CS 312: Artificial Intelligence Laboratory
Lab 9 Report

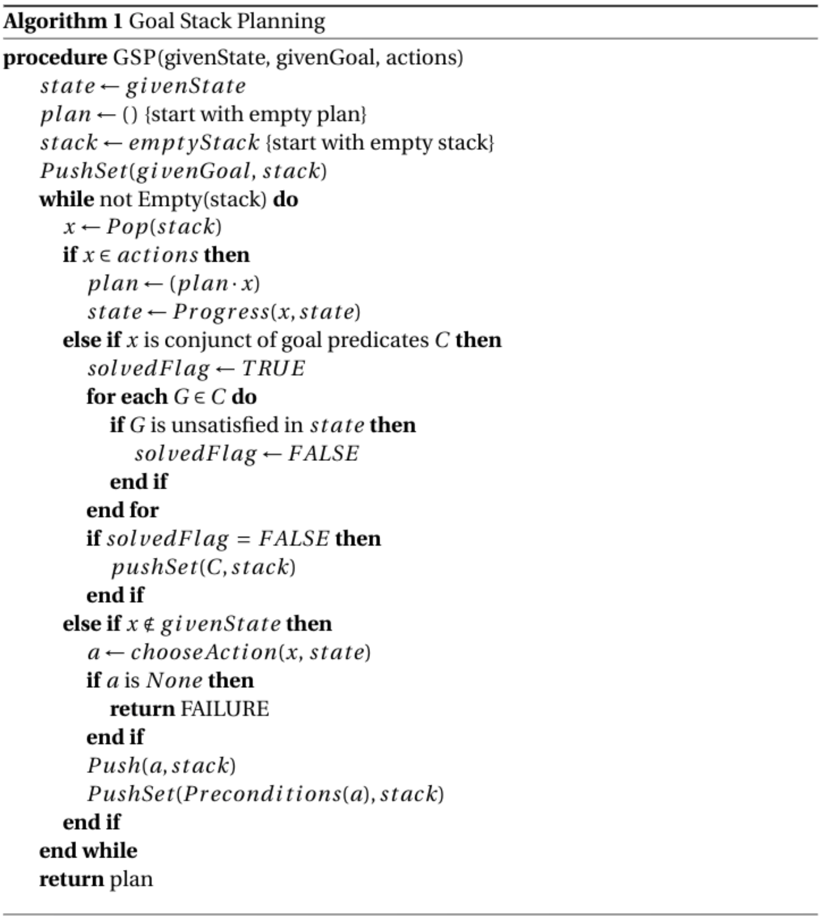
S V Praveen - 170010025

April 12, 2020

1 Introduction

The objective of this task is to simulate goal stack planning in the block world domain for the given start state and goal state.

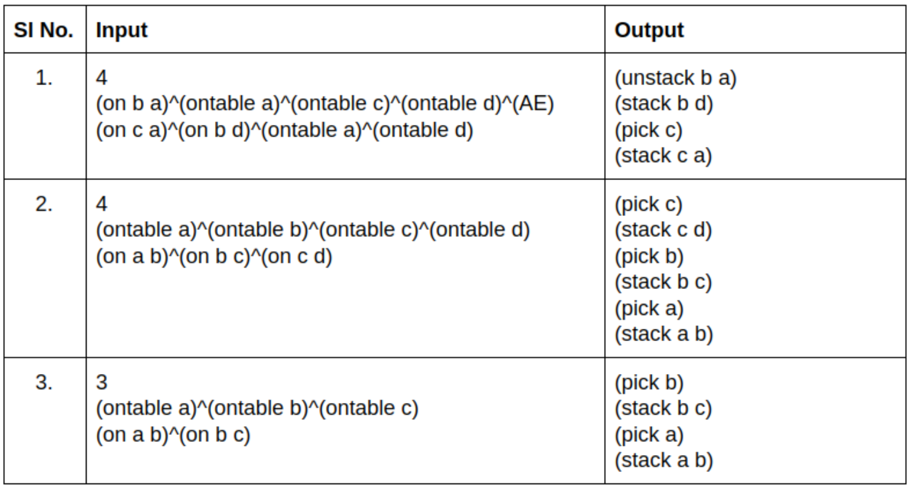
2 Pseudo Code



Algorithm 1 Goal Stack Planning

```
procedure GSP(givenState, givenGoal, actions)
    $state \leftarrow givenState$
    $plan \leftarrow ()$ {start with empty plan}
    $stack \leftarrow emptyStack$ {start with empty stack}
    $PushSet(givenGoal, stack)$
    while not Empty(stack) do
        $x \leftarrow Pop(stack)$
        if $x \in actions$ then
            $plan \leftarrow (plan \cdot x)$
            $state \leftarrow Progress(x, state)$
        else if $x$ is conjunct of goal predicates $C$ then
            $solvedFlag \leftarrow TRUE$
            for each $G \in C$ do
                if $G$ is unsatisfied in $state$ then
                    $solvedFlag \leftarrow FALSE$
                end if
            end for
            if $solvedFlag = FALSE$ then
                $pushSet(C, stack)$
            end if
        else if $x \notin givenState$ then
            $a \leftarrow chooseAction(x, state)$
            if $a$ is $None$ then
                return FAILURE
            end if
            $Push(a, stack)$
            $PushSet(Preconditions(a), stack)$
        end if
    end while
    return plan
```

3 Input-Output for Given Examples



Sl No.	Input	Output
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) (stack b d) (pick c) (stack c a)
2.	4 (ontable a)^(ontable b)^(ontable c)^(ontable d) (on a b)^(on b c)^(on c d)	(pick c) (stack c d) (pick b) (stack b c) (pick a) (stack a b)
3.	3 (ontable a)^(ontable b)^(ontable c) (on a b)^(on b c)	(pick b) (stack b c) (pick a) (stack a b)

4 Example 1: Stack Visualization

1. **Stack:**

   EMPTY

2. **Stack:** pushSet(Goal)

   (on c a)∧(on b d)∧(ontable a)∧(ontable d)
   (on c a)
   (on b d)
   (ontable a)

3. **Stack:** pop()

   (on c a)∧(on b d)∧(ontable a)∧(ontable d)
   (on c a)
   (on b d)

4. **Stack:** pop()

   (on c a)∧(on b d)∧(ontable a)∧(ontable d)
   (on c a)

5. **Stack:**
   push(stack b d) and preconditions
   pop()

   (on c a)∧(on b d)∧(ontable a)∧(ontable d)
   (on c a)
   (stack b d)
   (hold b)∧(clear d)
   (hold b)

6. **Stack:** pop()

   (on c a)∧(on b d)∧(ontable a)∧(ontable d)
   (on c a)
   (stack b d)
   (hold b)∧(clear d)

7. **Stack:**
   push(unstack b a) and preconditions
   pop()

   (on c a)∧(on b d)∧(ontable a)∧(ontable d)
   (on c a)
   (stack b d)
   (hold b)∧(clear d)
   (unstack b a)
   (on b a)∧(clear b)∧(AE)
   (on b a)
   (clear b)

8. **Stack:** pop()

   (on c a)∧(on b d)∧(ontable a)∧(ontable d)
   (on c a)
   (stack b d)
   (hold b)∧(clear d)
   (unstack b a)
   (on b a)∧(clear b)∧(AE)
   (on b a)

9. **Stack:** pop()

   (on c a)∧(on b d)∧(ontable a)∧(ontable d)
   (on c a)
   (stack b d)
   (hold b)∧(clear d)
   (unstack b a)
   (on b a)∧(clear b)∧(AE)

10. **Stack:** pop()

    (on c a)∧(on b d)∧(ontable a)∧(ontable d)
    (on c a)
    (stack b d)
    (hold b)∧(clear d)
    (unstack b a)

11. **Stack:**
    pop()
    Progress(unstack b a, state)

(on c a)∧(on b d)∧(ontable a)∧(ontable d)

(on c a)

(stack b d)

(hold b)∧(clear d)

12. **Stack:** pop()

(on c a)∧(on b d)∧(ontable a)∧(ontable d)

(on c a)

(stack b d)

13. **Stack:**

pop()

Progress(stack b d, state)

(on c a)∧(on b d)∧(ontable a)∧(ontable d)

(on c a)

14. **Stack:**

pop()

(on c a)∧(on b d)∧(ontable a)∧(ontable d)

15. **Stack:**

push(stack c a) and preconditions

pop()

(on c a)∧(on b d)∧(ontable a)∧(ontable d)

(stack c a)

(hold c)∧(clear a)

(hold c)

16. **Stack:** pop()

(on c a)∧(on b d)∧(ontable a)∧(ontable d)

(stack c a)

(hold c)∧(clear a)

17. **Stack:**

push(pick c) and preconditions

pop()

(on c a)∧(on b d)∧(ontable a)∧(ontable d)

(stack c a)

(hold c)∧(clear a)

(pick c)

(ontable c)∧(clear c)∧(AE)

(ontable c)

(clear c)

18. **Stack:** pop()

    (on c a)∧(on b d)∧(ontable a)∧(ontable d)
    (stack c a)
    (hold c)∧(clear a)
    (pick c)
    (ontable c)∧(clear c)∧(AE)
    (ontable c)

19. **Stack:** pop()

    (on c a)∧(on b d)∧(ontable a)∧(ontable d)
    (stack c a)
    (hold c)∧(clear a)
    (pick c)
    (ontable c)∧(clear c)∧(AE)

20. **Stack:** pop()

    (on c a)∧(on b d)∧(ontable a)∧(ontable d)
    (stack c a)
    (hold c)∧(clear a)
    (pick c)

21. **Stack:**
    pop()
    Progress(pick c, state)

    (on c a)∧(on b d)∧(ontable a)∧(ontable d)
    (stack c a)
    (hold c)∧(clear a)

22. **Stack:** pop()

    (on c a)∧(on b d)∧(ontable a)∧(ontable d)
    (stack c a)

23. **Stack:**
    pop()
    Progress(stack c a, state)

    (on c a)∧(on b d)∧(ontable a)∧(ontable d)

24. **Stack:**
    pop()

    EMPTY