

# PDDL

---

This assignment is **individual**.

**Deadline:** The deadline is **October 23, 2020, 11:59pm** and it is strict.

**Submission:** Please, submit your solution to Exercise 1.1. as a text file delivery-domain.pddl in PDDL1.1 assignment and to Exercise 1.2 as a text file problem.pddl in PDDL1.2 assignment.

---

# 1 DELIVERY TASK

In this exercise, we are going to look into encoding a planning domain using PDDL. Consider a delivery company with distribution centers (DCs) in different cities, each of which services areas of its corresponding city. Long-range (LR) vehicles, such as trains and airplanes, are used to transport parcels between connected DCs, while short-range (SR) vehicles are used to travel from DCs to its corresponding (and connected) areas.

The four basic actions to solve this problem are:

- **Load parcel:** the parcel is loaded into a vehicle;
- **Unload Parcel:** the parcel is unloaded from a vehicle;
- **Travel long:** a long-range vehicle travels between DCs;
- **Travel short:** a short-range vehicle travels on local roads.

## 1.1 DEFINING THE DOMAIN

Download the package **Delivery.zip** from Canvas. Your task is to complete the file **delivery-domain.pddl** to formalize the planning domain. The syntax used in the file is a standardized syntax used in state-of-the-art PDDL solvers, such as in this on-line editor and solver [2]. There are numerous examples of problems encoded in this syntax under the Import tab in this tool. There are also numerous tutorials on this syntax, for instance this one [1]. The relevant tab to explore there is PDDL Background.

Note that the file `delivery-domain.pddl` will only contain the definition of the *domain*. The *problem instance* including the definition of objects in the world, the initial state and the goal specification are given in a separate file. You can find one problem instance for the domain in this exercise in **problem-1.pddl**, illustrated in Figure 1.1.a).

Your task is to complete only the code for each of the four actions, which involves writing the parameters, the precondition and the effect. Note that each action comes with a comment that gives more details than the brief domain introduction above. All the predicates you are allowed to use are already given in the file. You will not need to define any requirements or functions. Figure 1.1a shows the road network and initial state of the objects in the world.

You can use the above mentioned on-line editor and solver [2] to see whether your domain definition allows to find a solution to `problem-1.pddl`. It should.

Please, submit your solution to Exercise 1.1. as a text file `delivery-domain.pddl` in PDDL1.1 assignment. Make sure that your submitted file does not contain syntax errors.

## 1.2 DESIGNING A PROBLEM

In the previous exercise you defined the planning domain and tested it in a given problem instance. Now the task is to adapt `problem-1.pddl` in order to reflect the objects and initial conditions defined in Figure 1.1.b. For convenience, dashed lines indicate the added elements on top of Figure 1.1.a. The goal is to have *parcel1* delivered to *A21*, and *parcel2* to *A12*.

Please, submit your solution to Exercise 1.2 as a text file `problem.pddl` in PDDL1.2 assignment.

## REFERENCES

- [1] A PDDL 2.1 tutorial, <https://www.cs.cmu.edu/afs/cs/project/jair/pub/volume20/fox03a-html/JAIRpddl.html>, Accessed: 2018-09-26

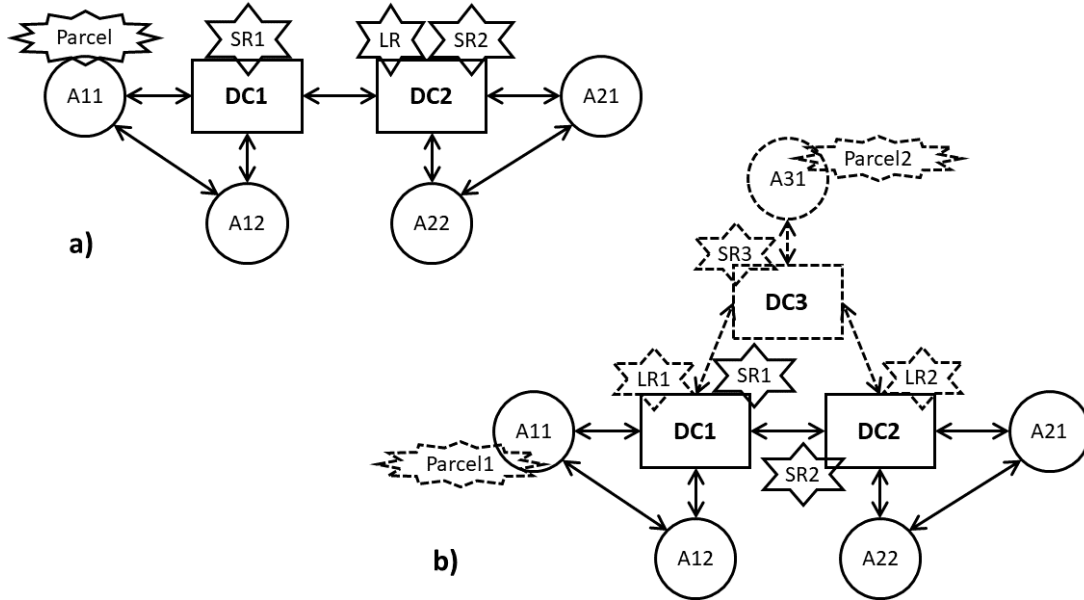


Figure 1.1: Road networks and initial conditions of two delivery problems. DCx stands for distribution center number 'x', LRx and SRx stand for long- and short-range vehicles, respectively, and Axy stands for area 'y' of DCx.

[2] An online PDDL editor and solver, <http://editor.planning.domains/>, Accessed: 2018-09-26