Planning

Anthony Kapolka
CS 340 – Fall 2011
Wilkes University

Problem Solving vs. Planning

In everyday language planning means to figure out several problem-solving steps before executing any of them...

... of course, the computer usually can't do anything but plan...

Planning Framework

- Describe initial world state.
- Describe goal world state.
- Identify actions that transform states.

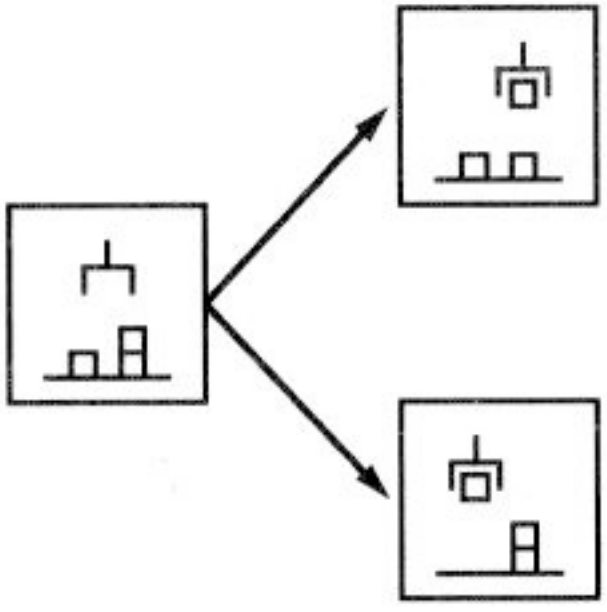
A plan is a sequence of actions which transforms initial state into goal state.

State Space or Plan Space?

State space – explore states

- Construct a list of states to achieve goal
- Apply actions/operators to generate new states
- Generate plan by identifying search path
- Extract plan by listing actions necessary for each state

State Space Search



Moves in the space:

- Modify world state via operator

Model of time:

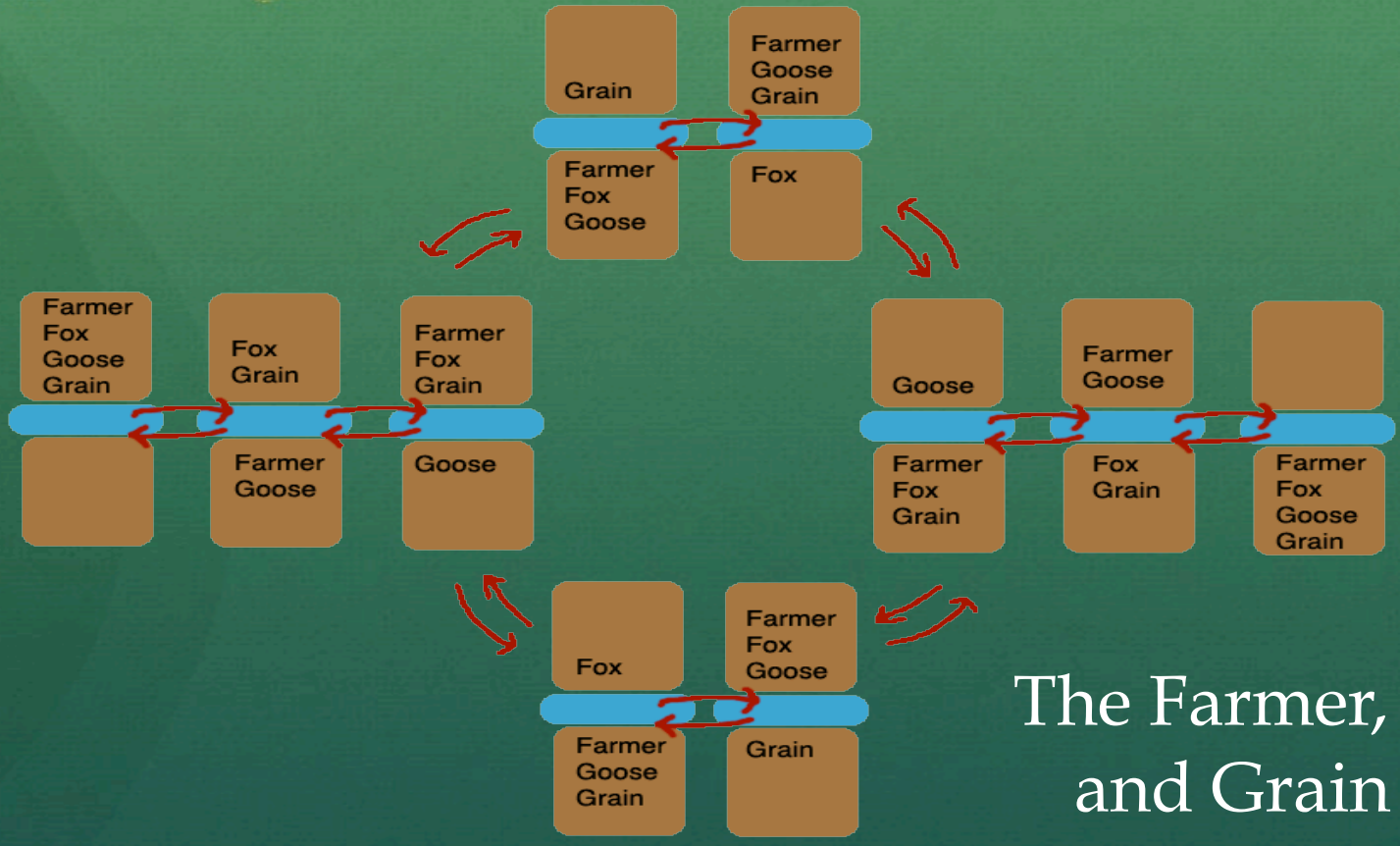
- Depth of node in search space

Plan stored in:

- Series of state transitions

[Rich & Knight 1991]

State Space Example



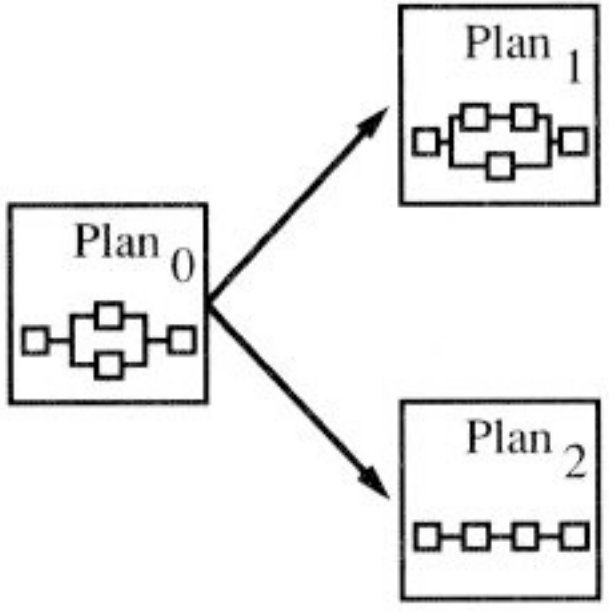
The Farmer, Fox, Goose and Grain problem.

State Space or Plan Space?

Plan space – consider actions

- States are plans (partial plans)
  - plans have actions, bindings, orderings
- Test to see if the plan achieves our goal
- Refine/extend plan with available actions as we search
- Extract plan by listing actions necessary for each state

State Space Search



Moves in the space:

- Add operators
- Order operators
- Bind variables
- Or otherwise constrain plan

Model of time:

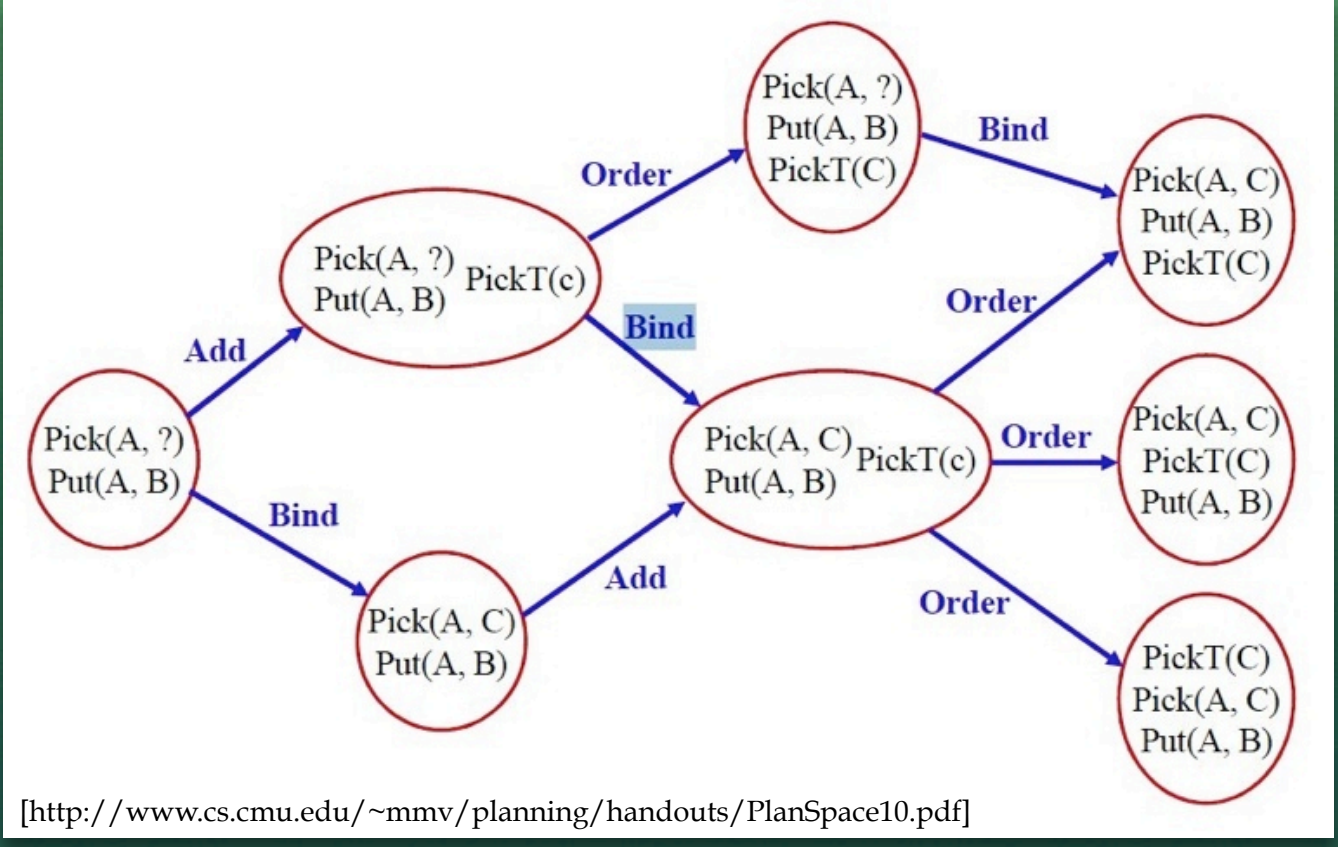
- Partially ordered set of operators

Plan stored in:

- Single node

[Rich & Knight 1991]

Plan Space Search

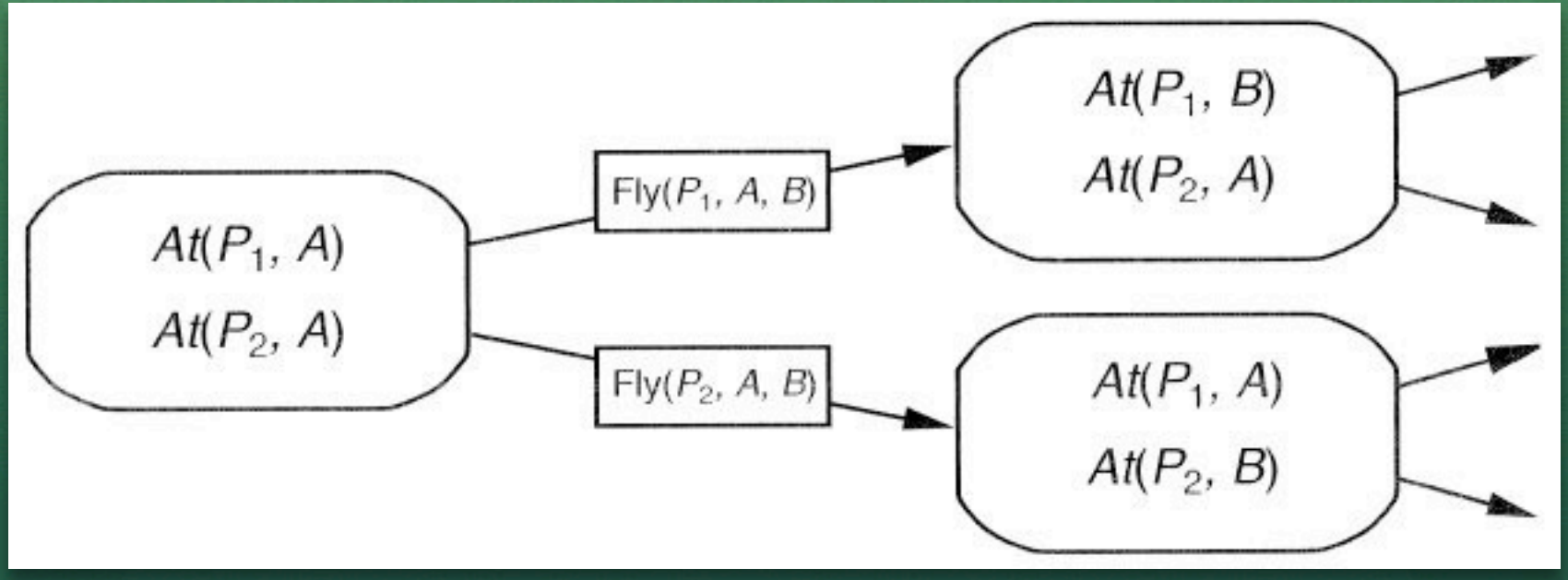


[http://www.cs.cmu.edu/~mmv/planning/handouts/PlanSpace10.pdf]

Forwards or Backwards?

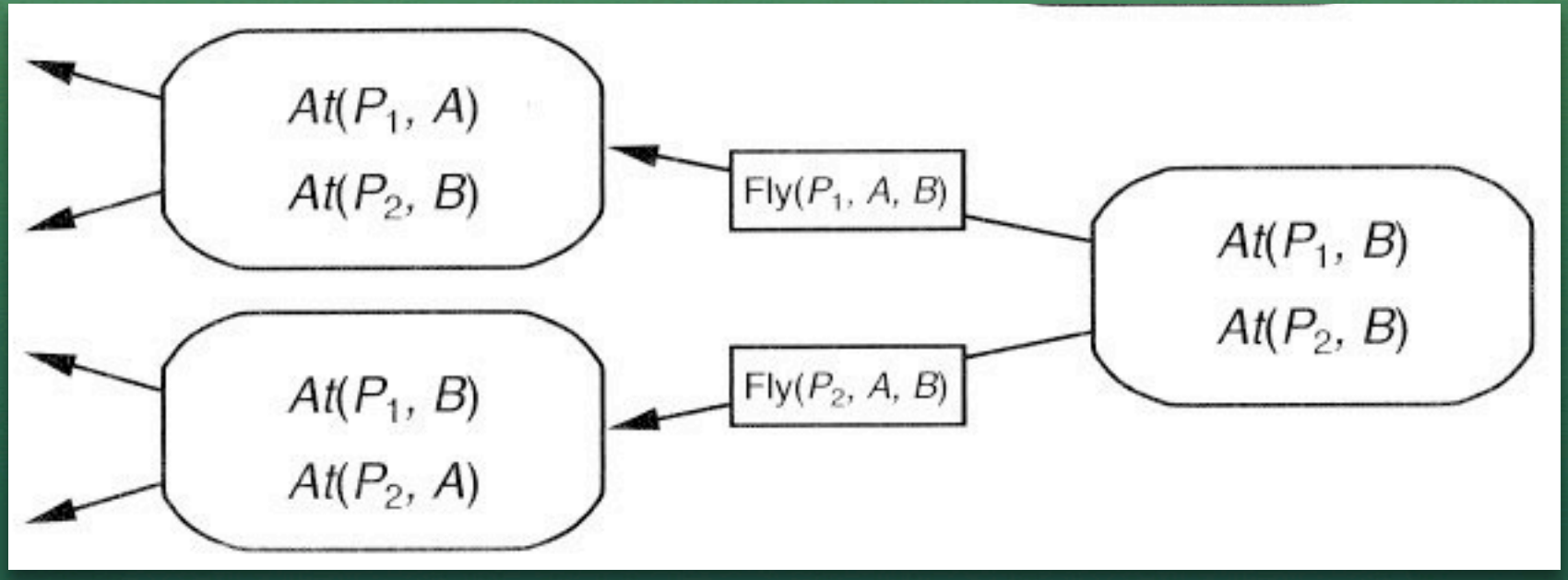
- Forward planning is usually more natural
- Backwards planning is often more efficient

Forward Planning



[Russel & Norvig 2010]

Backward Planning



[Russel & Norvig 2010]

From Whence Complexity?

- Searching the LARGE space
  - worst case, exponential in plan length
- Costs Associated with each Node
  - generating next (successor) nodes
  - identifying when goal state is reached
  - computing heuristics

usually a tradeoff between search time and work per node

Problem Decomposition

For more complicated problem domains

- Break problem into smaller parts
- Solve parts of the problem
- Recombine parts into complete solution

Problem Decomposition

Avoid recomputing things

- *The frame problem* – identifying constants

Divide difficult problem into smaller problems.

- Many problems are *nearly decomposable*.
- Identify and handle potential interactions.

Sussman's Anomaly

illustrates why planning
is non-trivial

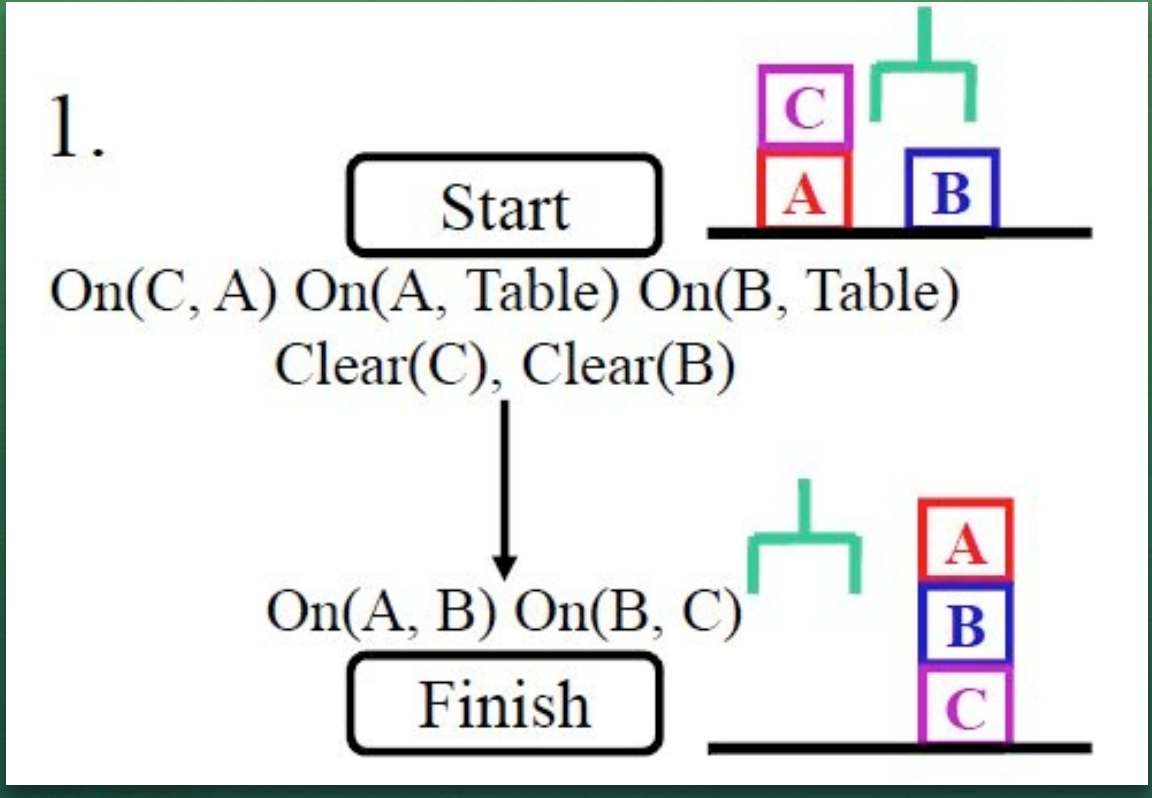
Gerald Sussman

MIT, under Marvin Minsky
developed Scheme
founding director, FSF

Sussman's Anomaly

[http://www.cs.cmu.edu/~mmv/planning/handouts/PlanSpace10.pdf]

Sussman's Anomaly

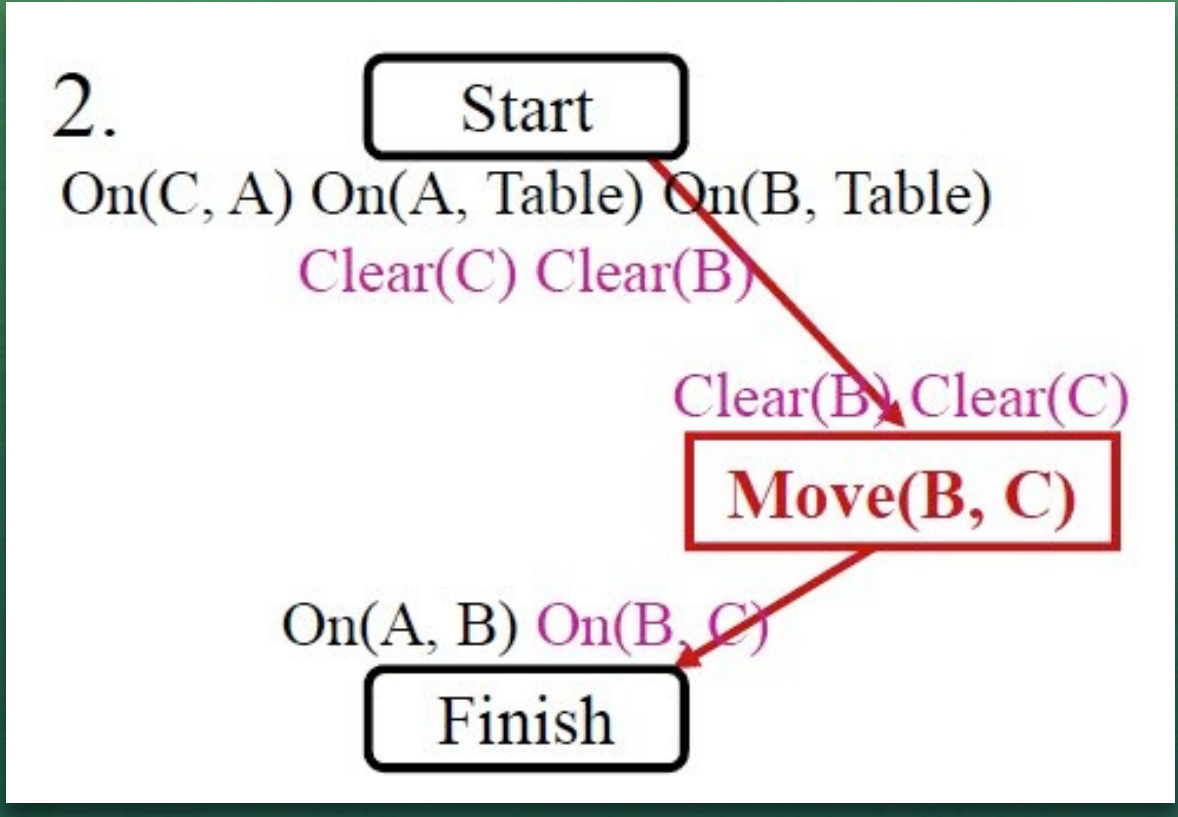


simple planners separate the goal into subgoals:

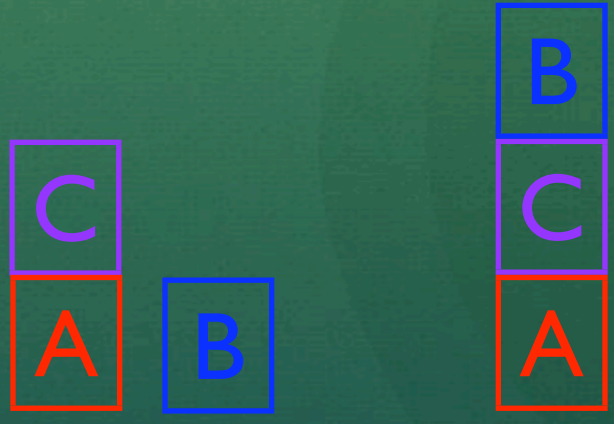
get A atop B
get B atop C

[http://www.cs.cmu.edu/~mmv/planning/handouts/PlanSpace10.pdf]

Sussman's Anomaly

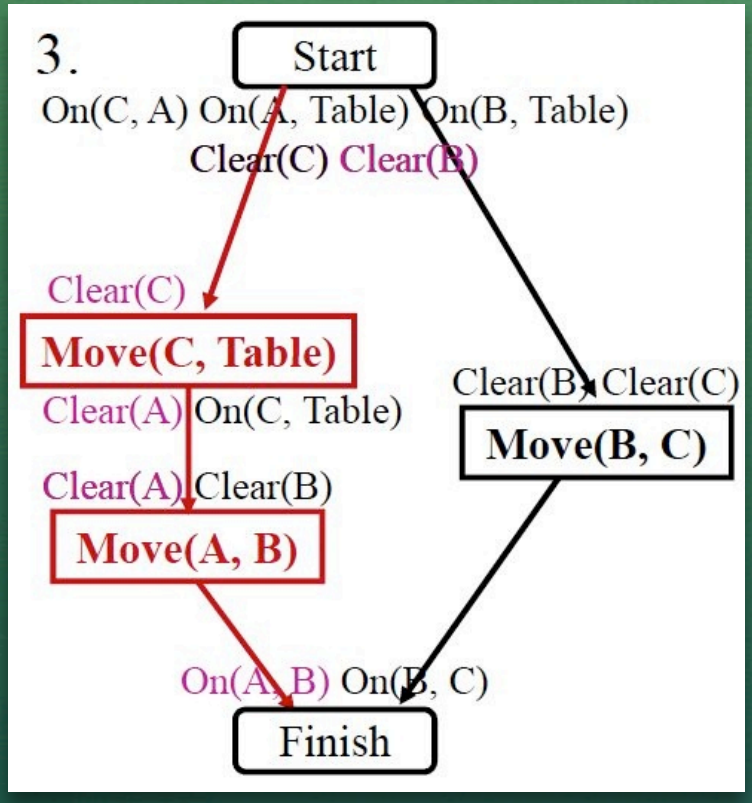


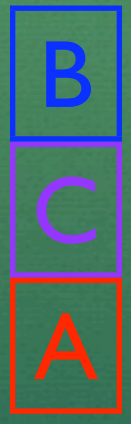
first, pursue
get B atop C



[http://www.cs.cmu.edu/~mmv/planning/handouts/PlanSpace10.pdf]

Sussman's Anomaly



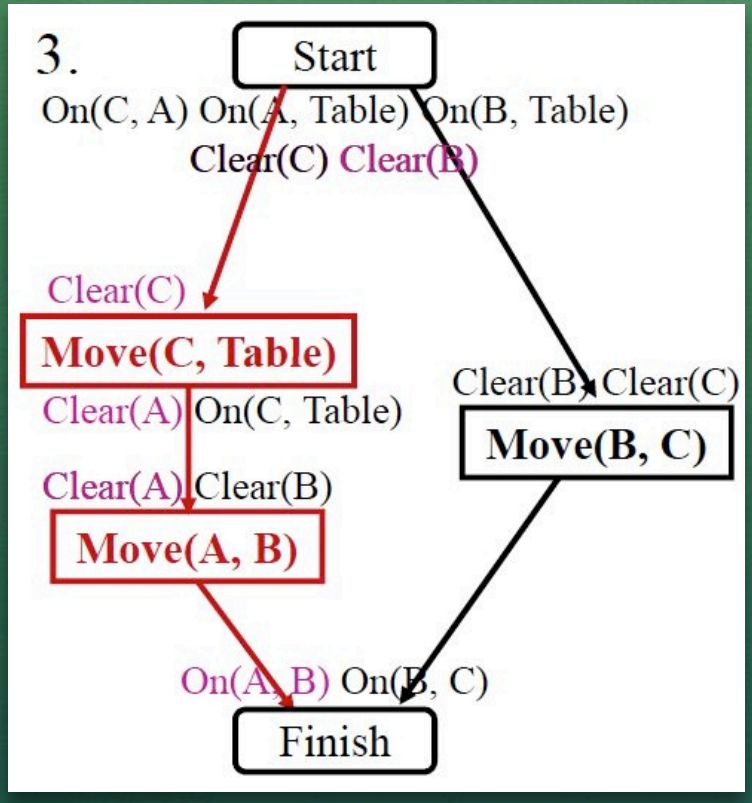


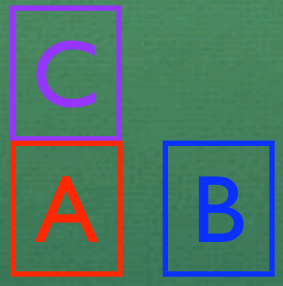
then, pursue
second subgoal
get A atop B

move(C, table)
move(A, B)

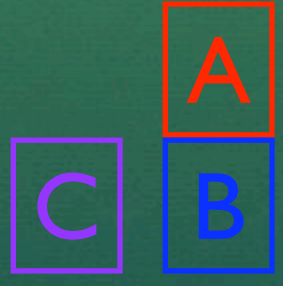
[http://www.cs.cmu.edu/~mmv/planning/handouts/PlanSpace10.pdf]

Sussman's Anomaly





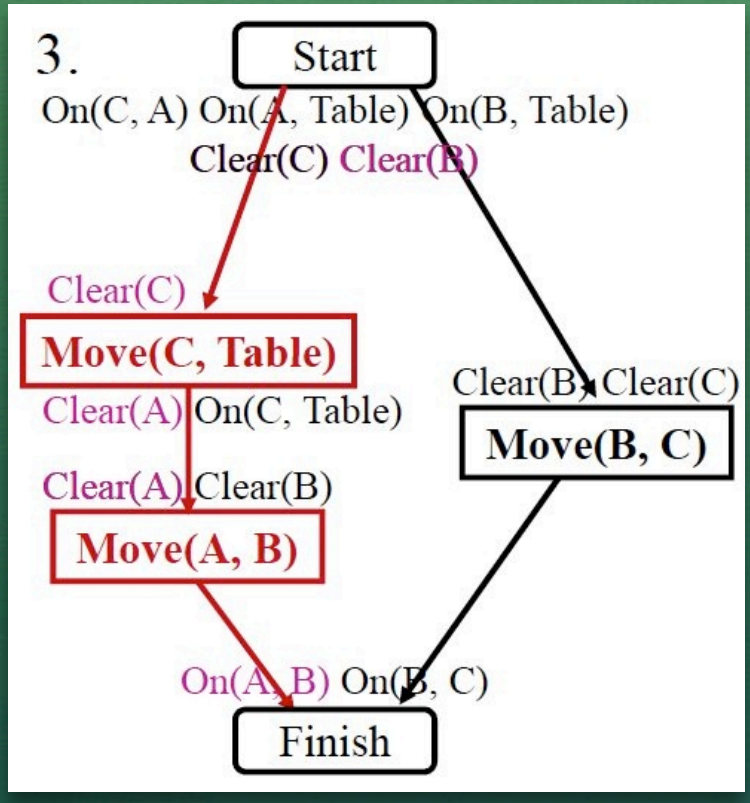
try second subgoal first
get A atop B

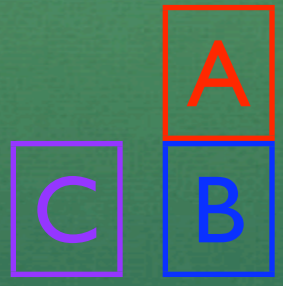


move(C, table)
move(A, B)

[http://www.cs.cmu.edu/~mmv/planning/handouts/PlanSpace10.pdf]

Sussman's Anomaly



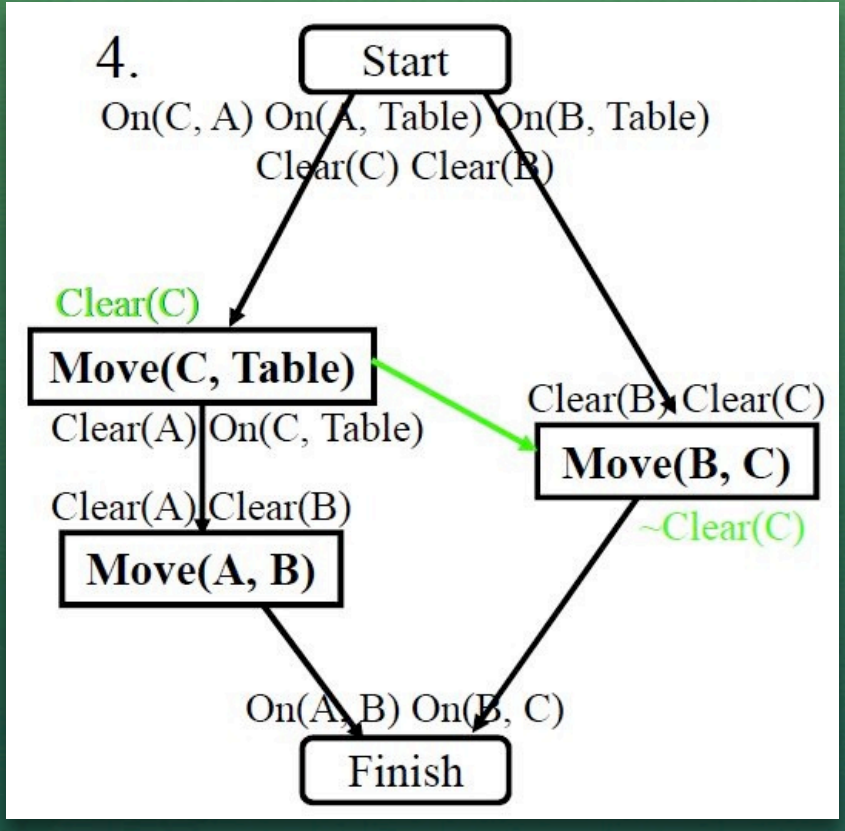


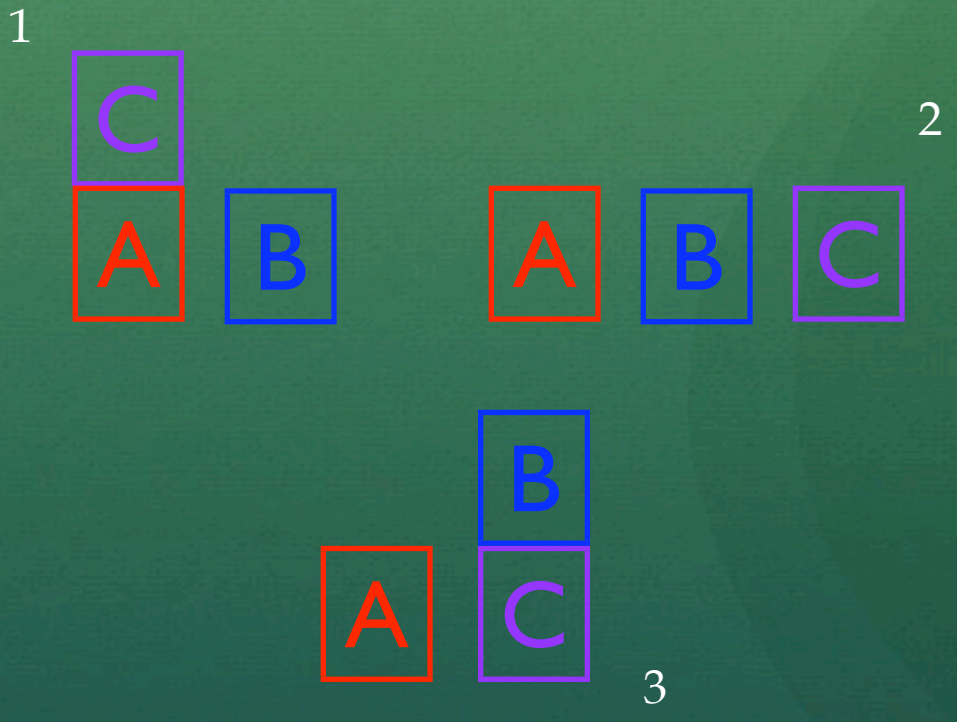
now first subgoal get B atop C

move(B, C)

[http://www.cs.cmu.edu/~mmv/planning/handouts/PlanSpace10.pdf]

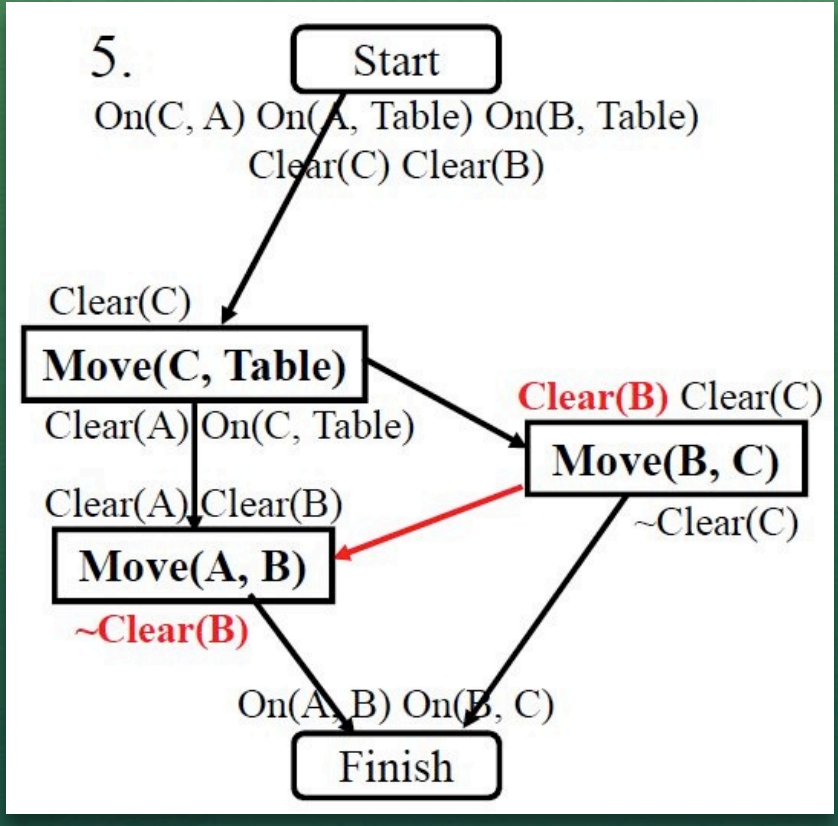
Sussman's Anomaly



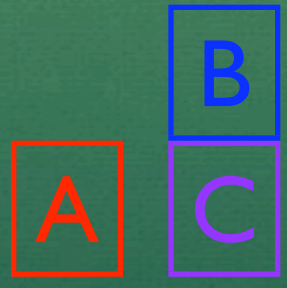


[http://www.cs.cmu.edu/~mmv/planning/handouts/PlanSpace10.pdf]

Sussman's Anomaly



3



4

required interleaving of subgoals

[http://www.cs.cmu.edu/~mmv/planning/handouts/PlanSpace10.pdf]

Plan Terminology

- Totally ordered plan

  - there is an ordering O of actions

- Fully instantiated plan

  - all variables are constrained, B

- Consistent plan

  - there are no contradictions in O or B

- Complete plan

  - all preconditions of all actions in O are met

Types of Planners

- Heuristic Search
- Planning Graph
- Satisfiability, Constraint Satisfaction
- First-order Logic Deduction
- Refining Partially Ordered Plans

Planning vs. Scheduling

- classic planning - what to do, in what order
- scheduling takes into account resource constraints
- one approach: plan first, schedule afterwards
- to minimize time, determine earliest start times for tasks
- want to find critical path
  - the path with the maximum shortest time to completion

Problems with Planning

- Symbolic approach poor for continuous problems
- Symbolic approach bad for making value judgments (like ranking two options)
- Assumptions: world well behaved, actions do succeed
- Ignores adversaries

More wrinkles

- Can steps be ignored or undone?
- In the real world, some actions are irrevocable
- Is the universe predictable?
- If not, computer simulation falters

So, in the real world…

- Don't plan too far ahead – just react to situation.
- Do our best – design a plan likely to succeed.
  - Still, plan may well fail.
  - Small surprises may not completely ruin plan.
  - How to react to salvage plan?

In games…

- We have a lot of control over the world
- Game actions are often monotonic
  (once performed effect is permanent.)
- Custom solutions implemented for hard subtasks
- Just have to “appear intelligent”