AI

Project 2 Report

Initially, we includes “KB.pl” to be able to use this knowledge base to initialize the search agent. This file contains four main predicates:

1. **goal(S):** Call ids predicate with S.
2. **Ids(X,L):** This predicate does depth limited search to backchain on the query provided as the first argument 1 of the predicate.
3. **goal2(result(A,R)):** This predicate starts by initializing the agent position with the station position to stop after dropping the passengers neither than returning to the position that the agent starts from and initialize the SHIPLIST with the positions of the ships then call goal2helper with these data.

4. **goal2helper(agent(X,Y),NOP,SHIPLIST,result(A,R)):** This predicate infers a new state using successor state axioms on the current situation. It returns a new state if the operator used on the inputted state is valid in accordance with the matrix rules. It keeps concatenating the actions the agent takes to a string result. Agent(X,Y) where X,Y represent the agent position, NOP represent the number of passengers that the agent can pickup , the *SHIPLIST* represents a list of the locations of the ships and result(A,R) represents the result where A is the action taken and R is the returned string.

- ## The positions of the new state need to exist within the matrix boundaries

  a. **UP** : To move up, we need to decrement the X value by 1 and check it's validation (by checkpositionV predicate) and set A with up.
  b. **DOWN** : To move down , we need to increment the X value by 1 and check it's validation (by checkpositionV predicate) and set A with down .
  c. **LEFT** : To move left, we need to decrement the Y value by 1 and check it's validation (by checkpositionV predicate) and set A with left.

d. **RIGHT** : To move right, we need to increment the Y value by 1 and check it's validation (by checkpositionV predicate) and set A with right.

If the agent took one of these actions, his x and y are updated for the new state and recall goal2helper with the updated attributes.

e. **PICKUP**: To pickup, first we check NOP that is greater than zero then call pickupfromship predicate if true the we will decrement NOP by 1 and set the A with pickup and recall goal2helper with the new NOP and SHIPLIST .

f. **Drop**: To drop, first we check if the position of the agent is the same as the station then set the A with drop and reset the NOP and recall goal2helper with the new NOP .

- This predicate is divided into two cases:

a. **Case 1:** Is the base case by initializes agent(X , Y) and set the agent capacity(NOP) by zero and empty SHIPLIST and initial state "firstaction".

b. **Case 2:** Recursively indicates a new state from the current situation using successor state axioms.

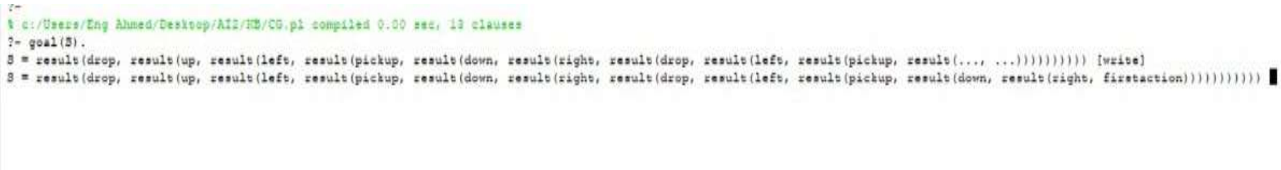
- **It contains some other helper functions which are stated below:**

1. **checkpositionV(X,Y,W,H):** This predicate checks if the agent position is a valid position in the given grid where W,H is the grid size "grid(W,H)", check if X and Y greater than or equal to zero and if X less than H and Y is less than W.
2. **pickupfromship(SList,X,Y,SList2):** This predicate checks if the agent position(X,Y) contain a ship by using predefined method called delete(SList,[X,Y],SList2) which check if [X,Y] is an element of list SList and if true if remove this element from the SList and return a new list in SList2.

Examples

Example 1

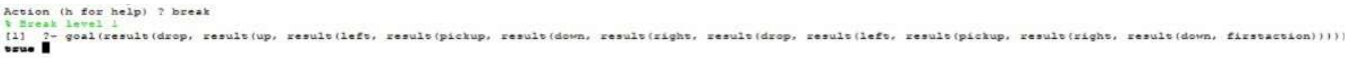
goal(S).



```
?-
% c:/Users/Eng Ahmed/Desktop/AI2/KB/CG.pl compiled 0.00 sec, 13 clauses
?- goal(S).
S = result(drop, result(up, result(left, result(pickup, result(down, result(right, result(drop, result(left, result(pickup, result(..., ...)))))))))) [write]
S = result(drop, result(up, result(left, result(pickup, result(down, result(right, result(drop, result(left, result(pickup, result(down, result(right, firstaction)))))))))))
```

Example 2

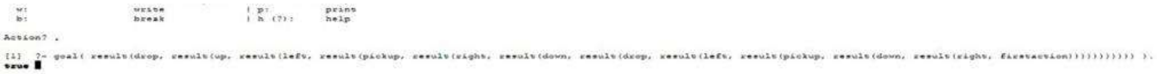
goal(result(drop, result(up, result(left, result(pickup, result(down, result(right, result(drop, result(left, result(pickup, result(right, result(down, firstaction)))))))))))).



```
Action (h for help) ? break
% Break level 1
[1]  ?- goal(result(drop, result(up, result(left, result(pickup, result(down, result(right, result(drop, result(left, result(pickup, result(right, result(down, firstaction))))
true █
```

Example 3

goal(result(drop, result(up, result(left, result(pickup, result(right, result(down, result(drop, result(left, result(pickup, result(down, result(right, firstaction)))))))))))).



```
 w:                write          | p:        print
 b:                break          | h (?):    help

Action? .

[1]  ?- goal( result(drop, result(up, result(left, result(pickup, result(right, result(down, result(drop, result(left, result(pickup, result(down, result(right, firstaction))))))))))) ).
true █
```