Sample STRIPS planning problem

Consider a robot whose behavior is described the following STRIPS operators:

$$Op(ACTION : Go(x, y), PRECOND : At(ROBOT, x), EFFECT : \neg At(ROBOT, x) \wedge At(ROBOT, y))$$
$$Op(ACTION : Pick(o), PRECOND : At(ROBOT, x) \wedge At(o, x), EFFECT : \neg At(o, x) \wedge Holding(o))$$
$$Op(ACTION : Drop(o), PRECOND : At(ROBOT, x) \wedge Holding(o), EFFECT : At(o, x) \wedge \neg Holding(o))$$

A. The operators allow the robot to hold more than object. Show how to modify them with an *EmptyHand* predicate for a robot that can only hold one object.

B. Assuming these are the only actions in the world, write a successor state axiom for *EmptyHand.*

C. Suppose the initial state is
$At(Apple, Room1) \wedge At(Orange, Room1) \wedge At(ROBOT, Room1) \wedge EmptyHand(\ )$
and the goal is
$At(Apple, Room2) \wedge At(Orange, Room2)$
Derive a total-order plan for this situation, and give the state after each action.

ANSWERS ARE OVER THE PAGE

Answers:

A.
$$Op\begin{pmatrix} ACTION: Pick(o), PRECOND: EmptyHand(\ ) \wedge At(ROBOT, x) \wedge At(o, x), \\ EFFECT: \neg EmptyHand(\ ) \wedge \neg At(o, x) \wedge Holding(o) \end{pmatrix}$$
$$Op\begin{pmatrix} ACTION: Drop(o), PRECOND: At(ROBOT, x) \wedge Holding(o), \\ EFFECT: EmptyHand(\ ) \wedge At(o, x) \wedge \neg Holding(o) \end{pmatrix}$$

Not that the STRIPS formulation does not allo negative preconditions.

B. In English, the hand is empty after doing an action if it was empty before and the action was not a successful *Pick* , or if an object was dropped.

$$EmptyHand(Rsult(a, s)) \Leftrightarrow \begin{bmatrix} (EmptyHand(s) \wedge \neg \exists o, x\ (At(ROBOT, x) \wedge At(o, x) \wedge a = Pick(o))) \\ \vee (Holding(o) \wedge a = Drop(o)) \end{bmatrix}$$

C.

$At(Apple, Room1) \wedge At(Orange, Room1) \wedge At(ROBOT, Room1) \wedge EmptyHand(\ )$

$Pick(Apple)$

$At(Orange, Room1) \wedge At(ROBOT, Room1) \wedge Holding(Apple)$

$Go(Room1, Room2)$

$At(Orange, Room1) \wedge At(ROBOT, Room2) \wedge Holding(Apple)$

$Drop(Apple),$

$At(Orange, Room1) \wedge At(ROBOT, Room2) \wedge At(Apple, Room2) \wedge EmptyHand(\ )$

$Go(Room2, Room1)$

$At(Orange, Room1) \wedge At(ROBOT, Room1) \wedge At(Apple, Room2) \wedge EmptyHand(\ )$

$Pick(Orange)$

$Holding(Orange) \wedge At(ROBOT, Room1) \wedge At(Apple, Room2)$

$Go(Room1, Room2), Drop(Orange)$

$Holding(Orange) \wedge At(ROBOT, Room2) \wedge At(Apple, Room2)$

$Drop(Orange)$

$At(Orange, Room2) \wedge At(ROBOT, Room2) \wedge At(Apple, Room2) \wedge EmptyHand(\ )$

Clearly the Apple and the Orange can be picked in the opposite order.