

Planning

What is planning ?

Suppose we have one specified goal and are given a set of operators, then a plan is a partially ordered sequence of operator instantiations that will achieve the goal.

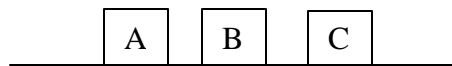
Assumptions.

1. We have a single goal in mind
2. Set of operators is complete and correct relative to the goal - complete in the sense the goal is achievable using the given operators.
3. Cannot add an operator arbitrarily : This is also called the qualification problem.

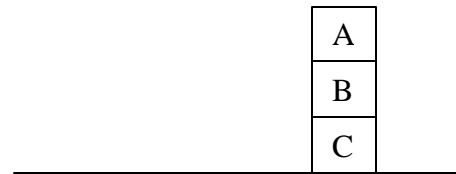
Why is the planning problem hard ?

1. Efficiency of planning should be high
2. How do we represent the operators ?

Blocks World Problem.



Initial State (I.S.)



Goal State (G.S.)

Constraints :

1. Move only 1 block at a time
2. Table is infinitely large
3. Position on table does not matter

Goals are represented using Initial and Final states.

I.S. : (ON A TABLE)
(ON B TABLE)
(ON C TABLE)

G.S. : (ON C TABLE)
(ON B C)
(ON A B)

What operators do we need ?

1. MOVE BLOCK X ON TOP OF BLOCK Z TO BLOCK Y
2. MOVE BLOCK X TO TABLE
3. MOVE BLOCK X FROM TABLE TO ON TOP OF BLOCK Y

(Operator MOVE BLOCK ON TABLE TO TABLE, not needed because of constraint no 3.)

We need to sequence operators. => Intelligence lies in the **use** of domain knowledge, not in the domain knowledge itself.

Encoding of Operators

Operator 2. MOVE BLOCK X TO TABLE

For every operator we specify the preconditions and postconditions. Pre-conditions are the constraints before the operator is applied and postconditions are the results of applying the operator.

Pre-condition : (ON X Y) , (CLEAR X)

Post-condition : 2 Types

Add-List : Condition becomes true after operator is applied

Delete-List : Condition that existed before but is no longer true after operator is applied.

Add-List : (ON X TABLE)

Delete-List : (ON X Y)

Note : (CLEAR X) is not an operator, its just a predicate (assertion).

Operator 1 : MOVE BLOCK X ON TOP OF BLOCK Z TO BLOCK Y

Precondition : (CLEAR X), (ON X Z) , (CLEAR Y)

Postcondition: Add-List : (ON X Y), (CLEAR Z)

Delete-List : (ON X Z), (CLEAR Y)

Operator 3 : MOVE BLOCK X FROM TABLE TO ON TOP OF BLOCK Y

Precondition : (CLEAR X), (CLEAR Y), (ON X TABLE)

Postcondition: Add-List : (ON X Y)

Delete-List : (ON X TABLE), (CLEAR Y)

1. Goal provides partial control knowledge , control knowledge here => which operator to chose next....
2. In case there are multiple goals we need to do goal ordering.
3. Failure recognition : Realize a failure and not repeat them again & again.

One difference between planning and search is that in search we can use any number of heuristics whereas planning deals with goal directed reasoning.

In the above example we have 3 initial states and 3 goals....

Initial State	Goal State
1. ON A TABLE	1. ON A B
2. ON B TABLE	2. ON B C
3. ON C TABLE	3. ON C TABLE

One way to begin is by seeing the differences between the initial and goal states.

Initial state 3 matches with Goal state 3.. so we have only 2 goals now, goal 1 and 2.

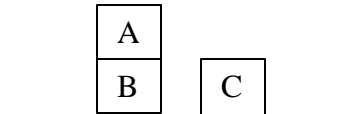
Agent has to pick up 1 of these 2 goals. ... Lets say the agent by random picks up goal 2 → (ON B C)

Now match this against the postconditions of the operators, especially the ADD-LIST. Goal 2 matches the post-conditions of operators 1 and 3 (ON X Y). Operator 1 eliminated.

Now check the pre-conditions, matches with operator 3 . Thus we chose the operator MOVE BLOCK X FROM TABLE TO ON TOP OF Y. **Bind** variables X to B and Y to C. Here we have introduced the new term “**Variable Binding**”.

We have met goal 2. Now go to the next goal (ON A B). Apply operator 3 with variable bindings X = A, Y = B. Thus we have achieved all 3 goals.

Now what if the agent picks up goal 1 first, instead of goal 2. By similar reasoning if the agent applies operator 3 first then we get the following intermediate state



Here we see that goal 1 clobbers goal 2. => Failure.

Planning => interleaving of execution and monitoring (monitoring required to detect failures).

Detecting Failures :

1. pick up goal at random, make partial plan, execute and monitor. Go to the next goal if ok.
2. do a simulation and detect failures.
3. clever simulation => detect failures before they occur. For multiple goals, create partial plans for each goal and then compare the preconditions with postconditions to see which goal clobbers which one.

Problems like the one above are called non-linear problems.

In non-linear problems we have

1. Multiple goals, have to achieve all of them.
2. These goals have interactions between them.
3. Any arbitrarily ordered goals will not work, so the goals have to be executed in a particular sequence to be able to achieve all of them.

Goal interactions are of 2 types, +ve interaction and –ve Interaction.

+ve interaction : Plan for goal X also achieves goal Y. ex. Sending a message and an ack in the same packet.

-ve interaction : Plan for goal X kills goal Y.

Non-Linear Problems always have –ve interactions.

So far we have seen 3 major classes of AI problems : Non-seperable, non-monotonic, non-linear.

How control knowledge can be learnt ?

Explanation based learning - Do an exhaustive search of different goal ordering. Some of these branches are going to lead to failures, one of them will succeed. Learn the correct ordering from this search.