

Problem Set: PDDL Modelling

Introduction

In this problem set, your task is to model a selection of planning problems from a given description, as well as from your own experience. The language you should use is PDDL 2.1¹ and PDDL 2.2², with the exception that you should *not* use derived predicates. Please read over these referenced documents to teach yourself PDDL. There are also some useful online tutorials available about PDDL.

To make the task of modelling easier, the end of this problem set contains an optional set of instructions for using an online PDDL editor that gives you syntax highlighting, some autocomplete features, etc. You are not required to use this, but it may help.

There are many ways to model any problem. You should strive to be as efficient and clear as you can with your models, and your mark will reflect how successful you are.

There are two parts to this problem. In the first part, you'll be asked to model a Star Trek³ scenario in PDDL. This is meant to get you warmed up with PDDL. In the second exercise, you'll be asked to model some real-world scenario of your choosing in PDDL.

Star Trek: Nairb's Mission to Levinia

Part I

In this part of the problem set you, you'll model a Star Trek scenario, and then modify it to add in additional complexity. In the first part, we give a number of hints on how you can model the elements of the problem, and in the second part you must come up with the encoding yourself.

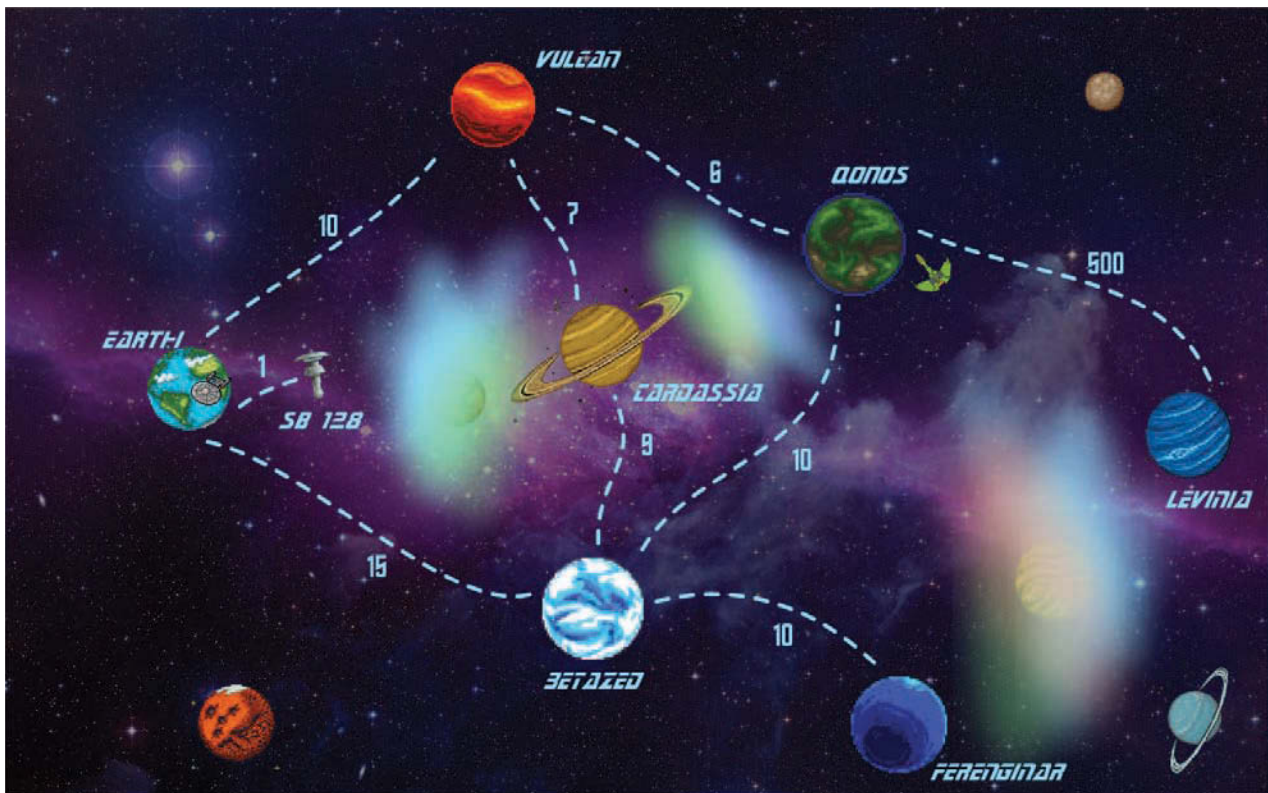
¹ Fox, M.; Long, D. (2003). "PDDL2.1: An Extension to PDDL for Expressing Temporal Planning Domains" (PDF). *Journal of Artificial Intelligence Research (JAIR)* 20: 61–124.

² Edelkamp, S.; Hoffmann, J. (2003). "PDDL2.2: The Language for the Classical Part of the 4th International planning Competition" (PDF). Technical Report No. 195 (Institut für Informatik)

³ For background material, which is optional but highly recommended by the course staff, please watch seasons 1-7 of Star Trek: The Next Generation before completing this problem set. ;-)

Scenario: Planet Levinia needs your help! As a freshly minted Starfleet Academy graduate in charge of your own galaxy-class starship, the USS Enterprise, your first mission has been assigned by Admiral Smailliw Nairb himself. You are to travel to the planet Levinia, and rescue a group of civilians that are in danger from temporal anomalies. There is a deadline though, after which you will be unable to rescue the population. Given the world-class training you received in course 16.412 at Starfleet Academy, you know exactly what to do: build a PDDL model! A planner should be able to find a solution and help you direct your crew to success.

Levinia is quite far away. Your new starship isn't finished being built yet, and so it can only travel at impulse speed. Please see the following chart for the travel times (in hours) to travel between different planets at impulse. Note that you can't travel directly between all planets due to other temporal anomalies.



(Map not to scale!)

To get to Levinia on time and save its population, you'll need to build a warp drive. Once built, you'll be able to travel 10 times faster than you could at impulse speed. For example, warping between Earth and Vulcan would take 1 hour instead of 10 hours.

However, warp drive supplies are scattered around the galaxy. Once obtained, to build a warp drive, you need to attach a plasma conduit on top of a plasma injector, which

must be attached on top of a warp coil, which must be attached on top of some dilithium. Once everything is attached in this manner, you'll be able to enable your warp drive to move faster through the galaxy, delivering medical supplies to Levinia.

You'll need to get to Levinia quickly, because after 400 hours, the temporal anomalies may grow too strong and you won't be able to save them anymore (plan failure).

In this first example, we're giving you a bit of PDDL modeling help. **You must use these types and actions (and only these ones) in your initial model.**

Please use the following types:

Type	Superclass type
location, locatable	object
ship, supply	locatable
plasmaconduit, plasmainjector, warpcoil, dilithium, medicalsupply	supply

This domain has the following locations:

Place	Type
earth	location
sb128 (Star Base 128)	location
vulcan	location
betazed	location
cardassia	location
qonos	location
ferenginar	location
levinia	location

This domain has the following objects, which are initially at these locations:

Thing	Type	Initial location
enterprise	ship	earth
plasmaconduit1	plasmaconduit	vulcan
plasmainjector1	plasmainjector	betazed
warpcoil1	warpcoil	qonos
dilithium1	dilithium	ferenginar
medicalsupply1	medicalsupply	sb128

You may use the following actions:

Action Name	Parameters & types	Description	Time (hr)
travel-impulse-speed	?s - ship, ?l1 - location, ?l2 - location	Ship ?s moves from ?l1 to ?l2 at impulse speed	(See map)
travel-warp-speed	?s - ship, ?l1 - location, ?l2 - location	Ship ?s moves from ?l1 to ?l2 at warp speed	(See map, divided by 10)
deploy-crew	?s - ship, ?l - location	Deploys the ship's away-team to ?l at the start of the action.. Once this action completes, the away team is back on board. So, this action should completely temporally encompass any activities involving the away team.	>= 4
beam-up-supplies	?s - ship, ?l - location, ?p - supply	Ship ?s at location ?l beams up supplies ?p. Once the action is complete, the supplies are onboard the ship. Requires the away team.	1
attach-above	?s - ship, ?p1 - supply, ?p2 - supply	Attach supply ?p1 above ?p2, both of which must be onboard ship ?s	1
detach-above	?s - ship, ?p1 - supply,	Detach supply ?p1 from	1

	?p2 - supply	the top of ?p2, both of which are on ship ?s	
enable-warp-drive	?s - ship, ?pc - plasmaconduit, ?pi - plasmainjector, ?wc - warpcoil, ?di - dilithium	All of these supplies must be onboard ship ?s and attached in the proper configuration. This action powers up the warp drive, enabling flight at warp speed afterwards	3
save-people	?s - ship, ?l - location, ?m - medicalsupply	Save the people at ?l, who are in danger! Requires the away team.	10

We're not restricting the set of predicates you use in your modeling, so add any predicates, functions, or other PDDL features as appropriate.

Once you're created a PDDL domain and problem file, run it with the following command in the VM. We'll be using an off-the-shelf, freely-available state-of-the-art temporal planner developed by a research group called Optic (if you're curious, [here](#) is the homepage).

```
optic-clp -N <domain file> <problem file>
```

This will print the plan out to the command line, along with other information. We've created a convenient tool that can parse this output and generate a pretty web-based visualization. To use it, pipe the input into the planviz.py script:

```
optic-clp -N <domain file> <problem file> | planviz.py
```

This will open up a new Firefox window showing a visualization across time of the plan.

Please model this PDDL domain, creating a PDDL domain and problem file. Then, run the optic planner to get a successful plan and visualize it.

Part II

New intelligence has been received, and there are more constraints than we originally anticipated! You'll need to update your PDDL model to account for the new information (new actions, fluents, etc):

- We have discovered that the target location on Levinia is larger than previously thought. You need to rescue civilians from a 3-by-3 city block: there are 9 locations in total that *all* need to be visited and cleared for victims, and they are connected in the 4 normal directions (north, south, east, and west). Traversing in each of these directions takes your away team 1 hour. Make sure your away team visits every one of the locations with a medkit in time. When your away team lands, they may beam down to any location on the 3-by-3 grid.
- There are storms and earthquakes being reported for Vulcan. Fortunately, our scientists have been able to pinpoint the times that are not safe for a surface mission. Make sure that your away team does not visit Vulcan until after 150 hours, when scientists predict that all storms will be over.

Modeling Another Domain

For the second part of this problem set, your task is to come up with a domain description and encoding from scratch. You are encouraged to draw inspiration from your own research or interests, and should strive to use the features of PDDL 2.1 / 2.2 as much as you can (part of your grade will depend on this!). PDDL domains have been used to model a vast array of problems, even ones outside of robotics (such as genetics!), so feel free to be creative.

In grading this question, we will be looking for elements such as (1) how novel and creative your problem and encoding are for the task of automated planning; (2) how efficient and intuitive your encoding is; and (3) how accurately your PDDL model captures the problem description.

For this problem, we ask that you provide (1) the description of the domain and problem; (2) the PDDL for the domain and at least two problem instances; and (3) an answer to each of the following questions:

- What are the aspects that were easy or difficult to model with PDDL? How would you change the language so it was easier?

- How well do you think your approach will scale to larger problem sizes?
- How well were you able to solve the problem(s) with the Optic planner?

Deliverables

For each part of this problem, you should provide the domain and problem PDDL files, and they should include sufficient comments to explain what is going on and how they work.

Please create a .zip archive, containing:

1. Your PDDL domain and problem for the first Star Trek scenario
2. Your PDDL domain and problem for the second, modified Star Trek scenario
3. Your PDDL domain and problem for the domain you modeled
4. The answers to all of the above questions, preferably in .PDF format

MIT OpenCourseWare
<https://ocw.mit.edu>

16.412J / 6.834J Cognitive Robotics
Spring 2016

For information about citing these materials or our Terms of Use, visit: <https://ocw.mit.edu/terms>.