Demonstration on Planning Domain Definition Language (PDDL)

COURSE: CS60045

By : Briti Gangopahyay, Sudipa Mandal and Sumanta Dey

Pallab Dasgupta
Professor,
Dept. of Computer Sc & Engg

Planning Domain Definition Language Introduction

PDDL is a standardise planning domain and problem description language for “classical” planning tasks. It was developed mainly for International Planning Competition (IPC) series (https://ipc2018.bitbucket.io/)

Components of a PDDL planning task:

- **Objects:** Things in the world that interest us.
- **Predicates**: Properties of objects that we are interested in; can be True or False.
- **Initial state:** The state of the world that we start in.
- **Goal specification:** A state of the world that we want to reach.
- **Actions/Operators:** Ways of changing the state of the world.

Parts of a PDDL Domain Definition

A PDDL definition consists of two parts: The *domain* and the *problem* definition. A domain file, typically named domain.pddl (e.g.: blocksworld.pddl), has the following structure:

```
;; domain.pddl for blocksworld
```

→ Comments

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

→ Name of the domain

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

→

```
<PDDL code for predicates>
(:predicates (PREDICATE_1_NAME ?A1 ?A2 ... ?AN)
             (PREDICATE_2_NAME ?A1 ?A2 ... ?AN)
             ...)
```

Parts of a PDDL Domain Definition

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

```
<PDDL code for first action>
(:action ACTION_1_NAME
[:parameters (?P1 ?P2 ... ?PN)]
[:precondition PRECOND_FORMULA]
[:effect EFFECT_FORMULA]
)
```

...

...

```
 (: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)))))
```

```
<PDDL code for last action>
```

Parts of a PDDL Problem Definition

```
;; problem file: blocksworld-prob1.pddl
(define (problem blocksworld-prob1)
        (:domain blocksworld)
        (:objects a b)
        (:init (on-table a) (on-table b) (clear a) (clear b))
        (:goal (and (on a b))))
```

```
(define (problem PROBLEM_NAME)
(:domain DOMAIN_NAME)
(:objects OBJ1 OBJ2 ... OBJ_N)
(:init ATOM1 ATOM2 ... ATOM_N)
(:goal CONDITION_FORMULA)
)
```

Recent versions of PDDL (PDDL 2.1) have a mechanism for expressing the measure of plan quality (metric) that planners should try to optimise. To specify action costs, it is necessary to add a "fluent" that keeps track of the cost.

```
(:functions
  (total-cost)
 )
```

```
:effect (and ... (increase (total-cost) C))
(= (total-cost) 0)
```

```
(define (problem ..)
.
(:metric minimize (total-cost)))
```

PDDL Demonstration Pre-requisites

- Online Editor and Solver
  - http://planning.domains/
  - Online PDDL editor: http://editor.planning.domains (may not always open)
  - Also plugin can be installed with vs code. Follow the below steps:
    - Download and Set up VS Code https://code.visualstudio.com/download
    - Go to Extensions → Search for PDDL → Install

- (Fast Forward) FF planner
  - Download FF v2.3 from https://fai.cs.uni-saarland.de/hoffmann/ff.html
    - Build the planner by typing make in Terminal which produces an executable named ff
    - Execute the PDDL files using the command "ff.exe -o <domain file> -f <problem file>"

Modelling Planning Problems in PDDL

1) He who must not be named (Lord Voldemort) has regained power and is terrorizing the Wizarding World. In order to destroy him Harry Potter, who is at Hogwarts, must first destroy the last two Horcruxes (Parts of Voldemort's soul). He also needs three magical objects (The Sword of Godric Gryffindor, Basilisk's fang, and The Elder Wand) in order to do so. For travelling from one location to another Harry uses his broom which lies at the room of requirements at Hogwarts. Salazar Slytherin's Locket is at Ministry of Magic which can be destroyed using The Sword of Godric Gryffindor (Hidden at Forest of Dean). The second Horcrux, Helga Hufflepuff's cup is at a Gringott's vault. The cup can be destroyed with Basilisk's fang in the Chamber of Secrets. Finally Harry must retrieve the Elder Wand from Olivander's to kill Lord Voldemort at the Forbidden Forest.

1) Model this problem in PDDL and check if there exists a plan to destroy Salazar Slytherin's Locket?
2) Can Harry kill Lord Voldemort?

Modelling Planning Problems in PDDL

2) John Snow, the king in the north, needs Longclaw, the Valyrian steel sword to kill the Night's King. He is at the Wall, ready for the war. But the sword is with Daenerys Targaryen, the mother of the dragons who is at Dragonstone and needs to bring the letter of Ser Jorah to Dragonstone from the Wall. It is impossible to travel from Dragonstone to the Wall unless Daenerys uses her dragon which can fly.

Model this problem in PDDL and check if there exists a plan such that John Snow can kill the Night's king?

Water Jug Problem 1

You are given a 4 liter jug and a 3 liter jug where. Both the jugs are initially full. The jugs don't have markings to allow measuring smaller quantities.

Goal: You have to measure exactly one liter of water and it should be in the second jug.

The operations you can perform are:

- Empty a Jug
- Pour water from one jug to the other until one of the jugs is either empty or full.

Water Jug Problem 2

You have three jugs with a capacity of eight, five and three liters, respectively. At the start the largest jug is completely filled rest of the jugs are empty.

Goal: You have to divide the contents in two, ie. four liters in two jugs.

The operations you can perform are:

- Pour water from one jug to the other until one of the jugs is either empty or full.

Resources

- PDDL Tooling Lectures : https://www.youtube.com/watch?v=XW0z8Oik6G8
- Writing Planning Domains and Problems in PDDL : http://users.cecs.anu.edu.au/~patrik/pddlman/writing.html
- PDDL The Planning Domain Definition Language : https://courses.cs.washington.edu/courses/cse473/06sp/pddl.pdf
- Planning problems in Fast Forward : http://fai.cs.uni-saarland.de/hoffmann/ff-domains.html
- PDDL using Fast Downward : http://csci431.artifice.cc/notes/pddl.html
- Fast Forward Homepage : https://fai.cs.uni-saarland.de/hoffmann/ff.html