CS217 **Artificial Intelligence and Machine Learning** **Quiz1**

30Jan25 **10.45 AM to 11.25 AM** **10 marks**

(**VIMP**: do rough work and then post the neat and clean answer to the two sides of the A4 size paper. Illegible, round-about, unnecessarily complicated answers will not be evaluated.)

Q1. In a problem, the search graph is a complete binary tree of depth d. Level 0 is the start node S. At depth d=1, there are 2 nodes, at d=2 4 nodes and so on. At depth d, there is the unique goal node G. Each arc in the search graph has cost 1. The search algorithm is a strange one. The open and closed lists are maintained. The search algorithm at any stage picks up a node for expansion randomly with uniform probability from the open list at that stage. Find the (a) probability of success and (b) probability of success without expanding any useless node.

2+3=5

Ans:

(a) The probability of success is 1. Remember the *definition* of success. The goal node getting transferred to CL is the condition of success. The OL has to contain G at some point and G has to get selected, since the OL gets depleted by nodes getting transferred to CL.

(b) The recurrence relation governing the number of nodes in the OL is

> N(i+1)=N(i)+1, since 1 node goes into the CL and 2 come into the OL, where i is the depth of the tree and is also the iteration number of the search algorithm, Clearly N(0)=1, N(1)=2, N(2)=3, N(3)=4,…

So the required probability=1/(d+1)!

Q2. An A* algorithm runs with a heuristic that satisfies monotone restriction (MR). At a stage of search, the open list contains two nodes N and M. N is expanded and is moved to the closed list. After some time, M is expanded. M re-generates N. Prove FROM FIRST PRINCIPLES that the parent pointer of N need not be changed to M. You are NOT allowed to use the monotone restriction theorem that under MR any expanded node has already found the optimal path from the start node S.

5 marks

Ans:

In general an involved proof. You have to use the proof given in the class for the special case that BOTH N and M are present in the OL.

But you get full marks if you consider the very special case of M being expanded immediately after N. The idea is to see your approach.

Since N was expanded before M

F(N) <= F(M)

i.e., G(N)+H(N) <= G(M)+H(M) (1)

Also by MR,

H(M) <= H(N)+C(M,N) (2)

(1)(2) gives

G(N)<=G(M)+C(M,N)

Hence no necessity to redirect the parent pointer of N.

==========Paper ends==========