1. Write an action schema for grabbing an object, one for carrying a grabbed object from a location to another, and one to go from a location to another.

We need to add another action for dropping an object, otherwise after grabbing an object there is no way of grabbing anything else, unless we allow to carry more than one object. This last idea will make writing the actions more complex because we need to keep track of all the objects that are being held. So, we will stick to carrying a single object at a time.

We need to decide what predicates to use. Let’s say we decide to use:

- \( \text{At}(\text{entity}, \text{location}) \)
- \( \text{Holding}(a, o) \)
- \( \text{EmptyHand}(a) \)

\( \text{Holding} \) and \( \text{EmptyHand} \) are almost opposite of each other. \( \text{EmptyHand} \) will tell us if a robot is not carrying anything (without having to use negation). \( \text{Holding}(a, o) \) will enable us to keep track of what object is being carried around. We can use it to ensure that the location of the object is updated correctly. When an object is held, we will remove from the world model the object location and add it back when the object is dropped.

**Grab** specifies that to grab object \( o \) agent \( a \) and the object have to be at the same location.

\[
\text{Op} \text{(Action:} \text{Grab}(a, o, x) \text{)}
\]

\[
\text{Precond:} \text{EmptyHand}(a) \land \text{At}(a, x) \land \text{At}(o, x) \land \text{Object}(o)
\]

\[
\text{Effect:} \neg \text{EmptyHand}(a) \land \neg \text{At}(o, x) \land \text{Holding}(a, o)
\]

**Carry** specifies that to carry a object from \( x \) to \( y \) agent \( a \) and the object have to be at \( x \).

\[
\text{Op} \text{(Action:} \text{Carry}(a, o, x, y) \text{)}
\]

\[
\text{Precond:} \text{Holding}(a, o) \land \text{At}(a, x) \land \text{Object}(o)
\]

\[
\text{Effect:} \text{At}(a, y) \land \neg \text{At}(a, x)
\]

**Go** specifies that agent \( a \) will move from its current location \( x \) to the target location \( y \).

\[
\text{Op} \text{(Action:} \text{Go}(a, x, y) \text{)}
\]

\[
\text{Precond:} \text{At}(a, x), \land \neg \text{At}(a, y)
\]

Effect: \( \neg \text{At}(a, x) \land \text{At}(a, y) \)

We add the action **Drop** to drop object \( o \) at position \( x \).

\[
\text{Op} \text{(Action:} \text{Drop}(a, o, x, ) \text{)}
\]

\[
\text{Precond:} \text{Holding}(a, o) \land \text{At}(a, x) \land \text{Object}(o)
\]

\[
\text{Effect:} \text{At}(o, y) \land \text{EmptyHand}(a) \land \neg \text{Holding}(a, o)
\]

2. using the actions you defined and the following initial state

\[
\text{At}(Me, Grocery) \land \text{At}(Egg, Grocery) \land \text{At}(Orange, Grocery) \land \text{At}(Ketchup, Grocery) \land \text{Object}(Egg) \land \text{Object}(Orange) \land \text{Object}(Ketchup)
\]

show a plan to achieve the goal

\[
\text{At}(Me, Home) \land \text{At}(Orange, Home) \land \text{At}(Egg, Home)
\]

Here is an example:

\[
\text{Go}(Me, Store), \text{Grab}(Me, Orange, Store), \text{Carry}(Me, Orange, Home), \text{Drop}(Me, Orange, Home), \text{Go}(Me, Store), \text{Grab}(Me, Egg, Store), \text{Carry}(Me, Egg, Home), \text{Drop}(Me, Orange, Home)
\]
3. write the same actions using the successor state axiom notation.

This is the successor state axiom for Go(p,x,y)

∀a, s, p, x, y:

\[ At(p, y, Result(a, s)) \leftrightarrow [(a = Go(x, y) \land At(p, x, s)) \]
\[ \lor (At(p, y, s) \land \neg(\exists z a = Go(y, z) \land z \neq y))] \]  

(1)

Write the successor state axiom for Grab.

∀a, s, p, o, x, z:

\[ Holding(p, o, Result(a, s)) \leftrightarrow [(a = Grab(p, o, x) \land At(p, x, s)) \land At(o, x, s)) \land EmptyHand(a) \]
\[ \lor (Holding(p, o, s) \land a \neq Drop(o, z)) \]  

(2)