Planning

A planning agent will construct plans to achieve its goals, and then execute them.

Analyze a situation in which it finds itself and develop a strategy for achieving the agent's goal.

Achieving a goal requires finding a sequence of actions that can be expected to have the desired outcome.

Problem Solving

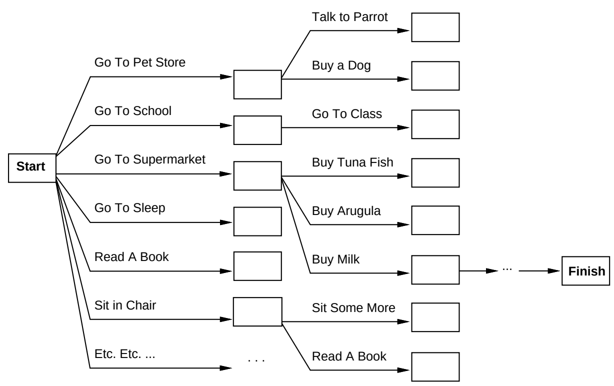
Representation of actions – actions generate successor states

Representation of states – all state representations are complete

Representation of goals – contained in goal test and heuristic function

Representation of plans – unbroken sequence of actions leading from initial to goal state.

Planning Example



GOAL: Get a quart of milk and a bunch of bananas and a variable-speed cord-less drill.

Planning Versus Problem Solving

(1) Open up the representation of states, goals and actions.

- States and goals represented by sets of sentences – $Have(Milk)$
- Actions represented by rules whose consequent specifies an action and its effect: $Buy(Milk) \rightarrow Have(Milk)$

This allows the planner to make direct connections between states and actions.

Planning Versus Problem Solving

(2) Planner is free to add actions to the plan wherever they are needed, rather than in an incremental sequence starting at the initial state.

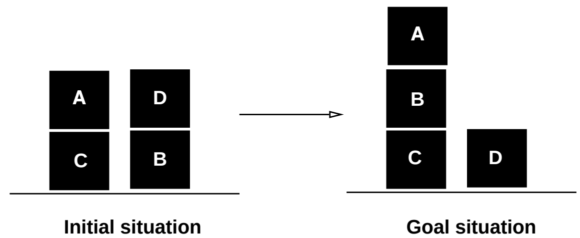
- No connection between the order of planning and the order of execution.
- Representation of states as sets of logical sentences makes this freedom possible.

Planning Versus Problem Solving

(3) Most parts of the world are independent of most other parts.

- Can solve $Have(Milk) \wedge Have(Bananas) \wedge Have(Drill)$ using divide-and-conquer strategy.
- Can re-use subplans (go to supermarket)

Planning as a Logical Inference Problem



Axioms:

on(a,c), on(c,table), on(d,b), on(b,table), clear(a), clear(d)
plus rules for moving things around...

Prove: on (a,b) $\wedge$ on(b,c)

Planning as Deduction: Situation Calculus

In first-order logic, once a statement is shown to be true, it remains true forever.

Situation calculus: way to describe change in first-order logic. Each assertion specifies the situation in which the assertion is true.

$on(A, C)$	$on(A, C, S_0)$
$on(C, Table)$	$\exists s_f on(C, Table, s_f)$

Actions: *place*-on-table

$\forall s \forall x[\neg on(x, Table, s) \rightarrow on(x, Table, place(x, s))]$

$\forall s \forall y \forall z[on(y, z, s) \wedge \neg equal(z, Table) \rightarrow \neg on(y, Table, s)]$

The Frame Problem and Its Relatives

Actions don't specify what happens to objects not involved in the action, but the logic framework requires that information.

$$\forall s \forall x [\neg on(x, Table, s) \rightarrow on(x, Table, place(x, s))]$$

Frame axioms: Inform the system about preserved relations.

$$\forall s \forall x \forall y \forall z [on(x, y, s) \wedge \neg equal(x, z) \rightarrow on(x, y, place(z, s))]$$

representational frame problem: proliferation of frame axioms

inferential frame problem: have to carry each property through all intervening situations during problem-solving, even if the property remains unchanged throughout

qualification problem: difficult, in the real world, to define the circumstances under which a given action is guaranteed to work

ramification problem: proliferation of *implicit* consequences of actions.

The Need for Special Purpose Algorithms

So...We have a formalism for expressing goals and plans and we can use resolution theorem proving to find plans.

Problems:

- frame problem
- time to find plan can be exponential
- logical inference is semi-decidable
- resulting plan could have many irrelevant steps

We'll need to:

- restrict language
- use a special purpose algorithm called a planner

The STRIPS Language

States and Goals: Conjunctions of function-free literals.

Have (Milk) $\wedge$ Have (Bananas) $\wedge$ Have (Drill)
$\wedge$ At (Home)

Operators:

action description: name for action; command to environment.

preconditions: conjunction of atoms that must be true before the operator can be applied.

effects: *add* list and *delete* list

STRIPS Operators

Op 1: Move block x from block y to block z

preconds: $on(x, y) \wedge clear(x) \wedge clear(z)$

effects: Add: on(x,z), clear(y)

Delete: on(x,y), clear(z)

Op 2: Move block x from block y to Table

preconds: $on(x, y) \wedge clear(x)$

effects: Add: on(x,Table), clear(y)

Delete: on(x,y)

Op 3: Move block x from Table to block z

preconds: $on(x, Table) \wedge clear(x) \wedge clear(z)$

effects: Add: on(x,z)

Delete: on(x,Table), clear(z)

Plan by Searching for a Satisfactory Sequence of Operators

situation space planner searches through space of possible situations

progression planner searches forward from the initial situation to the goal situation

regression planner search backwards from the goal state to the initial state

Searching Plan Space

Alternative is to search through the space of *plans* rather than the space of *situations*.

Start with simple, incomplete **partial plan**; expand until complete.

Operators: add a step, impose an ordering on existing steps, instantiate a previously unbound variable.

Refinement Operators take a partial plan and add constraints

Modification Operators are anything that is not a refinement operator; take an incorrect plan and debug it.

Representation for Plans

Goal: $RightShoeOn \wedge LeftShoeOn$

Initial state: λ

Operators:

Action	Preconds	Effect
RightShoe	RightSockOn	RightShoeOn
RightSock	λ	RightSockOn
LeftShoe	LeftSockOn	LeftShoeOn
LeftSock	λ	LeftSockOn

Partial Plans

Partial Plan: RightShoe LeftShoe

Principle of **Least Commitment** says to only make choices about things that you currently care about.

Partial order planner – can represent plans in which some steps are ordered and others are not.

Total order planner considers a plan a simple list of steps

A linearization of P is a totally ordered plan that is derived from a plan P by adding ordering constraints.

Defer Variable Binding

Planners must commit to bindings for variables

Example: Goal: Have(Milk) Action: Buy(item,store)

Principle of Least Commitment: Only make choices about things that you care about, leaving other details to be worked out later.

Buy(Milk,K-MART) versus Buy(Milk,store)

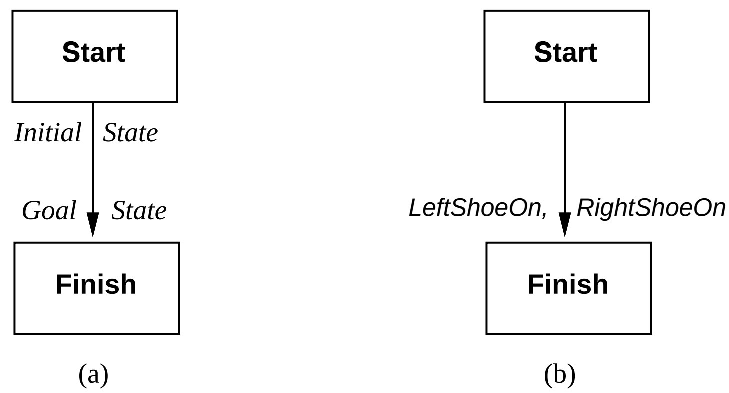
Fully instantiated plan: every variable is bound to a constant.

Definition of a Plan

- A set of plan steps (operators).
- A set of step ordering constraints of the form $S_i \prec S_j$
- A set of variable binding constraints
- A set of causal links, written as $S_i \xrightarrow{c} S_j$

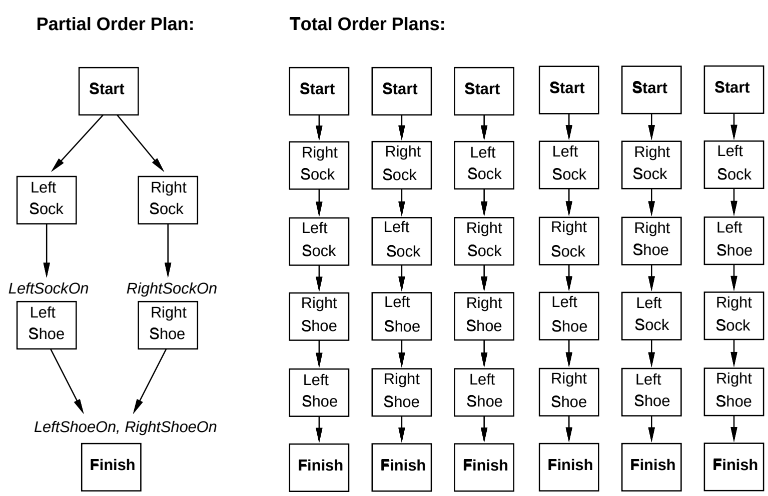
Initial Plan for Shoes and Socks

Initial plan: $Start \prec Finish$



Slide CS472 – Planning 21

Partial Plan for Shoes and Socks



Slide CS472 – Planning 22

Planner Output

A solution is a complete, consistent plan.

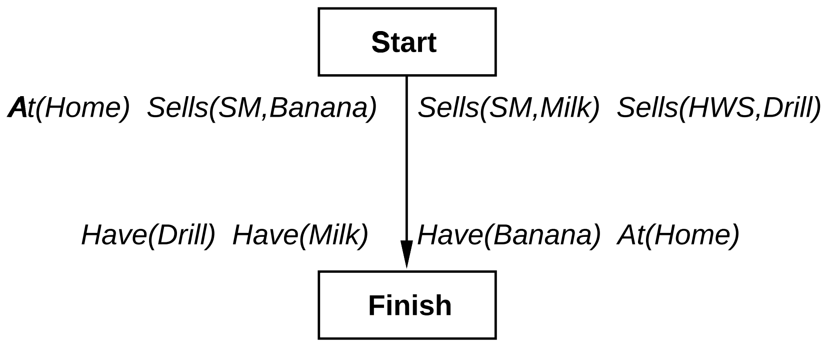
A complete plan: every precondition of every step is achieved by some other step.

A consistent plan: there are no contradictions in the ordering or binding constraints. Contradiction occurs when both $S_i \prec S_j$ and $S_j \prec S_i$.

POP Example: STRIPS Operators

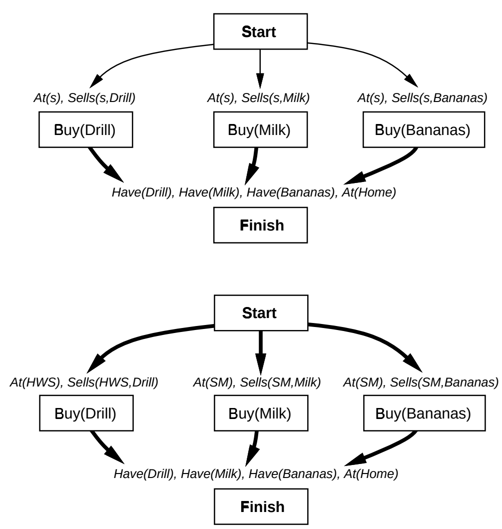
Action	PreCond	Effect
Go(*there*)	At(*here*)	At(*there*) $\land$ $\neg$At(*here*)
Buy(x)	At(*store*) $\land$ Sells(*store*, x)	Have(x)

POP Example: Initial Plan



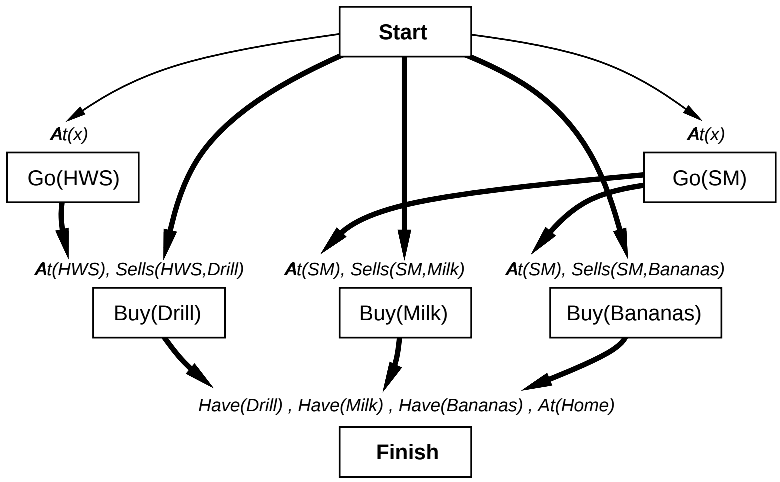
Slide CS472 – Planning 25

A Partial Plan I



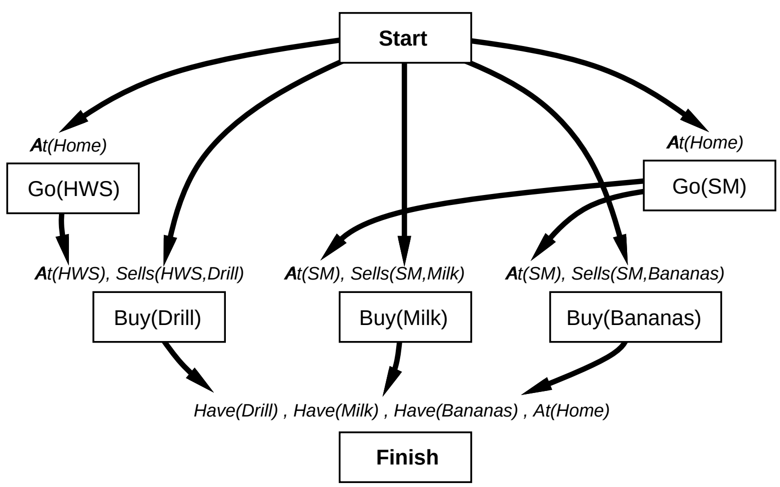
Slide CS472 – Planning 26

A Partial Plan II



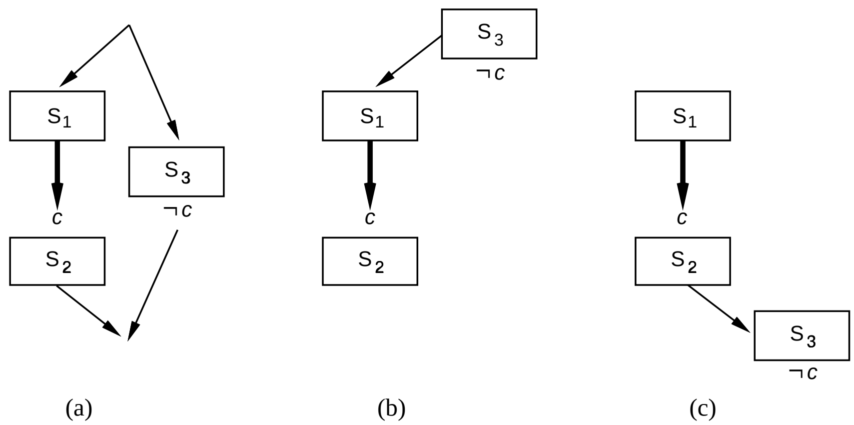
Slide CS472 – Planning 27

A Partial Plan III



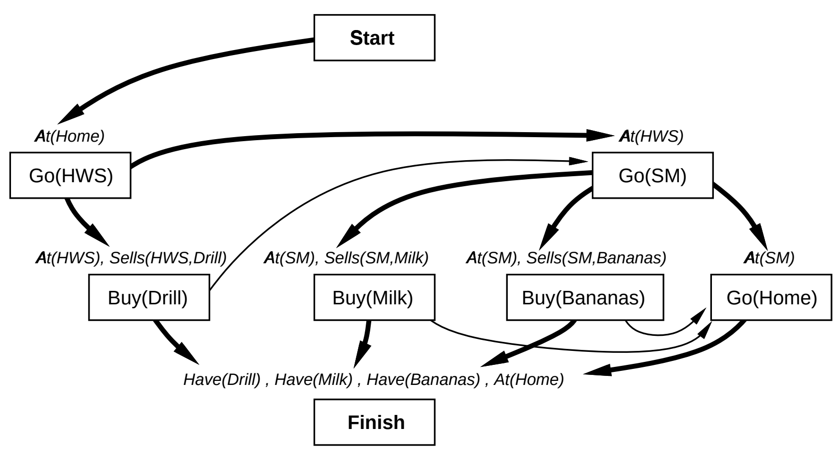
Slide CS472 – Planning 28

Protecting Causal Links



Slide CS472 – Planning 29

A Partial Plan III



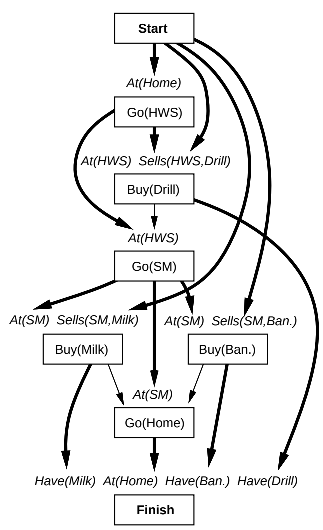
Slide CS472 – Planning 30

Achieving At(Home)

Candidate link	Threats
At(x) to initial state	Go(HWS), Go(SM)
At(x) to Go(HWS)	Go(SM)
At(x) to Go(SM)	At(SM) preconds of Buy(Milk), Buy(Bananas)

Solution: Link At(x) to Go(SM), but order Go(Home) to come after Buy(Bananas) and Buy(Milk).

A final Plan



function POP(*initial, goal, operators*) **returns** *plan*

plan ← MAKE-MINIMAL-PLAN(*initial, goal*)
loop do
  if SOLUTION?(*plan*) **then return** *plan*
  S_{need}, c ← SELECT-SUBGOAL(*plan*)
  CHOOSE-OPERATOR(*plan, operators*, S_{need}, c)
  RESOLVE-THREATS(*plan*)
end

---

function SELECT-SUBGOAL(*plan*) **returns** S_{need}, c

pick a plan step S_{need} from STEPS(*plan*)
  with a precondition c that has not been achieved
return S_{need}, c

---

procedure CHOOSE-OPERATOR(*plan, operators*, S_{need}, c)

choose a step S_{add} from *operators* or STEPS(*plan*) that has c as an effect
if there is no such step **then fail**
add the causal link $S_{add} \xrightarrow{c} S_{need}$ to LINKS(*plan*)
add the ordering constraint $S_{add} \prec S_{need}$ to ORDERINGS(*plan*)
if S_{add} is a newly added step from *operators* **then**
  add S_{add} to STEPS(*plan*)
  add $Start \prec S_{add} \prec Finish$ to ORDERINGS(*plan*)

---

procedure RESOLVE-THREATS(*plan*)

for each S_{threat} that threatens a link $S_i \xrightarrow{c} S_j$ in LINKS(*plan*) **do**
  choose either
    Promotion: Add $S_{threat} \prec S_i$ to ORDERINGS(*plan*)
    Demotion: Add $S_j \prec S_{threat}$ to ORDERINGS(*plan*)
  if not CONSISTENT(*plan*) **then fail**
end

Slide CS472 – Planning 33

Strengths of Partial-Order Planning Algorithm

- Takes a huge state space problem and solves in only a few steps.
- Least commitment strategy means that search only occurs in places where sub-plans interact.
- Causal links allow planner to recognize when to abandon a doomed plan without wasting time exploring irrelevant parts of the plan.

Slide CS472 – Planning 34

Practical Planners

STRIPS approach is insufficient for many practical planning problems. Can't express:

resources: Operators should incorporate resource consumption and generation. Planners have to handle constraints on resources efficiently.

time: Real-world planners need a better model of time.

hierarchical plans: Need the ability to specify plans at varying levels of detail.

Also need to incorporate heuristics for guiding search.

Spacecraft Assembly, integration and verification (AIV)

- OPTIMUM-AIV used by the European Space Agency to AIV spacecraft.
- Generates plans and monitors their execution – ability to re-plan is the principle objective.
- Uses O-Plan architecture – like partial-order planner, but can represent time, resources and hierarchical plans. Accepts heuristics for guiding search and records its reasons for each choice.

Scheduling for Space Missions

- Planners have been used by the ground teams for the Hubble space telescope and for the Voyager, Uosat-II and ERS-1.
- Goal: coordinate the observational equipment, signal transmitters and altitude and velocity-control mechanism in order to maximize the value of the information gained from observations while obeying resource constraints on time and energy.

Planning in a Hierarchy of Abstraction Spaces

- Ignore the detail of the full problem.
- Ideally work done at one level is not disturbed by the next level.
- If this assumption is true and if level i's plan creates subproblems of equal size in level $i + 1$, then exponential speedup ensues.

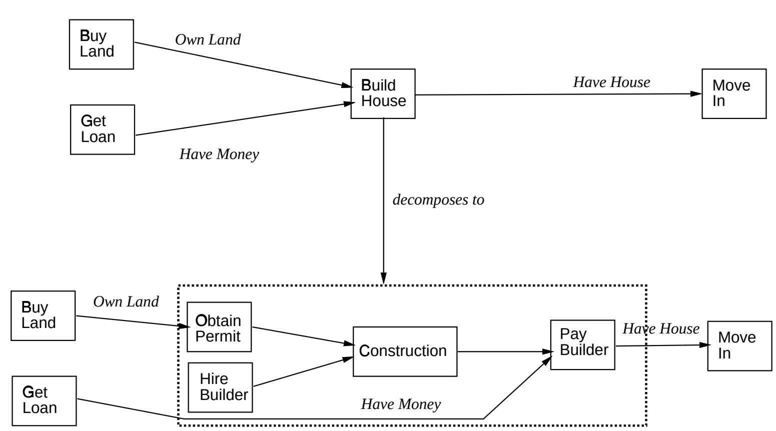
Hierarchical Decomposition

hierarchical decomposition: an *abstract operator* can be decomposed into a group of steps that forms a plan that implements the operator.

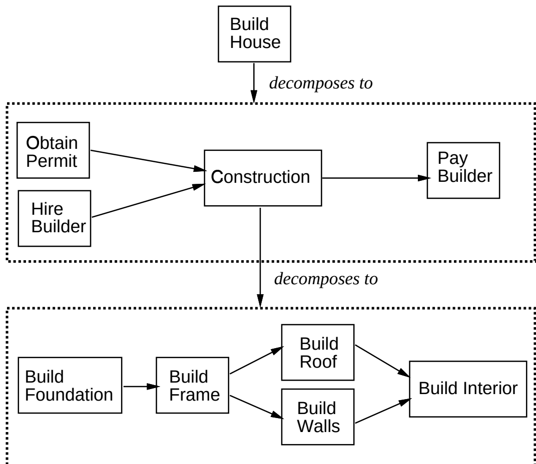
Requires two extensions to existing planner:

1. Extend STRIPS language to allow for nonprimitive operators.
2. Modify algorithm to allow the replacement of a nonprimitive operator with its decomposition.

Hierarchical Decomposition



Hierarchical decomposition of ***Build House*** operator



Slide CS472 – Planning 41

Rules for Decomposition

A plan p correctly implements an operator o if it is a complete and consistent plan for the problem of achieving the effects of o given the preconditions of o:

1. p must be consistent.
2. Every effect of o must be asserted by at least one step of p (and not denied by some other, later step of p).
3. Every precondition of the steps in p must be achieved by a step in p or be one of the preconditions of o.

Slide CS472 – Planning 42

Adapting Previously Generated Plans

1. Given a plan, how can you *index* it so you can retrieve it for future situations?
2. Given a problem to solve, how can you retrieve the most similar stored plan?
3. How can you adapt the retrieved plan for the new situation?