What Is the Role of Planning In Artificial Intelligence?

Artificial intelligence is an important technology in the future. Whether it is intelligent robots, self-driving cars, or smart cities, they will all use different aspects of artificial intelligence!!! But Planning is very important to make any such AI project.

Even Planning is an important part of Artificial Intelligence which deals with the tasks and domains of a particular problem. Planning is considered the logical side of acting.

Everything we humans do is with a definite goal in mind, and all our actions are oriented towards achieving our goal. Similarly, Planning is also done for Artificial Intelligence.

① Find best route
② identify set of actions.
③ identify timing.

For example, Planning is required to reach a particular destination. It is necessary to find the best route in Planning, but the tasks to be done at a particular time and why they are done are also very important.

That is why Planning is considered the logical side of acting. In other words, Planning is about deciding the tasks to be performed by the artificial intelligence system and the system's functioning under domain-independent conditions.

→ deciding a set of actions to be performed in AGENT ENVIRONMENT based on the PERCEPTION to achieve the decided GOAL.

What is a Plan?

for any planning system, we need
A plan is • domain description
– action specification
– goal description

We require domain description, task specification, and goal description for any planning system. A plan is considered a sequence of actions, and each action has its preconditions that must be satisfied before it can act and some effects that can be positive or negative.

So, we have Forward State Space Planning (FSSP) and Backward State Space Planning (BSSP) at the basic level.

sequence of actions.

Two types of planning in AI

each action has it own preconditions.

all preconditions must be satisfied before performing the action.

Forward state space planning (FSSP)   **Backward state space planning (BSSP)**

1. Forward State Space Planning (FSSP)

FSSP behaves in the same way as forwarding state-space search. It says that given an initial state S in any domain, we perform some necessary actions and obtain a new state S' (which also contains some new terms), called a progression. It continues until we reach the target position. Action should be taken in this matter.

- Disadvantage: Large branching factor

- Advantage: The algorithm is Sound

Planning
→ classical
fully observable environment
deterministic
finite
static & discrete.

→ Non classical
partially observable environment
involves diff. setup algorithms
& agent designs.

2. Backward State Space Planning (BSSP)

BSSP behaves similarly to backward state-space search. In this, we move from the target state g to the sub-goal g, tracing the previous action to achieve that goal. This process is called (regression) (going back to the previous goal or sub-goal). These sub-goals should also be checked for consistency. The action should be relevant in this case.

- Disadvantages: not sound algorithm (sometimes inconsistency can be found)
- Advantage: Small branching factor (much smaller than FSSP)

So for an efficient planning system, we need to combine the features of FSSP and BSSP, which gives rise to target stack planning which will be discussed in the next article.

What is planning in AI?

Planning in artificial intelligence is about decision-making actions performed by robots or computer programs to achieve a specific goal.

Execution of the plan is about choosing a sequence of tasks with a high probability of accomplishing a specific task.

Block-world planning problem

- The block-world problem is known as the Sussmann anomaly.
- The non-interlaced planners of the early 1970s were unable to solve this problem. Therefore it is considered odd.
- When two sub-goals, G1 and G2, are given, a non-interleaved planner either produces a plan for G1 that is combined with a plan for G2 or vice versa.
- In the block-world problem, three blocks labeled 'A', 'B', and 'C' are allowed to rest on a flat surface. The given condition is that only one block can be moved at a time to achieve the target.

The start position and target position are shown in the following diagram.

	precondition	action
pick up(x)	• arm empty • on(x, table) • clear(x)	• holding (x)
put down(x)	• holding (x)	• arm empty • on (x, table) • clear (x)
stack (x, y)	• holding (x)	• on (x, b) • clear (x) • Arm empty
unstack(x, y)	• on (x, y) • clear (x) • arm empty	• holding (x) • clear (y)

START

GOAL

Start State　　　　　Goal State
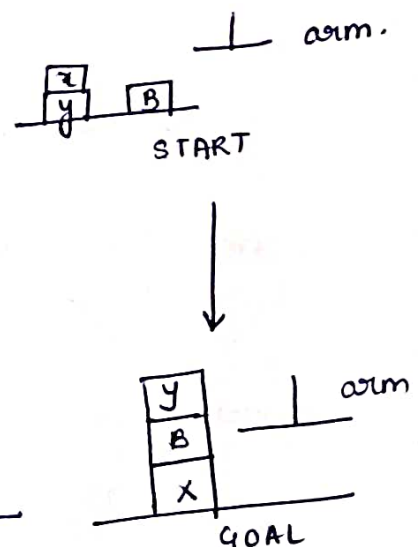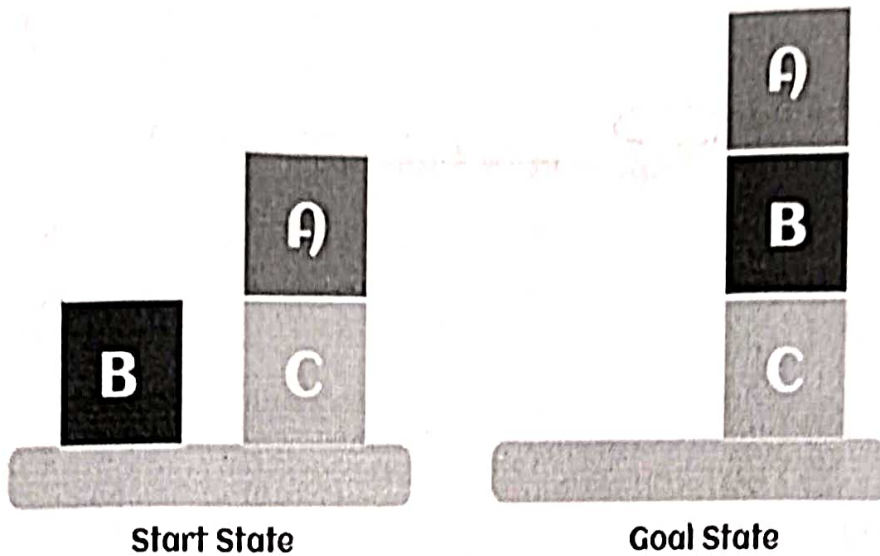
Word planning problem

Components of the planning system

The plan includes the following important steps:

- Choose the best rule to apply the next rule based on the best available guess.

- Apply the chosen rule to calculate the new problem condition.

- Find out when a solution has been found.

- Detect dead ends so they can be discarded and direct system effort in more useful directions.

- Find out when a near-perfect solution is found.

Target stack plan

- It is one of the most important planning algorithms used by STRIPS.

- Stacks are used in algorithms to capture the action and complete the target. A knowledge base is used to hold the current situation and actions.

- A target stack is similar to a node in a search tree, where branches are created with a choice of action.

The important steps of the algorithm are mentioned below:

1. Start by pushing the original target onto the stack. Repeat this until the pile is empty. If the stack top is a mixed target, push its unsatisfied sub-targets onto the stack.

2. If the stack top is a single unsatisfied target, replace it with action and push the action precondition to the stack to satisfy the condition.

III. If the stack top is an action, pop it off the stack, execute it and replace the knowledge base with the action's effect.

If the stack top is a satisfactory target, pop it off the stack.

Non-linear Planning

This Planning is used to set a goal stack and is included in the search space of all possible sub-goal orderings. It handles the goal interactions by the interleaving method.

Advantages of non-Linear Planning

Non-linear Planning may be an optimal solution concerning planning length (depending on the search strategy used).

Disadvantages of Nonlinear Planning

It takes a larger search space since all possible goal orderings are considered.

Complex algorithm to understand.

Algorithm

1. Choose a goal 'g' from the goal set

2. If 'g' does not match the state, then
   - Choose an operator 'o' whose add-list matches goal g
   - Push 'o' on the OpStack
   - Add the preconditions of 'o' to the goal set

3. While all preconditions of the operator on top of OpenStack are met in a state
   - Pop operator o from top of opstack
   - state = apply(o, state)
   - plan = [plan; o]