BBM 405 – Fundamentals of Artificial Intelligence – Spring 2020

Homework 3

Goal:

In this assignment you will learn how to model and solve planning problems.

Introduction:

PDDL (the "Planning Domain Definition Language") was designed to standardise planning domain and problem description languages for the International Planning Competitions (IPL). It is based on the STRIPS language developed Stanford Research Institute (SRI) to be used in Shakey the Robot. Besides STRIPS, PDDL contains features from ADL and more. You will need only the STRIPS subset of PDDL in this homework.

There are many planners available. Some recent planners are given in https://ipc2018.bitbucket.io/

In this homework you will use a planner that is accessible on http://editor.planning.domains/.

Syntax of PDDL

Planning problems represented in PDDL are separated into two parts: the domain and the problem definition.

The domain definition lists the available predicates and actions. A problem definition describes the objects, initial state and goal criteria for a particular scenario. There can be multiple problem scenarios on the same domain.

Please refer to http://users.cecs.anu.edu.au/~patrik/pddlman/writing.html for the details of writing PDDL files. More examples can be found in the listed references.

The domain file has the following format

```
(define (domain <DOMAIN_NAME>)
 (:requirements [:strips] [:equality] [:typing] [:adl])
 (:predicates (<PREDICATE_1_NAME> ?<P1> ... ?<PN>)
              (<PREDICATE_2_NAME> ?<P1> ?<PN>)
              ...)

 (:action <ACTION_1_NAME>
   [:parameters (?<P1> ... ?<PN>)]
   [:precondition <PRECOND>]
   [:effect <EFFECT>]
  )

<OTHER ACTIONS>
)
```

The problem file has the following format:

```
(define (problem <PROBLEM_NAME>)
  (:domain <DOMAIN_NAME>)
  (:objects O1 O2 ... ON)
  (:init A1 A2 ... AN)
  (:goal CONDITION)
  )
```

Example: Blocksworld from http://csci431.artifice.cc/notes/pddl.html

Domain file:

```
(define (domain blocksworld)
  (:requirements :strips)

 (:predicates (clear ?x)
              (on-table ?x)
              (holding ?x)
              (on ?x ?y))

 (:action pickup
          :parameters (?ob)
          :precondition (and (clear ?ob) (on-table ?ob))
          :effect (and (holding ?ob) (not (clear ?ob)) (not (on-table ?ob))))

  (:action putdown
           :parameters (?ob)
           :precondition (and (holding ?ob))
           :effect (and (clear ?ob) (on-table ?ob)
                        (not (holding ?ob))))

  (:action stack
           :parameters (?ob ?underob)
           :precondition (and  (clear ?underob) (holding ?ob))
           :effect (and (clear ?ob) (on ?ob ?underob)
                        (not (clear ?underob)) (not (holding ?ob))))

  (:action unstack
           :parameters (?ob ?underob)
           :precondition (and (on ?ob ?underob) (clear ?ob))
           :effect (and (holding ?ob) (clear ?underob)
                        (not (on ?ob ?underob)) (not (clear ?ob)))))
```

Problem file:

```
(define (problem blocksworld-prob1)
  (:domain blocksworld)
  (:objects a b)
  (:init (on-table a) (on-table b) (clear a) (clear b))
  (:goal (on a b)))
```

Plan found

(pickup a) (stack a b)

```
(:action pickup
    :parameters (a)
    :precondition
      (and
        (clear a)
        (on-table a)
      )
    :effect
      (and
        (holding a)
        (not
          (clear a)
        )
        (not
          (on-table a)
        )
      )
  )
```

```
(:action stack
    :parameters (a b)
    :precondition
      (and
        (clear b)
        (holding a)
      )
    :effect
      (and
        (clear a)
        (on a b)
        (not
          (clear b)
        )
        (not
          (holding a)
        )
      )
  )
```

Task1 (30 points)

Your first task in this homework is to apply PDDL for the `gripper` problem.

In this problem, a robot moves between several rooms and can pick-up balls and put them down. For this task you will use the descriptions by Manuela Veloso in
http://www.cs.toronto.edu/~sheila/384/w11/Assignments/A3/veloso-PDDL_by_Example.pdf

First, using the problem domain given in that page try to obtain the same resulting plan.

Then, your only duty is to define two additional problem scenarios and report the plans found.

Provided Files :
gripper-domain.pddl
gripper-problem-1.pddl

Required files:
gripper-problem-2.pddl
gripper-plan-2.pdf
gripper-problem-3.pddl
gripper-plan-3.pdf

Task2 (45 points)

Next, you will apply planning to Wumpus world.

Use the domain file (3rd try) written in
http://users.cecs.anu.edu.au/~patrik/pddlman/wumpus.html.

First, using the problem domain given in that page try to obtain the same resulting plan.

Then, generate three other problem scenarios by yourself and report the resulting plans. Try to create plans as challenging as possible!

Provided Files :
wumpus-domain.pddl
wumpus-problem-1.pddl

Required files:
wumpus-problem-2.pddl
wumpus-plan-2.pdf
wumpus-problem-3.pddl
wumpus-plan-3.pdf
wumpus-problem-4.pddl
wumpus-plan-4.pdf

Task3 (25 points)

Finally, you will write your own PDDL files for a new problem.

Choose a problem from the following page other than the examples that we discussed, and write the PDDL files. Note that, the examples given in that page are written in STRIPS and therefore have slight notational differences.

https://stripsfiddle.herokuapp.com/

Required files

task3-domain.pddl

task3-problem.pddl

task3-plan.pdf

Submission

Send your homework in a zip file named as BBM405_HW3_Name_Surname.zip
Where Name and Surname is yours ☺

You should submit the files shown as required in each task.
<task-name>.plan.pdf files will show the generated plans.

Please send your homeworks to pinar@cs.hacettepe.edu.tr
In your header please include BBM405-HW3

Good luck

References:

http://www.cs.toronto.edu/~sheila/384/w11/Assignments/A3/veloso-PDDL_by_Example.pdf

https://www.cs.toronto.edu/~sheila/2542/s14/A1/introtopddl2.pdf

http://users.cecs.anu.edu.au/~patrik/pddlman/writing.html

http://csci431.artifice.cc/notes/pddl.html

https://homepages.inf.ed.ac.uk/mfourman/tools/propplan/pddl.pdf

https://github.com/pellierd/pddl4j/wiki/A-tutorial-to-start-with-PDDL

https://en.wikipedia.org/wiki/Planning_Domain_Definition_Language