50.021 – Artificial Intelligence

Kwan Hui

Week 10 Coding Homework - Planning

[The following notes are compiled from various sources such as textbooks, lecture materials, Web resources and are shared for academic purposes only, intended for use by students registered for a specific course. In the interest of brevity, every source is not cited. The compiler of these notes gratefully acknowledges all such sources. ]

Due: 5th Apr, 11:59pm

Submission: via eDimension

**Name: Sharryl Seto**

**ID: 1005523**

**Class: Cl03**

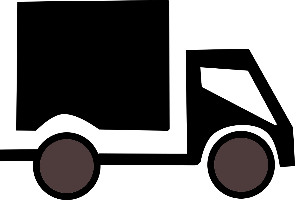
# Logistic Problem: PDDL Domain File

Read through the PDDL lecture notes (“Week 10 - Planning II.pdf”) provided in the lecture. Planning tasks in PDDL are based on two main types of files: a domain file and a problem file. In a domain file, you specify the predicates (similar to facts or propositional variables) and actions. An example is given for the gripper domain and problem discussed in the lecture.

**Tampines**

**Changi**

**Bedok**



Package 1

Consider the above logistic problem. There are three locations with a truck and a package. The truck is able to move from one location to another and load/unload the package. Based on this information, do the following:

1. Define (and submit) a PDDL domain file that describes this type of problem. Hint: the exact locations are not important at this point of time.

For the sake of coherency, I have combined my answers for Question 1 and 2. I have noted that question 1 does not require an obvious goal. **Question 1: (for domain). Question 2: (for problem)**

* **Objects:** The package, the truck, and 3 locations.
  1. **Locations**: Tampines, Bedok, Changi
  2. **Package:** Package1
  3. **Truck:** Truck1

**In PDDL (for problem):**

(:objects locationa locationb locationc

package1

truck1)

* **Predicates:** Is *x* a location? Is *x* a package? ~~Is truck~~ *~~x~~* ~~empty?~~ (assuming truck has no known capacity for packages) ~~Is package~~ *~~x~~* ~~at location~~ *~~y~~*? (since locations are not important)
  1. **LOCATION(x)** - true iff x is a location
  2. **PACKAGE (x)** - true iff x is a package
  3. **TRUCK(x)** - true iff x is a truck
  4. **at-location(x)** - true iff x is a location, and truck is at x
  5. **at-package(x, y)** - true iff x is a package, y is a location, and x is in y
  6. **carry(x, y)** - true iff x is a truck, y is a package, and x contains y

**In PDDL (for domain):**

(:predicates (LOCATION ?x) (PACKAGE ?x) (TRUCK ?x)

(at-location ?x) (at-package ?x ?y) (carry ?x ?y) )

* **Initial state:** Truck is at Tampines, and package is at Bedok. Truck is empty.
  1. LOCATION(Tampines), LOCATION(Bedok), LOCATION(Changi) are true
  2. PACKAGE(Package1) is true
  3. TRUCK(Truck 1), free(Truck 1) are true
  4. at-location(Tampines), at-package(Bedok) are true
  5. Everything else is false.

**In PDDL (for problem):**

(:init (LOCATION Tampines) (LOCATION Bedok) (LOCATION Changi)

(PACAKGE Package1)

(TRUCK Truck1)

(at-location Tampines)

(at-package Bedok) )

* **Goal specification:** Truck to load package at Bedok and deliver to Changi (question 2).
  1. at-package(Package1, Changi) must be true.
  2. We don’t care about everything else.

**In PDDL (for problem):**

(:goal (and (at-package Package1 Changi) ))

* **Actions/Operators:** The truck can move between locations, load and unload a package.
  1. Description: the truck can move from x to y
     + Precondition: LOCATION(x), LOCATION(y), and at-location(x) are true
     + Effect: at-location(y) becomes true. at-location(x) becomes false. All else doesn’t change.
  2. Description: the truck can load x at y.
     + Precondition: PACKAGE(x), LOCATION(y), at-package(x, y), at-location(y) are true.
     + Effect: carry(x, y) becomes true. at-package(x, y) becomes false. All else doesn’t change
  3. Description: the truck can unload x at y.
     + Precondition: PACKAGE(x), LOCATION(y), at-location(y) are true.
     + Effect: carry(x, y) becomes false. at-package(x, y) becomes true. All else doesn’t change

**In PDDL (for domain):**

(:action move :parameters (?x ?y)

:precondition (and (LOCATION ?x) (LOCATION ?y)

(at-location ?x) )

:effect (and (at-location ?y) (not (at-location ?x)) ) )

(:action pick-up :parameters (?x ?y)

:precondition (and (PACKAGE ?x) (LOCATION ?y)

(at-package ?x) (at-location ?y) )

:effect (and (carry ?x ?y) (not (at-package ?x ?y)) ) )

(:action drop :parameters (?x ?y)

:precondition (and (PACKAGE ?x) (LOCATION ?y)

(at-location ?y)

:effect (and (not (carry ?x ?y)) (at-package ?x ?y) ) )

# Logistic Problem: PDDL Problem File

Consider the same logistic problem in Task 1 (Logistic Problem: PDDL Domain File). From this diagram, the truck is at Tampines, the package is at Bedok. The goal is to deliver the package to Changi. Based on this information and your domain file from Task 1, do the following:

1. Define (and submit) a PDDL problem file that describes this specific problem.

**See answers above for objects, init, and goal.**

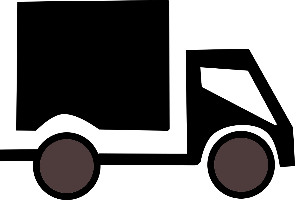
1. Using both your PDDL domain file and problem file, solve for this problem using any PDDL solver (such as http://editor.planning.domains/). What is the solution you obtained?

# Logistic Problem II: PDDL Problem File

**Tampines**

**Changi**

**Bedok**



Package 2



Package 1

Consider the above logistic problem, with a similar domain as that in Task 1 (Logistic Problem: PDDL Domain File). From this diagram, the truck is at Tampines, and there are two packages at Bedok (Package 1) and Changi (Package 2). The goal is to deliver package 1 to Changi and package 2 to Bedok. Based on this information and your domain file from Task 1, do the following:

1. Define (and submit) a PDDL problem file that describes this specific problem.
2. Using both your PDDL domain file and problem file, solve for this problem using any PDDL solver (such as http://editor.planning.domains/). What is the solution you obtained?