



# Theory Assignment-I

---

*Prashanth.S 19MID0020*

## Practical Planners

---

The domain definition, action specification, and objective description are required for any planning system. A plan is supposed to be a series of activities, each of which has its own set of preconditions that must be met before the action can be carried out, as well as certain repercussions that can be good or negative. At the most basic level, we have Forward State Space Planning (FSSP) and Backward State Space Planning (BSSP).

### Forward State Space Planning (FSSP)

FSSP works in the same way as forward state space search. It states that given a start state  $S$  in any domain, we undertake the appropriate activities and acquire a new state  $S'$  (which includes some additional conditions as well) termed progress, which continues until we reach the goal state. In this instance, the activities must be appropriate.

Advantage: Algorithm is Sound

Disadvantage : Large branching factor

### Backward State Space Planning (BSSP)

Backward state space search is similar to how BSSP works. We proceed from goal state  $g$  to sub-goal  $g$ , which is determining the previous action to be taken in order to attain that specific objective. This is referred to as regression (moving back to the previous goal or sub-goal). These sub-goals must also be double-checked for consistency. In this scenario, the acts must be pertinent.

Advantage: Small branching factor (very small compared to FSSP)

Disadvantage : Not a sound algorithm (some-times inconsistency can be found)

# THEORY ASSIGNMENT-I

Prashanth.S 19MID0020

TO MAKE PLANNING PRACTICAL WE NEED TO:

→ Restrict the language with which we define problems.

With a restrictive language, there are fewer possible solutions to search through.

→ Use a special-purpose algorithm called a planner rather than a general purpose theorem prover.

## General Language Features:

### A. REPRESENTATION OF STATES

- Decompose the world into logical conditions and portray a state as a positive literal conjunction.
- PROPOSITIONAL LITERALS : (Poor  $\wedge$  Unknown) - FO-literals.  
(grounded and function-free): At(Plane1, Melbourne)  $\wedge$  At(Plane2, Sydney)
- Closed world assumption.

### B. REPRESENTATION OF GOALS

- Partially specified state and represented as a conjunction of positive ground literals
- A goal is satisfied if the state contains all literals in goal.

### C. REPRESENTATION OF ACTIONS

ACTION + PRECOND + EFFECT

- Action (Fly(p, from, to)
- PRECOND : At (p, from) Plane (p)  $\wedge$  Airport (from) – Airport (to)
- EFFECT : At (p, from)  $\wedge$  At (p, to)

### D. REPRESENTATION OF PLANS

- A SET OF PLAN STEPS: Each step is one of the operators for the problem.
- A SET OF STEP ORDERING CONSTRAINTS: Each ordering constraint is of the form Si-Sj indicating Si must occur sometime before Sj.
- A SET OF VARIABLE BINDING CONSTRAINTS OF THE FORM V = X: where v is a variable in some step, where v is a variable in some step, and x is either a constant or another variable.
- A SET OF CAUSAL LINKS WRITTEN AS S  $\rightarrow$  C : S' indicating S satisfies the precondition c for S'.

## STRIPS

- STANFORD RESEARCH INSTITUTE PROBLEM SOLVER
- Many planners today use specification languages that are variants of the one used in STRIPS.

## ADL

- ACTION DESCRIPTION LANGUAGE
- In artificial intelligence action description language(ADL) is an automated planning and scheduling system in particular for robots. It is considered an advancement of STRIPS.

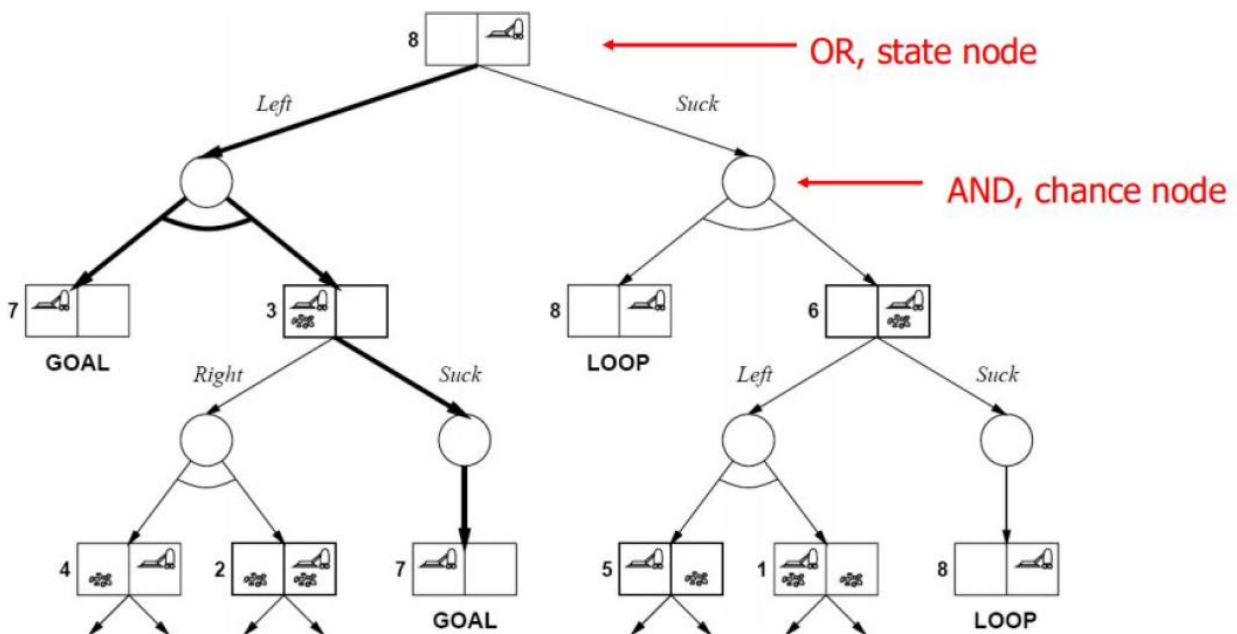
## Conditional Planning

- Constructing a conditional plan that accounts for every possible event or contingency that could emerge is how conditional planning deals with insufficient knowledge.
- It must function independently of the action's outcome.
- Conditional planning occurs in a Completely Observable Environment, in which the agent's current state is fully observable in a known environment.
- Because the consequence of actions is unpredictable, the environment is described as non-deterministic.
- At preset points in the strategy, we can verify what is happening in the environment to cope with unclear activities.
- It must be able to take certain actions in each stage and manage every outcome of the activities it takes.
- A "state node" is symbolized by a "square," whereas a "chance node" is symbolized by a "circle."

Example:

Consider the following **Vacuum Cleaner Robot**:

- In **Initial state**, robot is in the **Right Square** of a clean world. The environment is fully observable.
- The **Goal state** has the robot in the **Left Square** of a clean world.
- It sometimes **Deposits Dirt** when it **moves to a clean destination square** and sometimes **if Suck is applied to a clean square**.



## THEORY ASSIGNMENT-I

*Prashanth.S 19MID0020*

- Every leaf contains a Goal Node.
- At each of its "state" nodes, one action (Right, Left, Suck) is given, as well as every result branch (Goal, Loop) at each of its "chance" nodes.
- To solve the plan for conditional planning, we employ a modified version of the min-max algorithm.
- The MAX and MIN nodes are renamed OR and AND, respectively. The strategy must take some action in each state, but it must also handle all of the possible outcomes of the actions it performs.
- At AND and OR nodes, the algorithm should produce a conditional plan rather than just a single move.
- Because every path must achieve a goal, a dead end, or a repeated state, i.e. a non-cyclic solution, the algorithm finishes in every finite state space.

### Replanning Agents

Replanning uses preceding techniques to construct a plan, but it monitors the execution process and replan when necessary.

EXAMPLE:

SCENARIO: -> Given an initial state with a chair, a table, and some cans of paint, with everything of unknown color, achieve the state where the chair and table have the same color.

- LookAt (Table), LookAt (Chair)
- If Color (Table, c) ^ Color (Chair, c) then NoOp
- Else RemoveLid (Can1), LookAt (Can1)
- If Color (Table, c) ^ Color (Can1, c) then Paint (Chair, Can1)
- Else replan

Example: Vacuum World

Single-state, start in #5. Solution?

[Right, Suck]

Multi-state, start in #[1, 2, ..., 8]. Solution?

[Right, Suck, Left, Suck]

