Planning

CS472/CS473 – Fall 2005

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

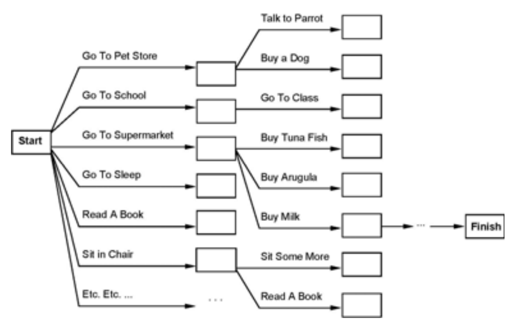
Representation of actions
actions generate successor states

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 vs. 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 that represent their preconditions and effects: *Buy*(*x*) achieves *Have*(*x*) and leaves everything else unchanged

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

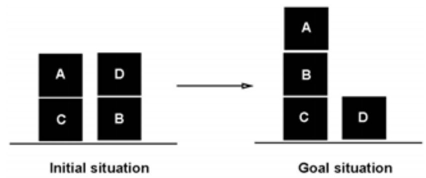
Planning vs. Problem Solving

2. **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 sub-plans (go to supermarket)

Planning vs. Problem Solving

3. **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 as a Logical Inference Problem



Axioms:

On(A,C) On(C,Table), On(D,B), On(B,Table), Clear(A), Clear(D)

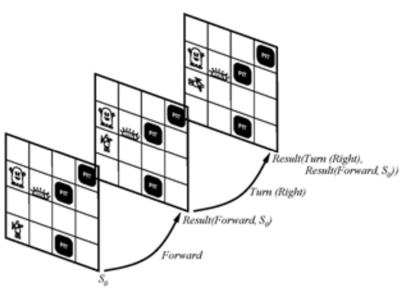
Plus rules for moving things around…

Prove: On (A,B) ∧ 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.



Situation Calculus

Fluents: functions and predicates that vary from one situation to the next

on(*A*,*C*)	*on*(A,C,S_0)
at(*agent*,[1,1])	*at*(*agent*, [1,1], S_0)

Atemporal functions and predicates: true in any situation

block(A)

gold(G_1)

Situation Calculus: Actions

Actions are described by stating their effects.

Possibility Axiom: *preconditions → Poss(a,s).*

$$\forall s \forall x \neg On(x, Table, s) \wedge Clear(x,s) \Rightarrow Poss(PlaceOnTable(x), s)]$$

Effect Axiom: *Poss(a,s) → Changes that result from action.*

$$\forall s \forall x Poss(PlaceOnTable(x), s) \Rightarrow On(x, Table, Result(PlaceOnTable(x), s))$$

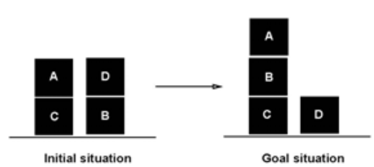
Situation Calculus: Action Sequences

We'd like to be able to prove:

$$\exists\, seq\ \ On(A,B,Result(seq,S_0)) \wedge On(B,C,Result(seq,S_0))$$

Which would produce, for example, the following:

$$On(A,B,Result(Put(A,B),Result(Put(B,C),Result(PoT(D),Result(PoT(A),S_0)))))$$
$$\wedge$$
$$On(B,C,Result(Put(A,B),Result(Put(B,C),Result(PoT(D),Result(PoT(A),S_0)))))$$



Situation Calculus: Problem

Axioms:

On(A,C,S_0) On(C,Table,S_0), On(D,B,S_0), On(B,Table,S_0), Clear(A,S_0), Clear(D,S_0)

$\forall s \forall x \neg On(x,Table,s) \wedge Clear(x,s) \Rightarrow Poss(PlaceOnTable(x),s)]$

$\forall s \forall x Poss(PlaceOnTable(x), s) \Rightarrow$
$On(x, Table, Result(PlaceOnTable(x), s))$

Prove:

1. On(A,Table,Result(PoT(A),S0))
2. On(D,B,Result(PoT(A),S0))

The Frame Problem

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

$$\forall s \forall x Poss(PoT(x), s)) \Rightarrow On(x, Table, Result(PoT(x), s))$$

Frame Axioms: Inform the system about preserved relations.

$$\forall s \forall x \forall y \forall z[(On(x, y, s) \wedge (x \neq z)) \Rightarrow On(x, y, Result(PoT(z), s))]$$

… and Its Relatives

Representational Frame Problem: proliferation of frame axioms.

Solution: use successor-state axioms

Action is possible $\Rightarrow$ *(Fluent is true in result state* $\Leftrightarrow$ *(Action's effect made it true* $\vee$ *It was true before and action left it alone)).*

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