AI Planning

Adil El Hakouni
Youssef Lamzaouak
Ecole des Mines de Saint-Etienne

1 Modeling Plans in PDDL

1.1 Problem Description

we have two (or many) different rooms and two (or many) balls, the aim is that the robot which has two arms moves the balls from a place to another according to a predefined goal.

1.2 Question 1

a- what are the initial and final states:

1. The initial state: The robot is in the room A, both of its arms are free and both the balls (ball 1 and ball 2) are in the same room A
2. The goal state: The goal is that the robot moves the two balls from room A to room B.

```
(define (problem pb1)
    (:domain gripper)
    (:objects roomA roomB Ball1 Ball2 left right)

    (:init
        (room roomA)
        (room roomB)
        (ball Ball1)
        (ball Ball2)
        (gripper left)
        (gripper right)
        (atRobby roomA)
        (free left)
        (free right)
        (at Ball1 roomA)
        (at Ball2 roomA))

    (:goal (and (at Ball1 roomB) (at Ball2 roomB)) )
)
```

b- The possible actions: our agent has three possible actions:

1-move: if a robot is in the from-room (room A in the initial state), it can move to the other room.

2- pick: if the robot is in a room, it's gripper is free, and there exists a ball in the same room, Robby can pick the ball.so it's no longer exist in the room and it's gripper is no longer free.

3- drop: The robot can drop the ball in a room so that it frees it's gripper.

```
INFOS: Running planner, maximum memory: 1á794MB
INFOS: Expanding graph
INFOS: Goals not possible with 1 steps
INFOS: Expanding graph
INFOS: Goals not possible with 2 steps
INFOS: Expanding graph
INFOS: Extracting solution
INFOS: Plan found with 3 steps
INFOS: Planning took 281ms ( 0s )
INFOS: Total memory used: 123MB
INFOS: Plan found:
pick(ball1,rooma,right)
pick(ball2,rooma,left)
move(rooma,roomb)
drop(ball1,roomb,right)
drop(ball2,roomb,left)

INFOS: Plan length: 5
```

c- the other problems pb.2 and pb.3 for example are more complicated versions of the first problem (more balls and more different states).

2 GraphPlan

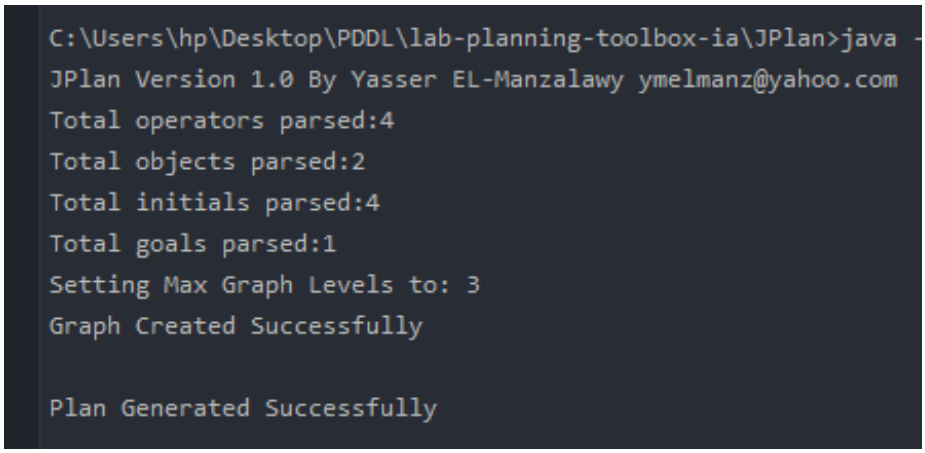
2.1 Question 1

a-The pb1.txt file describes the problem of blocksworld for a given initial state and goal state. in our case there exist two blocks (a) and (b) and both of them are clear and on the table, goal is that the block (a) should be on the block (b).it is called 'the problem definition".

The blocksworld.txt contains the possible actions (pick up, put down, stack and unstack) related to the problem described in the file pb1.txt.it is called "the domain definition".

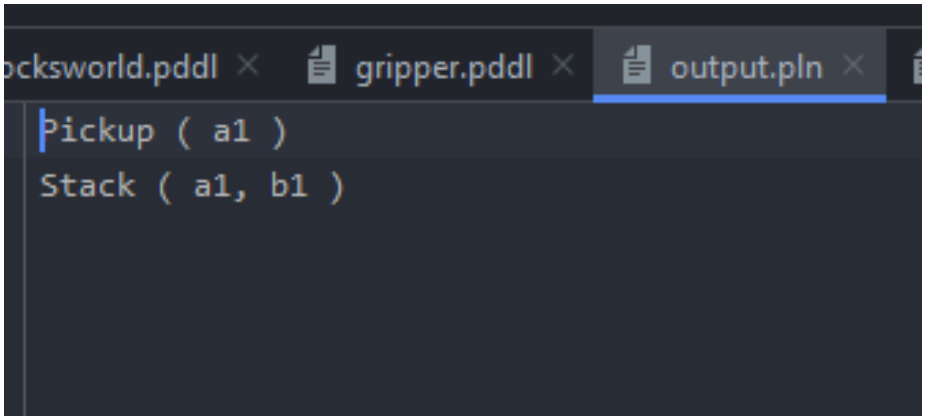
b-The meaning of the three lines between []: the first line contains the preconditions of the action the second one contains the positive effects of the action. the third line contains the negative effects of the action.

c-We execute JPlan with the default depth 3 using the following command : java -classpath ./lib/jplan.jar JPlan examples/BlocksWorld/blocksworld.txt examples/BlocksWorld/pb1.txt 3 And the result is :



2.2 Question 2

a-Yes the graph is correct since to put a1 on b1 ,we should first pick a1 and then put it on b1.the output.pln gives the following results:



b-Since we have made 2 actions to achieve our goalstate,then the depth is 2..

d-we can interpret the GraphPlan in output.gp by saying that as we could see the size of both the Action layer and the proposition layer increased as long

as we move toward the solution.AT first the proposition layer contained just Ontable(a1),Ontable(b1),Clear(a1);Clear(b1) but by the end the proposition layer contained lot of states because eacu time we add to our first preconditions ,the effects that the actions in our action layer will have.For example the second Proposition layer contained in addition t the first proposition layer Holding(a1)and Holding(a2) and these are the effects of the actions Pickup(a1)and Pickup(a2).We can see also the no op and the Mutex in both Proposition Layer and Action layer .For instance Pickup(b1) was Mutex with noopOntable(b1)and noopClear(b1) and this is because these actions are dependant as we can not pick up b1 if it is not Clear and Ontable.

2.3 Question 3

a-the proposition Layer:is a like a memory that stocks our states since the initial state till the goal state.it starts with our initial state and add as long as we move forward to the goalstate,the effects of our actions.

b-The action Layer:is the set of Actions that we take to move from an intial state toward a goalstate.

c-We can divide the Mutex Expressions to the mutex Actions and the MutexPropositions; for the Mutex propositions,I think defining when propositopn P and Q will not be Mutex if they are produced by a pair of independent actions or by a single action. Mutex Actions :Two Actions a1 and a2 in Action layer Aj are Mutex if a1and a2 are dependant or if a precondition of a1 is Mutex with a precondition of a2.

d-The no op term means the no operation action ,it is an action that carries forward a proposition p from one propositional layer to the next propositional layer.it's aim is to simplify the incompatibility test to determine the Mutex Propositions.

e-the number of Action layers is consistent with the plan as we have two action layers so we find our solution before the maximum number of layers 3.Sometimes ,the given maximum number of actions Layers to find the solution could be superior to depth.like here the max number of action layers is 3 and the depth is 2.

f-no op Clear(a1) is Mutex with Pickup(a1) because there are dependant why the other Mutex is to see if there is other actions than no opClear(a1) that depends on on Pickup(a1).and of course if p and q are Mutex ,then q and p are Mutex so the fact that Pickup(a1) is clearly Mutex with noopClear(a1),we want just to see if there is more.

h- we have tested the planning algorithm wth the depth 5 for example .Here is the result:

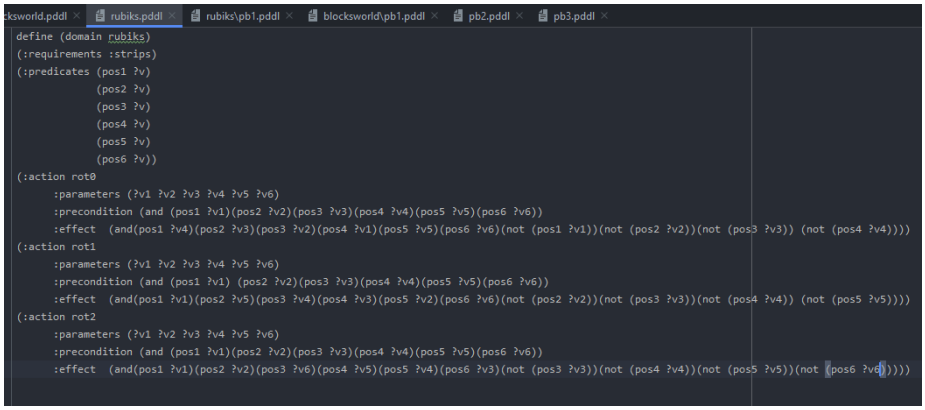
```
JPlan Version 1.0 By Yasser EL-Manzalawy ymelmanz@yahoo.com
Total operators parsed:4
Total objects parsed:2
Total initials parsed:4
Total goals parsed:1
Setting Max Graph Levels to: 5
Graph Created Successfully

Plan Generated Successfully
```

3 Modeling Problems and Domains with Graph Plan

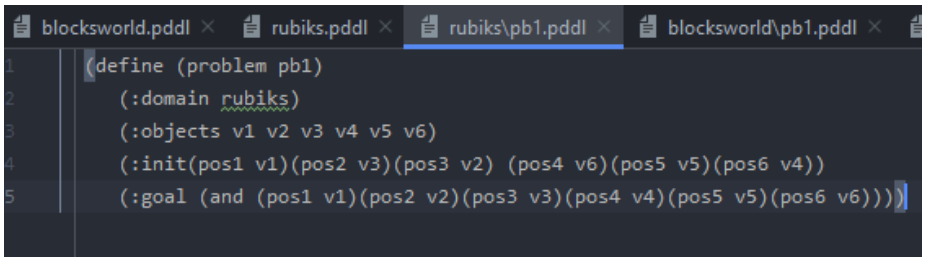
3.1 Question1

We have Modeled the 1D Rubik's Doamin and its three Problems using the PDDL language and Strips Language as you can find in te attached javafile.but we will display here just an example of the first Problem Definition.

A-Domain Definition:



B-Problem Definition:



3.2 Question 2

We have tested the plan it is valid and similar to the one suggested in the Example.Here is the Plan Generated:

```
C:\Users\hp\Desktop\PDDL\lab-planning-toolbox-ia\javagp>java -jar javagp
INFOS: Running planner, maximum memory: 1á794MB
INFOS: Expanding graph
INFOS: Goals not possible with 1 steps
INFOS: Expanding graph
INFOS: Goals not possible with 2 steps
INFOS: Expanding graph
INFOS: Extracting solution
INFOS: Plan found with 3 steps
INFOS: Planning took 2078ms ( 2s )
INFOS: Total memory used: 220,5MB
INFOS: Plan found:
rot1(v1,v3,v2,v6,v5,v4)
rot2(v1,v5,v6,v2,v3,v4)
rot1(v1,v5,v4,v3,v2,v6)

INFOS: Plan length: 3
```