



SNS COLLEGE OF TECHNOLOGY



Coimbatore-35

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A++' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF INFORMATION TECHNOLOGY

19CSE303 - ARTIFICIAL INTELLIGENCE
III YEAR IV SEM

UNIT III – PLANNING

TOPIC – **Partial Order Planning**



The Planning Problem

Formally a planning algorithm has three inputs:

1. A description of the world in some formal language,
2. A description of the agent's goal in some formal language, and
3. A description of the possible actions that can be performed.

The planner's o/p is a sequence of actions which when executed in any world satisfying the initial state description will achieve the goal.



Representation for states and goals

In the STRIPS language, states are represented by conjunctions of function-free ground literals, that is, predicates applied to constant symbols, possibly negated. For example,

$\text{At(Home)} \wedge \neg \text{Have(Milk)} \wedge \neg \text{Have(Bananas)} \wedge \neg \text{Have(Drill)} \wedge \dots$

Goals are also described by conjunctions of literals. For example,

$\text{At(Home)} \wedge \text{Have(Milk)} \wedge \text{Have(Bananas)} \wedge \text{Have(Drill)}$

Goals can also contain variables. For example, the goal of being at a store that sells milk would be represented as



Representations for actions

Our STRIPS operators consist of three components:

- the *action description* is what an agent actually returns to the environment in order to do something.
- the *precondition* is a conjunction of atoms (positive literals) that says what must be true before the operator can be applied.
- the *effect* of an operator is a conjunction of literals (positive or negative) that describes how the situation changes when the operator is applied.

Here's an example for the operator for going from one place to another:

**Op(Action:Go(there), Precond:At(here)^Path(here, there),
Effect:At(there)^ ¬At(here))**



Search through World Space

The simplest way to build a planner is to cast the planning problem as search through the space of world states. Each node in the graph denotes a state of the world, and arcs connect worlds that can be reached by executing a single action. We would call it a **situation space** planner because it searches through the space of possible situations.

There are two types of planners:

Progression Planner: it searches forward from the initial situation to the goal situation.

Regression Planner: it searches backward from the goal situation to the initial situation.



Search through the Space of Plans

An alternative is to search through space of plans rather than space of situations i.e plan-space node denote plans

- Start with a simple partial plan
- Expand the plan until the complete plan is developed
- Operators in this step:
 - Adding a step
 - Imposing an ordering that puts one step after another
 - Instantiating a previously unbound variable
- The solution \rightarrow final plan Path \rightarrow irrelevant



Representation of Plans

Consider a simple problem:

- Putting on a pair of shoes
- Goal \rightarrow RightShoeOn \wedge LeftShoeOn
- Four operators:
 - **Op(Action:RightShoe,PreCond:RightSockOn,Effect:RightShoeON)**
 - **Op(Action:RightSock , Effect: RightSockOn)**
 - **Op(Action:LeftShoe, Precond:LeftSockOn, Effect:LeftShoeOn)**
 - **Op(Action:LeftSock,Effect:LeftSockOn)**



Least Commitment –

One should make choices only about things that you currently care about ,leaving the others to be worked out later.

Partial Order Planner –

A planner that can represent plans in which some steps are ordered (before or after) w.r.t each other and other steps are unordered.

Total Order Planner—

Planner in which plans consist of a simple lists of steps

Linearization of P—

A totally ordered plan that is derived from a plan P by adding constraints



Partial Order Plans:



Total Order Plans:





Plans, Causal Links and Threats

Representing a Plan as a three tuple :-- $\langle A, O, L \rangle$

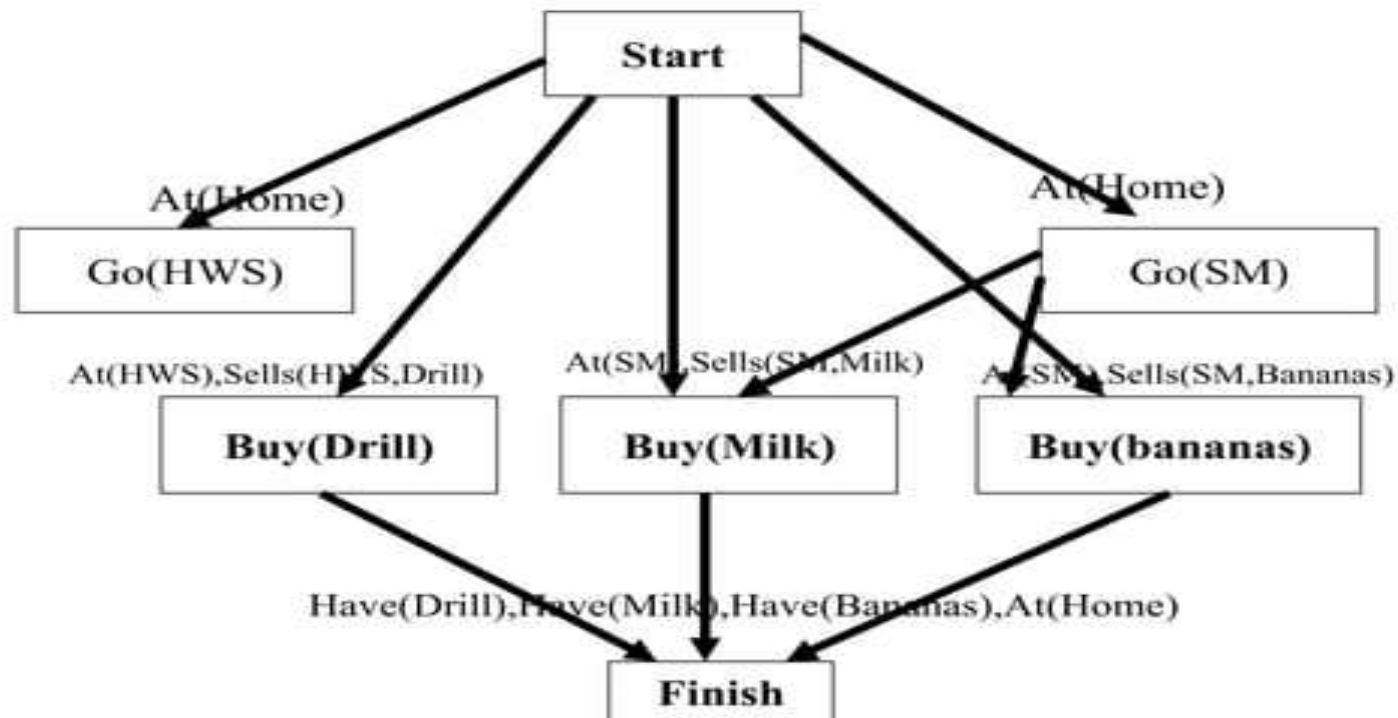
- $A \rightarrow$ Set of Actions
- $L \rightarrow$ Set of Causal Links
- $O \rightarrow$ Set of Ordering constraints over A

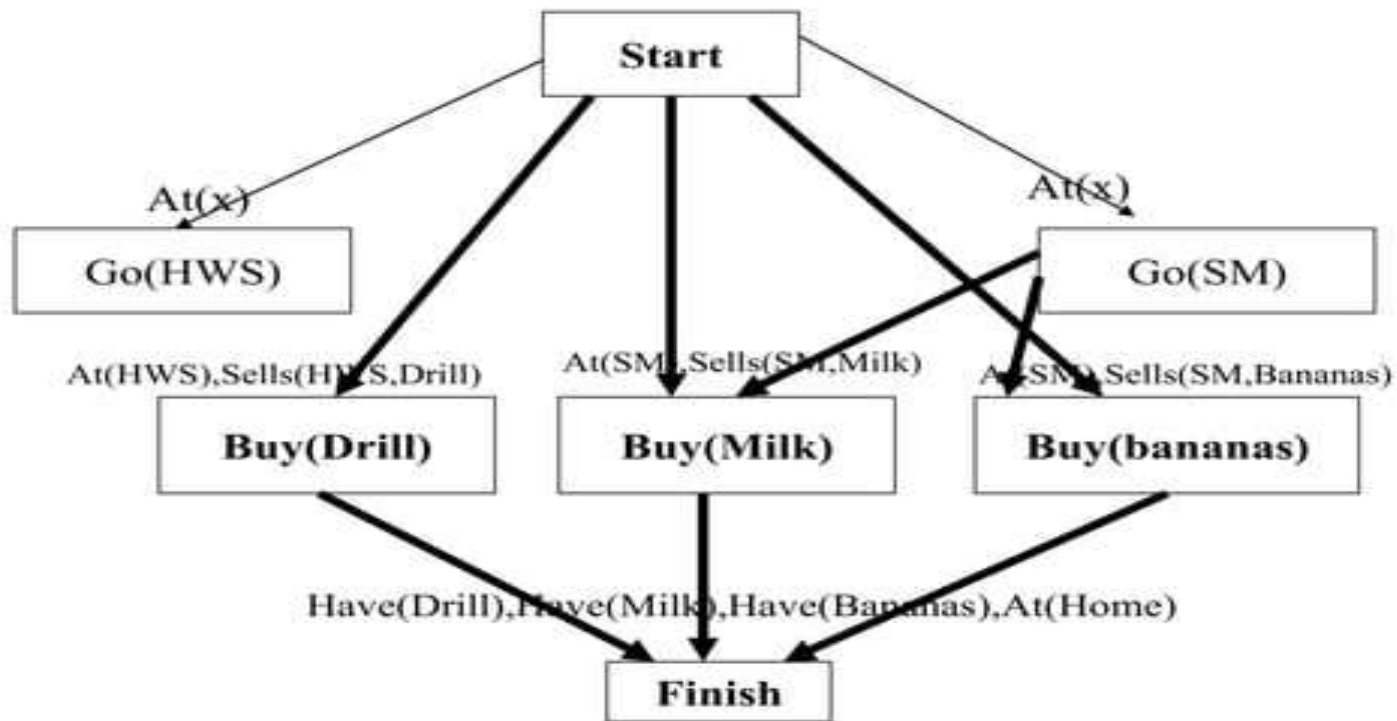
If $A = \{A1, A2, A3\}$ then O might be set $\{A1 < A3, A2 < A3\}$

These constraints specify a plan in which $A3$ is necessarily the last action but does not commit to a choice of which of the three actions comes first .

As least commitment planners refine their plans, they must do constraint satisfaction to ensure consistency of O









Causal Links—

A Causal Link is written as $S_i \xrightarrow{c} S_j$ and read as “ S_i achieves c for S_j “. Causal Links serve to record the purpose of steps in the plan: here a purpose of S_i is to achieve the precondition c of S_j

Threat –

Causal Links are used to detect when a newly introduced action interferes with past decision. Such an action is called a Threat.

Let A_t be a different action in A : we say that A_t *threatens* $A_p \xrightarrow{Q} A_c$ when the two criteria are met:

- $\{A_p < A_t < A_c\}$ is consistent
- A_t has $\neg Q$ as an effect.



Resolving Threats

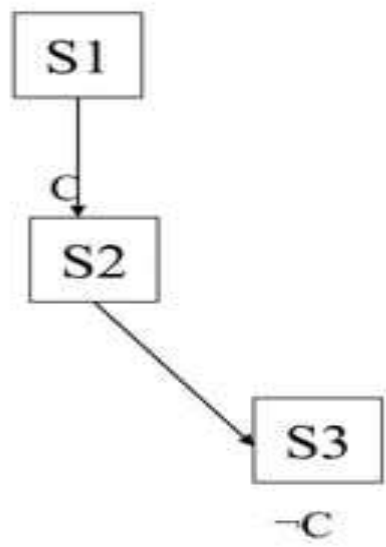
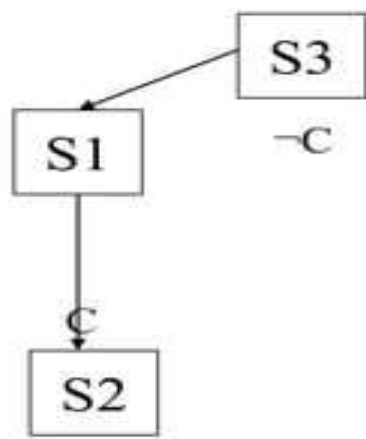
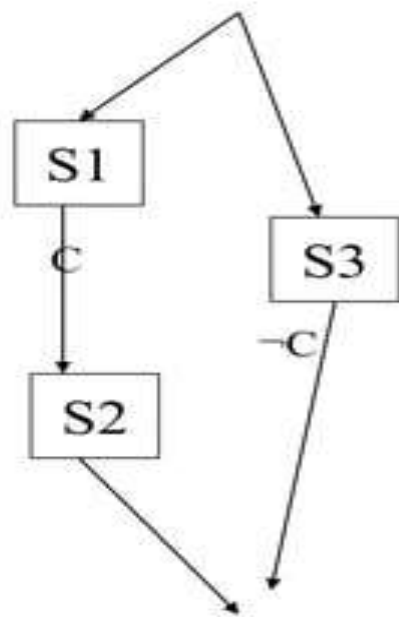
When a plan contains a threat, then there is a danger that the plan won't work as anticipated i.e. We have reached a dead-end in the search.

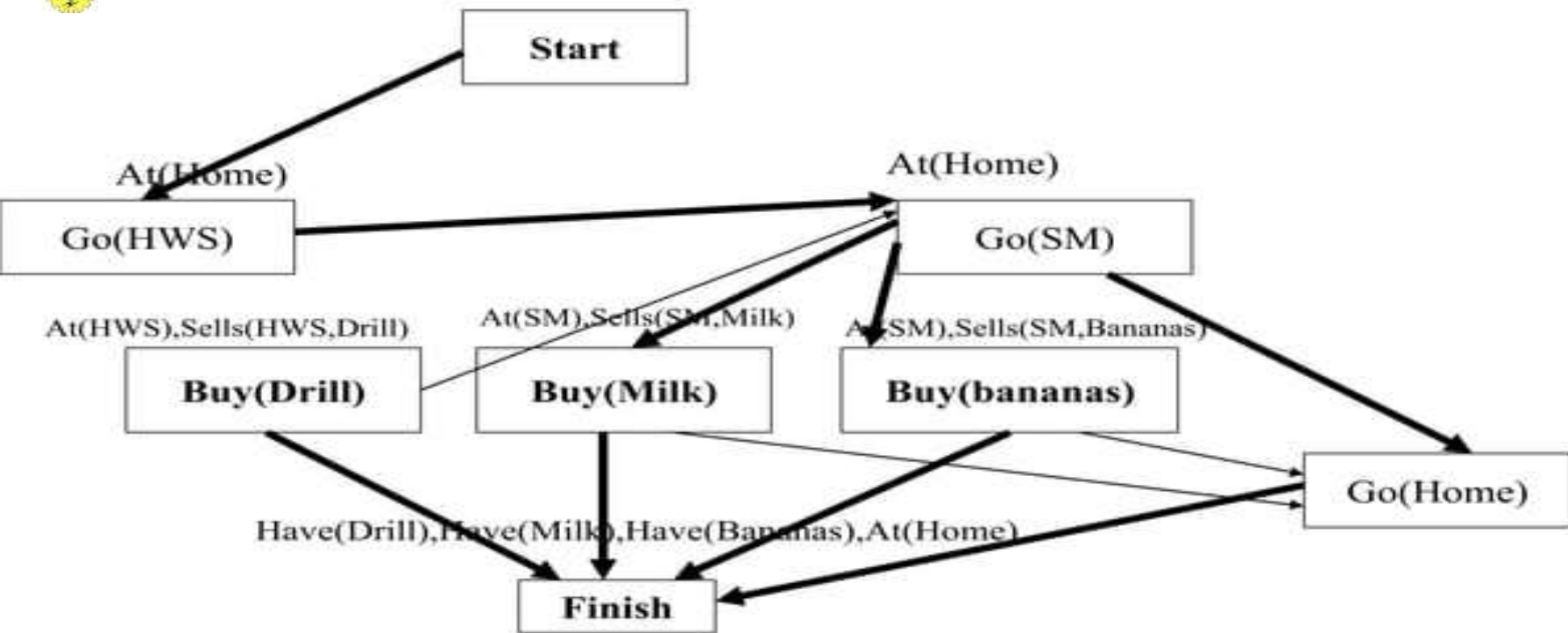
The causal link $S1 \overset{c}{\rightarrow} S2$ is threatened by a new step $S3$ because one effect of $S3$ is to delete c .

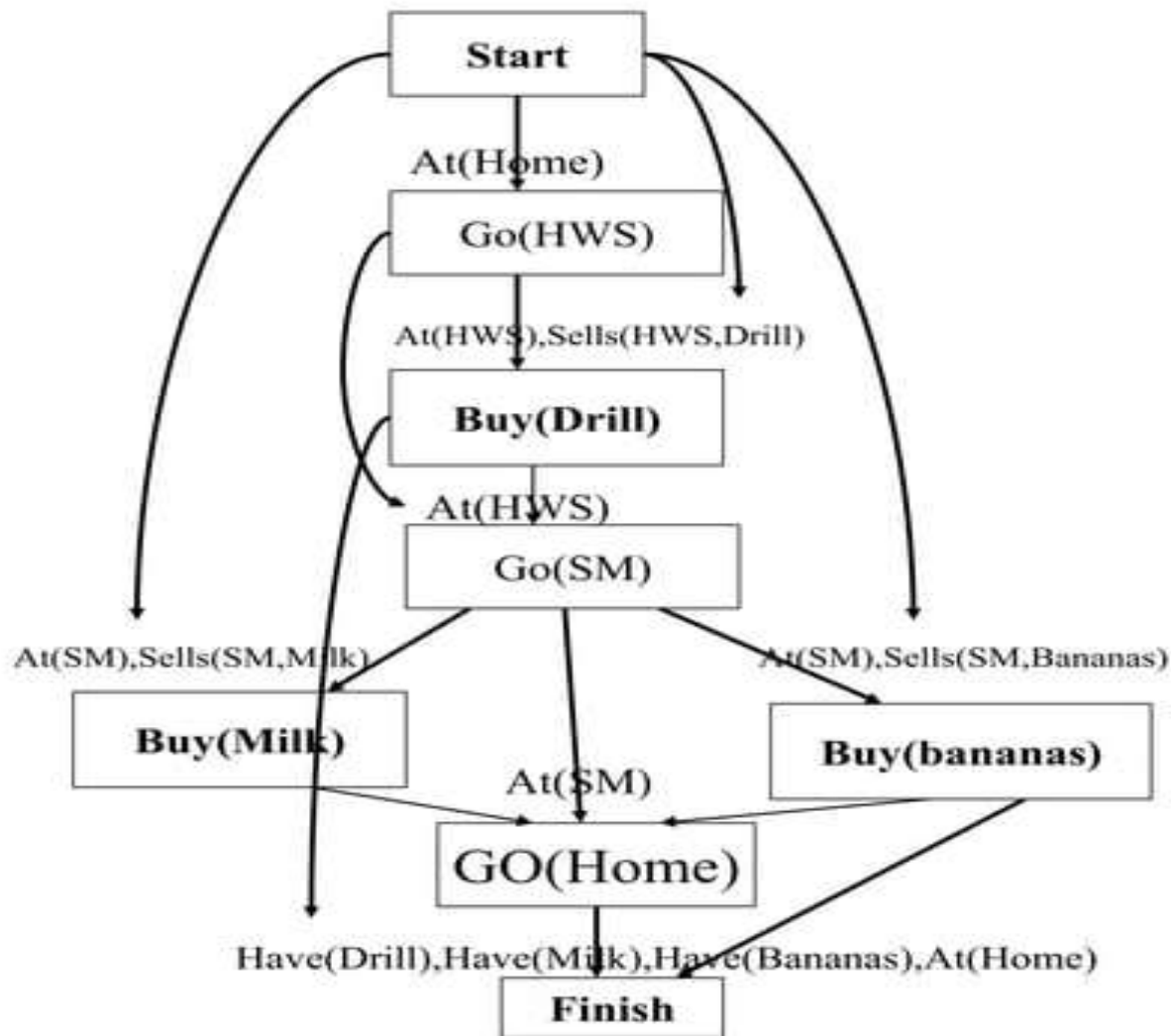
The way to resolve the Threat is to add ordering constraints to make sure that $S3$ does not intervene between $S1$ and $S2$

Demotion -- $S3$ is placed before $S1$.

Promotion – $S3$ is placed after $S2$.









THANK YOU