that child node that has the least cost from the heuristic estimates. Thus Best-first search focuses on expanding that node that is estimated closest to the goal. It evaluates nodes by using just the heuristic cost function; that is, $f(n) = h(n)$.

    b. Best-first search is useful when we have many child nodes and need to find a quick path to the goal node without evaluating the cost of reaching a node.

(iii)

    a. A* search considers the cost of reaching a child node, represented as $g(n)$, and the estimated cost of reaching the goal node from that child node, represented as $h(n)$ in order to compute the overall cost $f(n)$ as $f(n) = g(n) + h(n)$ to a goal node via that node. A* then selects that child node which has the lowest cost $f(n)$.

    b. The difference between A* search and Best-first search is that A* considers the ==cost of reaching a child node in addition to the estimated cost of reaching the goal node from that child node==, whereas Best-first does not consider the cost of reaching a particular node but only focuses on selecting that node that is estimated to reach the goal faster.

    c. A* search is better as it will return the optimal cost path from the start node to a goal node.

(iv)   Dijkstra's algorithm focuses on finding the path that has the cheapest cost of reaching the goal node, assuming that all the actual costs are known upfront. It does ==not consider the estimated cost of reaching a goal that can be computed from heuristics==. In real-world, since the environment is dynamic, the estimated cost can be computed from heuristics and this helps select the most appropriate path for a goal.

Q3: Answer the following: (4 marks)

    (i)   Describe Modus Tollens and explain the rationale of its derivation.

    (ii)   What is Resolution? Show in logic form.

    (iii)   What is Hypothetical Syllogism? Explain its derivation using resolution.

    (iv)   What knowledge does the following logic statement express? Is it a valid statement?

$$\exists\, x\ (StudiesAt(x,\ IIITD)) \rightarrow Smart(x))$$

Answers for Q3

    (i)   Modus Tollens states that given a proposition $P \rightarrow Q$ and given the fact that $\neg Q$ is TRUE then we can derive that $\neg P$ is TRUE. This is because if P was TRUE then Q would have been derived, but since $\neg Q$ is given then $\neg P$ has to be TRUE else the proposition $P \rightarrow Q$ would be violated.

    (ii)   In Resolution when we have a clause of the form $(\neg P \vee Q)$ and we have a fact P then we can derive Q from that through the resolution of P with $\neg P$. In logical form we have P and $(\neg P \vee Q)$, which can be written as $(P \wedge \neg P) \vee Q$ which is equivalent to $(\mathbb{F}) \vee Q$ which is equivalent to Q. Thus, we can derive Q.

    (iii)   Hypothetical Syllogism states that given two statements, S1: $P \rightarrow Q$ and S2: $Q \rightarrow R$ then we can derive $P \rightarrow R$. This is possible using resolution as follows: S1 can be rewritten as $(\neg P \vee Q)$, and S2 can be rewritten as $(\neg Q \vee R)$. Thus, we have $(\neg P \vee Q) \wedge (\neg Q \vee R)$ which can be rewritten as $\neg P \vee (Q \wedge \neg Q) \vee R$ where $(Q \wedge \neg Q)$ resolve to give $\neg P \vee (\mathbb{F}) \vee R$ which gives us $(\neg P \vee R)$ and that is equivalent to $P \rightarrow R$.

    (iv)   This statement means that "there exists someone who if he/she studies in IIITD then that x is smart". This is an invalid logical statement.

Q4: Describe Hill-climbing method. What is the problem with Hill-climbing? Why is Hill climbing method required when we have Best-first search? (4 marks)

Answer for Q4

    a.   The Hill climbing method evaluates all the child nodes of the current node and selects that child node that give the highest gradient or 'gain' (or steepest-ascent) towards the goal. Hill climbing terminates when it reaches a "peak" where no unexplored child node gives a higher value or gain as compared to the current state.

b. The problem with hill climbing is that since it does not look ahead beyond the current set of child nodes it can get stuck at the local maxima, especially when the immediate child nodes do not give a 'gain' or when the child nodes are giving a loss.

c. Hill climbing search is required when we do not have all the heuristic information of all the nodes except for the immediate child nodes. This is useful in various practical scenarios when it is not possible to evaluate the heuristic cost for achieving the goal state from the current set of states except the local gains/ losses.

Q5: You are given the road distance between a few cities below. Describe A* search of finding a route from Calicut to Amritsar in the data given below. Write out the initial state, actions, state-space, path, goal-test for the search. Create your own reasonable heuristic.   (7 marks)

Distance in Kilometres	Ahmedabad	Bangalore	Bhubaneshwar	Bombay	Calcutta	Chandigarh
Ahmedabad	-	1490	1697	552	2068	1157
Amritsar	1356	2496	2224	1849	1919	239
Bangalore	1490	-	1538	1013	1961	2296
Bombay	552	1013	1678	-	2012	1645
Calcutta	2068	1461	423	2012	-	1721
Calicut	1648	520	1923	1171	2346	2741

Answer

Assuming heuristic h(n) as (given distance – 200) between city to Amritsar. For cities not directly specified we take an intermediate city and compute the distance from city to Amritsar. Thus, we have h(n) as follows:

City (from Calicut)	g(n)	h(n) (to Amritsar)	f(n)	f*(n)
Ahmedabad	1648	1156	2804	3004
Amritsar	2980 (via Chandigarh)	0	2938	2938
Bangalore	520	2296	2816	3016
Bhubaneshwar	1923	2024	3947	4147
Bombay	1171	1649	2820	3020
Calcutta	2346	1719	4065	4265
Calicut	0	2780	2780	2980
Chandigarh	2741	39	2780	2980

We can see that h(n) is admissible and consistent.

Cycle 1:

Initial State: OPEN: {Calicut}; CLOSED: {}; Goal-test: {Amritsar} fails

Actions: Select and remove node from OPEN – Calicut. Expand Calicut with g(n), h(n) and f(n) for each calculated. Now OPEN is { Ahmedabad (1648,1156,2804) , Bangalore (520,2296,2816), Bhubhaneshwar (1923,2024,3947), Bombay (1171,1649,2820), Calcutta (2346,1719,4065), Chandigarh(2741,39,2780) }. Add Calicut to CLOSED.

State space: {Ahmedabad, Bangalore, Bhubhaneshwar, Bombay, Calcutta, Chandigarh}

Goal test: CLOSED {Calicut} not the Goal state

Cycle 2:

Initial State: OPEN is { Ahmedabad (1648,1156,2804) , Bangalore (520,2296,2816), Bhubhaneshwar (1923,2024,3947), Bombay (1171,1649,2820), Calcutta (2346,1719,4065), Chandigarh(2741,39,2780) }; CLOSED: {Calicut}; Goal-test: {Amritsar} fails

Actions: Select lowest f value from OPEN and remove node from OPEN – Chandigarh. Expand Chandigarh with g(n), h(n) and f(n) for each calculated. So we get { Ahmedabad(3893,1156,5054), Amritsar(2980,0,2980), Bangalore(5037,2296,7333), Bombay(4386,1649,6035), Calcutta(4462,1719,6181) }.

We note that the parent of Amritsar is Chandigarh while for the rest the parent node is Calicut.

Since Calicut is already in CLOSED list, it is not considered.   Now OPEN already has { Ahmedabad (1648,1156,2804) , Bangalore (520,2296,2816), Bhubhaneshwar (1923,2024,3947), Bombay (1171,1649,2820), Calcutta (2346,1719,4065) } and each of these nodes already have a lower g(n) cost that the ones expanded from Chandigarh these nodes are not updated as the route via that city through Chandigarh is more expensive to what already exists. Thus, only Amritsar(2980,0,2980) is added to OPEN list. Thus the OPEN list now is { Ahmedabad (1648,1156,2804) , Bangalore (520,2296,2816), Bhubhaneshwar (1923,2024,3947), Bombay (1171,1649,2820), Calcutta (2346,1719,4065), Amritsar(2980,0,2980) } and the parents of all these nodes except Amritsar is Calicut.

Add Chandigarh to CLOSED list.

State space: {Ahmedabad, Bangalore, Bhubhaneshwar, Bombay, Calcutta, Amritsar}.

Goal test: CLOSED {Calicut, Chandigarh} not the Goal state

Cycle 3:

Initial State: OPEN { Ahmedabad (1648,1156,2804) , Bangalore (520,2296,2816), Bhubhaneshwar (1923,2024,3947), Bombay (1171,1649,2820), Calcutta (2346,1719,4065), Amritsar(2980,0,2980) }. CLOSED: {Calicut, Chandigarh}; Goal-test: {Amritsar} fails

Actions: Select lowest f value from OPEN and remove node from OPEN – Ahmedabad. This is not the goal state. Expand Ahmedabad with g(n), h(n) and f(n) for each calculated. So we get { Amritsar(3004,0,3004), Bangalore(3138,2296,5434), Bhubhaneshwar(3345,2024,5369), Bombay(2200,1649,3849), Calcutta(4414,1719,6133) }. Now OPEN has { Bangalore (520,2296,2816), Bhubhaneshwar (1923,2024,3947), Bombay (1171,1649,2820), Calcutta (2346,1719,4065), Amritsar(2980,0,2980) }. CLOSED has {Calicut, Chandigarh, Ahmedabad}

State space: {Amritsar, Bangalore, Bhubhaneshwar, Bombay, Calcutta}. Goal-test fails.

Cycle 4:

Initial State: OPEN has { Bangalore (520,2296,2816), Bhubhaneshwar (1923,2024,3947), Bombay (1171,1649,2820), Calcutta (2346,1719,4065), Amritsar(2980,0,2980) }. Choose cheapest, Bangalore, remove from OPEN and add to closed. CLOSED {Calicut, Chandigarh, Ahmedabad, Bangalore}. Goal-test: fails

Actions: Choose lowest f(n) node, so we choose Bangalore and expand. We get {Bhubhaneshwar(2058,2024,4082), Bombay(1533,1649,3182), Calcutta(1981,1719,3700), Amritsar(3016,0,3016)}. Now we see that in OPEN list none of the nodes will be updated in the cost or their parents changed.

Cycle 5:

OPEN has { Bhubhaneshwar (1923,2024,3947), Bombay (1171,1649,2820), Calcutta (2346,1719,4065), Amritsar(2980,0,2980) }. Choose cheapest – Bombay and remove from OPEN and add to CLOSED and expand that node as it is not the goal node.

Actions: Expand Bombay we get {Bhubhaneshwar(2849,2024,4873), Calcutta(3183,1719,4902), Amritsar(3020,0,3020)}. Now we see that none of the nodes in OPEN has a greater f(n) value than these and thus not updated.

Cycle 6:

OPEN has { Bhubhaneshwar (1923,2024,3947), Calcutta (2346,1719,4065), Amritsar(2980,0,2980) }.

We pick the cheapest node that is Amritsar. This is the goal-node and its parent is Chandigarh. And Chandigarh node parent is Calicut. So the goal-test succeeds and we return the path {Calicut, Chandigarh, Amritsar}. The search terminates successfully.

We can see that h(n) is admissible and consistent from all the values as for any node h(s) ≤ h*(s). It is consistent as h(n) ≤ h(n) ≤ c(n, n') + h(n').

Search cost: Examined 6 nodes. In general, it can be $O(b^d)$ where b is the branching factor and d the depth.

Q6: There are two parts to this question.

(a) Assume the following search graph is given. The path cost and heuristics are given with the nodes. S is the start node and G is the goal node. What are the paths returned for DFS, BFS, Best-first and A* search.   (2 marks)
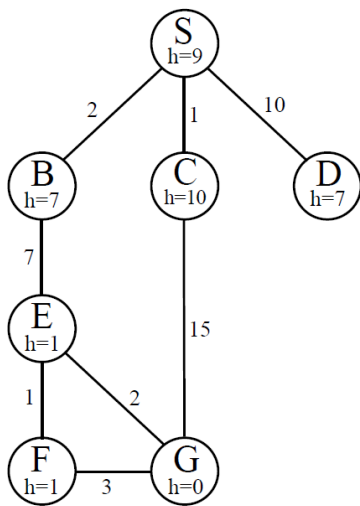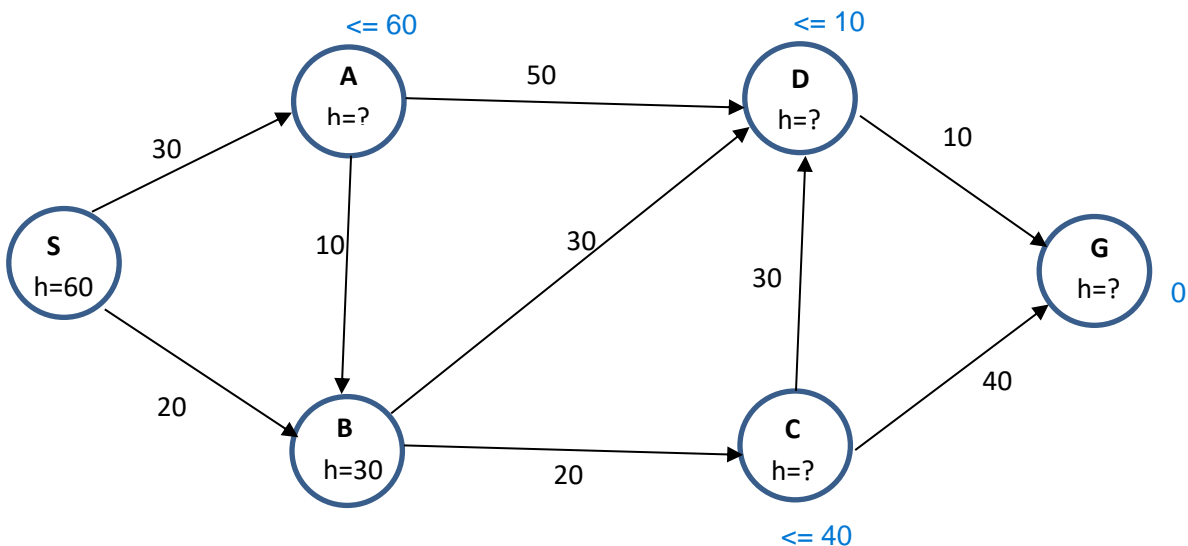

DFS: S --> B --> E --> F --> G
BFS: S --> C --> G

Best First:
1. B-7, C-10, D-7
2. E-1, C-10, D-7
3. G-0, C-10, D-7, F-1
Ans = S --> B --> E --> G

A*:
1. B-9, C-11, D-17
2. Pick B, C-11, D-17, E-10
3. Pick E, C-11, D-17, F-11. G-11,
4. Pick C, D-17, F-11, G-11
5. Pick F, D-17, G-11
6. Pick G

b) Consider the following graph.   (3 marks)

   i)   Create heuristics such that in A* search the heuristic is admissible.
   ii) Can you show that your heuristics is consistent?
   iii) If incidentally your heuristics is not consistent then what should be done to make it consistent?
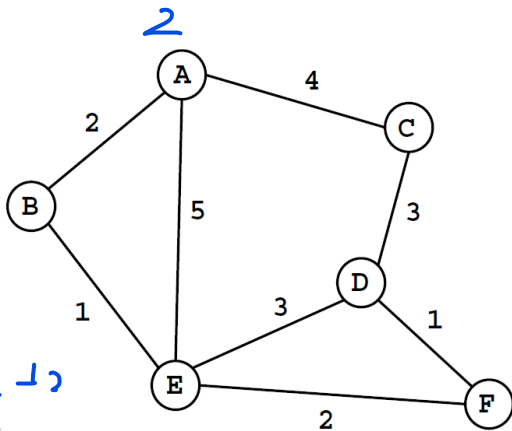
Answer for Q6

a) Paths returned are
   i.   Depth-first search: S-B-E-F-G
   ii.   Breadth-first search: S-C-G
   iii. Best-first search: S-B-E-G
   iv.   A* search: S-B-E-G

b) Heuristics are:
   i.   h(G) = 0, h(D) = 10, h(C) = 40, h(A) = 40. Since each heuristic is less than equal to the true cost to reach the goal, it never overestimates the cost to the goal and thus this heuristic is admissible.
   ii.   This heuristic is not consistent since the h(S) = 60 and it is greater than 20 (the cost to the next node B) plus 30 (the heuristic at node B).
   iii. In order to make this heuristic consistent we need to change h(S) to 50. Otherwise this heuristic cannot be made consistent.

Q7: For the given graph below in which A is the start node and F is the goal node, the cost of traversing to a node is given. Let the heuristic function be the minimum number of arcs between node n and the goal node. Is this an admissible heuristic? Justify. Also show in what order does A* visit the nodes and what are the estimated values when they are visited?   (3 marks)

$h(A) = 2$; $h(B) = 2$; $h(C) = 2$; $h(D) = 1$; $h(E) = 1$; $h(F) = 0$

0. Open = {f(A) = 2}
1. Remove A, f(B) = 4; f(E) = 6; f(C) = 6
2. Remove B, New f(E) = 4 < Old f(E) --> Update
3. Remove E, f(D) = 7. f(F) = 5. {C = 6, D = 7, F = 5}
4. Remove F, if F is goal: hogya
                    .
For admissible,
For all nodes n, h(n) <= actual cost h*(n) .

Answer for Q7

The heuristic values are h(A) = 2, h(B) = 2, h(C) = 2, h(D) = 1, h(E) = 1, h(F) = 0.

The heuristic is admissible since the heuristic value at each node is less than or equal to the optimal value to the goal node.

A* visits the nodes and gives the path as A-B-E-F and the estimated values are at f(A) = 0+2 = 2, f(B) = 2+2 = 4, f(E from A) = 5+1 = 6, f(C) = 4+2 =6. First node B is expanded and then f(E from B) = 3+1 =4 is expanded since it is the cheapest.

Q8: In the AND-OR graph given below find the value of the root node for the optimal solution for the graph. Note that the values given inside all single circle nodes is the estimate of the cost or the heuristic value, while the value given inside all double circle nodes is the actual value of the SOLVED node. The cost of each edge is 10 units. Show all your steps for computation. (5 marks)

A 152 / 160 = 152
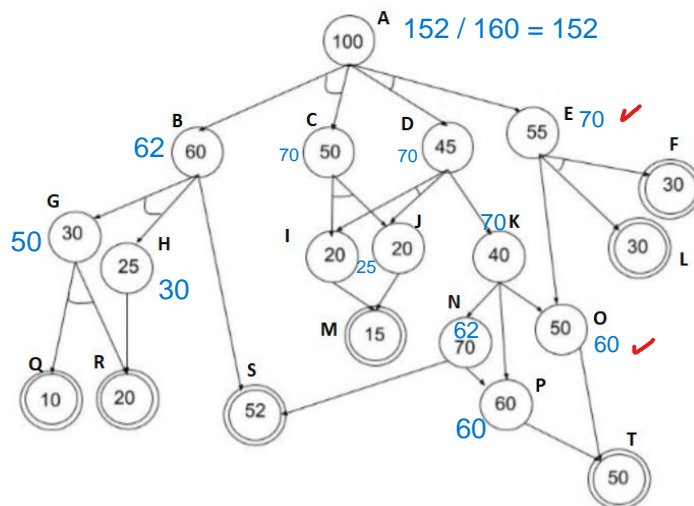
Answer for Q8

Starting from Solved nodes we compute the value of intermediate nodes for optimal solution as follows:

i) O=60, E (from F,L since it is AND node) = 30+30+10+10 = 80, E (from O since it is OR node) = 60+10 = 70, thus min cost of E = 70.

ii) N = min (path from P, path from S) = min(70, 62) = 70, K = min(path from O, P, N) = min(70, 70, 70) = 70. D = min(K, (J,K)) = min(70, (35+35)) = 70, C = 70.

iii) H = 30, G = 30+20 = 50, B= min(path from S, (G,H)) = min(62, (80))=62, A = min(path from (B,C), path from (D,E)) = min((152), (160)) = 152.

iv) Thus A = 152 and gives the optimal solution as A-B-C-S-I-J-M

Q9. This question has two parts. (4 marks)

i) Given the propositional rule A → B and given proposition B can we conclude A? Is this conclusion sound? If yes, explain why with an example. And if no, explain why with an example.

ii) Given the following propositions: A, B, C, D, ¬E, F, ¬G determine the truth value of the following sentences:

$$E1: B \land D \land (C \land E \rightarrow F)$$
$$E2: C \land E \land (B \land D \rightarrow G)$$

Answer for Q9

i)   No, we cannot conclude A. If we conclude A it won't be sound. Since A → B reduces to ¬A v B and we are given B then if we try to resolve and prove A, we are unable to prove A. Thus, we cannot say about A. We note that A may be TRUE or FALSE, so A cannot be concluded. For example, given the sentence S: If it is rainy then you cannot go for a picnic. And given "you cannot go for a picnic" then you cannot derive that "it is rainy". The sentence S is however TRUE since "you cannot go for a picnic" is TRUE.

ii)

        a.   E1 evaluates to TRUE as C ∧ E → F is equivalent to (¬C v ¬E v F) and since ¬E and F is given then this evaluates to TRUE and since B and D are given then E1 evaluates to TRUE.

        b.   E2 evaluates to FALSE since (B ∧ D → G) is equivalent to (¬B v ¬D v G). Here G is FALSE as ¬G is given and since B and D are TRUE this evaluates to FALSE. Further since ¬E is given then E2 will evaluate to FALSE. Thus, E2 evaluates to FALSE.

Q10: Assume that the spacecraft Chandrayaan-5 is going to be launched to the Moon and its rover Pragyan-5 is being designed to drive around the Moon's surface, collect rock samples and then come back to the Vikram-5 lander. It has the following aspects:

The rover's batteries can be charged by the solar cells and it needs to do that. It can move around. It can pick up a rock. The rover has a map that indicates the type of rock expected in a location and the expected weight of rocks at that location.

Formulate the PEAS for the Pragyan-5 rover. What kind of Agent will you design for this rover? Specify the Agent details. Specify all the characteristics of the environment that the rover will encounter.

(8 marks)

Answer for Q10

Agent: Pragyan-5 rover

Kind of Agent: Goal-based agent

Performance Measure:

    i)   Types of rocks collected
    ii)   Number of rocks collected
    iii)   Locations visited
    iv)   Able to charge its batteries before it runs out of charge
    v)   Able to plan its moving based on amount of charge left

Environment:

    i)   Rocky surface mixed with mounds
    ii)   Solar light power

iii)   Different types of rocks

iv)   Estimated weight of rocks

v)   Distinguish between rocks and mounds

Actuators

i)   Robotic arm and robotic hands to grab/lift rocks

ii)   Location identification through cameras

iii)   Charging through solar cells

iv)   Wheels to move around

Sensors

i)   Cameras at various points of the rover

ii)   Infrared sensors for distance measurement

iii)   Battery charge detection sensor

iv)   Sensor for Location with respect to lander

v)   Map correlation sensor

vi)   Sensors to identify between rocks and mounds

Environment type

i)   Fully observable

ii)   Deterministic

iii)   Episodic

iv)   Static (nothing changes on the moon)

v)   Discrete (actions like pick up, move, charge)

vi)   Single Agent