CS 312: Artificial Intelligence Laboratory

Task 9: Planning

Note: This assignment is to be done individually.

Problem Statement

For the block world problem simulate goal stack planning for given start state and goal state.

Description

Predicates	1. on b a 2. ontable a 3. hold a 4. clear a 5. AE
Actions	1. stack a b : Precond: Clear(A) ^ Clear(B) ^ AE() Push Order: ● AE() ● Clear(B) ● Clear(A) Effects: AE() ^ !Clear(B) ^ !Hold(A) ^ On(A, B) ^ Clear(A) 2. unstack a b: Precond: On(A, B) ^ Clear(A) ^ AE() Push Order: ● AE() ● Clear(A) ● On(A, B) Effects: Clear(B) ^ !Clear(A) ^ Hold(A) ^ !AE() ^ !On(A, B) 3. pick a: Precond: OnTable(A) ^ Clear(A) ^ AE() Push Order: ● AE() ● Clear(A) ● OnTable(A) Effects: Hold(A) ^ !AE() ^ !Clear(A) ^ !OnTable(A) 4. putdown a: Precond: Hold(A) Effects: Clear(A) ^ AE() ^ OnTable(A) ^ !Hold(A)
Start State (Example)	(on b a)^(ontable a)^(ontable c)^(ontable d)^(AE)

Goal State (Example)	(on c a)^(on b d)^(ontable a)^(ontable d)

- The program should take input the number of blocks, start state as given above and goal state as given from an input.txt file.
- The output of the program must be the plan in terms of the actions to be carried from start state to goal state and should be written into output.txt file.

PFA input.txt and output.txt file formats.

Submission:

1. Code files and Report
2. run.sh script
3. input.txt and output.txt for 3 examples given below (name it input1.txt , input2.txt so on)

Evaluation Criteria:
Correctness: 25 (will be tested against hidden test cases)
Report: 20
Code Quality: 5

Deadline: 11:55 PM 13 April 2020
Late Submission Policy: 5% of marks will be deducted per day late.

Report Format:

1. Pseudo code (5 marks)
2. Input and output for 3 given examples (9 marks)
3. Stack visualization for first example(6 marks)

Inputs and Outputs:

1	4 (on b a)^(ontable a)^(ontable c)^(ontable d)^(AE) (on c a)^(on b d)^(ontable a)^(ontable d)	(unstack b a) (putdown b) (stack c a) (stack b d)
2	4 (ontable a)^(ontable b)^(ontable c)^(ontable d) (on a b)^(on b c)^(on c d)	(stack a b) (unstack a b) (putdown a) (stack b c) (unstack b c) (putdown b) (stack c d) (stack a b) (unstack a b) (putdown a) (stack b c) (stack a b)

3	3 (ontable a)^(ontable b)^(ontable c) (on a b)^(on b c)	(stack a b) (unstack a b) (putdown a) (stack b c) (stack a b)

NOTE :

- Input and outputs are case sensitive , any mistakes in the format will be considered as completely incorrect.
- Newline and space characters also should be handled with care.
- Submit code after checking it against the given test files.
- Refer to the order of preconditions to be added to the stack in the description table above. Follow the same order to get unique results.
- For the predicates clear and hold follow the below procedure:
    - If Clear A is popped out of stack:
        - If hold A is true then “putdown” action to be pushed else Unstack.
    - If Hold A is popped out of stack:
        - If ontable A is true push action “pick” else unstack.
- At the start AE is true for the initial state construction (if hold is not present in the given start state).
- The program will be tested only for valid inputs.
- While pushing the goal state predicates,see that the first one is on the top of the stack.

Reference:

Goal Stack Planning :
https://www.youtube.com/watch?time_continue=1648&v=nvRmeHyklIA&feature=emb_logo