University of Pittsburgh
CS 2740 Knowledge Representation
Professor Milos Hauskrecht

Handout 14
December 5, 2008

Problem set 5 solutions

STRIPS Planning

Problem 1

Consider a simple blocks world problem:

We use predicates $On(x, y)$ and $Clear(x)$ to describe the states of the world. $On(x, y)$ says that block x is directly atop block y and $Clear(x)$ says that the top of the block x is clear. The initial state is:

$$On(B, A), On(A, C), On(C, Table), On(D, Table), Clear(B), Clear(D).$$

The goal condition is:

$$On(C, B), On(B, A), On(A, D).$$

Part a. Write two STRIPS operators that apply to the blocks world:

- **put-on**(x,y) for stacking a block x on another block y, and
- **put-table**(x) to put the block on the table.

Solution:
put-on(x,y):
Precondition: Clear(x), Clear(y) On(x,z), Add: On(x,y),Clear(z) Delete: Clear(y), On(x,z)
put-table(x):
Precondition: Clear(x), On(x,z) Add: On(x,Table), Clear(z) Delete: On(x,z)

Remark. To be perfectly consistent, one should make sure that Table does not interfere with our moving and stacking. After all, to the planner Table is just another constant. Right now, just a minor conflict results if, say, put-on(A,Table) is applied. Then, On(A, Table) is added and deleted at the same time and the result might depend on implememtation, which is not a desirable thing. But if we decide to add other operations, we might risk inconsistencies. Therefore, the operators should be restricted to operate only on blocks. This can be accomplished by adding Block(A) thru Block(D) to the fact base and using Block(x) in precondition of both **put-on**(x,y) and **put-table**(x).

Part b. Assume we search for the plan using forward (goal-progression) search. Describe the state we obtain after the operator **put-table**(B) is applied to the initial state.

Solution: The state obtained is:

$On(A, C), On(B, Table), On(C, Table), On(D, Table), Clear(A), Clear(B), Clear(D)$

Part c. An alternative to the forward search is the backward or goal regression search. Assume the operator to be applied just before the goal state is reached is **put-on**(C,B). Describe the new goal that results from the selection of this operator.

Solution: $Clear(B), Clear(C), On(B, A), On(A, D), On(C, z)$

Problem 2

Consider a robot whose operation is described by the following STRIPS operators:

- $Action : Go(x, y), Precondition : At(Robot, x), Add : At(Robot, y), Delete : At(Robot, x),$
- $Action : Pick(o), Precondition : At(Robot, x) \wedge At(o, x), Add : Holding(Robot, o),$ $Delete : At(o, x)$
- $Action : Drop(o), Precondition : At(Robot, x) \wedge Holding(Robot, o), Add : At(o, x),$ $Delete : Holding(Robot, o)$

Assume the initial state is described as:
$At(Apple, Room1) \wedge At(Orange, Room1) \wedge At(Robot, Room1)$

and the goal state is:
$At(Apple, Room2) \wedge At(Orange, Room2)$.

Part a. Draw a complete partial-order plan the POP algorithm would find. Note that there can be more complete partial order plans that are consistent with the problem. You are asked to give only one of these complete plans. Show clearly all causal and ordering links between operators. Give a list of all threats resolved through ordering.

Solution: Figure 1.

Part b. List all plans consistent with your partial order plan.

Solution:

- $Pick(Apple), Pick(Orange), Go(Room_1, Room_2), Drop(Apple), Drop(Orange)$
- $Pick(Apple), Pick(Orange), Go(Room_1, Room_2), Drop(Orange), Drop(Apple)$
- $Pick(Orange), Pick(Apple), Go(Room_1, Room_2), Drop(Apple), Drop(Orange)$
- $Pick(Orange), Pick(Apple), Go(Room_1, Room_2), Drop(Orange), Drop(Apple)$

Part c. Assume the modified operation set that prevents the robot to carry more than one objects:

- $Action : Go(x, y), Precondition : At(Robot, x),$
  $Add : At(Robot, y), Delete : At(Robot, x),$
- $Action : Pick(o), Precondition : At(Robot, x) \wedge At(o, x) \wedge EmptyHand(Robot)$
  $Add : Holding(Robot, o), Delete : At(o, x), EmptyHand(Robot)$
- $Action : Drop(o), Precondition : At(Robot, x) \wedge Holding(Robot, o),$
  $Add : At(o, x), EmptyHand(Robot); Delete : Holding(Robot, o)$

Assume the initial state is:
$At(Apple, Room1) \wedge At(Orange, Room1) \wedge At(Robot, Room1) \wedge EmptyHand(Robot)$
and the goal state is:
$At(Apple, Room2) \wedge At(Orange, Room2)$.

Draw a complete partial order plan the POP algorithm would find. Show clearly all causal links and ordering links between operators. Give all the threats resolved through ordering.

Solution: Figure 2.

Part d. List all plans consistent with your partial order plan.

Solution:

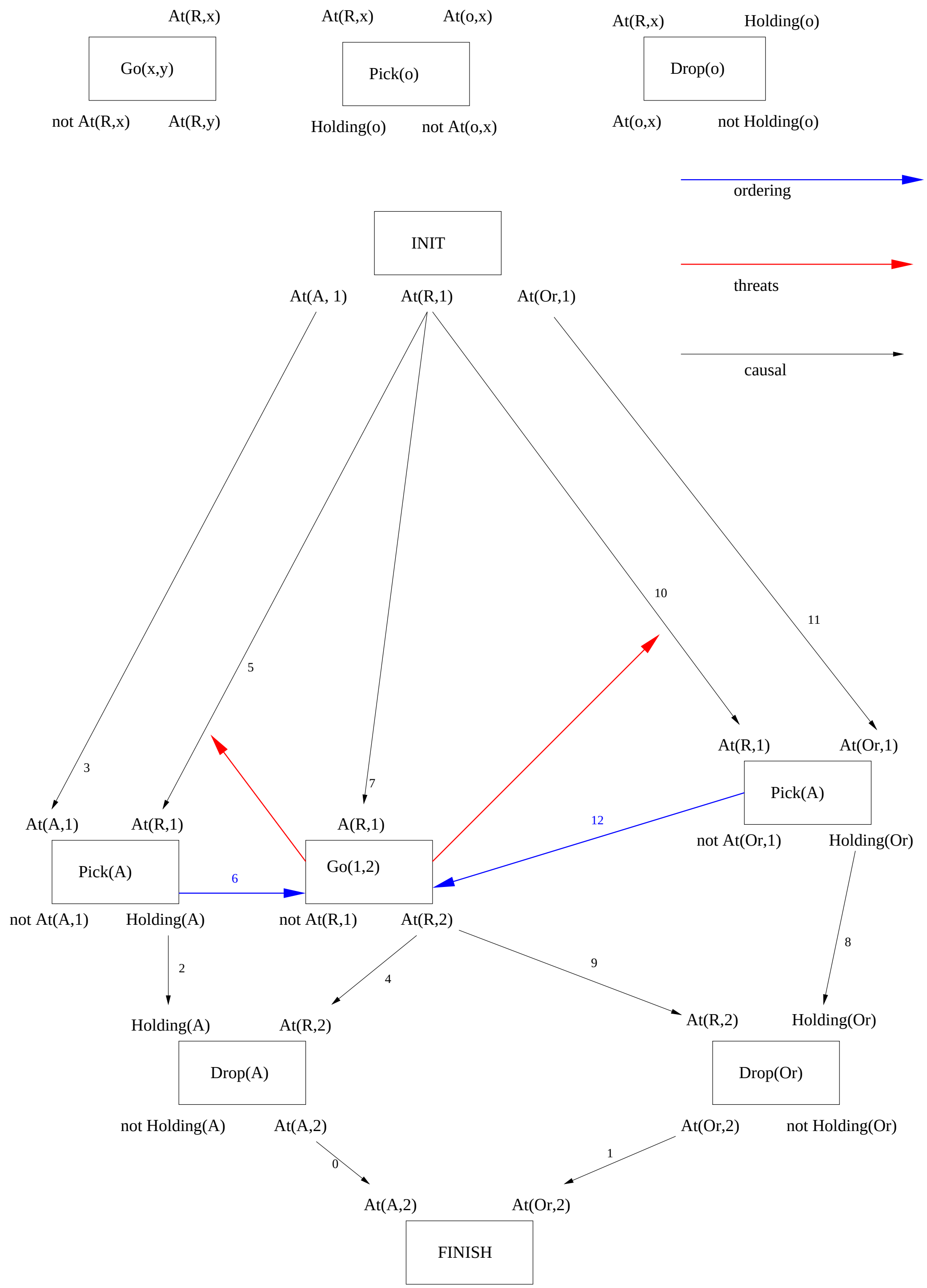

Figure 1: A possible solution to part 2a. The numbers on links give the order in which they were incorporated into the partial plan.

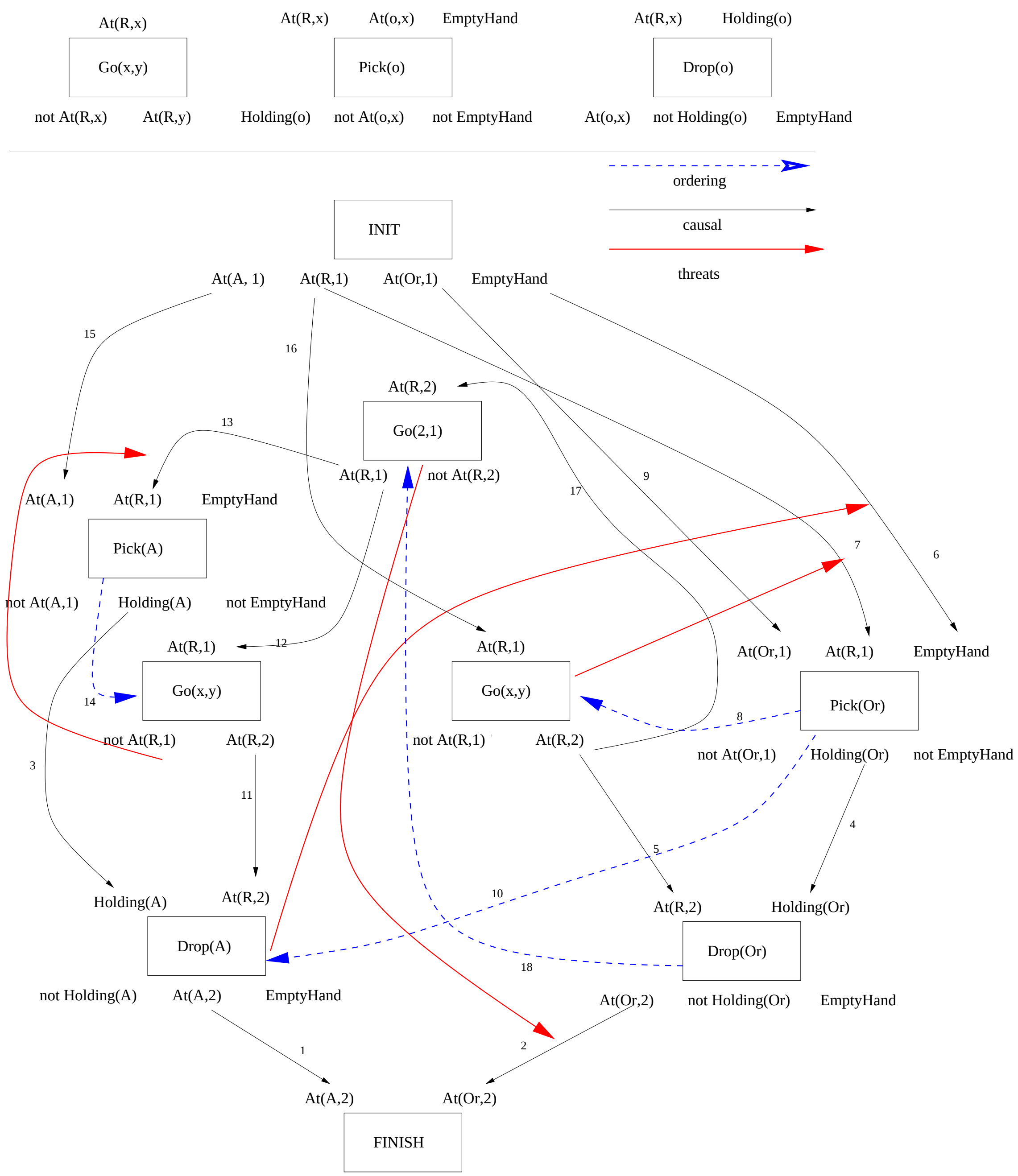

Figure 2: A possible solution to part 2c. The numbers on links give the order in which they were incorporated into the partial plan.

- $Pick(Orange), Go(Room_1, Room_2), Drop(Orange), Go(Room_2, Room1),$ $Pick(Apple), Go(Room_1, Room_2), Drop(Apple)$