```
# Define a function to represent the state of the block world
def print_state(state):
    for key in state:
        print(f"{key}: {state[key]}")
    print()
# Function to check if a block is clear (no other block is on top of it)
def is_clear(block, state):
    return state['clear'][block]
# Function to move a block from one position to another
def move_block(block, from_pos, to_pos, state):
    state['clear'][from_pos] = True
    state['clear'][block] = False
    state['clear'][to_pos] = False
    state['on'][block] = to_pos
    print(f"Move {block} from {from_pos} to {to_pos}")

# Goal Stack Planning algorithm
def goal_stack_planning(initial_state, goal_state):
    state = initial_state
    goal_stack = []

    # Initialize the goal stack with the goal conditions
    for goal in goal_state['on']:
        goal_stack.append(('ON', goal, goal_state['on'][goal]))
    while goal_stack:
        subgoal = goal_stack.pop()
        if subgoal[0] == 'ON':
            block, target = subgoal[1], subgoal[2]
            if state['on'][block] != target:
                if not is_clear(target, state):
                    # If the target position is not clear, add a subgoal to clear it
```

```
                goal_stack.append(('CLEAR', target))
            if not is_clear(block, state):
                # If the block itself is not clear, add a subgoal to clear it
                goal_stack.append(('CLEAR', block))
            # Add an action to move the block to the target position
            goal_stack.append(('MOVE', block, state['on'][block], target))
        elif subgoal[0] == 'CLEAR':
            block = subgoal[1]
            if not is_clear(block, state):
                # If the block is not clear, move the top block to the table
                for b in state['on']:
                    if state['on'][b] == block:
                        goal_stack.append(('MOVE', b, block, 'table'))
                        break
        elif subgoal[0] == 'MOVE':
            block, from_pos, to_pos = subgoal[1], subgoal[2], subgoal[3]
            move_block(block, from_pos, to_pos, state)
            print_state(state)
    print("Goal state reached.")

# Define the initial state and goal state
initial_state = {
    'on': {'A': 'table', 'B': 'A', 'C': 'table'},  # 'on' dictionary represents where each block is located
    'clear': {'A': False, 'B': True, 'C': True}    # 'clear' indicates if a block has nothing on top of it
}
goal_state = {
    'on': {'A': 'B', 'B': 'C', 'C': 'table'}       # The goal is to have A on B, B on C, and C on the table
}
# Run the Goal Stack Planning algorithm
goal_stack_planning(initial_state, goal_state)
```