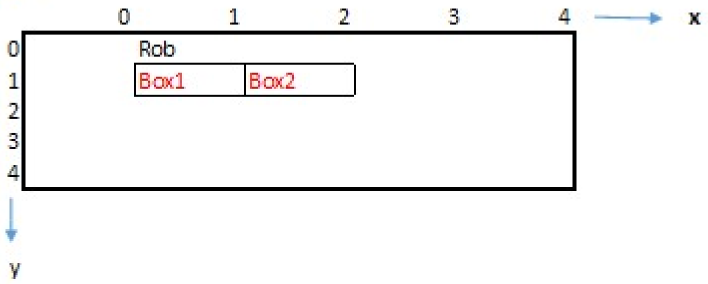
Name: Zhihan Wang

Student ID: 17272381

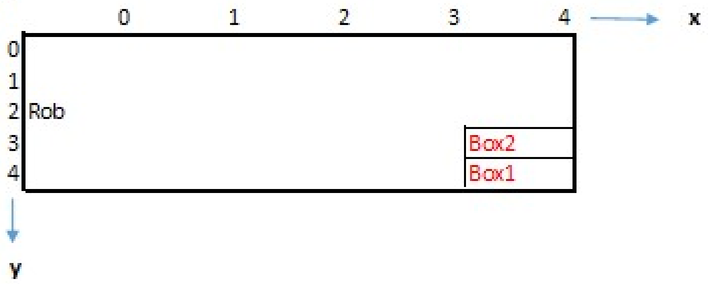
Problem 2:

Solving the planning problem for a robot trying to push two boxes to their designated location and return to a designated empty location in a x and y size rectangular room, as per an example graph below, and its source code is in 'Problem2_Boxes' file .

Initial State



Goal State



Primitive Features:

Feature	**Comment**
Rob : { room(row,col) for row in range(x) for col in range(y)}	Rob's location at room (row,col) in a x and y size rectangular room
Box1 : { room(row,col) for row in range(x) for col in range(y)}	Box1's location at room (row,col) in a x and y size rectangular room
Box2: { room(row,col) for row in range(x) for col in range(y)}	Box2's location at room (row,col) in a x and y size rectangular room

Actions:

Operator: Rob_push_box1_from_room_(row,col)_to_room_(row,col+1)_and_ Rob_move_from_room_(row,col-1)_to_room_(row,col)_box2_is_at_(box2_row,box2_col)
Pre- Condition: {'Rob': room (row,col-1) , 'Box1': room (row,col) , 'Box2': room(box2_row,box2_col) }
Effect: {'Rob': room (row,col) , 'Box1': room (row,col+1), 'Box2': room(box2_row,box2_col) }
Comment: In a x and y size rectangular room, Rob push box1 right , it uses 3 rooms horizontally for Rob move from (row,col-1) to (row,col) and box1 from (row,col) to (row,col+1),and enumerate all box2 possible locations. However, box1 has constraint that it can not be located at 0 and y of column which are at the edge of room , box2 can not be at the 3 rooms where Rob push box1 right either.

Operator: Rob_push_box2_from_room_(row,col)_to_room_(row,col+1)_and_ Rob_move_from_room_(row,col-1)_to_room_(row,col)_box1_is_at_(box1_row,box1_col)

Pre- Condition: {'Rob': room (row,col-1) , 'Box2': room (row,col) , 'Box1': room(box1_row,box1_col) }

Effect: {'Rob': room (row,col) , 'Box2': room (row,col+1), 'Box1': room(box1_row,box1_col) }

Comment: Rob push box2 right, similar constraint as comment in Rob push box1 right, but swap their conditions

Operator: Rob_push_box1_from_room_(row,col)_to_room_(row,col-1)_and_ Rob_move_from_room_(row,col+1)_to_room_(row,col)_box2_is_at_(box2_row,box2_col)

Pre- Condition: {'Rob': room (row,col+1) , 'Box1': room (row,col) , 'Box2': room (box2_row,box2_col) }

Effect: {'Rob': room (row,col) , 'Box1': room (row,col-1), 'Box2': room(box2_row,box2_col) }

Comment: In a x and y size rectangular room, Rob push box1 left , it uses 3 rooms horizontally for Rob move from (row,col+1) to (row,col) and box1 from (row,col) to (row,col-1), and enumerate all box2 possible locations. However, box1 has constraint that it can not be located at 0 and y of column which are at the edge of room , box2 can not be at the 3 rooms where Rob push box1 left either.

Operator: Rob_push_box2_from_room_(row,col)_to_room_(row,col-1)_and_ Rob_move_from_room_(row,col+1)_to_room_(row,col)_box2_is_at_(box1_row,box1_col)

Pre- Condition: {'Rob': room (row,col+1) , 'Box2': room (row,col) , 'Box1': room(box1_row,box1_col) }

Effect: {'Rob': room (row,col) , 'Box2': room (row,col-1), 'Box1': room(box1_row,box1_col) }

Comment: Rob push box2 left , similar constraint as comment in Rob push box1 left but swap their conditions

Operator: Rob_push_box1_from_room_(row,col)_to_room_(row-1,col)_and_ Rob_move_from_room_(row+1,col)_to_room_(row,col)_box2_is_at_(box2_row,box2_col)

Pre- Condition: {'Rob': room (row+1,col) , 'Box1': room (row,col) , 'Box2': room(box2_row,box2_col) }

Effect: {'Rob': room (row,col) , 'Box1': room (row-1,col), 'Box2': room(box2_row,box2_col) }

Comment: In a x and y size rectangular room, Rob push box1 up , it uses 3 rooms vertically for Rob move from (row+1,col) to (row,col) and push box1 from (row,col) to (row-1,col) , and enumerate all box2 possible locations. However, box1 has constraint that it can not be located at 0 and x of rows which are at the edge of room , box2 can not be at the 3 rooms where Rob push box1 up either.

Operator: Rob_push_box2_from_room_(row,col)_to_room_(row-1,col)_and_ Rob_move_from_room_(row+1,col)_to_room_(row,col)_box1_is_at_(box1_row,box1_col)

Pre- Condition: {'Rob': room (row+1,col) , 'Box2': room (row,col) , 'Box1': room(box1_row,box1_col) }

Effect: {'Rob': room (row,col) , 'Box2': room (row-1,col), 'Box1': room(box1_row,box1_col) }

Comment: Rob push box2 up , similar constraint as comment in Rob push box1 up but swap their conditions

Operator: Rob_push_box1_from_room_(row,col)_to_room_(row+1,col)_and_ Rob_move_from_room_(row-1,col)_to_room_(row,col)_box2_is_at_(box2_row,box2_col)

Pre- Condition: {'Rob': room (row-1,col) , 'Box1': room (row,col) , 'Box2': room(box2_row,box2_col) }

Effect: {'Rob': room (row,col) , 'Box1': room (row+1,col), 'Box2': room(box2_row,box2_col) }

Comment: In a x and y size rectangular room, Rob push box1 down , it uses 3 rooms vertically for Rob move from (row-1,col) to (row,col) and push box1 from (row,col) to (row+1,col) , and enumerate all box2 possible locations. However, box1 has constraint that it can not be located at 0 and x of rows which are at the edge of room , box2 can not be at the 3 rooms where Rob push box1 down either.

Operator: Rob_push_box2_from_room_(row,col)_to_room_(row+1,col)_and_ Rob_move_from_room_(row-1,col)_to_room_(row,col)_box1_is_at_(box1_row,box1_col)

Pre- Condition: {'Rob': room (row-1,col) , 'Box2': room (row,col) , 'Box1': room(box1_row,box1_col) }

Effect: {'Rob': room (row,col) , 'Box2': room (row+1,col), 'Box1': room(box1_row,box1_col) }

Comment: Rob push box2 down , similar constraint as comment in Rob push box1 up but swap their conditions

Operator:'Rob_move_from_room_(row,col)_to_room_(row,col+1)_and_box1_at_room_(box1_row,box1_col)_box2_at_(box2_row,box2_col)

Pre- Condition: {'Rob': room (row,col), 'Box1' : room (box1_row,box1_col) , 'Box2' : room (box2_row,box2_col)

Effect: {'Rob': room (row,col+1),'Box1' : room (box1_row,box1_col),'Box2' : room (box2_row,box2_col) }

Comment: In a x and y size rectangular room, Rob move right and do not push Box1 and Box 2, it uses 2 rooms horizontally for Rob move from (row,col) to (row,col+1) , and enumerate all box1 and box2 possible locations. However, box1 and box2 have constraints that they can not have overlap locations, and box1 and box2 can not be at the 2 rooms where Rob move either.

Operator:'Rob_move_from_room_(row,col)_to_room_(row,col-1)_and_box1_at_room_(box1_row,box1_col)_box2_at_(box2_row,box2_col)

Pre- Condition: {'Rob': room (row,col), 'Box1' : room (box1_row,box1_col) , 'Box2' : room (box2_row,box2_col)

Effect: {'Rob': room (row,col-1),'Box1' : room (box1_row,box1_col),'Box2' : room (box2_row,box2_col) }

Comment: In a x and y size rectangular room, Rob move left and do not push Box1 and Box 2, it uses 2 rooms horizontally for Rob move from (row,col) to (row,col-1) , and enumerate all box1 and box2 possible locations. However, box1 and box2 have constraints that they can not have overlap locations, and box1 and box2 can not be at the 2 rooms where Rob move either.

Operator:'Rob_move_from_room_(row,col)_to_room_(row-1,col)_and_box1_at_room_(box1_row,box1_col)_box2_at_(box2_row,box2_col)

Pre- Condition: {'Rob': room (row,col), 'Box1' : room (box1_row,box1_col) , 'Box2' : room (box2_row,box2_col)

Effect: {'Rob': room (row-1,col),'Box1' : room (box1_row,box1_col),'Box2' : room (box2_row,box2_col) }

Comment: In a x and y size rectangular room, Rob move up and do not push Box1 and Box 2, it uses 2 rooms horizontally for Rob move from (row,col) to (row-1,col) , and enumerate all box1 and box2 possible locations. However, box1 and box2 have constraints that they can not have overlap locations, and box1 and box2 can not be at the 2 rooms where Rob move either.

Operator:'Rob_move_from_room_(row,col)_to_room_(row+1,col)_and_box1_at_room_(box1_row,box1_col)_box2_at_(box2_row,box2_col)

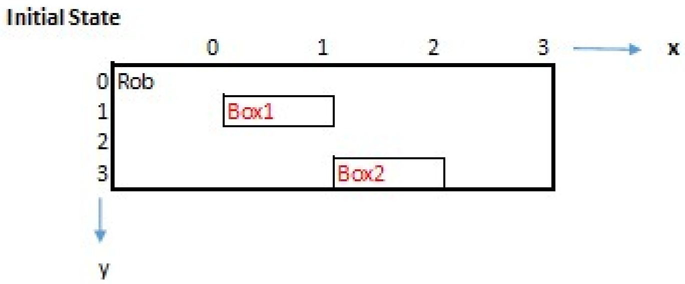
Pre- Condition: {'Rob': room (row,col), 'Box1' : room (box1_row,box1_col) , 'Box2' : room (box2_row,box2_col)

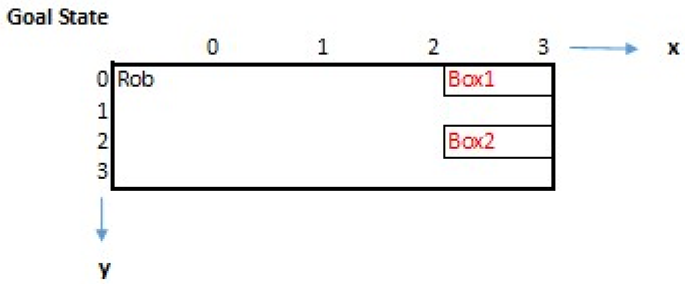
Effect: {'Rob': room (row+1,col),'Box1' : room (box1_row,box1_col),'Box2' : room (box2_row,box2_col) }

Comment: In a x and y size rectangular room, Rob move down and do not push Box1 and Box 2, it uses 2 rooms horizontally for Rob move from (row,col) to (row+1,col) , and enumerate all box1 and box2 possible locations. However, box1 and box2 have constraints that they can not have overlap locations, and box1 and box2 can not be at the 2 rooms where Rob move either.

Problem Solutions:

Case1: Robot start at room(0,0) , push box1 from room (1,1) to room (0,3) , box2 from room (3,2) to room(2,3) and return to room(0,0) in a 4 x 4 room as graph below



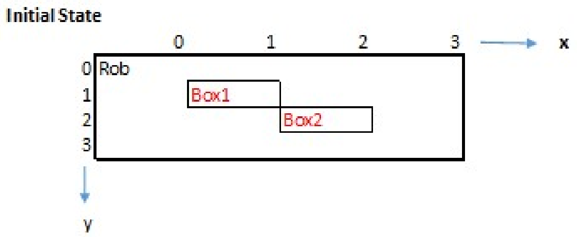


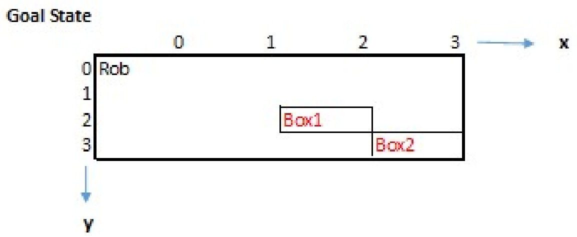
Initial State: { 'Rob' : room(0,0), 'Box1':room(1,1),'Box2':room(3,2) }

Goal State: { 'Rob' : room(0,0), 'Box1':room(0,3),'Box2':room(2,3) }

Comment: No solution was found by both STRIPS forward and regression planners, since there is no route to move Box2 up to its designated location.

Case2: Robot start at room(0,0) , push box1 from room (1,1) to room (2,2) , box2 from room (2,2) to room(3,3) and return to room(0,0)in a 4 x 4 room as graph below





Initial State: { 'Rob' : room(0,0), 'Box1':room(1,1),'Box2':room(2,2) }

Goal State: { 'Rob' : room(0,0), 'Box1':room(2,2),'Box2':room(3,3) }

Comment: STRIPS forward and regression planners have found different solutions

One of solutions from STRIPS forward planner:

Step	Actions
0	Rob_move_from_room_(0,0)_to_room_(0,1)_and_box1_at_room_(1,1)_box2_at_(2,2)
1	Rob_push_box1_down_from_room_(1,1)_to_room_(2,1)_and_Rob_move_down_from_room_(0,1)_to_room_(1,1)_box2_is_at_(2,2)
2	Rob_move_from_room_(1,1)_to_room_(1,2)_and_box1_at_room_(2,1)_box2_at_(2,2
3	Rob_push_box2_down_from_room_(2,2)_to_room_(3,2)_and_Rob_move_down_from_room_(1,2)_to_room_(2,2)_box1_is_at_(2,1)
4	Rob_move_from_room_(2,2)_to_room_(1,2)_and_box1_at_room_(2,1)_box2_at_(3,2)
5	Rob_move_from_room_(1,2)_to_room_(1,1)_and_box1_at_room_(2,1)_box2_at_(3,2)
6	Rob_move_from_room_(1,1)_to_room_(1,0)_and_box1_at_room_(2,1)_box2_at_(3,2)
7	Rob_move_from_room_(1,0)_to_room_(2,0)_and_box1_at_room_(2,1)_box2_at_(3,2)
8	Rob_push_box1_rigt_from_room_(2,1)_to_room_(2,2)_and_Rob_move_right_from_room_(2,0)_to_room_(2,1)_box2_is_at_(3,2)
9	Rob_move_from_room_(2,1)_to_room_(3,1)_and_box1_at_room_(2,2)_box2_at_(3,2)
10	Rob_push_box2_right_from_room_(3,2)_to_room_(3,3)_and_Rob_move_right_from_room_(3,1)_to_room_(3,2)_box1_is_at_(2,2)
11	Rob_move_from_room_(3,2)_to_room_(3,1)_and_box1_at_room_(2,2)_box2_at_(3,3)
12	Rob_move_from_room_(3,1)_to_room_(3,0)_and_box1_at_room_(2,2)_box2_at_(3,3)
13	Rob_move_from_room_(3,0)_to_room_(2,0)_and_box1_at_room_(2,2)_box2_at_(3,3)
14	Rob_move_from_room_(2,0)_to_room_(1,0)_and_box1_at_room_(2,2)_box2_at_(3,3)
15	Rob_move_from_room_(1,0)_to_room_(0,0)_and_box1_at_room_(2,2)_box2_at_(3,3)

One of solutions from STRIPS regression planner:

Step	Actions
0	Rob_move_from_room_(0,0)_to_room_(1,0)_and_box1_at_room_(1,1)_box2_at_(2,2)
1	Rob_push_box1_right_from_room_(1,1)_to_room_(1,2)_and_Rob_move_right_from_room_(1,0)_to_room_(1,1)_box2_is_at_(2,2)
2	Rob_move_from_room_(1,1)_to_room_(2,1)_and_box1_at_room_(1,2)_box2_at_(2,2)
3	Rob_push_box2_rigt_from_room_(2,2)_to_room_(2,3)_and_Rob_move_right_from_room_(2,1)_to_room_(2,2)_box1_is_at_(1,2)
4	Rob_move_from_room_(2,2)_to_room_(2,1)_and_box1_at_room_(1,2)_box2_at_(2,3)
5	Rob_move_from_room_(2,1)_to_room_(1,1)_and_box1_at_room_(1,2)_box2_at_(2,3)
6	Rob_move_from_room_(1,1)_to_room_(0,1)_and_box1_at_room_(1,2)_box2_at_(2,3)
7	Rob_move_from_room_(0,1)_to_room_(0,2)_and_box1_at_room_(1,2)_box2_at_(2,3)
8	Rob_push_box1_down_from_room_(1,2)_to_room_(2,2)_and_Rob_move_down_from_room_(0,2)_to_room_(1,2)_box2_is_at_(2,3)
9	Rob_move_from_room_(1,2)_to_room_(1,3)_and_box1_at_room_(2,2)_box2_at_(2,3)
10	Rob_push_box2_down_from_room_(2,3)_to_room_(3,3)_and_Rob_move_down_from_room_(1,3)_to_room_(2,3)_box1_is_at_(2,2)
11	Rob_move_from_room_(2,3)_to_room_(1,3)_and_box1_at_room_(2,2)_box2_at_(3,3)
12	Rob_move_from_room_(1,3)_to_room_(0,3)_and_box1_at_room_(2,2)_box2_at_(3,3)

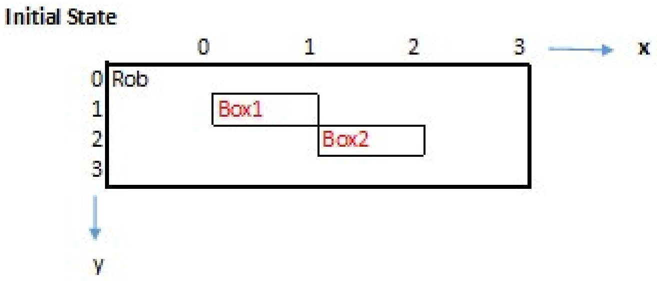
13	Rob_move_from_room_(0,3)_to_room_(0,2)_and_box1_at_room_(2,2)_box2_at_(3,3)
14	Rob_move_from_room_(0,2)_to_room_(0,1)_and_box1_at_room_(2,2)_box2_at_(3,3)
15	Rob_move_from_room_(0,1)_to_room_(0,0)_and_box1_at_room_(2,2)_box2_at_(3,3)

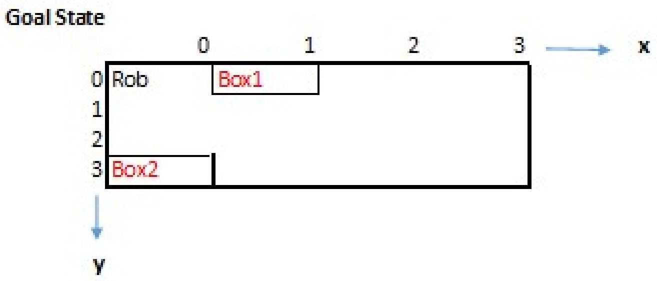
Case3: Robot start at room(0,0) , push box1 from room (1,1) to room (0,1) , box2 from room (2,2) to room(3,0) and return to room(0,0)in a 4 x 4 room as graph below

Initial State: { 'Rob' : room(0,0), 'Box1':room(1,1),'Box2':room(2,2) }

Goal State: { 'Rob' : room(0,0), 'Box1':room(0,1),'Box2':room(3,0) }

Comment: STRIPS forward and regression planners have found different solutions





One of solutions from STRIPS forward planner:

Step	Actions
0	Rob_move_from_room_(0,0)_to_room_(0,1)_and_box1_at_room_(1,1)_box2_at_(2,2)
1	Rob_move_from_room_(0,1)_to_room_(0,2)_and_box1_at_room_(1,1)_box2_at_(2,2)
2	Rob_move_from_room_(0,2)_to_room_(1,2)_and_box1_at_room_(1,1)_box2_at_(2,2)
3	Rob_push_box2_down_from_room_(2,2)_to_room_(3,2)_and_Rob_move_down_from_room_(1,2)_to_room_(2,2)_box1_is_at_(1,1)
4	Rob_move_from_room_(2,2)_to_room_(2,3)_and_box1_at_room_(1,1)_box2_at_(3,2)
5	Rob_move_from_room_(2,3)_to_room_(3,3)_and_box1_at_room_(1,1)_box2_at_(3,2)
6	Rob_push_box2_left_from_room_(3,2)_to_room_(3,1)_and_Rob_move_left_from_room_(3,3)_to_room_(3,2)_box1_is_at_(1,1)
7	Rob_push_box2_left_from_room_(3,1)_to_room_(3,0)_and_Rob_move_left_from_room_(3,2)_to_room_(3,1)_box1_is_at_(1,1)
8	Rob_move_from_room_(3,1)_to_room_(2,1)_and_box1_at_room_(1,1)_box2_at_(3,0)
9	Rob_push_box1_up_from_room_(1,1)_to_room_(0,1)_and_Rob_move_up_from_room_(2,1)_to_room_(1,1)_box2_is_at_(3,0)
10	Rob_move_from_room_(1,1)_to_room_(1,0)_and_box1_at_room_(0,1)_box2_at_(3,0)

11	Rob_move_from_room_(1,0)_to_room_(0,0)_and_box1_at_room_(0,1)_box2_at_(3,0)

One of solutions from STRIPS regression planner:

Step	Actions
0	Rob_move_from_room_(0,0)_to_room_(0,1)_and_box1_at_room_(1,1)_box2_at_(2,2)
1	Rob_push_box1_down_from_room_(1,1)_to_room_(2,1)_and_Rob_move_down_from_room_(0,1)_to_room_(1,1)_box2_is_at_(2,2)
2	Rob_move_from_room_(1,1)_to_room_(1,2)_and_box1_at_room_(2,1)_box2_at_(2,2)
3	Rob_push_box2_down_from_room_(2,2)_to_room_(3,2)_and_Rob_move_down_from_room_(1,2)_to_room_(2,2)_box1_is_at_(2,1)
4	Rob_move_from_room_(2,2)_to_room_(2,3)_and_box1_at_room_(2,1)_box2_at_(3,2)
5	Rob_move_from_room_(2,3)_to_room_(3,3)_and_box1_at_room_(2,1)_box2_at_(3,2)
6	Rob_push_box2_left_from_room_(3,2)_to_room_(3,1)_and_Rob_move_left_from_room_(3,3)_to_room_(3,2)_box1_is_at_(2,1)
7	Rob_push_box2_left_from_room_(3,1)_to_room_(3,0)_and_Rob_move_left_from_room_(3,2)_to_room_(3,1)_box1_is_at_(2,1)
8	Rob_push_box1_up_from_room_(2,1)_to_room_(1,1)_and_Rob_move_up_from_room_(3,1)_to_room_(2,1)_box2_is_at_(3,0)
9	Rob_push_box1_up_from_room_(1,1)_to_room_(0,1)_and_Rob_move_up_from_room_(2,1)_to_room_(1,1)_box2_is_at_(3,0)
10	Rob_move_from_room_(1,1)_to_room_(1,0)_and_box1_at_room_(0,1)_box2_at_(3,0)
11	Rob_move_from_room_(1,0)_to_room_(0,0)_and_box1_at_room_(0,1)_box2_at_(3,0)

Additional Comments:

If it is required to solve the problem for a Robert moving more than two boxes in a room or irregular/non symmetric size of room , it needs to enumerate and hard coding all the subset of proposed boxes and robot location plus their constraints when robot pushing boxes and robot moving in the room which increasing the search spaces exponentially. A good heuristic function will increase its search efficiency. The multi path pruning searcher has been used for this STRIP planner, multiple solutions have been found.