Introduction to Artificial Intelligence

Project 2: ∃a, s[MissionCompleted(Ethan, Result(a, s))]

Report.

1- Introduction.

In this section, a brief description of the code is shown below:

- goal(S) calls iterative deepening search if S is a free variable, and calls the axiom state if S is a compound "non-variable".
- state(Ex,Ey,Ec,Agents,result(A,S)), the axiom, holds true iff:
  a. Ethan could move to the corresponding valid cell.
  b. Ethan could stay in the current cell and carry if the cell has an un-carried agent.
  c. Ethan could drop if he is at the submarine carrying some agent(s).

2- Actions and Predicate Terms:

a. Actions:

Actions in the program are *(right,left,up,down,carry,drop)*. They are assigned accordingly in the state-handling Logic inside the Successor State Axiom.

b. Predicate Terms:

The following predicates are used in the program:

- **in_grid(X,Y):** Checks if the X and Y coordinates in the boundaries of the grid.
- **carried(X,Y,AgentsList,AgentsListUpdated):** assigns the carry status of the agent at position (X,Y) in AgentsList with carry status of 0, the value of 1, and returns the list AgentsListUpdated with the updated carry status of this agent.
- **dropped(X,Y,AgentsList,AgentsListUpdated):** assigns the carry status of the the agent at position (X,Y) with carry status of 1, the value of 2, and returns the list AgentsListUpdated with the updated carry status of this agent.
- **create_agents_array_with_carry(AgentsListKB,AgentsListWithCarry):** initializes a copy AgentsListWithCarry of the agents list in the Knowledge Base AgentsListKB, but with an additional parameter with each agent, C = 0, where C is the carry status of the agent (C=0 "Not Carried", C=1 "Carried", C=2 "Dropped").
- **create_agents_array_with_drop(AgentsListKB,AgentsListWithDrop):** initializes a copy AgentsListWithDrop of the agents list in the Knowledge Base AgentsListKB, but with an additional parameter with each agent, C = 2, where C is the carry status of the agent (C=0 "Not Carried", C=1 "Carried", C=2 "Dropped").
- **reverse2(L,R):** reverses the compound output L of the algorithm into R, to match the required format.

- **iterative_deepening(Goal,Limit):** increment the limit of the search if the R of the call_with_depth_limit(state,Limit,R) is depth_limit_exceeded, and returns the result of the state otherwise. Hence, performing iterative deepening search on the query starting with limit 1 till the query is found, and it will be found as IDS is complete.

3- Successor State Axiom:

a. Variables:

The program has only **One** successor state axiom, called *state*, and it keeps track of the following:

- Ethan's position: the X and Y coordinates of Ethan in the search problem Grid (Ex,Ey).
- Ethan's current carry count: The number of agents Ethan is currently carrying (Ec).
- Agents: the List of the agents positions, with their carry status.

b. Logic:

The successor state axiom state(Ex,Ey,Ec,Agents,result(A,S)) holds true for ethan to be in (Ex,Ey), with a carry count Ec, if and only if:

- He can go right to (Ex,Ey+1), if the next situation position is not out of the grid bounds.
- He can go left to (Ex,Ey-1), if the next situation position is not out of the grid bounds.
- He can go up to (Ex-1,Ey), if the next situation position is not out of the grid bounds.
- He can go down to (Ex+1,Ey), if the next situation position is not out of the grid bounds.
- He can stay in (Ex,Ey), and if there's an agent with a carry status 0 "Not Carried" in the same position, and he has room for more agents to carry "Ec < Capacity", he can Carry.
- He can stay in (Ex,Ey), and if (Ex,Ey) is the position of the submarine, and has agents carried "Ec > 0 and their carry status is 1 "Carried" ", He can drop.

4- The Goal Predicate:

The program runs using the goal predicate *goal(S)*, that takes a situation S, and returns true if it's a valid goal situation in the search problem.

goal(S) calls the initial state from the KB. The successor state axiom "state" is then started to query, and has a base case that holds true when ethan is at the submarine, with all agents of carry status: 2 "Dropped".

5- Running Examples:

KB:

```
ethan_loc(0,0).

members_loc([[1,1],[1,2]]).

submarine(0,2).

capacity(1).
```

goal(S).

S =
result(drop,result(up,result(carry,result(down,result(drop,result(up,result(right,result(carry,result(down,result(right,s0))))))))) .

goal(result(drop,result(up,result(carry,result(down,result(drop,result(up,result(right,result(carry,result(down,result(right,s0))))))))))).

true.

goal(result(up,result(carry,result(down,result(drop,result(up,result(right,result(carry,result(down,result(right,s0)))))))))).

false.