CS440 : Introduction to Artificial Intelligence **Spring 2014**

Problem Set 1

Dan McQuillan *Handed In: February 3, 2014*

1. **Solution to problem 1**

(a) (i) depth-first

n	**q**
-	(A)
A	(B, C, D)
B	(E, F, C, D)
E	(K, F, C, D)
K	(F, C, D)

(ii) breadth-first

n	**q**
-	(A)
A	(B, C, D)
B	(C, D, E, F)
C	(D, E, F, G, H)
D	(E, F, G, H, I, J)
E	(F, G, H, I, J, K)
F	(G, H, I, J, K, L)
G	(H, I, J, K, L)

(iii) uniform-cost

n	**q**
-	(A)
A	(B, C, D)
B	(C, E, F, D)
C	(E, F, H, D, G)
E	(F, H, D, G, K)
F	(H, L, D, G, K)
H	(L, D, N, G, M, K)
L	(D, N, G, M, K)
D	(N, G, M, I, K, J)
N	(G, M, I, K, J)

(iv) greedy/best-first

n	**q**
-	(A)
A	(C, D, B)
C	(G, H, D, B)
G	(H, D, B)

(v) A^*

n	**q**
-	(A)
A	(C, B, D)
C	(B, H, D, G)
B	(H, D, G, E, F)
H	(D, G, N, E, F, M)
D	(G, N, E, F, I, J, M)
G	(N, E, F, I, J, M)

(b) (i) Depth first is not admissible since it does not take weights or distance into consideration and could easily not choose the cheapest goal.

(ii) Breadth first search is not admissible for the same reason as depth first search since it does not consider weights or distance when finding a goal state.

(iii) Uniform cost search is admissible since it is using the actual cost to the node and recalculates and sorts the order of the queue based upon the total cost from the root node.

(iv) Greedy / Best-First search is not admissible since it does not take the cost of getting from $n \rightarrow n'$ into consideration and it only depends on how close the node is estimated to be from the goal.

(v) A^* search is admissible because when it terminates its search, it has found a path whose cost is lower than the estimated cost of any path through any open node. Although, since those estimates are optimistic, A^* can safely ignore those nodes. Therefore, A^* will never overlook the possibility of a lower cost and is admissible.

(c) We know that for all heuristic functions, it is consistent if for every node n and every successor n' of n generated by any action a, the estimated cost of reaching the goal from n is no greater than the step cost of getting to n' plus the estimated cost of reaching the goal from n'. Or more formally:

$$h(n) \leq c(n, a, n') + h(n')$$

Suppose:
B is an unreachable node and therefore the cost from any node, $n \in nodes$, the cost will be infinite.
Then:

$$h(n) \leq c(n, a, B) + h(B)$$

h(B) must therefore also be infinite since it is an unreachable and infinitely far away from any goal node

$\therefore h(n) \leq \infty + \infty$
$h(n) \leq \infty \rightarrow$ the heuristic is consistent

2. **Solution to problem 2**

(a) (i) A data structure to represent this structure could be a Tuple. Each state would be a tuple3, where it would contain three slots which would be filled with an enum value where the enumeration is as follows:

```
StateAttributeType {
    Empty,
    Box,
    Agent,
    Agent_With_Box
}
```

A goal state for this data structure would be defined as the following tuple3:

```
Tuple3 {
    StateAttributeType.Box,
    StateAttributeType.Box,
    StateAttributeType.Agent
}
```

(ii) Pickup and Drop Preconditions and Effects

Pickup Operator preconditions:

- Agent must not be holding a box
- Agent must be in a square with a box

effects:

- Agent now has a box.
- Box is removed from square

Drop Operator preconditions:

- Agent must be holding a box

effects:

- Box replaces empty state
- Agent state replaces agent with box state

(b) (i)

(ii)

(iii)