



# Coast Guard

24.12.2022

—

## Names :

Team Name: Natural Stupidity

Team Num: 37

Mahmoud Mohamed Aly

46-18406

T-18

Abdelrahman Khater Elsaied Selim

46-18829

T-13

Samer Walaa Gabra

46-18648

T - 12

## Overview

The Coast Guard problem is a problem where we have a grid (3 x 3) or (4 x 4) represents a sea area and there are 1 or 2 ships each has 1 passenger, our agent (Rescue Boat) task is to try to investigate the grid, reach the ships positions and save people by taking them to Stations. Our agent is a blind one, it doesn't know in the beginning where it should go, but it has a KB(knowledge base) which informs it of the grid objects positions.

## Fluents

- `fluent(location(X,Y)) :- agent_loc(X,Y) .`

This defines the location of the coast guard along the grid, and it has two attributes:

X: the vertical location of the boat on the grid.

Y: the horizontal location of the boat on the grid.

- `fluent(ships(X)) :- ships_loc(X) .`

This defines the location of the ships which haven't been collected by the boat yet, and it has one attribute:

X: list of the ships' locations along the grid.

- `fluent(holding(nothing)) .`

This defines the objects that the coast guard is carrying right now (if it's carrying something), and it has one attribute:

X: list of the locations of the ships that the boat is carrying its passengers right now(or nothing if it's not carrying anything).

## Successor State Axioms

```
execute_process([], [], S1, s0) :-
    s0(S1). % Nothing to do

execute_process(S1, [Action|Process], S2, S) :-
    execute_process(S1, Process, Sd, NxtS),
    poss(Action, Sd),
    result(Sd, Action, S2, Type),
```

```
S=result (Type,NxtS) .
```

We only have one successor state axiom which is (execute\_process), and it works as follows:

We firstly go deep in the DFS to start from the leaf of the tree.

Then, using the successor state axiom logic, we check the possibility of the action to be applied on the agent using (poss) which takes the inputs and the action and validates if it's possible to do such action on the agent like (traverse, pickup or drop).

If (poss) returns that it's a valid action we go to the next step which is getting the result of applying this action on the agent (remove the old states from the list and replace it with the new ones).

Finally we return the result set of fluents and the type of the last action applied on it.

## Test Cases

**We run the code on a machine with the following specifications:**

**Processor:** Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz 2.40 GHz

**RAM:** 8.00 GB

**SSD:** 256 GB . Reads(3.1GB/s) Write(1.2 GB/s)

**GPU:** Nvidia GTX 1650 (4GB)

**Test Case:**

```
grid(3,3).
agent_loc(0,1).
ships_loc([[2,2],[1,2]]).
station(1,1).
capacity(1).
```

**Result:**

- result(drop,result(left,result(pickup,result(right,result(drop,result(left,result(up,result(pickup,result(right,result(down,result(down,s0))))))))))
- result(drop,result(left,result(pickup,result(right,result(drop,result(up,result(left,result(pickup,result(right,result(down,result(down,s0))))))))))
- others

**Time:**

3.46 s

\*\*\*\*\*

**Test Case:**

```
grid(4,4).
agent_loc(0,2).
ships_loc([[1,2], [3,2]]).
station(1,1).
capacity(2).
```

**Result:**

- result(drop,result(left,result(pickup,result(right,result(drop,result(left,result(up,result(up,result(pickup,result(down,result(down,result(down,s0))))))))))))))
- result(drop,result(left,result(pickup,result(right,result(drop,result(left,result(up,result(up,result(pickup,result(down,result(down,result(down,s0))))))))))))))
- others

**Time:**

28.43 s

\*\*\*\*\*

**Test Case:**

```
grid(4,4).
agent_loc(0,2).
ships_loc([[1,2], [3,2]]).
station(1,1).
capacity(2).
goal(result(drop,result(left,result(pickup,result(right,result(drop,result(left,result(up,result(up,result(pickup,result(down,result(down,result(down,s0)))))))))))))).
```

**Result:**

true

**Time:**

26.65

\*\*\*\*\*

**Test Case:**

```

grid(4,4).
agent_loc(0,2).
ships_loc([[1,2], [3,2]]).
station(1,1).
capacity(2).
goal(result(left,result(left,result(pickup,result(right,result(drop,result(left,result(up,
result(up,result(pickup,result(down,result(down,result(down,s0)))))))))))))).

```

**Result:**

- It is a false plan.
- > 2 min. With no result

**Time:**

----

\*\*\*\*\*

**Test Case:**

```

grid(4,4).
agent_loc(0,2).
ships_loc([[1,2], [3,2]]).
station(1,1).
capacity(1).

```

**Result:**

- result(drop,result(left,result(pickup,result(right,result(drop,result(left,result(up,result(up,result(pickup,result(down,result(down,result(down,s0))))))))))))))
- result(drop,result(left,result(pickup,result(right,result(drop,result(up,result(left,result(up,result(pickup,result(down,result(down,result(down,s0))))))))))))))
- Others

**Time:**

20.91 s