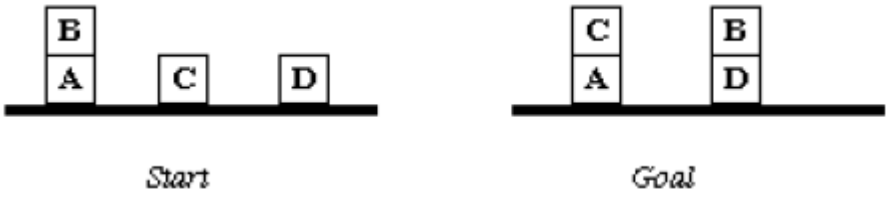
ASSIGNMENT AIR-3

Roll No: 41205

Problem Statement:

Implement goal stack planning for the following configurations from the blocks world,



Objective:

1. Understand goal stack planning
2. Apply the algorithm to the configurations mentioned above

Outcome: One will be able to apply goal stack planning for any block world configuration

Pre-requisites:

1. 64-bit Linux OS
2. Programming Languages: Python

Hardware Specification:

1. x86_64 bit
2. 2/4 GB DDR RAM
3. 80 - 500 GB SATA HD
4. 1GB NIDIA TITAN X Graphics Card

Software Specification:

1. Ubuntu 14.04

Theory:

- Goal Stack Planning is one of the earliest methods in artificial intelligence in which we work backwards from the goal state to the initial state.

- We start at the goal state, and we try fulfilling the preconditions required to achieve the initial state. These preconditions in turn have their own set of preconditions, which are required to be satisfied first. We keep solving these “goals” and “sub-goals” until we finally arrive at the Initial State. We make use of a stack to hold these goals that need to be fulfilled the actions as well that

we need to perform for the same.

- Apart from the “Initial State” and the “Goal State”, we maintain a “World State” configuration as well. Goal Stack uses this world state to work its way from Goal State to Initial State. World State on the other hand starts off as the Initial State and ends up being transformed into the Goal state.

- At the end of this algorithm, we are left with an empty stack and a set of actions which helps us navigate from the Initial State to the World State.
- Representing the configurations as a list of “predicates”

- Predicates can be thought of as a statement which helps us convey the information about a configuration in Blocks World.

- Given below are the list of predicates as well as their intended meaning

    - 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

- Using these predicates, we represent the Initial State and the Goal State in our example like this:

- Initial State — ON(B,A) ∧ ONTABLE(A) ∧ ONTABLE(C) ∧ ONTABLE(D) ∧ CLEAR(B) ∧ CLEAR(C) ∧ CLEAR(D) ∧ ARMEMPTY
- Initial State

- Goal State — ON(C,A) ∧ ON(B,D) ∧ ONTABLE(A) ∧ ONTABLE(D) ∧ CLEAR(B) ∧ CLEAR(C) ∧ ARMEMPTY
- Goal State

- Thus, a configuration can be thought of as a list of predicates describing the current scenario.

- “Operations” performed by the robot arm

- The Robot Arm can perform 4 operations:

    - STACK(X,Y) : Stacking Block X on Block Y
    - UNSTACK(X,Y) : Picking up Block X which is on top of Block Y
    - PICKUP(X) : Picking up Block X which is on top of the table
    - PUTDOWN(X) : Put Block X on the table

- All the four operations have certain preconditions which need to be satisfied to

perform the same. These preconditions are represented in the form of predicates.

- The effect of these operations is represented using two lists ADD and DELETE. DELETE List contains the predicates which will cease to be true once the operation is performed. ADD List on the other hand contains the predicates which will become true once the operation is performed.

Input:

Configuration as mentioned in the problem statement.

Output:

Steps to reach from initial to final state.

```
[PICKUP(C), PUTDOWN(C), UNSTACK(B,A), PUTDOWN(B), PICKUP(C), STACK(C,A), PICKUP(B), STACK(B,D)]
```

Conclusion: Thus, we were able to apply goal stack planning for the block worlds problem to reach to the final state from the initial state.