

## Artificial Intelligence Laboratory 9

COMPILED BY MEHUL BOSE (170030010)

### Pseudo Code:

```

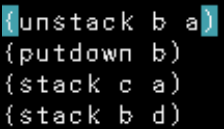
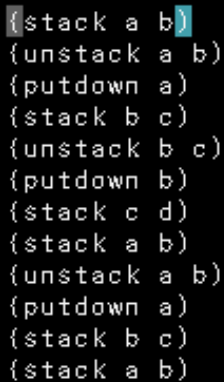
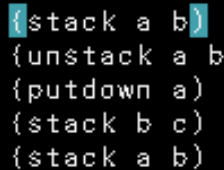
class Arm:
    # Defines an arm class:
    def __init__(self):
        # Contains two attributes, holding and empty
        self._holding = None
        self._empty = True
        # Both attributes have a set and get function,
        # Additionally functions are created to pick up a given block
        # or put down a held block down onto the table

class Block:
    def __init__(self, name):
        # Contains 5 attributes as shown
        self._name = name
        self._clear = True
        self._onTable = True
        self._on = None
        self._top = None
        # All of these attributes except name have a set and get function
        # Name does not require a set function as a block, once named, cannot
        # be named again
        # Additionally there are functions to stack and unstack given blocks on
        # top of other blocks

generate_starting_state(filename):
    # Generates the starting state by defining individual blocks and the arm
    # (essentially the starting state)
    actions = []
    # Generate a blank list which will store the required actions
    stack = []
    stack.append(goals)
    while len(stack) > 0:
        # While stack is not empty
        # pop an item
        # if the item is a set of conditions:
        # check if they are all true:
        # if yes, continue
        # else, add the set of condition followed by the individual condition
        # If the item is a single condition:
        # check if it is true:
        # if yes, continue
        # else, add an action to the stack to make the condition true.
        # actions for each individual condition:
        # on: stack
        # ontable: putdown
        # hold: unstack / pick (depending on the state)
        # clear: unstack
        # AE: putdown
        # If the item is an action
        # Check if the preconditions for the action are satisfied or not.
        # if yes, add action to actions list
        # else, add the preconditions in and form (C1^C2^C3)
    # Store into output file

```

**Input and output for 3 examples:**

S. No.	Input	Output	Terminal output screenshot
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)	

**Stack visualisation for the first input:**

Initialised with goal state:

$[(\text{on } c \ a) \wedge (\text{on } b \ d) \wedge (\text{ontable } a) \wedge (\text{ontable } d)]$

Added each condition indivisually as all conditions are not satisfied:

$[(\text{on } c \ a) \wedge (\text{on } b \ d) \wedge (\text{ontable } a) \wedge (\text{ontable } d), (\text{ontable } d), (\text{ontable } a), (\text{on } b \ d), (\text{on } c \ a)]$

Added stack operation to satisfy on condition:

$[(\text{on } c \ a) \wedge (\text{on } b \ d) \wedge (\text{ontable } a) \wedge (\text{ontable } d), (\text{ontable } d), (\text{ontable } a), (\text{on } b \ d), \text{stack } c \ a]$

Added conditions to allow stack operation to occur:

$[(\text{on } c \ a) \wedge (\text{on } b \ d) \wedge (\text{ontable } a) \wedge (\text{ontable } d), (\text{ontable } d), (\text{ontable } a), (\text{on } b \ d), \text{stack } c \ a, (\text{clear } c) \wedge (\text{clear } a), (AE), (\text{clear } c), (\text{clear } a)]$

Added unstack operation to satisfy clear condition

$[(\text{on } c \ a) \wedge (\text{on } b \ d) \wedge (\text{ontable } a) \wedge (\text{ontable } d), (\text{ontable } d), (\text{ontable } a), (\text{on } b \ d), \text{stack } c \ a, (\text{clear } c) \wedge (\text{clear } a), (AE), (\text{clear } c), \text{unstack } b \ a]$

Unstacked block b from block a

$[(\text{on } c \ a) \wedge (\text{on } b \ d) \wedge (\text{ontable } a) \wedge (\text{ontable } d), (\text{ontable } d), (\text{ontable } a), (\text{on } b \ d), \text{stack } c \ a, (\text{clear } c) \wedge (\text{clear } a), (AE), (\text{clear } c)]$

Block c is already clear

$[(\text{on } c \ a) \wedge (\text{on } b \ d) \wedge (\text{ontable } a) \wedge (\text{ontable } d), (\text{ontable } d), (\text{ontable } a), (\text{on } b \ d), \text{stack } c \ a, (\text{clear } c) \wedge (\text{clear } a), (AE)]$

Added putdown condition to satisfy AE condition

$[(\text{on } c \ a) \wedge (\text{on } b \ d) \wedge (\text{ontable } a) \wedge (\text{ontable } d), (\text{ontable } d), (\text{ontable } a), (\text{on } b \ d), \text{stack } c \ a, (\text{clear } c) \wedge (\text{clear } a), \text{putdown } b]$

Put down block b on table

$[(\text{on } c \ a) \wedge (\text{on } b \ d) \wedge (\text{ontable } a) \wedge (\text{ontable } d), (\text{ontable } d), (\text{ontable } a), (\text{on } b \ d), \text{stack } c \ a, (\text{clear } c) \wedge (\text{clear } a)]$

Preconditions of stack operation are met

$[(\text{on } c \ a) \wedge (\text{on } b \ d) \wedge (\text{ontable } a) \wedge (\text{ontable } d), (\text{ontable } d), (\text{ontable } a), (\text{on } b \ d), \text{stack } c \ a]$

Stacked block c on block a

$[(\text{on } c \ a) \wedge (\text{on } b \ d) \wedge (\text{ontable } a) \wedge (\text{ontable } d), (\text{ontable } d), (\text{ontable } a), (\text{on } b \ d)]$

Added stack operation to satisfy on condition:

$[(\text{on } c \ a) \wedge (\text{on } b \ d) \wedge (\text{ontable } a) \wedge (\text{ontable } d), (\text{ontable } d), (\text{ontable } a), \text{stack } b \ d]$

Stacked block b on block d

$[(\text{on } c \ a) \wedge (\text{on } b \ d) \wedge (\text{ontable } a) \wedge (\text{ontable } d), (\text{ontable } d), (\text{ontable } a)]$

Block a is already on the table

$[(\text{on } c \ a) \wedge (\text{on } b \ d) \wedge (\text{ontable } a) \wedge (\text{ontable } d), (\text{ontable } d)]$

Block d is already on the table

$[(\text{on } c \ a) \wedge (\text{on } b \ d) \wedge (\text{ontable } a) \wedge (\text{ontable } d)]$

Goal State is satisfied

[]