Advance AI
Nishank Singhal
Chapter 11 Exercise

Q 11.2) You have a number of trucks with which to deliver a set of packages. Each package starts at some location on a grid map, and has a destination somewhere else. Each truck is directly controlled by moving forward and turning. Construct a hierarchy of high-level actions for this problem. What knowledge about the solution does your hierarchy encode?

Solution:
A High-level action(HLA) is an action performed to make a task. Each HLA is characterized by one or more possible refinements into a sequence of action which may be HLA or an action with no refinement.

We first need to specify the primitive actions: for movement we have Forward (t), TurnLeft (t), and TurnRight(t) where t is a truck, and for package delivery we have Load(p,t) and Unload (p, t) where p is a package and t is a truck. These can be given PDDL descriptions in the usual way.

The hierarchy can be built in a number of ways, but one is to use the HLA Navigate(t, [x, y]) to take a truck t to coordinates [x,y], and Deliver(t,p) to deliver package p to its destina- tion with truck t. We assume the fluent At(o,[x,y]) for trucks and packages o records their current position [x, y], the predicate Destination(p, $[x', y']$) gives the package's destination.

This hierarchy encodes the knowledge that trucks can only carry one package at a time, that we need only drop packages off at their destinations not intermediate points, and that we can serialize deliveries (in reality, trucks would move in parallel, but we have no representation for parallel actions here). From a higher-level, the hierarchy says that the planner needs only to choose which trucks deliver which packages in what order, and trucks should navigate given their destinations.

Q 11.7) Some of the operations in standard programming languages can be modeled as actions that change the state of the world. For example, the assignment operation changes the con- tents of a memory location, and the print operation changes the state of the output stream. A program consisting of these operations can also be considered as a plan, whose goal is given by the specification of the program. Therefore, planning algorithms can be used to construct programs that achieve a given specification.

a. Write an action schema for the assignment operator (assigning the value of one variable to another). Remember that the original value will be overwritten!
b. Show how object creation can be used by a planner to produce a plan for exchanging the values of two variables by using a temporary variable.

Answer:

A) Write an action schema for the assignment operator (assigning the value of one variable to another). Remember that the original value will be overwritten!

- ➔ Action schema for assignment operator:
  - Action: Assign (var, value)
  - Precondition: True
  - Effect: var = value

This action schema represents an action called Assign, which takes two parameters: var (the variable to be assigned) and value (the value to be assigned to var). The precondition of this action is trivially true (since no condition needs to be satisfied before performing an assignment), and the effect is to update the value of the variable var to the given value.

B) Show how object creation can be used by a planner to produce a plan for exchanging the values of two variables by using a temporary variable.

- ➔ Plan for exchanging the values of two variables using a temporary variable: Assuming we have two variables x and y, the following plan can be generated by a planner that uses object creation to create a temporary variable temp to exchange the values:
  - Step 1: Assign (temp, x)
  - Step 2: Assign (x, y)
  - Step 3: Assign (y, temp)

  The plan first assigns the value of x to temp (using the Assign action schema), then assigns the value of y to x, and finally assigns the value of temp to y. At the end of this plan, the values of x and y have been exchanged.