# Planning

Lee Spector, lspector@hampshire.edu

# The General Problem Solver (GPS)

From Norvig: The General Problem Solver, developed in 1957 by Alan Newell and Herbert Simon, embodied a grandiose vision: a single computer program that could solve any problem, given a suitable description of the problem.

This is a piece of "real AI," at least in the eyes of Newell and Simon.

"It is not my aim to surprise or shock you. ... But the simplest way I can summarize is to say that there are now in the world machines that think, that learn and create. Moreover, their ability to do these things is going to increase rapidly until--in a visible future--the range of problems they can handle will be coextensive with the range to which the human mind has been applied." <quoted in Norvig p.109>

# Means-Ends Analysis

The problem is to model a general form of reasoning called "means-ends analysis." Newell and Simon trace means-ends analysis back to Aristotle, and they believe it to be a central form of reasoning in human thought. Here is an example of means-ends analysis in action:

*I want to take my son to nursery school. What's the difference between what I have and what I want? One of distance. What changes distance? My automobile. My automobile won't work. What is needed to make it work? A new battery. What has new batteries? An auto repair shop. I want the repair shop to put in a new battery; but the shop doesn't know that I need one. What is the difficulty? One of communication. What allows communication? A telephone... and so on.* <Newell & Simon, quoted in Norvig p.111>

# Action Representation

“STRIPS” operators consist of preconditions, add-lists, and delete-lists:

- The precondition states what must be true for the operator to be applicable.
- The add-list specifies things that the operator will make true.
- The delete-list specifies things that the operator will make false.

# Example STRIPS Operator

Operator: Eat-Macaroni
Preconditions: Have-Macaroni, Hungry
Add-List: Macaroni-In-Stomach
Delete-List: Hungry, Have-Macaroni

# GPS/STRIPS Planning Algorithm

Achieve the conditions in the goal list one at a time.

Each goal condition that is already true by the time it is considered can simply be skipped.

For goals that must be achieved, find an operator with the goal condition in its add-list and recursively achieve all of its preconditions. Then “apply” the operator by changing the state of the world according to its add- and delete-lists.

# GPS/STRIPS Planning Algorithm

```
INPUT: a goal list, an initial state description, a set of operators

Set the World to be the initial state description
Set the Stack to contain the goal list
Set the Plan to be an empty list

LOOP
    If the Stack is empty terminate and report success with Plan as OUTPUT
    Else if an operator is on the top of the stack, then perform the operator:
        Delete the operator’s delete list from the World
        Add the operator’s add list to the World
        Add the operator to the end of the Plan
        Remove the operator from the Stack
    Else if the goal on the top of the Stack is satisfied in the World
        Remove the goal from the Stack
    Else
        Find an operator with the top goal in its add list
        Push the operator onto the top of the Stack
        Push all of its preconditions onto the top of the stack
```

# STRIPS

STRIPS is GPS w/variables & a theorem prover

example STRIPS style operator:

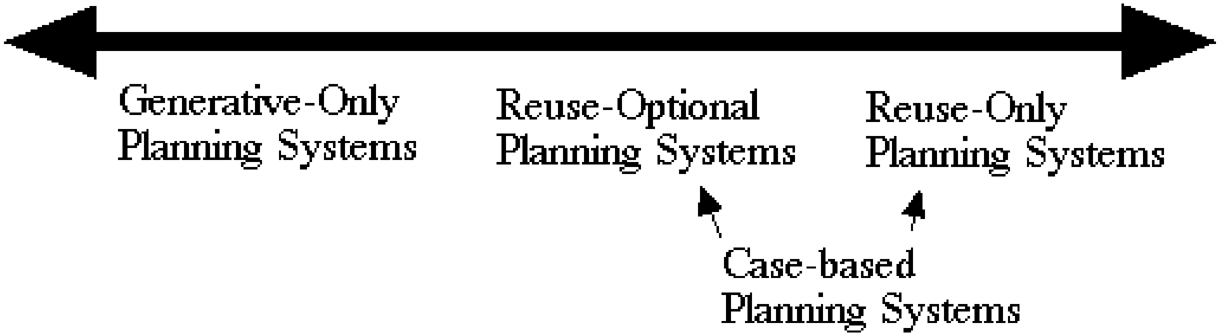
pickup (x)
precondition: ontable(x) & handempty & clear(x)
delete list: ontable(x), handempty, clear(x)
add list: holding(x)

# ABSTRIPS

- Abstraction-Based STRIPS
- Important stuff vs. details: "criticality"
- Complexity win by concentrating on important ("critical") stuff first.
- "skeletal plans"
- See Sacerdoti for details

# Case-Based Planning

- Planning from experience, mostly
- The Plan Reuse Spectrum of Planning Systems" [Kettler 1995]



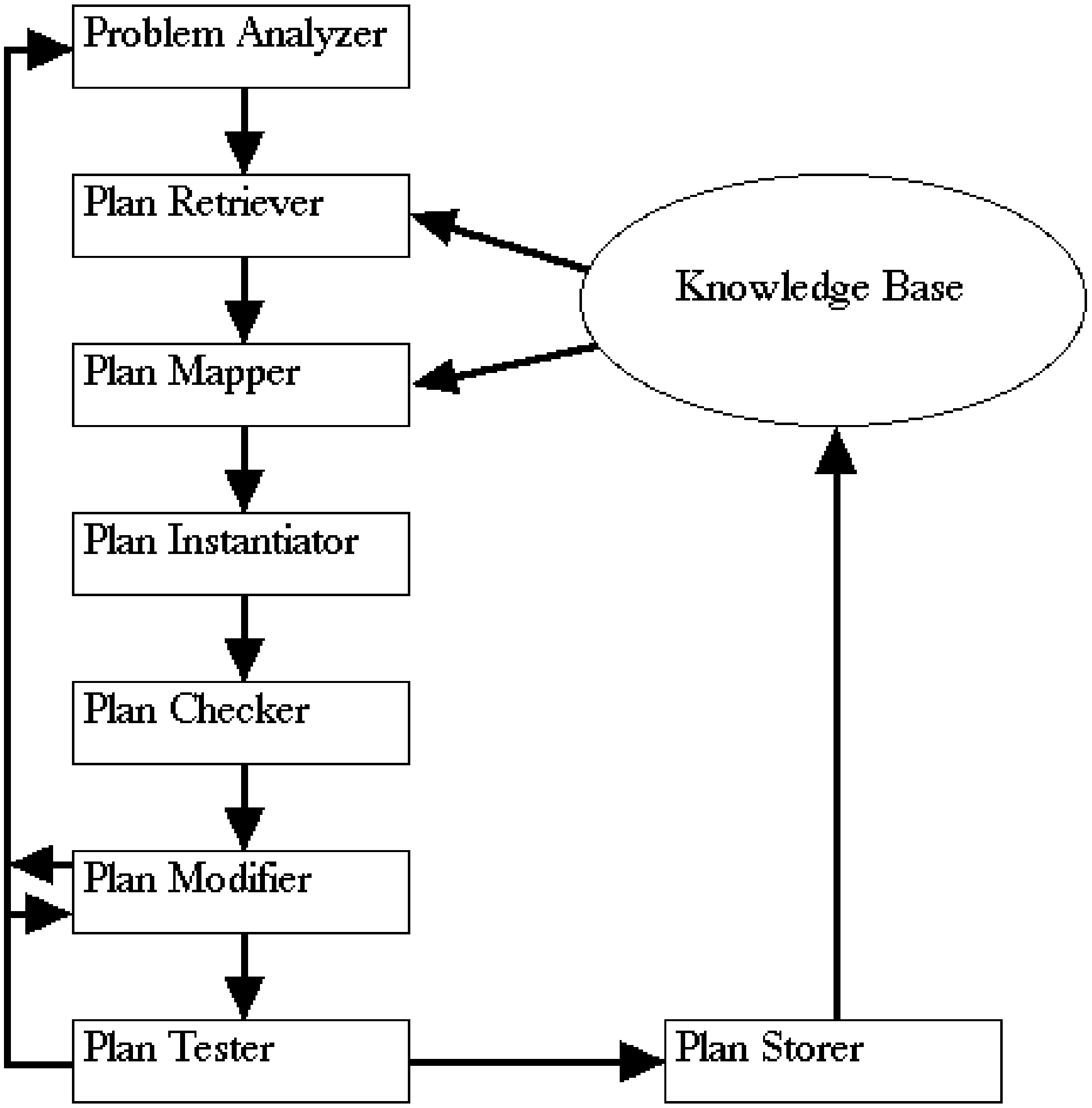
# Hacker: *The virtuous nature of bugs*

- Check for stored plan
- If not found, run planner
- "Criticize" for known bugs
- Store and index plan
- Try it
- If buggy classify mode of failure, summarize bug as new critic
- Patch plan

Kinds of bugs anticipated: failures of the "linear plan theory" that multiple subgoals can be achieved independently.

Prototypical bug patterns:
PREREQUISITE-CLOBBERS-BROTHER-GOAL (PCBG)
PREREQUISITE-CLOBBERS-SIBLING-GOAL (PCSG)

# CAPER [Kettler 1995]



# NOAH

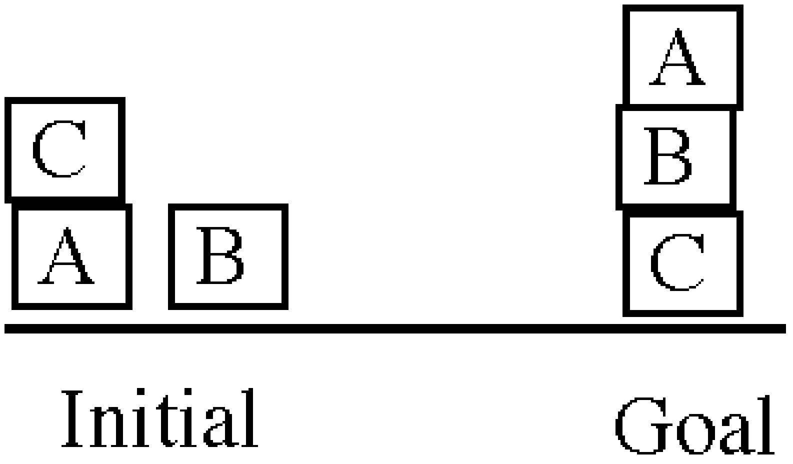
- Plans need not be linear -- use partial order and avoid ordering commitments until necessary.
- Plan execution linear.
- Plan generation need not be.

# Sussman Anomaly

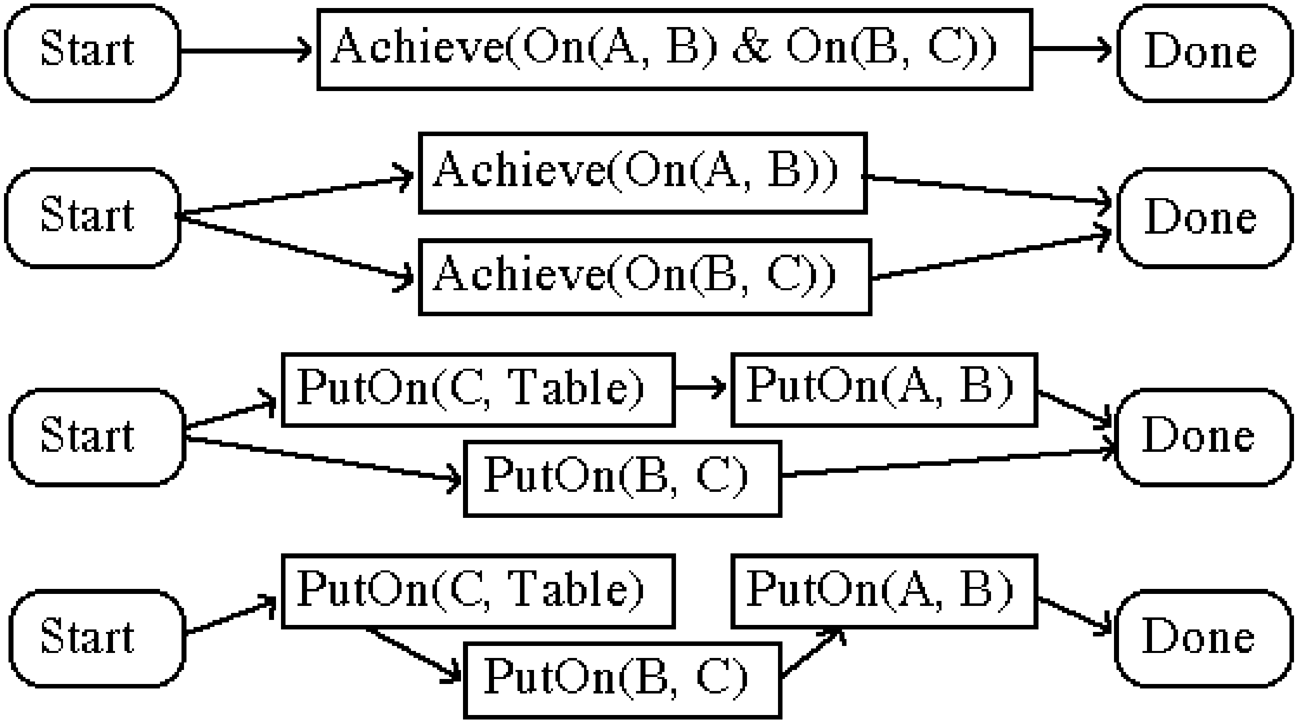
INITIAL: ((ON C A)(ON A TABLE)(ON B TABLE)(CLEAR C)(CLEAR B))

GOALS: ((ON A B) (ON B C) (ON C TABLE))

Single operator:
puton (x,y)
pre: (clear x), (clear y)
del: (clear y)
add: (on x y)



# Partial Order Planning



# More Recent Planners

- GraphPlan
- SATPlan
- Integrated planning/execution systems.

# The Frame Problem

- More knowledge -> more to consider before acting.
- Relevance filter necessary: "it does not appear that people take more time to make simple decisions about acts like reaching for the salt when they know more about, say, chemistry." <Pylyshyn>
- Frame problem: knowing what will not change as a result of a given action.
- Frame axioms.

# Janlert on the Frame Problem

The *prediction problem* and the *revision problem*: how to make predictions reasonably accurate (in spite of the necessity of representing the world coarsely) and how to revise beliefs after failures.

The *general bookkeeping problem*: how to keep track of all of the plans and situations. "A symptom of deeper problems, one of which is the frame problem."

The *qualification problem*: how to specify all (∞) of the qualifications under which an action will work. (Rowboat example, p. 13).

The **"frames" solution**: independent chunks of the world, specified by properties. e.g., the property is-red belongs to chunk 1, the property is-heavy belongs to chunk 2, etc. Problems: Chameleon.

Hayes's **"causal connection" solution**: independent chunks of the world specified by causal connection. Still problematic and ad hoc.

The **"consistency" solution**: "The world is basically inert."

New World := Max-Consistent(new-stuff U old-stuff)

problems: undecidability of consistency for 1st order theories, precise definition of Max-Consistent is problematic, more detail needed.

The **STRIPS solution**: "All that is not explicitly changed by an action remains unchanged." Brute force consistency solution. One problem: "parallel effects" that depend on context. Example: pushing saucer or cup when cup is on saucer. Deeper statement of problem: context independence of effects.