COMP 476 Theory Assignment 2

For Daniel Rinaldi

George Mavroeidis

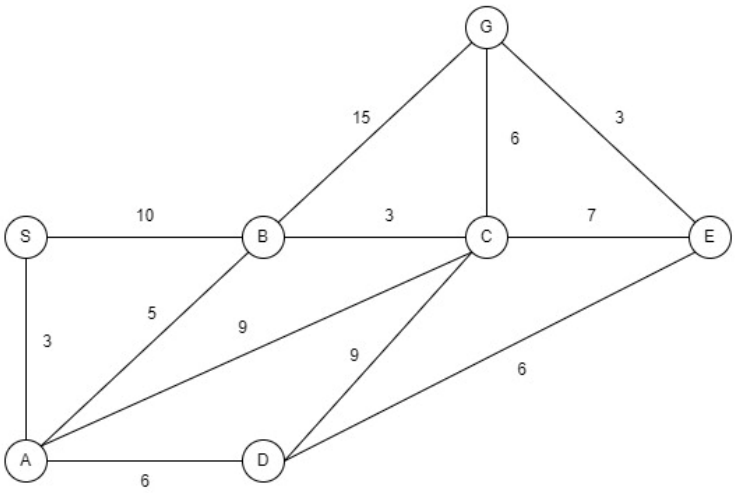
Question #1:

We can represent a weighted directed graph like as a set of nodes and edges (with assigned weights indicated):

- **nodes = {S, A, B, C, D, E, G}**
- **edges = { SA: 3, SB: 10, AB: 5, AD: 6, AC: 9, BC: 3, BG: 15, DE: 6, CD: 9, CE: 7, CG: 6, EG: 3}.**

In the graph, assume that S is the start node and G is the goal node.

a) **If we use Dijkstra's algorithm to find the minimum cost path from S to G, then the following table shows the contents of the open and closed lists for the first 2 steps of the algorithm. Fill in the remaining lines. Each entry in the lists is of the following format: (Node, Cost-So-Far, Connection). Stop when the guaranteed shortest path has been found.**



Current Node	Open List	Closing List
-	(S, 0, -)	-
S	(A,3,SA), (B,10,SB)	(S,0,-)
A	(B,8,SAB), (C,12,SAC), (D,9,SAD)	(S,0,-), (A,3,SA)
B	(G,23,SABG), (C,11,SABC), (D,9,SAD)	(S,0,-), (A,3,SA), (B,8,SAB)
D	(E,15,SADE), (C,11, SABC), (G,23,SABG)	(S,0,-), (A,3,SA), (B,8,SAB), (D,9,SAD)
C	(E,15,SADE), (G,17,SABCG)	(S,0,-), (A,3,SA), (B,8,SAB), (D,9,SAD), (C,11,SABC)
E	(G,17,SABCG)	(S,0,-), (A,3,SA), (B,8,SAB), (D,9,SAD), (C,11,SABC), (E,15,SADE)
G	-	(S,0,-), (A,3,SA), (B,8,SAB), (D,9,SAD), (C,11,SABC), (E,15,SADE), (G,17,SABCG)

The guaranteed shorted path is: S → A → B → C → G (read the path from the goal node visited)

b) When can we stop if we are not interested in guaranteed shortest path?

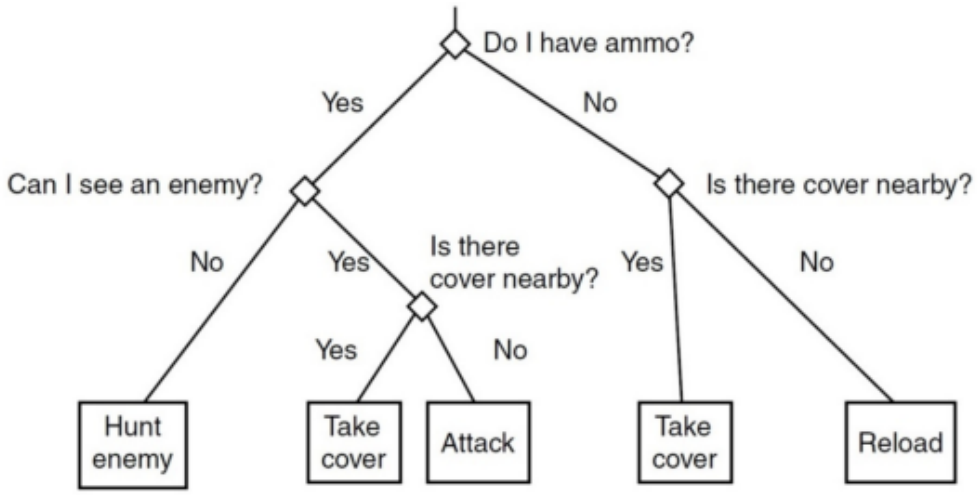
We can stop when G first appears on the open list, specifically when B is the current node. G had a cost of 23, going along the path S → A → B → G

c) How many paths to the goal node are evaluated if guaranteed shortest path is to be found?

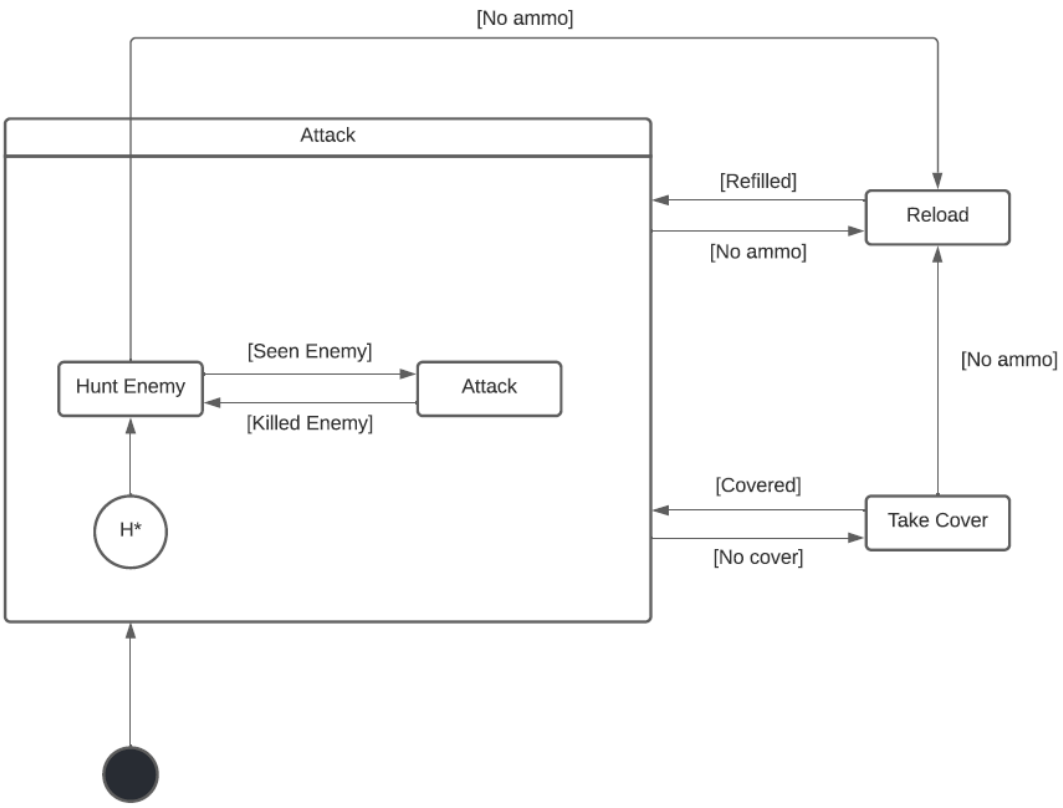
2 paths were found that reach the goal node when looking for guaranteed shortest path: path S → A → B → G that had a cost of 23 and path S → A → B → C → G that had a cost of 17. The second one is actually the guaranteed shortest path.

Question #2:

Consider the decision tree from Figure 6.6. of Artificial Intelligence for Games 2nd Edition, by Millington and Funge, reproduced below.



a) Design a hierarchical finite state machine that would produce behavior similar to that of this decision tree.



b) Add an alarm behavior/mechanism (see Slide 30 of decision-making lecture notes). To your hierarchical finite state machine from a), add alarm behavior such that the NPC, regardless of what is going on, will go to sleep (somewhere relatively safe) if too tired. Additionally, add the option of taking a quick nap if no enemies have been visible for a while and the NPC has ammo.