**UNIT-4**

**Planning**

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.

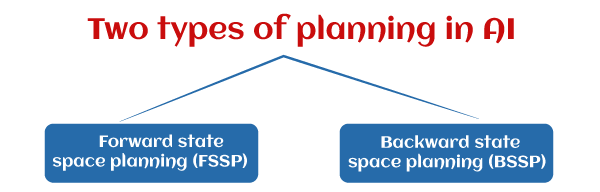
**For example**, Planning is required to reach a particular destination. It is necessary to find the best route in Planning, but the tasks to be done at a particular time and why they are done are also very important.

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.

What is a Plan?

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.



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

### 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?

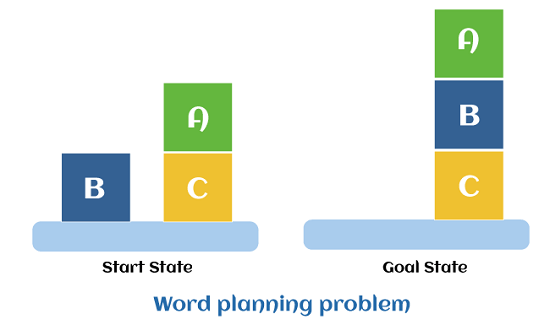
Planning in artificial intelligence is about decision-making actions performed by robots or computer programs to achieve a specific goal.

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.**



**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]

# Hierarchical Planning

***Hierarchical Planning in Artificial Intelligence***is a problem-solving and decision-making technique employed to reduce the computational expense associated with planning. The article provides an overview of hierarchical planning in AI, discussing its components, techniques, applications in autonomous driving and robotics, advantages, and challenges.

## What is Hierarchical Planning in AI?

In [artificial intelligence (AI)](https://www.geeksforgeeks.org/artificial-intelligence-an-introduction/), hierarchical planning is a [planning](https://www.geeksforgeeks.org/what-is-the-role-of-planning-in-artificial-intelligence/)methodology that entails grouping tasks and actions into several abstraction levels or hierarchies, with higher-level jobs being broken down into a series of lower-level tasks. It offers a method for effectively using a hierarchy of goals and sub-goals to reason and plan in complex contexts.

[AI systems](https://www.geeksforgeeks.org/difference-between-ai-and-expert-system/) can effectively handle complicated tasks and surroundings because of hierarchical planning, which enables them to make decisions at many levels of abstraction. Compared to flat planning systems, which treat tasks at the same level of abstraction, this approach differs. AI systems can efficiently handle relationships, prioritize tasks, and distribute resources thanks to the structured method of hierarchical planning, which makes it very useful in complicated contexts.

**Components of Hierarchical Planning**

Artificial intelligence (AI) hierarchical planning usually entails the following essential elements:

* **High-Level Goals:** High-level goals provide the initial direction for the planning process and guide the decomposition of tasks into smaller sub-goals.
* **Tasks:** Tasks are actions that need to be performed to accomplish the high-level goals.
* **Sub-Goals:**Sub-goals are intermediate objectives that contribute to the accomplishment of higher-level goals. Sub-goals are derived from decomposing high-level goals into smaller, more manageable tasks.
* **Hierarchical Structure:**Hierarchical planning organizes tasks and goals into a hierarchical structure, where higher-level goals are decomposed into sub-goals, and sub-goals are further decomposed until reaching primitive actions that can be directly executed.
* **Task Dependencies and Constraints**: Hierarchical planning considers dependencies and constraints between tasks and sub-goals. These dependencies determine the order in which tasks need to be executed and any preconditions that must be satisfied before a task can be performed.
* **Plan Representation**: Plans in hierarchical planning are represented as hierarchical structures that capture the sequence of tasks and sub-goals required to achieve the high-level goals. This representation facilitates efficient plan generation, execution, and monitoring.
* **Plan Evaluation and Optimization**: Hierarchical planning involves evaluating and optimizing plans to ensure they meet the desired criteria, such as efficiency, feasibility, and resource utilization. This may involve iteratively refining the plan structure or adjusting task priorities to improve performance.

**Hierarchical Planning Techniques in AI**

In section, we are going to discuss the hierarchical planning techniques that are leveraged for organizing and executing hierarchical structures:

**1. HTN (Hierarchical Task Network) Planning**

***HTN planning***decomposing high-level tasks into simpler sub-tasks using hierarchical structures called task networks. HTN planning enables the representation of complex tasks as hierarchical networks of actions and conditions, allowing for flexible and modular planning.

**2. Hierarchical Reinforcement Learning (HRL)**

***HRL***is extension of [reinforcement learning](https://www.geeksforgeeks.org/what-is-reinforcement-learning/), it leverages hierarchical structures to facilitate learning and decision-making in complex environments. In HRL, tasks are organized into a hierarchy of sub-goals, and the agent learns policies for achieving these sub-goals at different levels of abstraction. By learning hierarchies of policies, HRL enables more efficient exploration and exploitation of the environment, leading to faster learning and improved performance.

**3. Hierarchical Task Networks (HTNs)**

***HTNs*** are used for representing and reasoning about hierarchical task decomposition. HTNs consist of a set of tasks organized into a hierarchy, where higher-level tasks are decomposed into sequences of lower-level tasks. HTNs provide a structured framework for planning and execution, allowing for the efficient generation of plans that satisfy complex goals and constraints.

**4. Hierarchical State Space Search**

***Hierarchical state space search*** is a planning technique that involves exploring the state space of a problem in a hierarchical manner. Instead of directly exploring individual states, hierarchical state space search organizes states into hierarchical structures, where higher-level states represent abstract representations of the problem space. This hierarchical exploration allows for more efficient search and pruning of the state space, leading to faster convergence and improved scalability.

**Hierarchical Planning in Autonomous Driving**

Let’s consider an example of autonomous driving car, here hierarchical planning is employed in the following manner:

1. **High-Level Goal:**safely navigate from A to B, following the traffic rules
2. **Major Steps:**
   1. **Route Planning:** determine optimal route to B
   2. **Obstacle Avoidance:** identify obstacles like vehicle, people, etc.
   3. **Traffic Signal Recognition:** detect traffic signals and signs
   4. **Lane Keeping:** stay in the designated lane and adjust the vehicle’s position to avoid collision
3. **Minor Steps:**
   1. **Route Planning:**
      * **Map Analysis:** analyze maps to find the optimal route
      * **Traffic Prediction:**predict traffic patterns to avoid traffic jams.
   2. **Obstacle Avoidance:**
      * **Sensor Data Processing:**process the data from onboard sensor to detect nearby objects
      * **Path Planning:**generate paths to avoid obstacles
   3. **Traffic Signal Recognition:**
      * **Image Recognition:** analyze images to detect traffic lights
      * **Traffic Rule Interpretation:** interpret and detect signal to determine the action
   4. **Lane Keeping:**
      * **Lane Detection:**use computer vision algorithms to detect lane markings
      * **Control Systems:**adjust the speed, steering, break command to keep the vehicle within the detected lane.
4. **Hierarchical Planning:**
   1. **First Level Plan**: Define the high-level goals and major steps, such as route planning, obstacle avoidance, traffic signal recognition, and lane keeping.
   2. **Second Level Plan**: Break down each major step into subtasks and minor steps, as described above, to handle the complexity of each component.
   3. **Third Level Plan**: Further decompose the minor steps into detailed actions and algorithms necessary to execute them effectively.

**Hierarchical Planning Techniques in Autonomous Driving**

In autonomous driving, hierarchical planning techniques are crucial for safe navigation.

* **HTN Planning**: Decomposes route planning into subtasks like map analysis and traffic prediction, ensuring optimal routes.
* **Hierarchical Reinforcement Learning (HRL)**: Learns hierarchical policies for obstacle avoidance, adjusting vehicle trajectory to avoid collisions.
* **Hierarchical Task Networks (HTNs)**: Decomposes traffic signal recognition into subtasks for accurate detection and rule interpretation.
* **Hierarchical State Space Search**: Explores state space of lane keeping, adjusting vehicle commands for effective lane-keeping strategies.

**Hierarchical Planning in Robotics**

Let’s consider an example of hierarchical planning applied to a robotic arm tasked with assembling electronic devices:

1. **High-Level Goal:**ensemble electronic devices following the provided assembly process
2. **Major Steps:**
   1. **Identifying Components:**identify and local the components required for assembling the device.
   2. **Planning Assembly Sequence:** determine optimal sequence of assembly steps to minimize assembly time and maximize efficiency.
   3. **Manipulation and Grasping:** manipulate the arm of the robot to grasp and manipulate the electronic components.
   4. **Quality Control:**ensure the quality of assembly step and detect and correct any errors.
3. **Minor Steps:**
   1. **Identifying Components:**
      1. **Object Recognition:** recognize components using computer vision
      2. **Inventory Management:**maintain inventory of the components
   2. **Planning Assembly Sequence:**
      1. **Task Planning:** break the process into smaller and sequential tasks
      2. **Motion Planning:** planning the arm movement to perform assembling
   3. **Manipulation and Grasping:**
      1. **Grasping strategy:**determine optimal grasping pose based on shape and side of the components.
      2. **Path Generation:** generate smooth trajectories for the robotic arm
   4. **Quality Control:**
      1. **Vision Inspection:**inspect assembled components using cameras
      2. **Feedback Control:** implement feedback mechanism to adjust assembly actions.
4. **Hierarchical Planning:**
   1. **First Level Plan**: Define the high-level goals and major steps, including component identification, assembly sequence planning, manipulation and grasping, and quality control.
   2. **Second Level Plan**: Break down each major step into subtasks and minor steps, such as object recognition, inventory management, task planning, motion planning, grasping strategy, path generation, vision inspection, and feedback control.
   3. **Third Level Plan**: Further decompose the minor steps into detailed actions and algorithms necessary to execute them effectively, such as specific image processing algorithms for object recognition or control algorithms for adjusting robotic arm movements based on inspection results.

**Hierarchical Planning Techniques in Robotics**

In robotic arm assembly, hierarchical planning techniques like HTN planning, hierarchical reinforcement learning (HRL), HTNs, and hierarchical state space search ensure efficient execution.

* **HTN Planning**: Organizes major steps hierarchically, facilitating task decomposition.
* **Hierarchical Reinforcement Learning (HRL)**: Learns hierarchical policies for manipulation and grasping, optimizing performance.
* **HTNs (Hierarchical Task Networks)**: Structures minor steps systematically, ensuring accurate component handling.
* **Hierarchical State Space Search**: Optimizes assembly sequences considering constraints, ensuring efficiency.

**Advantages of Hierarchical Planning**

Here, the advantages of hierarchical planning:

* **Users capability:** Planning and reasoning at various levels of abstraction are made possible by hierarchical planning, which makes it possible to handle complicated tasks and situations effectively.
* **Internal Flexibility:** Plans can be adjusted to reflect changes in the environment or goals thanks to the flexibility that hierarchical planning offers, which strengthens and adapts the plans.
* **Personal reuse and abstraction:** By employing a hierarchy of activities or sub goals, plans can be repurposed and abstracted, increasing planning effectiveness and decreasing the need for duplicate planning.
* **Higher-level reasoning addaptibility:** AI systems are able to make strategic decisions and coordinate actions at a higher degree of abstraction thanks to hierarchical planning, which facilitates higher-level reasoning and decision-making.

**Challenges and Limitations of Hierarchical Planning**

Although hierarchical planning has many benefits, there are some challenges and limitations as well:

* **Planning Complexity:** As the number of tasks rises, both the initial decomposition and the following planning may become computationally demanding.
* **Adaptability:** Modifications to the environment or the main objectives may call for a thorough re-planning process that may demand a large amount of resources.