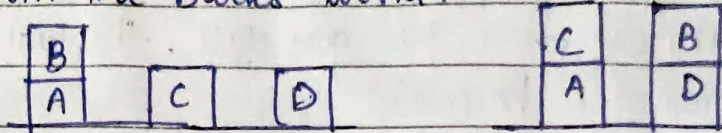


Assignment - AIR3

* TITLE: Goal Stack Planning

* PROBLEM STATEMENT: Implement goal stack planning for the following configuration from the blocks world:



* OBJECTIVE: To learn & implement goal stack planning.

* OUTCOME: Be able to learn and successfully implement goal stack planning.

* SOFTWARE AND HARDWARE REQUIREMENTS: Python 3, UNIX/LINUX OS, 64 bit CPU, 8 GB RAM

* THEORY:

Blocks World Problem is described as follows: There are N blocks, a table and a robotic arm. Blocks are identified by integers $1 \dots N$.

Each block can sit on top of another block, or on the block table; there can be a stack of blocks of arbitrary height.

However, only one block can be directly on another block. No 2 blocks can be sitting directly on the same block.

The bottom-most block of a stack must be on the table. The table can hold any number of blocks.

If there is no block on top of table block, then the block is clear. The robotic arm can only hold one. If robotic arm does not hold any block, it is empty.

GOAL STACK PLANNER:

Goal stack planning integrates the advantages of forward & backward.

An action is added to the plan only if its preconditions are satisfied. If any precondition is not satisfied, then we add a relevant action for that precondition on stack and repeat the same process.

There might be more than one relevant action for some predicates; but since Goal stack planning tries to choose only the relevant action starting from the goal state, the time it takes is smaller than forward & backward.

Algorithm:

1. Push the goal state on stack.

2. Repeat until the stack is empty:

a) If stack top is a compound goal,

i) push its subgoals on stack.

b) If stack top is a single unsatisfied goal

i) replace it by an action that makes it satisfied.

ii) push the action's performed precondition on stack.

c) If stack top is an action

i) check for unsatisfied prerequisites.

ii) if all prerequisites are satisfied

i) pop action from stack

ii) execute it

iii) change the knowledge base by action's effects.

else

i) push unsatisfied preconditions on stack.

d) If stack top is a satisfied goal

i) pop it from stack.

* CONCLUSION:

Successfully implemented goal stack planning.