Planning

Planning problem: find sequence of actions that lead to a goal
  – Is a classical search problem

• Typical (real-world) planning problem:
  – Large state descriptions: may represent a number entities and relations that hold in the world at specific point in time
  – Large number of operators (actions) one can apply at any point
  – Goals may be conditions on only a few entities, relations of the world.

• Problem: an extremely large search problem
• Solution: Open state, action and goal representations to allow selection, reasoning. Make things visible and expose the structure. Use the structure actively in the solution.
### Backward search (goal regression)

- Use operators to generate new goals
- Check whether the initial state satisfies the goal

![Diagram of Backward search](image)

### Divide and conquer - Sussman’s anomaly

- **Divide and conquer strategy** – divide the problem to a set of smaller subproblems
  - Can be applied safely if we can achieve each of the goals separately while not affecting other goals
- Case in which there is no order in which goals can be solved:

![Diagram of Divide and conquer](image)
**Sussman’s anomaly**

1. Assume we want to satisfy \( On(A, B) \) first

   ![Initial state](image1)

   But now we cannot satisfy \( On(B, C) \) without undoing \( On(A, B) \)

2. Assume we want to satisfy \( On(B, C) \) first.

   ![Initial state](image2)

   But now we cannot satisfy \( On(A, B) \) without undoing \( On(B, C) \)

---

**State-space vs. plan-space**

- An alternative to planning algorithms that search states (configurations of world) is to search the space of plans
- **Plan:**
  - Defines sequences of operators to be performed
- **Partial plans:**
  - plans are not complete
  - Some orderings of operators are not finalized
- **State-space vs Plan-space search:**
  - State-space search – a node is a configuration of the world
  - Plan-space search – a node is a partial plan
State-space vs. plan-space search

State-space search

- STRIPS operator
- State (set of formulas)

Plan-space search

- Plan transformation operator
- Incomplete (partial) plan

Plan transformation operators

Examples:

- Add an operator to a plan

  \[
  \text{Move}(A,x,B)
  \]

- Order (reorder) operators

  \[
  \begin{array}{c}
  \text{Move}(A,x,B) \\
  \text{Move}(A,C,B)
  \end{array}
  \]

- Instantiate an operator

  \[
  \begin{array}{c}
  \text{Move}(A,x,B) \\
  \text{Move}(A,C,B)
  \end{array}
  \]
**Partial-order planners**

- Also called Non-linear planners
- STRIPS-like operators

Illustration on Sussman’s anomaly case

Graphical representation of an operator

```
On(x,z)  Clear(y)  add list
Move(x,y,z)
On(x,y)  Clear(x)  Clear(z)  preconditions
```

**Partial order planning - Start and finish**

```
Start
On(C,A)  Clear(Fl)  On(A,Fl)  Clear(B)  On(B,Fl)  Clear(C)
```

```
Finish
On(A,B)  On(B,C)
```

Goal

```
Goal
C  A  B
```

```
Partial order planning

Finish

On(A,B)

Clear(y)

On(A,B)

Move(A,y,B)

Clear(A)

On(A,y)

Clear(B)

On(C,A)

Clear(Fl)

On(A,Fl)

Clear(B)

On(B,Fl)

Clear(C)

Start

Move(A,Fl,B)

Clear(A)

On(A,Fl)

Clear(B)

Move(C,A,Fl)

Start
Partial order planning

Partial order planning - Threats
Partial order planning - Directions

Partial order planning - Result plan

Plan: a topological sort of a graph
Partial order planning

- **Remember** we search the space of partial plans

![Diagram of partial order planning]

Hierarchical planners

Extension of STRIPS planners. Example ABSTRIPS.

**Idea:**
- Assign criticality level to each conjunct in preconditions of the operator
- Planning process refines the plan gradually based on criticality threshold, starting from the highest criticality value:
  - Develop the plan ignoring preconditions of criticality less than the criticality threshold value (assume that preconditions for lower criticality levels are true)
  - Lower the threshold value by one and repeat previous step
Hierarchical planning

Assume:

- the largest disk – criticality level 2
- the medium disk – criticality level 1
- the smallest disk – criticality level 0