

Planning

Classical Planning

Classical planning refers to the process of generating a sequence of actions to achieve specific goals in a deterministic, fully observable environment. It operates under the assumptions that:

- **Deterministic Actions:** Each action's outcome is predictable and leads to a specific state.
- **Fully Observable Environment:** The planner has complete knowledge of the current state and can observe changes caused by actions.

Key Concepts:

1. **State:** Represents the configuration of the world at a particular time, often described using variables and their values.
2. **Action:** A transition from one state to another, typically defined by preconditions (conditions that must be true for the action to be applicable) and effects (changes to the state after the action is executed).
3. **Goal:** The desired state or conditions that the planner aims to achieve through a sequence of actions.

Algorithms for Planning with State Space Search

Planning algorithms use state space search techniques to find a sequence of actions that transform the initial state into a state satisfying the goal conditions. Common algorithms include:

1. Forward State Space Search

- **Description:** Begins from the initial state and explores possible actions to reach the goal state.
- **Process:**
 - Start from the initial state.
 - Generate successor states by applying applicable actions.
 - Repeat until a state satisfying the goal conditions is reached.
- **Search Strategies:** Can use strategies like Breadth-First Search (BFS) or Depth-First Search (DFS) depending on memory constraints and desired completeness.

2. Backward State Space Search

- **Description:** Starts from the goal state and works backwards to determine a sequence of actions leading to the initial state.
- **Process:**
 - Begin with the goal state.
 - Identify actions that could lead to the goal state.
 - Repeat recursively until the initial state is reached.
- **Search Strategies:** Often uses techniques such as Goal Regression or Goal Stack Planning, focusing on achieving sub-goals leading to the main goal.

3. Heuristic Search

- **Description:** Uses heuristic functions to guide the search towards more promising states, potentially improving efficiency.
- **Process:**
 - Incorporates heuristic estimates of distance to the goal into the search.
 - Examples include A* search, which balances the cost of actions with an estimate of remaining cost to achieve the goal.

4. Graph-Based Planning

- **Description:** Represents the planning problem as a graph, where nodes represent states and edges represent actions.
- **Process:**
 - Utilizes algorithms like Graph Plan or SAS+ that manipulate the state space graph to generate plans efficiently.
 - Optimizes plan quality by considering dependencies and constraints between actions.

Certainly! Here's a detailed overview of Planning Graphs, Other Classical Planning Approaches, Analysis of Planning Approaches, and Hierarchical Planning in Markdown format:

Planning Graphs

Definition: Planning Graphs are a structured representation of the planning problem, aiming to simplify and expedite the process of finding solutions. They consist of two interleaved layers:

1. **Level 0:** Represents the initial state of the world.

2. **Level 1, 2, ...**: Alternating layers of state and action sets, where each level is derived from the previous one.

Key Concepts:

- **State Layer**: Represents possible states of the world at different time steps.
- **Action Layer**: Lists actions applicable to the states of the previous layer.

Advantages:

- **Efficiency**: Reduces the complexity of state space exploration by limiting the search to relevant subsets of states and actions.
- **Heuristic Estimation**: Enables heuristic evaluation by estimating the distance to the goal based on the level of the planning graph.

Other Classical Planning Approaches

1. STRIPS (Stanford Research Institute Problem Solver):

- **Description**: One of the earliest planning formalisms, focuses on actions with preconditions and effects.
- **Features**: Simple representation of actions and states, emphasizing changes caused by actions.

2. ADL (Action Description Language):

- **Description**: Extends STRIPS by allowing for more expressive action descriptions, including quantifiers and complex conditions.
- **Features**: Enhances flexibility in representing domain-specific knowledge and action constraints.

3. Partial Order Planning:

- **Description**: Plans actions without strictly ordering them, accommodating parallel execution and temporal constraints.
- **Features**: Suitable for domains where actions can be executed concurrently or in arbitrary order.

Analysis of Planning Approaches

Criteria for Analysis:

- **Expressiveness**: Ability to represent complex domains and actions.

- **Computational Complexity:** Efficiency in finding solutions, especially in large state spaces.
- **Scalability:** Performance under varying problem sizes and constraints.
- **Domain Specificity:** Suitability for different types of planning problems and real-world applications.

Comparison:

- **Graph-Based vs. Heuristic Search:** Graph-based approaches like Planning Graphs offer structured exploration, while heuristic search methods like A* balance completeness and efficiency.
- **Domain-Specific Formalisms:** STRIPS and ADL cater to different levels of domain complexity, influencing representation and planning efficiency.

Hierarchical Planning

Definition: Hierarchical Planning organizes planning tasks into hierarchical structures, simplifying complex problems by decomposing them into manageable sub-problems.

Levels:

- **High-Level Plans:** Define goals and sub-goals.
- **Low-Level Plans:** Specify detailed actions and their execution sequences.

Advantages:

- **Abstraction:** Focuses on high-level objectives while handling implementation details separately.
- **Reusability:** Encourages reuse of sub-plans across different planning scenarios.

Approaches:

- **HTN (Hierarchical Task Network):** Represents plans as hierarchical task networks, allowing for task decomposition and refinement.
- **Planning with Task Decomposition:** Divides tasks into sub-tasks, optimizing planning efficiency and flexibility.

Considerations:

- **Complexity Management:** Balancing between high-level goal achievement and low-level action execution.
- **Plan Refinement:** Ensuring consistency and correctness across different levels of planning abstraction.