National University of Singapore
School of Computing

Semester 2, AY2023-24

CS4246/CS5446 AI Planning and Decision Making

Tutorial Week 4: Hierarchical Planning

Guidelines

- You can discuss the content of the questions with your classmates.
- However, everyone should work on and be ready to present ALL the solutions.
- Your attendance is marked in the tutorial and participation noted to award class participation marks.

Package delivery

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. The package can be loaded to the truck and unloaded from the truck.

1) *Forward* action:
   If you model the *Forward* action as a primitive action, what would be the precondition and effect? Assume that addition is defined and available as an operator. State any assumptions you make in the modeling.
   Solution:

   - We need to define a fluent *At(o, [x, y])* that defines the current location (the *x,y*-coordinates) of the object o.
   - Addition is defined and available; hence, we can define the next location of the object using $x + \delta, y + \rho$. Now, we can define the precondition and effects as follows.
   - *Precondition:* $\{$*At(t, [x,y]), Truck(t), Coordinate(x), Coordinate(y)*$\}$
     Effect: At(t, [$(x + \delta), (y + \rho)$*]).*

2) What other primitive actions are needed define the planning problem? (no need to write the PDDL definition)
   Solution:

   - *Turnright(t), Turnleft(t), Load(p, t), Unload(p, t)* where t is the truck and p is the package.

3) Construct a hierarchy of high-level actions for this problem.
Solution:

   - Use the HLA *Navigate(t, [x, y])* to take a truck t to coordinates *[x, y]*.
   - HLA *Deliver(t, p)* to deliver package p to its destination with truck t.
   - *At(o, [x, y])* defines the current location (the *x,y*-coordinates) of the object o.
   - Fluent *Destination(p, [x1, y1])* defines the package's destination.

4) What knowledge about the solution does your hierarchy encode?
Solution:
The hierarchy encodes the following information:
   - Trucks can only carry one package at a time.
   - We need to only drop packages off at their destinations not intermediate points.
   - We can serialize deliveries.
   - 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.

5) What are some shortcomings (in terms of real-life implementation) of the hierarchy defined above?
Solution:
The hierarchy has the following shortcomings:
   - In reality, trucks would move in parallel, but we have no representation for parallel actions in the hierarchy.
   - The solution manadates that one truck can deliver only one package; in real life this is hardly the case.

Air cargo problem

Consider the air-cargo problem that we have seen in the lecture.

$$
\begin{aligned}
&\mathit{Init}(\mathit{At}(C_1,\ \mathit{SFO}) \wedge \mathit{At}(C_2,\ \mathit{JFK}) \wedge \mathit{At}(P_1,\ \mathit{SFO}) \wedge \mathit{At}(P_2,\ \mathit{JFK}) \\
&\quad \wedge \mathit{Cargo}(C_1) \wedge \mathit{Cargo}(C_2) \wedge \mathit{Plane}(P_1) \wedge \mathit{Plane}(P_2) \\
&\quad \wedge \mathit{Airport}(\mathit{JFK}) \wedge \mathit{Airport}(\mathit{SFO})) \\
&\mathit{Goal}(\mathit{At}(C_1,\ \mathit{JFK}) \wedge \mathit{At}(C_2,\ \mathit{SFO})) \\
&\mathit{Action}(\mathit{Load}(c,\ p,\ a), \\
&\quad \text{PRECOND: } \mathit{At}(c,\ a) \wedge \mathit{At}(p,\ a) \wedge \mathit{Cargo}(c) \wedge \mathit{Plane}(p) \wedge \mathit{Airport}(a) \\
&\quad \text{EFFECT: } \neg\, \mathit{At}(c,\ a) \wedge \mathit{In}(c,\ p)) \\
&\mathit{Action}(\mathit{Unload}(c,\ p,\ a), \\
&\quad \text{PRECOND: } \mathit{In}(c,\ p) \wedge \mathit{At}(p,\ a) \wedge \mathit{Cargo}(c) \wedge \mathit{Plane}(p) \wedge \mathit{Airport}(a) \\
&\quad \text{EFFECT: } \mathit{At}(c,\ a) \wedge \neg\, \mathit{In}(c,\ p)) \\
&\mathit{Action}(\mathit{Fly}(p,\ \mathit{from},\ \mathit{to}), \\
&\quad \text{PRECOND: } \mathit{At}(p,\ \mathit{from}) \wedge \mathit{Plane}(p) \wedge \mathit{Airport}(\mathit{from}) \wedge \mathit{Airport}(\mathit{to}) \\
&\quad \text{EFFECT: } \neg\, \mathit{At}(p,\ \mathit{from}) \wedge \mathit{At}(p,\ \mathit{to}))
\end{aligned}
$$

1) Write the successor state axiom for the fluent $At(P_1, SFO)$.

   Solution:

   $$At(P_1, SFO)^{t+1} \Leftrightarrow Fly(P_1, JFK, SFO)^t \vee (At(P_1, SFO)^t \wedge \neg Fly(P_1, SFO, JFK)^t)$$

2) Desribe how you will modify the problem so that each plane can carry only one cargo.

   Solution:
   We need to introduce a fluent $Full(p)$ to indicate the plane is full. Add $Full(p)$ to the add list of $Load(c, p, a)$ and to the delete list of $Unload(c, p, a)$ actions.