

**CSD 311**  
**ARTIFICIAL INTELLIGENCE**  
**ASSIGNMENT-2 REPORT**

Aditi Singhal, 1910110030  
Rahul Madan, 1910110300  
Shambhavi Sud, 1910110365

**GOAL STACK TECHNIQUE**

In AI, we use the Goal Stack Technique to move backwards from the goal state to the initial state.

We begin at the goal state and attempt to meet the preconditions necessary to reach the initial/start state. These preconditions, in turn, have their own set of preconditions that must be met before proceeding. We continue to solve these "goals" and "subgoals" until we reach the Initial State. We use a stack to keep track of the goals that need to be met as well as the actions that need to be taken to achieve them.

We have a "World State" option in addition to the "Initial State" and "Goal State." This world state is used by Goal Stack to progress from Goal State to Initial State. The World State, on the other hand, begins as the Initial State and progresses to the Goal State.

We are left with an empty stack and a set of actions at the end of this procedure to help us transit from the Initial State to the World State.

The configurations are represented as a list of "predicates."  
Predicates are statements that aid in the communication of information about a configuration in Blocks World.

The list of predicates as well as their intended meaning is given below :

The program recursively builds a stack, we do not maintain one specifically.

**ON(A,B)** : Block A is on B

**ONTABLE(A)** : A is on table

**CLEAR(A)** : Nothing is on top of A

**HOLDING(A)** : Arm is holding A.

**ARMEMPTY** : Arm is holding nothing

There are a total of 3 lists `on_table_elements`, `middle_elements`, `clear_elements` maintaining a record of those on table, on top of another and those that are in the middle respectively. Here we hold the assumption that within a single stack the last box will be on the table and the topmost one will always be clear on top.

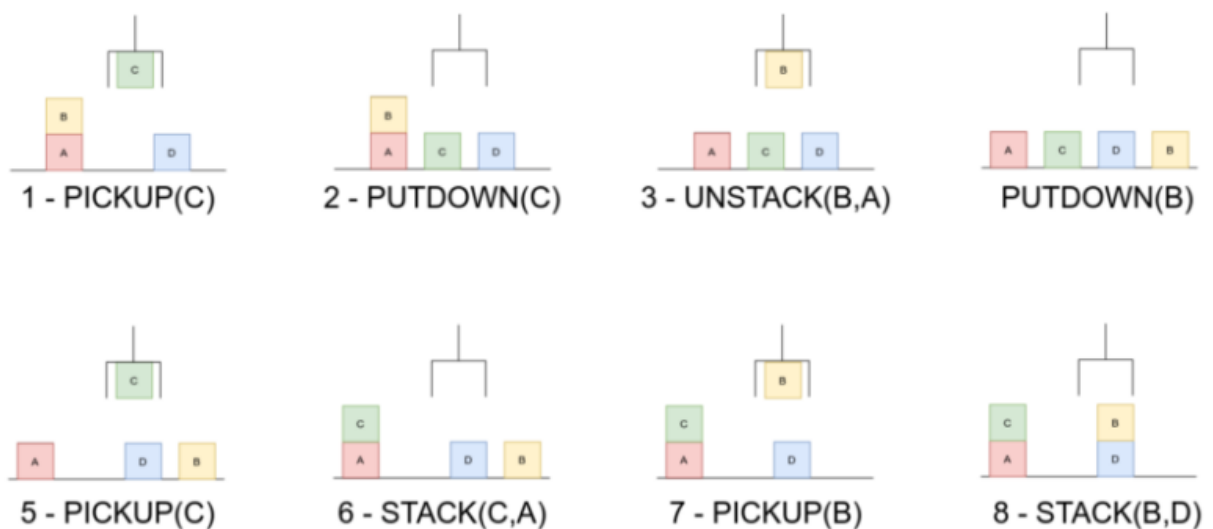
The actions used are :

**UNSTACK(A,B)** -- pick up and clear block A from block B;

**STACK(A,B)** -- place block A using the arm onto clear block B;

**PICKUP(A)** -- lift clear block A with the empty arm;

**PUTDOWN(A)** -- place the held block A onto a free space on the table.



## **BLOCKS WORLD PROBLEM**

There is a table with some blocks set on it. Some blocks can be stacked on top of others, while some cannot. To pick up or place the blocks, we have a robot arm. The robot arm can only move one block at a time, and no other blocks should be put on top of the robot arm's moving block.

Our goal is to modify the block arrangement from the Initial State to the Goal State.

### **Percentage Contributions of each member :**

Shambhavi = 33.333333%

Aditi = 33.333333%

Rahul = 33.333333%

### **References :**

- 1) <https://www.d.umn.edu/~gshute/cs2511/projects/Java/assignment6/blocks/blocks.x.html>
- 2) <https://aithefuture.wordpress.com/2018/02/26/goal-stack-planning/>
- 3) <https://users.cs.cf.ac.uk/Dave.Marshall/AI2/node121.html>