

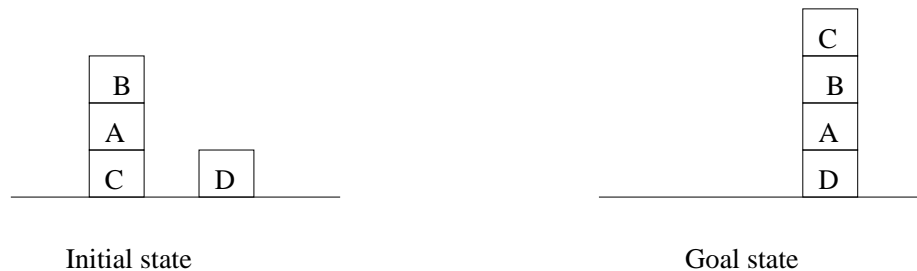
Problem assignment 7

Due: October 23, 2019

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.

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.

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.

Part d. We have decided to use forward search to solve the planning problem. What uninformed method would you consider to solve the problem? Justify.

Part e. Assume we want to enhance the forward search with a heuristic. To estimate the remaining length of the plan we consider the number of subgoals that remain to be satisfied from the current state. We incorporate the heuristic through an evaluation-function driven search procedure. The evaluation function for the state is $f(s) = g(s) + h(s)$, where $g(s)$ is the length of the path from the initial state and the heuristic is the number of subgoals of the goal state that remain to be satisfied by the current state. Draw the search tree generated by the search procedure. Is the solution found optimal?

Part f. We are thinking about adopting the number of subgoals heuristic to an arbitrary STRIPS planning problem. Is the evaluation-function search with $f(s) = g(s) + h(s)$ in combination with such a heuristic always optimal? Explain your answer.

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.

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