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**

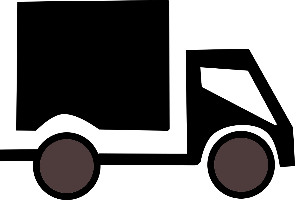
# Logistic Problem: PDDL Domain File

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.

**Predicates:** Is *x* a location? Is *x* a package? Is truck *x* empty? ~~Is package~~ *~~x~~* ~~at location~~ *~~y~~*? (since locations are not important)

* **location(x)** - true iff x is a location
* **package (x)** - true iff x is a package
* **truck(x)** - true iff x is a truck
* **truck-at-loc(x)** - true iff x is a location, and truck is at x
* **package-at-loc(x, y)** - true iff x is a package, y is a location, and x is in y
* **free(x)** - true iff x is a truck, and is free to carry packages (has capacity)
* **carry(x, y)** - true iff x is a truck, y is a package, and x contains y

**In PDDL:**

(:predicates (location ?l)

(package ?p)

(truck ?t)

(truck-at-loc ?l)

(package-at-loc ?p ?l)

(free ?t)

(carry ?p ?t))

**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 truck-at-loc(x) are true
     + Effect: truck-at-loc (y) becomes true. truck-at-loc(x) becomes false. All else doesn’t change.
  2. Description: the truck can load x at y with z.
     + Precondition: package(x), location(y), truck(z), package-at-loc(x, y), truck-at-loc(y) are true.
     + Effect: carry(x, y) becomes true. package-at-loc(x, y), and free(z) becomes false. All else doesn’t change.
  3. Description: the truck can unload x at y.
     + Precondition: package(x), location(y), truck-at-loc(y) are true.
     + Effect: carry(x, y) becomes false. package-at-loc(x, y), and free(z) becomes true. All else doesn’t change.

**In PDDL (for domain):**

(:action move

:parameters (?from ?to)

:precondition (and (location ?from) (location ?to) (truck-at-loc ?from))

:effect (and (truck-at-loc ?to)

(not (truck-at-loc ?from))))

(:action load

:parameters (?package ?location ?truck)

:precondition (and (package ?package) (location ?location) (truck ?truck)

(package-at-loc ?package ?location) (truck-at-loc ?location) (free ?truck))

:effect (and (carry ?package ?truck)

(not (package-at-loc ?package ?location))

(not (free ?truck))))

(:action unload

:parameters (?package ?location ?truck)

:precondition (and (package ?package) (location ?location) (truck ?truck)

(carry ?package ?truck) (truck-at-loc ?location))

:effect (and (package-at-loc ?package ?location)

(free ?truck)

(not (carry ?package ?truck))))

# 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.

**Objects:** The package, the truck, and 3 locations.

* + **Locations**: tampines, bedok, changi
  + **Package:** package1
  + **Truck:** truck1

**In PDDL:**

(:objects tampines bedok changi package1 truck1)

**Initial state:** Truck is at Tampines, and package is at Bedok.

* + location(tampines), location(bedok), location(changi) are true
  + package(package1) is true
  + truck(truck1) and free(truck1) is true
  + truck-at-loc(tampines), package-at-loc(bedok) are true
  + Everything else is false.

**In PDDL:**

(:init (location tampines)

(location bedok)

(location changi)

(package package1)

(truck truck1)

(truck-at-loc tampines)

(free truck1)

(package-at-loc package1 bedok))

**Goal specification:** Truck to load package at Bedok and deliver to Changi

* package-at-loc(package1, changi) must be true. We don’t care about everything else.

**In PDDL:**

(:goal (and (package-at-loc package1 changi)))

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?

**Solution:**

**A screenshot of a computer

Description automatically generated**

**(move tampines bedok)**

(:action move  
 :parameters (tampines bedok)  
 :precondition  
 (and  
 (location tampines)  
 (location bedok)  
 (truck-at-loc tampines)  
 )  
 :effect  
 (and  
 (truck-at-loc bedok)  
 (not  
 (truck-at-loc tampines)  
 )  
 )  
 )

**(load package1 bedok truck1)**

(:action load  
 :parameters (package1 bedok truck1)  
 :precondition  
 (and  
 (package package1)  
 (location bedok)  
 (truck truck1)  
 (package-at-loc package1 bedok)  
 (truck-at-loc bedok)  
 (free truck1)  
 )  
 :effect  
 (and  
 (carry package1 truck1)  
 (not  
 (package-at-loc package1 bedok)  
 )  
 (not  
 (free truck1)  
 )  
 )  
 )

**(move bedok changi)**

(:action move  
 :parameters (bedok changi)  
 :precondition  
 (and  
 (location bedok)  
 (location changi)  
 (truck-at-loc bedok)  
 )  
 :effect  
 (and  
 (truck-at-loc changi)  
 (not  
 (truck-at-loc bedok)  
 )  
 )  
 )

**(unload package1 changi truck1)**

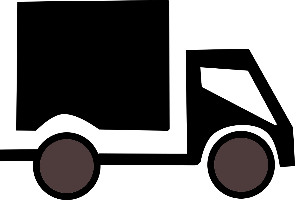
(:action unload  
 :parameters (package1 changi truck1)  
 :precondition  
 (and  
 (package package1)  
 (location changi)  
 (truck truck1)  
 (carry package1 truck1)  
 (truck-at-loc changi)  
 )  
 :effect  
 (and  
 (package-at-loc package1 changi)  
 (free truck1)  
 (not  
 (carry package1 truck1)  
 )  
 )  
 )

# 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.

**Objects:** The package, the truck, and 3 locations.

* + **Locations**: tampines, bedok, changi
  + **Package:** package1, package2
  + **Truck:** truck1

**In PDDL:**

(:objects tampines bedok changi package1 package2 truck1)

**Initial state:** Truck is at Tampines, and package is at Bedok.

* + location(tampines), location(bedok), location(changi) are true
  + package(package1), package(package2) is true
  + truck(truck1) and free(truck1) is true
  + truck-at-loc(tampines), package-at-loc(bedok) are true
  + Everything else is false.

**In PDDL:**

(:init (location tampines)

(location bedok)

(location changi)

(package package1)

(package package2)

(truck truck1)

(truck-at-loc tampines)

(free truck1)

(package-at-loc package1 bedok))

(package-at-loc package2 changi))

**Goal specification:** Truck to deliver package 1 to Changi and package 2 to bedok

* package-at-loc(package1, changi) must be true. We don’t care about everything else.

**In PDDL:**

(:goal (and (package-at-loc package1 changi) (package-at-loc package2 bedok) ))

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?

**A screenshot of a computer program

Description automatically generatedSolution (not the only possible solution):**

**(move tampines changi)**

(:action move  
 :parameters (tampines changi)  
 :precondition  
 (and  
 (location tampines)  
 (location changi)  
 (truck-at-loc tampines)  
 )  
 :effect  
 (and  
 (truck-at-loc changi)  
 (not  
 (truck-at-loc tampines)  
 )  
 )  
 )

**(load package2 changi truck1)**

(:action load  
 :parameters (package2 changi truck1)  
 :precondition  
 (and  
 (package package2)  
 (location changi)  
 (truck truck1)  
 (package-at-loc package2 changi)  
 (truck-at-loc changi)  
 (free truck1)  
 )  
 :effect  
 (and  
 (carry package2 truck1)  
 (not  
 (package-at-loc package2 changi)  
 )  
 (not  
 (free truck1)  
 )  
 )  
 )

**(move changi bedok)**

(:action move  
 :parameters (changi bedok)  
 :precondition  
 (and  
 (location changi)  
 (location bedok)  
 (truck-at-loc changi)  
 )  
 :effect  
 (and  
 (truck-at-loc bedok)  
 (not  
 (truck-at-loc changi)  
 )  
 )  
 )

**(unload package2 bedok truck1)**

(:action unload  
 :parameters (package2 bedok truck1)  
 :precondition  
 (and  
 (package package2)  
 (location bedok)  
 (truck truck1)  
 (carry package2 truck1)  
 (truck-at-loc bedok)  
 )  
 :effect  
 (and  
 (package-at-loc package2 bedok)  
 (free truck1)  
 (not  
 (carry package2 truck1)  
 )  
 )  
 )

**(load package1 bedok truck1)**

(:action load  
 :parameters (package1 bedok truck1)  
 :precondition  
 (and  
 (package package1)  
 (location bedok)  
 (truck truck1)  
 (package-at-loc package1 bedok)  
 (truck-at-loc bedok)  
 (free truck1)  
 )  
 :effect  
 (and  
 (carry package1 truck1)  
 (not  
 (package-at-loc package1 bedok)  
 )  
 (not  
 (free truck1)  
 )  
 )

**(move bedok changi)**

(:action move  
 :parameters (bedok changi)  
 :precondition  
 (and  
 (location bedok)  
 (location changi)  
 (truck-at-loc bedok)  
 )  
 :effect  
 (and  
 (truck-at-loc changi)  
 (not  
 (truck-at-loc bedok)  
 )  
 )  
 )

**(unload package1 changi truck1)**

(:action unload  
 :parameters (package1 changi truck1)  
 :precondition  
 (and  
 (package package1)  
 (location changi)  
 (truck truck1)  
 (carry package1 truck1)  
 (truck-at-loc changi)  
 )  
 :effect  
 (and  
 (package-at-loc package1 changi)  
 (free truck1)  
 (not  
 (carry package1 truck1)  
 )  
 )  
 )