Computer and Information Sciences
University of Strathclyde

Lab 1 – Introduction to PDDL with Switches

Aim

This exercise is an introduction (or refresher) to modelling planning problems using the Planning Domain Definition Language (PDDL). By following this tutorial you will learn the basics of PDDL, create a model for various levels of the Simple Switches domain, and generate plans using an online solver. **This lab will run for more than one week, so do not worry if you do not complete it in the first session.**

Tools and software required:

Everything can be run online using http://editor.planning.domains

However, we will use Visual Studio Code (installed in the lab computers). You can install VS code on your own computers and also search for the PDDL plug in:

https://marketplace.visualstudio.com/items?itemName=jan-dolejsi.pddl

Other references:

The PDDL wiki, which includes a reference and guide: https://planning.wiki

An Introduction to the Planning Domain Definition Language. Haslum, P., Lipovetzky, N., Magazzeni, D.,

The Online Editor

While we will be using VS Code, you do have the option to use the fully online editor.

http://editor.planning.domains

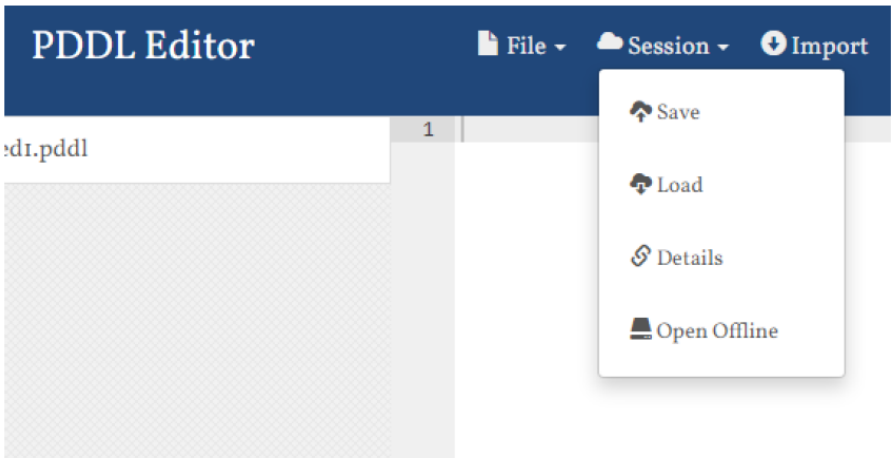
Sessions

The online editor allows you to save and share sessions. This is a useful way to work together, or to share read-only copies of your work when asking for help or giving examples.

- Saving a Read only session will not affect the original session.
- Saving a Read/Write session will overwrite the session with your current workspace.
- Be careful to share the correct link!

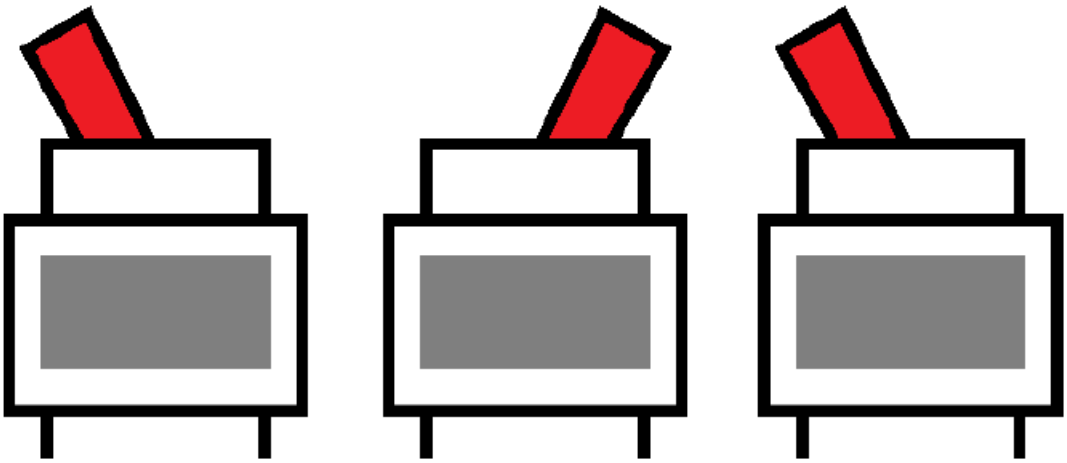
Modified September 2022
By Dr Andrew Abel, based on Dr Michael Cashmore

Computer and Information Sciences
University of Strathclyde



Switches

Your task is to create a planning model that executes a set of simple actions to toggle switches. You can have different initial states (on or off), and a specific goal in mind (i.e. all on).



The main components of a planning problem are:

- A set of things that make up the world (the objects)
- The properties that are used to describe the state of those things (the predicates)
- A description of how the world behaves and the capabilities of the agent (the actions)
- A description of the initial situation (the initial state)
- A description of the desired situation (the goal)

Switches

http://editor.planning.domains/#read_session=jfespcjFc3

Using the link above take a look at the domain and problem PDDL files in the editor. Note that the predicates and actions are described in the domain file. The objects, their initial situation, and the goal are described in the problem file.

Extend the domain and problem so they describe the following scenario:

1. There are three switches, all initially off.

2. Switches can be turned on and off again.

3. The goal is to turn all three switches on.

The output should be a plan listing the actions taken to reach the goal state

Tricky Switches

Update the problem so that:

1. There are five switches.

2. A switch can only be switched on if it has a neighbour that is already on.

3. The five switches are in a row so that each switch has two neighbours, except the two at the ends of the row which only have one.

4. The five switches are in initial positions: {off, off, on, off, off}.

You may have to create a new predicate to describe that two switches are neighbours. For example:

```
(neighbours 1 ?s1 ?s2 - switch)
```

How can you test your plan? How do you know that the output is valid? You can implement the plan yourself using pen and paper!

More Challenging Switches

Update the problem so that:

1. There are ten switches in a row.
2. The ten switches are in initial positions: {on, off, on, off, off, on, off, off, on, off}.
3. A switch can only be switched **on** if it has **exactly one** neighbour that is already on.

Test your output manually!

Computer and Information Sciences
University of Strathclyde

Counting Switches

Update the problem so that:

1. There are ten switches in a row.
2. The ten switches are in initial positions: {on, off, on, off, off, on, off, off, on, off}.
3. A switch can only be switched on if it has exactly one neighbour that is already on.
4. The goal is to have exactly eight switches on - but it can be any eight.