CPSC 322
Introduction to Artificial Intelligence

November 19, 2004

Chapter 6

Chapter 6 in your textbook starts out by talking about expert systems, but it quickly turns into a discussion of how to build the CILOG meta-interpreter

Rule-based systems are similar in some ways to deductive reasoning systems, but rule-based systems aren't restricted to logical inference

Still, we could write simple rule-based systems in CILOG if CILOG could write to a data base, but no...

...so we'll wave goodbye to expert systems and move ahead to Chapter 8, which is about...

Actions and Planning

There are a number of situations where you'd like your man-made artifacts to have some ability to make a plan and then act according to the plan

Actions and Planning

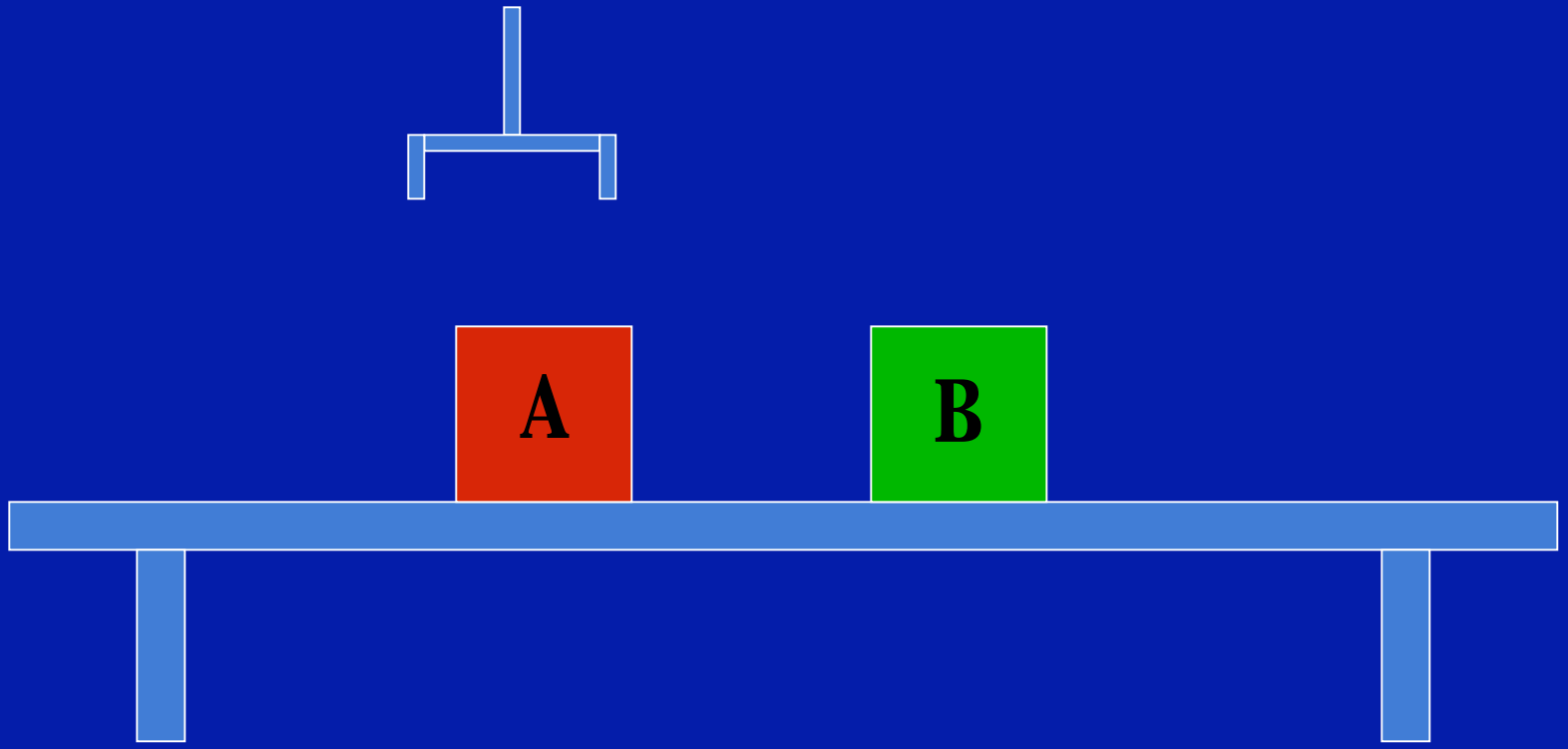
Planners that can do stuff like this....

Actions and Planning

...have their beginnings in stuff like this...

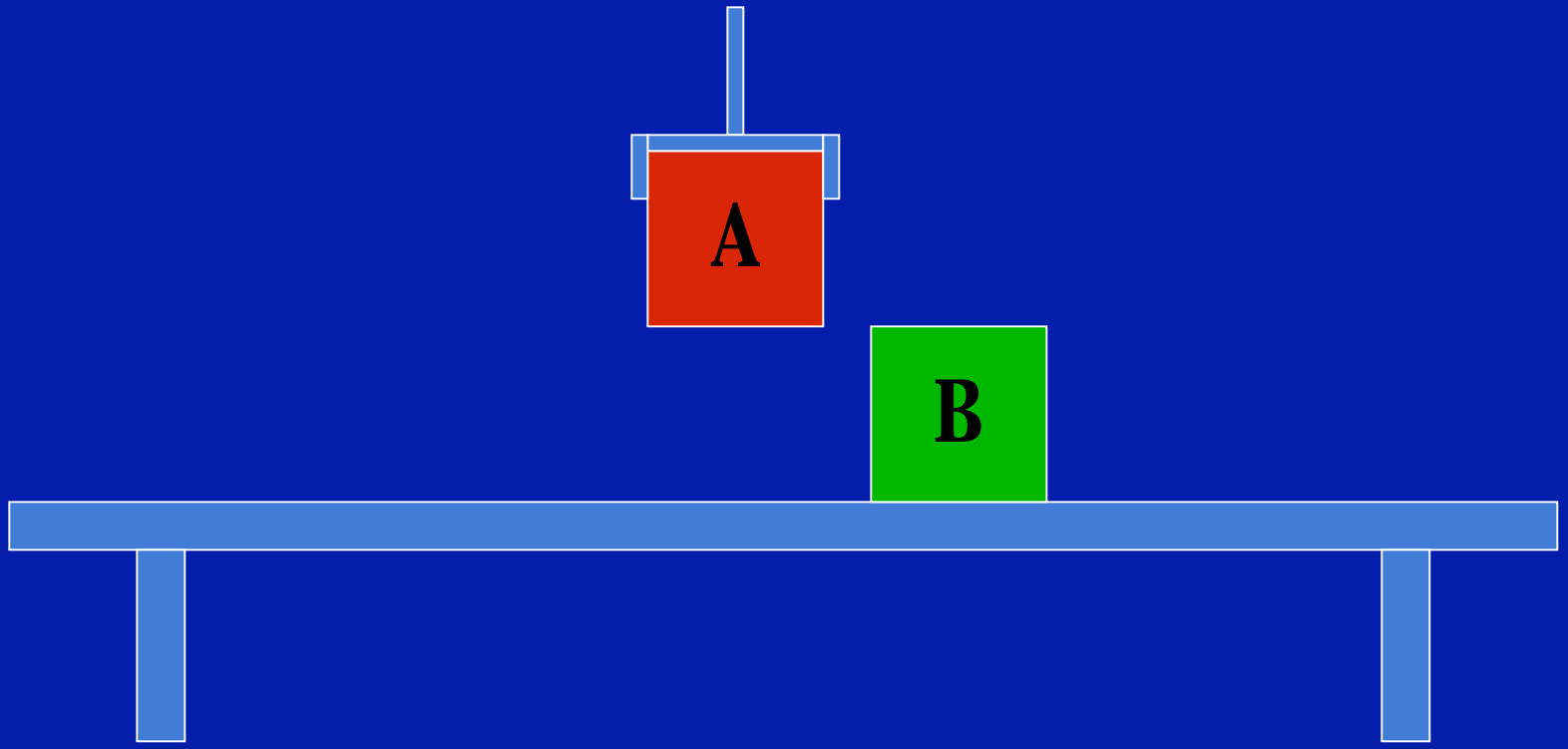
Actions and Planning

So we'll start at the beginning, and try to figure out how to get a computer to generate a plan to get from here....



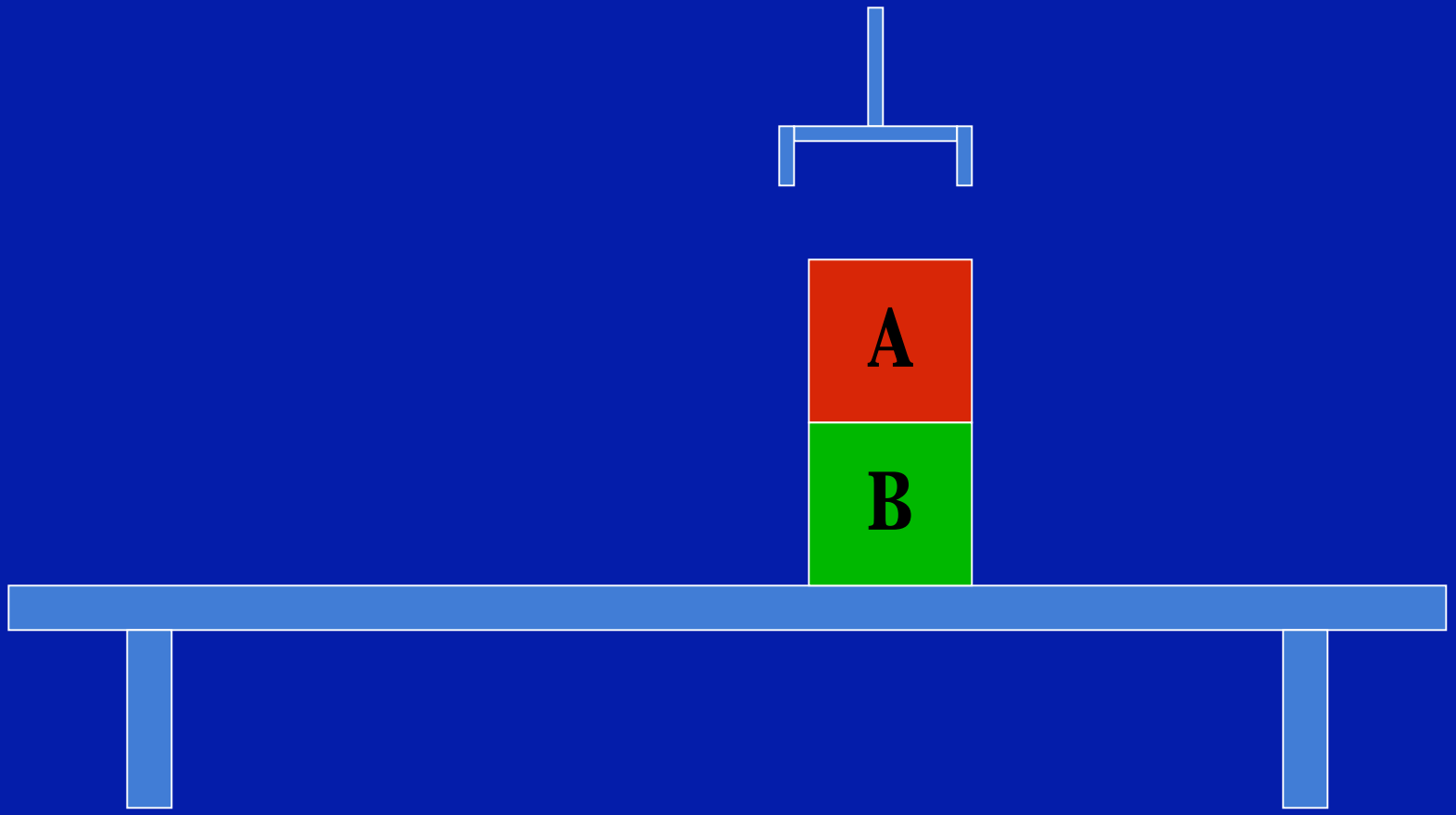
Actions and Planning

...to here...



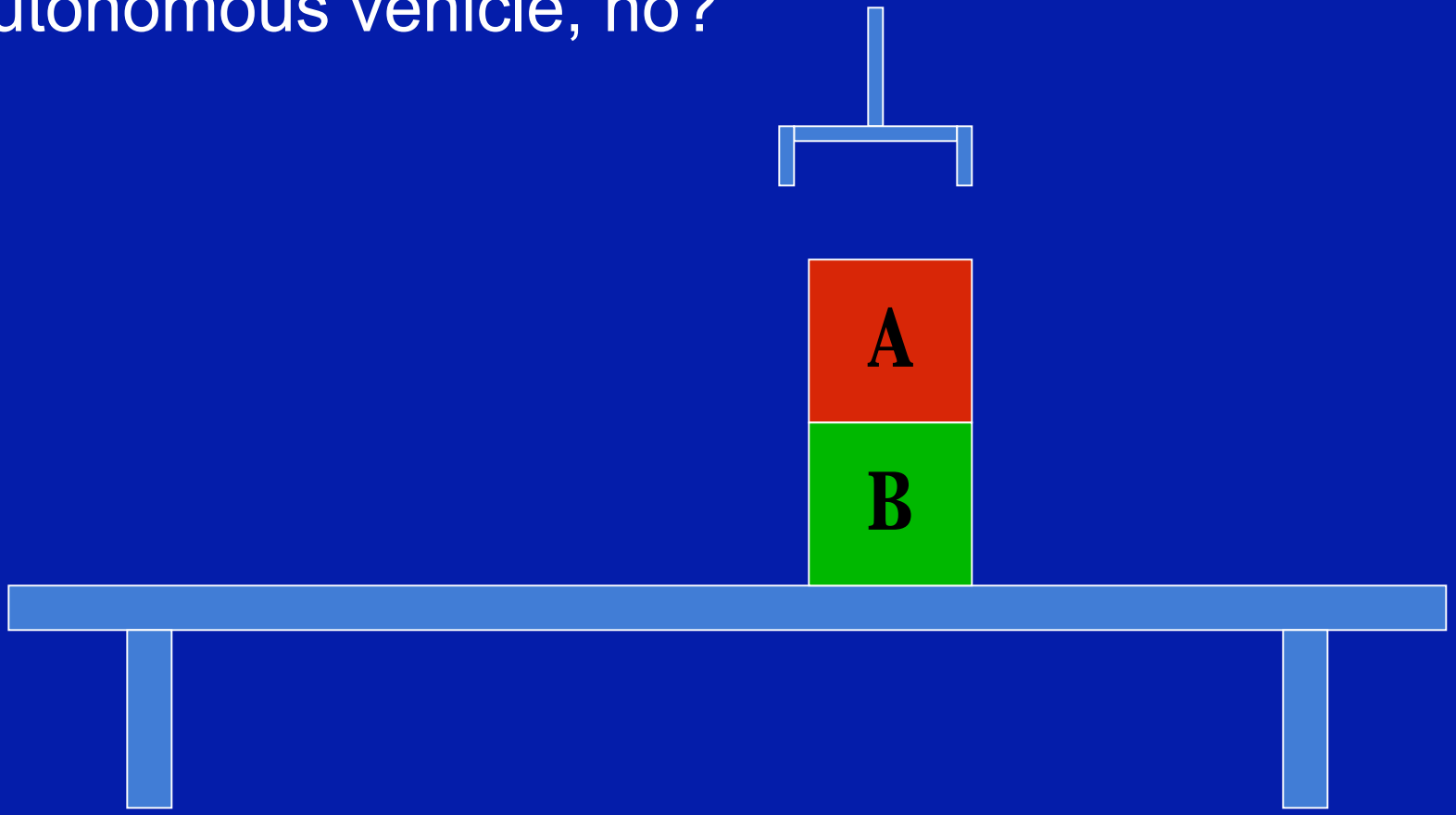
Actions and Planning

...and finally to here



Actions and Planning

Then you can scale this up to build your own autonomous vehicle, no?



Search in more robust problems

As we've seen in our little forays into language processing and rule-based systems, the solution is still search, but the nature of the search changes...

- Required knowledge gets bigger
- Our operators become more complex
- We mix our representation schemes

Search and planning

If we add to that bigger, more complicated domain the notion of an intelligent agent (a robot) that acts in and changes its world...

...along with the reality that the agent's actions can be expensive, irrevocable, and maybe even fatal...

...we then think about these problems and their solutions as "planning" instead of "search". It's still search, but the nature of the problem is different.

Planning

So planning is about computing some or all the steps of a solution to a problem before executing any of them in the real world. The sequence of steps that gets the agent from a start state to a goal state is a plan.

Planning

Good planning requires lots of knowledge:

- Current state
- Goal state
- Operators
    - when to use them (preconditions)
    - how they change the world (postconditions)
- Knowledge of the entire problem domain in advance (a “map” of the entire “terrain”, for instance)

Planning

This last nugget is problematic:

- Knowledge of the entire problem domain in advance (a “map” of the entire “terrain”, for instance)

because for all that knowledge to remain useful, the agent has to accurately track what is changed by an action as well as what isn’t.

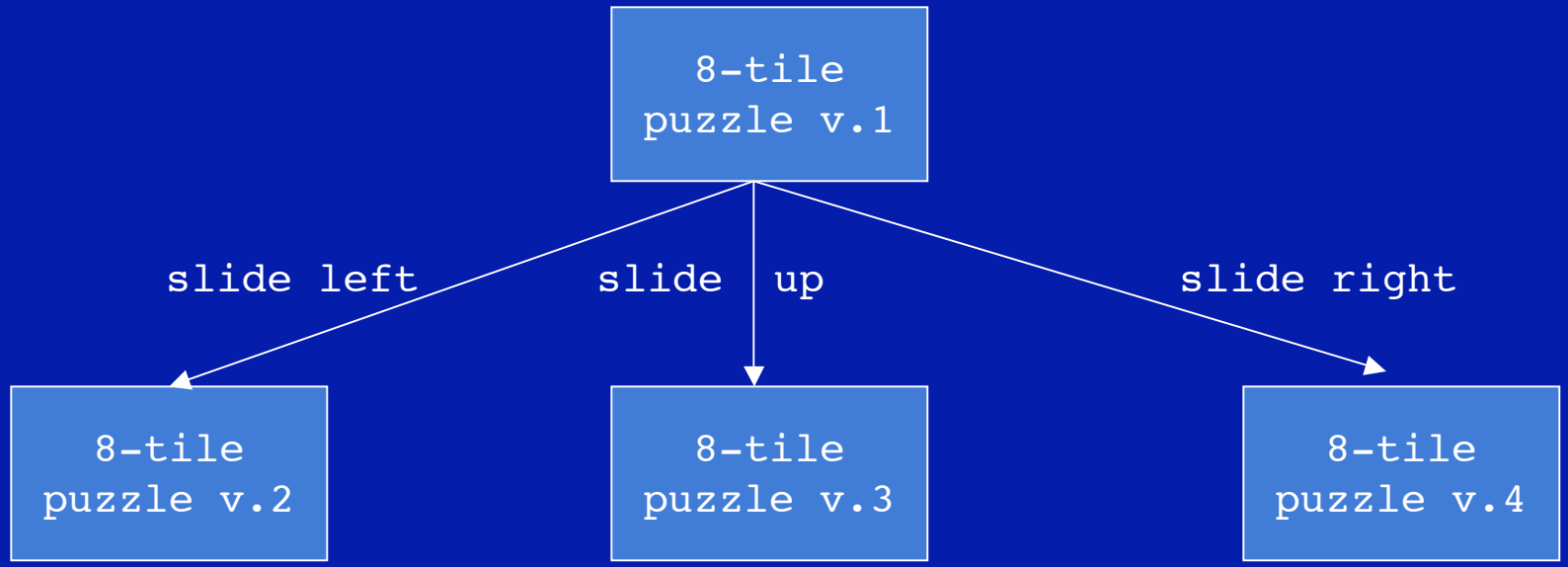
Planning

That hasn't been a problem for us so far, because our domains have been so small. When you're searching through possible changes to the 8-tile puzzle, you keep track of what changed and what didn't by generating a copy of the entire domain with the changes represented explicitly.

But what if the domain is, say, all the geological and topographic features of the surface of Mars for a 20km radius?

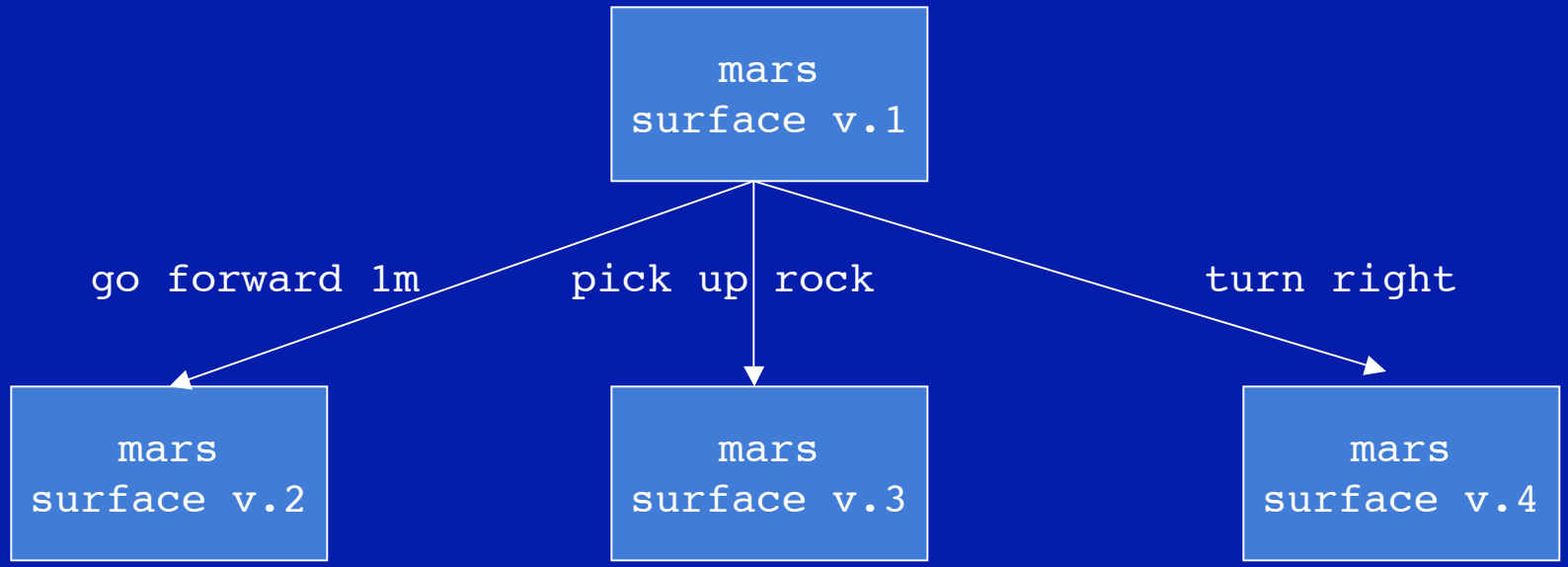
Planning

This is ok...



Planning

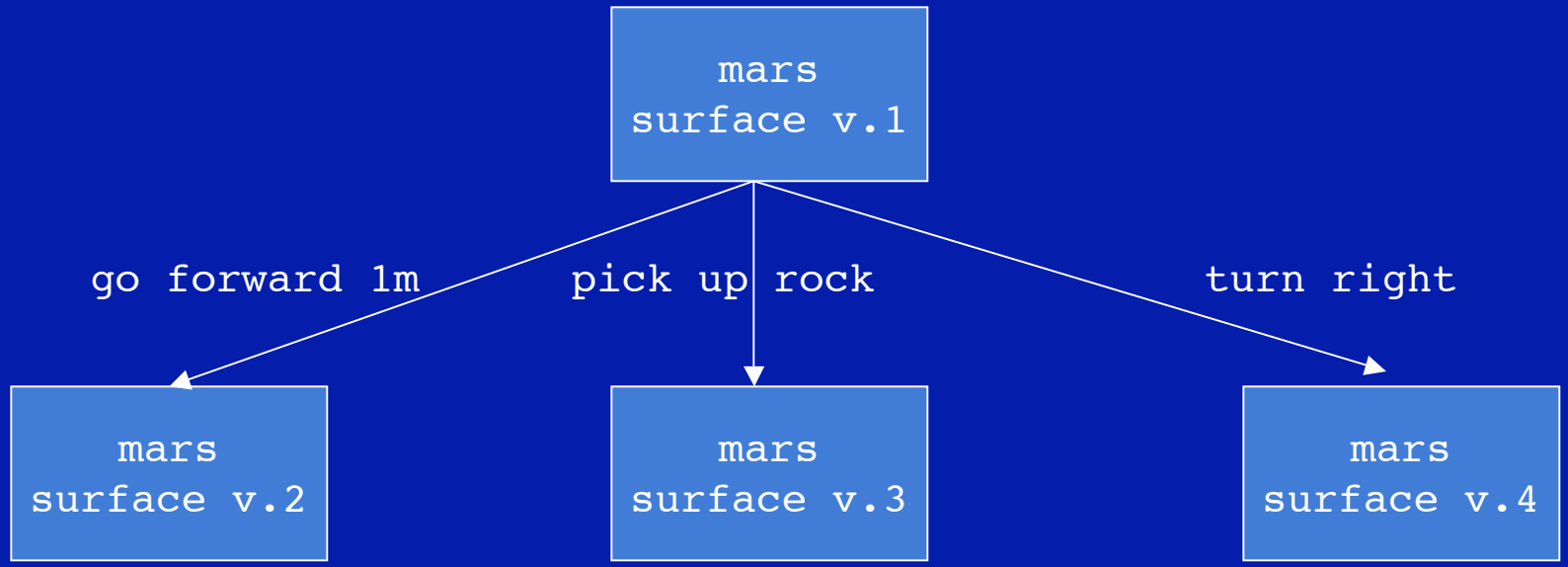
This is nasty...



...and it has a name

The frame problem

With massive amounts of knowledge to deal with...



...how do you concisely specify what doesn't change when an action is performed?

A simple solution to the frame problem

One way to solve this problem:

- Represent the initial state of the planning problem as you'd expect (lists of facts, relations, etc.)
- Represent subsequent states not as copies of the initial state with small changes, but as the sequence of actions that get you from the initial state to that new state
- Each action carries with it a list of things that the action changes when it's applied to a state
  - an add list says what's added to the description
  - a delete list says what's taken away

STRIPS in the blocks world

This approach was first used by the people who built the robot Shakey (you saw Shakey in the movie) to navigate from room to room.

The representation for actions and the algorithm for planning with that representation is collectively called STRIPS (STanford Research Institute Problem Solver)

Here's an application of STRIPS to a simpler blocks world problem (which doesn't really justify the use of this approach but it's easy to follow)

STRIPS: relations in blocks world

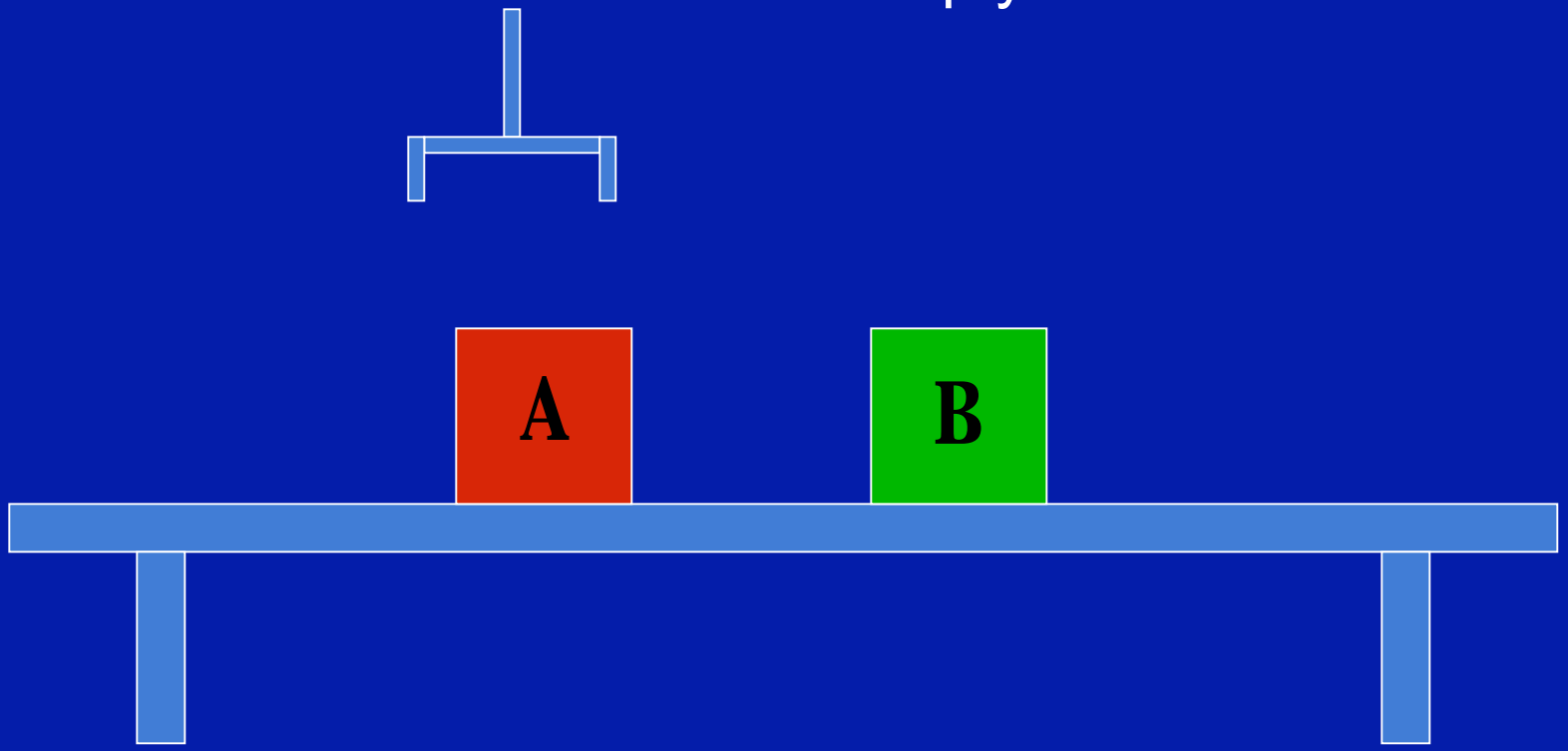
the objects are a table, a gripper arm, and some blocks

the relations that can exist between the objects are:

on(X,Y) -	block X is on block Y
ontable(X) -	block X is on the table
clear(X) -	there is nothing on top of block X
holding(X) -	the arm is holding block X
armempty -	the arm is holding nothing

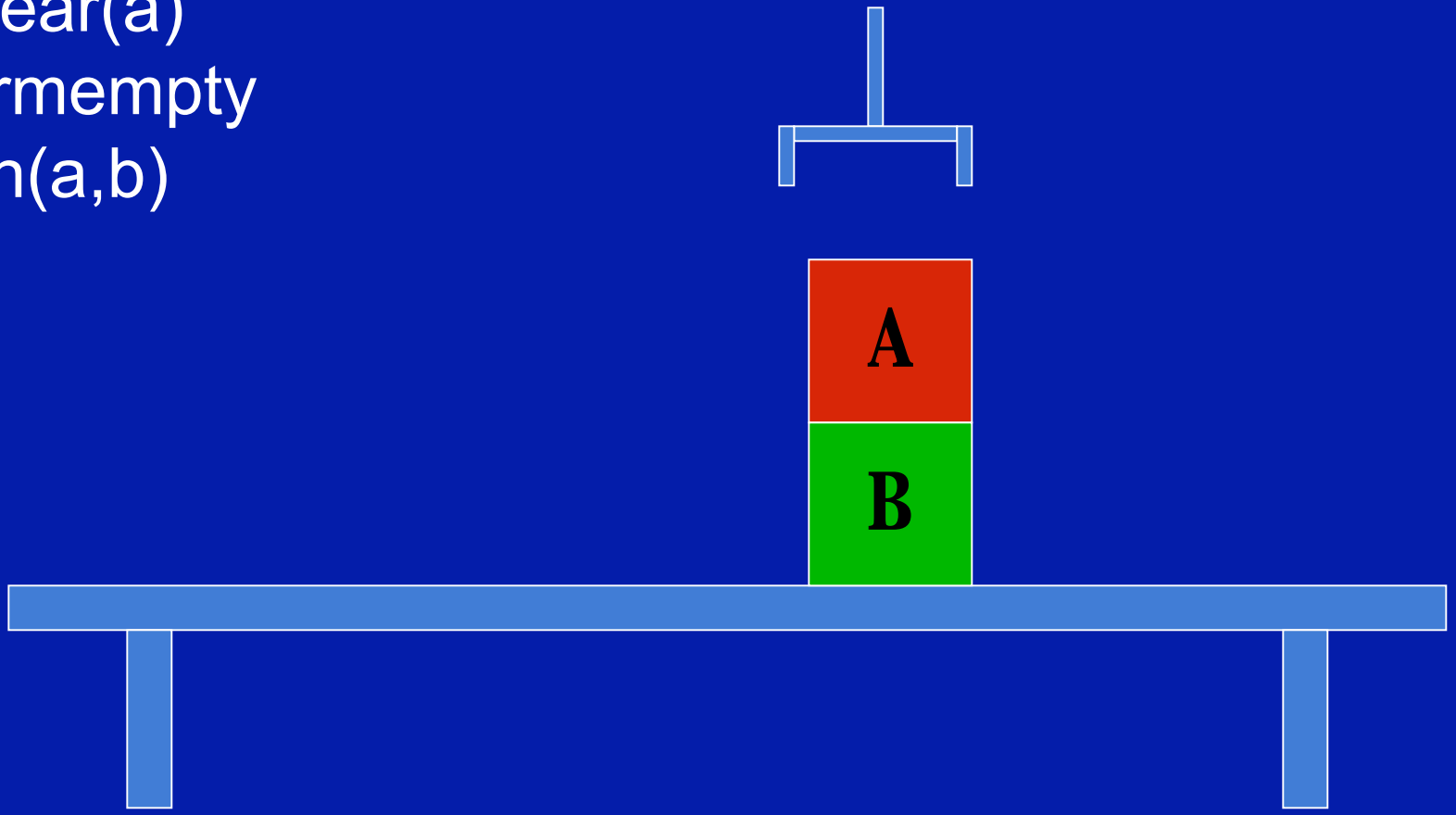
The start state

ontable(a)
ontable(b)
clear(a)
clear(b)
armempty



The goal state

ontable(b)
clear(a)
armempty
on(a,b)



STRIPS: actions in blocks world

stack(X,Y) - place block X on block Y. The arm must already be holding X and the top of Y must be clear

unstack(X,Y) - pick up block X from its current position on top of block Y. The arm must be empty and the top of block X must be clear.

pickup(X) - pick up block X from the table and hold it. The arm must be empty and the top of X must be clear.

putdown(X) - put block X down on the table. The arm must have been holding X.

STRIPS: actions as lists

stack(X,Y) - preconds: clear(Y) & holding(X)
delete: clear(Y) & holding(X)
add: armempty & on(X,Y)

unstack(X,Y) - preconds: on(X,Y) & clear(X) & armempty
delete: on(X,Y) & armempty
add: holding(X) & clear(Y)

pickup(X) - preconds: clear(X) & ontable(X) & armempty
delete: ontable(X) & armempty
add: holding(X)

putdown(X) - preconds: holding(X)
delete: holding(X)
add: ontable(X) & armempty

Finding the plan

stack(X,Y) - preconds: clear(Y) & holding(X)
delete: clear(Y) & holding(X)
add: armempty & on(X,Y)

unstack(X,Y) - preconds: on(X,Y) & clear(X) & armempty
delete: on(X,Y) & armempty
add: holding(X) & clear(Y)

pickup(X) - preconds: clear(X) & ontable(X) & armempty
delete: ontable(X) & armempty
add: holding(X)

putdown(X) - preconds: holding(X)
delete: holding(X)
add: ontable(X) & armempty

plan:

start: ontable(a)
ontable(b)
clear(a)
clear(b)
armempty

goals:
ontable(b)
clear(a)
armempty
on(a,b)

Finding the plan

stack(X,Y) - preconds: clear(Y) & holding(X)
delete: clear(Y) & holding(X)
add: armempty & on(X,Y)

unstack(X,Y) - preconds: on(X,Y) & clear(X) & armempty
delete: on(X,Y) & armempty
add: holding(X) & clear(Y)

pickup(X) - preconds: clear(X) & ontable(X) & armempty
delete: ontable(X) & armempty
add: holding(X)

putdown(X) - preconds: holding(X)
delete: holding(X)
add: ontable(X) & armempty

plan:

start: ontable(a)
ontable(b)
clear(a)
clear(b)
armempty

goals:
ontable(b)
clear(a)
armempty
on(a,b)

if a goal is fulfilled in the current state, then don't worry about it

Finding the plan

stack(X,Y) - preconds: clear(Y) & holding(X)
delete: clear(Y) & holding(X)
add: armempty & on(X,Y)

unstack(X,Y) - preconds: on(X,Y) & clear(X) & armempty
delete: on(X,Y) & armempty
add: holding(X) & clear(Y)

pickup(X) - preconds: clear(X) & ontable(X) & armempty
delete: ontable(X) & armempty
add: holding(X)

putdown(X) - preconds: holding(X)
delete: holding(X)
add: ontable(X) & armempty

plan:

start: ontable(a)
ontable(b)
clear(a)
clear(b)
armempty

goals:
ontable(b)
clear(a)
armempty
on(a,b)

if a goal is fulfilled in the current state, then don't worry about it

Finding the plan

stack(X,Y) - preconds: clear(Y) & holding(X)
delete: clear(Y) & holding(X)
add: armempty & on(X,Y)

unstack(X,Y) - preconds: on(X,Y) & clear(X) & armempty
delete: on(X,Y) & armempty
add: holding(X) & clear(Y)

pickup(X) - preconds: clear(X) & ontable(X) & armempty
delete: ontable(X) & armempty
add: holding(X)

putdown(X) - preconds: holding(X)
delete: holding(X)
add: ontable(X) & armempty

plan:

start: ontable(a)
ontable(b)
clear(a)
clear(b)
armempty

goals:

on(a,b)

if a goal is fulfilled in the current state, then don't worry about it

Finding the plan

stack(X,Y) - preconds: clear(Y) & holding(X)
delete: clear(Y) & holding(X)
add: armempty & on(X,Y)

unstack(X,Y) - preconds: on(X,Y) & clear(X) & armempty
delete: on(X,Y) & armempty
add: holding(X) & clear(Y)

pickup(X) - preconds: clear(X) & ontable(X) & armempty
delete: ontable(X) & armempty
add: holding(X)

putdown(X) - preconds: holding(X)
delete: holding(X)
add: ontable(X) & armempty

plan:

start: ontable(a)
ontable(b)
clear(a)
clear(b)
armempty

goals:

on(a,b)

for remaining goals, reduce difference between goal and current state

Finding the plan

stack(X,Y) - preconds: clear(Y) & holding(X)
delete: clear(Y) & holding(X)
add: armempty & on(X,Y)

unstack(X,Y) - preconds: on(X,Y) & clear(X) & armempty
delete: on(X,Y) & armempty
add: holding(X) & clear(Y)

pickup(X) - preconds: clear(X) & ontable(X) & armempty
delete: ontable(X) & armempty
add: holding(X)

putdown(X) - preconds: holding(X)
delete: holding(X)
add: ontable(X) & armempty

plan:

start: ontable(a)
ontable(b)
clear(a)
clear(b)
armempty

goals:

on(a,b)

find an action with add list that contains goal...

Finding the plan

stack(X,Y) - preconds: clear(Y) & holding(X)
delete: clear(Y) & holding(X)
add: armempty & on(X,Y)

unstack(X,Y) - preconds: on(X,Y) & clear(X) & armempty
delete: on(X,Y) & armempty
add: holding(X) & clear(Y)

pickup(X) - preconds: clear(X) & ontable(X) & armempty
delete: ontable(X) & armempty
add: holding(X)

putdown(X) - preconds: holding(X)
delete: holding(X)
add: ontable(X) & armempty

plan:

start: ontable(a)
ontable(b)
clear(a)
clear(b)
armempty

goals:

clear(b)
holding(a)

post that action's preconditions as new goals...

Finding the plan

stack(X,Y) - preconds: clear(Y) & holding(X)
delete: clear(Y) & holding(X)
add: armempty & on(X,Y)

unstack(X,Y) - preconds: on(X,Y) & clear(X) & armempty
delete: on(X,Y) & armempty
add: holding(X) & clear(Y)

pickup(X) - preconds: clear(X) & ontable(X) & armempty
delete: ontable(X) & armempty
add: holding(X)

putdown(X) - preconds: holding(X)
delete: holding(X)
add: ontable(X) & armempty

plan:

stack(a,b)

start: ontable(a)
ontable(b)
clear(a)
clear(b)
armempty

goals:

clear(b)
holding(a)

post that action as a step in the plan...

Finding the plan

stack(X,Y) - preconds: clear(Y) & holding(X)
delete: clear(Y) & holding(X)
add: armempty & on(X,Y)

unstack(X,Y) - preconds: on(X,Y) & clear(X) & armempty
delete: on(X,Y) & armempty
add: holding(X) & clear(Y)

pickup(X) - preconds: clear(X) & ontable(X) & armempty
delete: ontable(X) & armempty
add: holding(X)

putdown(X) - preconds: holding(X)
delete: holding(X)
add: ontable(X) & armempty

plan:

stack(a,b)

start: ontable(a)
ontable(b)
clear(a)
clear(b)
armempty

goals:

clear(b)
holding(a)

if a goal is fulfilled in the current state, then don't worry about it

Finding the plan

stack(X,Y) - preconds: clear(Y) & holding(X)
delete: clear(Y) & holding(X)
add: armempty & on(X,Y)

unstack(X,Y) - preconds: on(X,Y) & clear(X) & armempty
delete: on(X,Y) & armempty
add: holding(X) & clear(Y)

pickup(X) - preconds: clear(X) & ontable(X) & armempty
delete: ontable(X) & armempty
add: holding(X)

putdown(X) - preconds: holding(X)
delete: holding(X)
add: ontable(X) & armempty

plan:

stack(a,b)

start: ontable(a)
ontable(b)
clear(a)
clear(b)
armempty

goals:

clear(b)
holding(a)

if a goal is fulfilled in the current state, then don't worry about it

Finding the plan

stack(X,Y) - preconds: clear(Y) & holding(X)
delete: clear(Y) & holding(X)
add: armempty & on(X,Y)

unstack(X,Y) - preconds: on(X,Y) & clear(X) & armempty
delete: on(X,Y) & armempty
add: holding(X) & clear(Y)

pickup(X) - preconds: clear(X) & ontable(X) & armempty
delete: ontable(X) & armempty
add: holding(X)

putdown(X) - preconds: holding(X)
delete: holding(X)
add: ontable(X) & armempty

plan:

stack(a,b)

start: ontable(a)
ontable(b)
clear(a)
clear(b)
armempty

goals:

holding(a)

if a goal is fulfilled in the current state, then don't worry about it

Finding the plan

stack(X,Y) - preconds: clear(Y) & holding(X)
delete: clear(Y) & holding(X)
add: armempty & on(X,Y)

unstack(X,Y) - preconds: on(X,Y) & clear(X) & armempty
delete: on(X,Y) & armempty
add: holding(X) & clear(Y)

pickup(X) - preconds: clear(X) & ontable(X) & armempty
delete: ontable(X) & armempty
add: holding(X)

putdown(X) - preconds: holding(X)
delete: holding(X)
add: ontable(X) & armempty

plan:

stack(a,b)

start: ontable(a)
ontable(b)
clear(a)
clear(b)
armempty

goals:

holding(a)

for remaining goals, reduce difference between goal and current state

Finding the plan

stack(X,Y) - preconds: clear(Y) & holding(X)
delete: clear(Y) & holding(X)
add: armempty & on(X,Y)

unstack(X,Y) - preconds: on(X,Y) & clear(X) & armempty
delete: on(X,Y) & armempty
add: holding(X) & clear(Y)

pickup(X) - preconds: clear(X) & ontable(X) & armempty
delete: ontable(X) & armempty
add: holding(X)

putdown(X) - preconds: holding(X)
delete: holding(X)
add: ontable(X) & armempty

plan:

stack(a,b)

start: ontable(a)
ontable(b)
clear(a)
clear(b)
armempty

goals:

holding(a)

find an action with add list that contains goal...

Finding the plan

stack(X,Y) - preconds: clear(Y) & holding(X)
delete: clear(Y) & holding(X)
add: armempty & on(X,Y)

unstack(X,Y) - preconds: on(X,Y) & clear(X) & armempty
delete: on(X,Y) & armempty
add: holding(X) & clear(Y)

pickup(X) - preconds: clear(X) & ontable(X) & armempty
delete: ontable(X) & armempty
add: holding(X)

putdown(X) - preconds: holding(X)
delete: holding(X)
add: ontable(X) & armempty

plan:

stack(a,b)

start: ontable(a)
ontable(b)
clear(a)
clear(b)
armempty

goals:

clear(a)
ontable(a)
armempty

post that action's preconditions as new goals...

Finding the plan

stack(X,Y) - preconds: clear(Y) & holding(X)
delete: clear(Y) & holding(X)
add: armempty & on(X,Y)

unstack(X,Y) - preconds: on(X,Y) & clear(X) & armempty
delete: on(X,Y) & armempty
add: holding(X) & clear(Y)

pickup(X) - preconds: clear(X) & ontable(X) & armempty
delete: ontable(X) & armempty
add: holding(X)

putdown(X) - preconds: holding(X)
delete: holding(X)
add: ontable(X) & armempty

plan:

pickup(a)
stack(a,b)

start: ontable(a)
ontable(b)
clear(a)
clear(b)
armempty

goals:

clear(a)
ontable(a)
armempty

post that action as a step in the plan...

Finding the plan

stack(X,Y) - preconds: clear(Y) & holding(X)
delete: clear(Y) & holding(X)
add: armempty & on(X,Y)

unstack(X,Y) - preconds: on(X,Y) & clear(X) & armempty
delete: on(X,Y) & armempty
add: holding(X) & clear(Y)

pickup(X) - preconds: clear(X) & ontable(X) & armempty
delete: ontable(X) & armempty
add: holding(X)

putdown(X) - preconds: holding(X)
delete: holding(X)
add: ontable(X) & armempty

plan:

pickup(a)
stack(a,b)

start: ontable(a)
ontable(b)
clear(a)
clear(b)
armempty

goals:

clear(a)
ontable(a)
armempty

if a goal is fulfilled in the current state, then don't worry about it

Finding the plan

stack(X,Y) - preconds: clear(Y) & holding(X)
delete: clear(Y) & holding(X)
add: armempty & on(X,Y)

unstack(X,Y) - preconds: on(X,Y) & clear(X) & armempty
delete: on(X,Y) & armempty
add: holding(X) & clear(Y)

pickup(X) - preconds: clear(X) & ontable(X) & armempty
delete: ontable(X) & armempty
add: holding(X)

putdown(X) - preconds: holding(X)
delete: holding(X)
add: ontable(X) & armempty

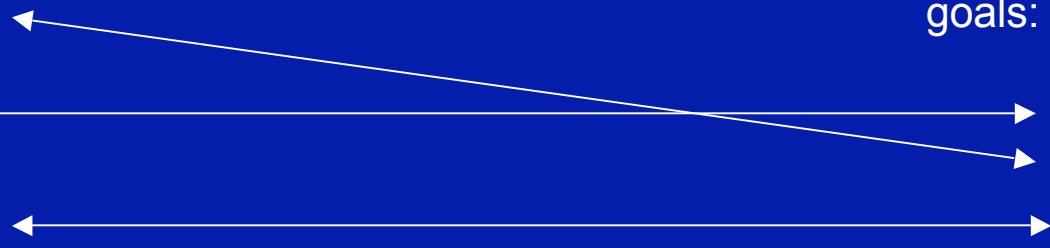
plan:

pickup(a)
stack(a,b)

start: ontable(a)
ontable(b)
clear(a)
clear(b)
armempty

goals:

clear(a)
ontable(a)
armempty



if a goal is fulfilled in the current state, then don't worry about it

Finding the plan

stack(X,Y) - preconds: clear(Y) & holding(X)
delete: clear(Y) & holding(X)
add: armempty & on(X,Y)

unstack(X,Y) - preconds: on(X,Y) & clear(X) & armempty
delete: on(X,Y) & armempty
add: holding(X) & clear(Y)

pickup(X) - preconds: clear(X) & ontable(X) & armempty
delete: ontable(X) & armempty
add: holding(X)

putdown(X) - preconds: holding(X)
delete: holding(X)
add: ontable(X) & armempty

plan:

pickup(a)
stack(a,b)

start: ontable(a)
ontable(b)
clear(a)
clear(b)
armempty

goals:

all the goals have been fulfilled...we now have our plan

Executing the plan

stack(X,Y) - preconds: clear(Y) & holding(X)
delete: clear(Y) & holding(X)
add: armempty & on(X,Y)

unstack(X,Y) - preconds: on(X,Y) & clear(X) & armempty
delete: on(X,Y) & armempty
add: holding(X) & clear(Y)

pickup(X) - preconds: clear(X) & ontable(X) & armempty
delete: ontable(X) & armempty
add: holding(X)

putdown(X) - preconds: holding(X)
delete: holding(X)
add: ontable(X) & armempty

plan:

pickup(a)
stack(a,b)

start: ontable(a)
ontable(b)
clear(a)
clear(b)
armempty

goals:

are the preconditions for the first step true? yes

Executing the plan

stack(X,Y) - preconds: clear(Y) & holding(X)
delete: clear(Y) & holding(X)
add: armempty & on(X,Y)

unstack(X,Y) - preconds: on(X,Y) & clear(X) & armempty
delete: on(X,Y) & armempty
add: holding(X) & clear(Y)

pickup(X) - preconds: clear(X) & ontable(X) & armempty
delete: ontable(X) & armempty
add: holding(X)

putdown(X) - preconds: holding(X)
delete: holding(X)
add: ontable(X) & armempty

plan:

pickup(a)
stack(a,b)

start: ontable(a)
ontable(b)
clear(a)
clear(b)
armempty

goals:

then do what the delete list says to do...

Executing the plan

stack(X,Y) - preconds: clear(Y) & holding(X)
delete: clear(Y) & holding(X)
add: armempty & on(X,Y)

unstack(X,Y) - preconds: on(X,Y) & clear(X) & armempty
delete: on(X,Y) & armempty
add: holding(X) & clear(Y)

pickup(X) - preconds: clear(X) & ontable(X) & armempty
delete: ontable(X) & armempty
add: holding(X)

putdown(X) - preconds: holding(X)
delete: holding(X)
add: ontable(X) & armempty

plan:

pickup(a)
stack(a,b)

start:

ontable(b)
clear(a)
clear(b)
armempty

goals:

then do what the delete list says to do...

Executing the plan

stack(X,Y) - preconds: clear(Y) & holding(X)
delete: clear(Y) & holding(X)
add: armempty & on(X,Y)

unstack(X,Y) - preconds: on(X,Y) & clear(X) & armempty
delete: on(X,Y) & armempty
add: holding(X) & clear(Y)

pickup(X) - preconds: clear(X) & ontable(X) & armempty
delete: ontable(X) & armempty
add: holding(X)

putdown(X) - preconds: holding(X)
delete: holding(X)
add: ontable(X) & armempty

plan:

pickup(a)
stack(a,b)

start:

ontable(b)
clear(a)
clear(b)

goals:

then do what the delete list says to do...

Executing the plan

stack(X,Y) - preconds: clear(Y) & holding(X)
delete: clear(Y) & holding(X)
add: armempty & on(X,Y)

unstack(X,Y) - preconds: on(X,Y) & clear(X) & armempty
delete: on(X,Y) & armempty
add: holding(X) & clear(Y)

pickup(X) - preconds: clear(X) & ontable(X) & armempty
delete: ontable(X) & armempty
add: holding(X)

putdown(X) - preconds: holding(X)
delete: holding(X)
add: ontable(X) & armempty

plan:

pickup(a)
stack(a,b)

start:

ontable(b)
clear(a)
clear(b)

goals:

and then do what the add list says to do...

Executing the plan

stack(X,Y) - preconds: clear(Y) & holding(X)
delete: clear(Y) & holding(X)
add: armempty & on(X,Y)

unstack(X,Y) - preconds: on(X,Y) & clear(X) & armempty
delete: on(X,Y) & armempty
add: holding(X) & clear(Y)

pickup(X) - preconds: clear(X) & ontable(X) & armempty
delete: ontable(X) & armempty
add: holding(X)

putdown(X) - preconds: holding(X)
delete: holding(X)
add: ontable(X) & armempty

plan:

pickup(a)
stack(a,b)

start: holding(a)
ontable(b)
clear(a)
clear(b)

goals:

and then do what the add list says to do...

Executing the plan

stack(X,Y) - preconds: clear(Y) & holding(X)
delete: clear(Y) & holding(X)
add: armempty & on(X,Y)

unstack(X,Y) - preconds: on(X,Y) & clear(X) & armempty
delete: on(X,Y) & armempty
add: holding(X) & clear(Y)

pickup(X) - preconds: clear(X) & ontable(X) & armempty
delete: ontable(X) & armempty
add: holding(X)

putdown(X) - preconds: holding(X)
delete: holding(X)
add: ontable(X) & armempty

plan:

pickup(a)
stack(a,b)

start: holding(a)
ontable(b)
clear(a)
clear(b)

goals:

are the preconditions for the first step true? yes

Executing the plan

stack(X,Y) - preconds: clear(Y) & holding(X)
delete: clear(Y) & holding(X)
add: armempty & on(X,Y)

unstack(X,Y) - preconds: on(X,Y) & clear(X) & armempty
delete: on(X,Y) & armempty
add: holding(X) & clear(Y)

pickup(X) - preconds: clear(X) & ontable(X) & armempty
delete: ontable(X) & armempty
add: holding(X)

putdown(X) - preconds: holding(X)
delete: holding(X)
add: ontable(X) & armempty

plan:

pickup(a)
stack(a,b)

start: holding(a)
ontable(b)
clear(a)
clear(b)

goals:

then do what the delete list says to do...

Executing the plan

stack(X,Y) - preconds: clear(Y) & holding(X)
delete: clear(Y) & holding(X)
add: armempty & on(X,Y)

unstack(X,Y) - preconds: on(X,Y) & clear(X) & armempty
delete: on(X,Y) & armempty
add: holding(X) & clear(Y)

pickup(X) - preconds: clear(X) & ontable(X) & armempty
delete: ontable(X) & armempty
add: holding(X)

putdown(X) - preconds: holding(X)
delete: holding(X)
add: ontable(X) & armempty

plan:

pickup(a)
stack(a,b)

start: holding(a)
ontable(b)
clear(a)

goals:

then do what the delete list says to do...

Executing the plan

stack(X,Y) - preconds: clear(Y) & holding(X)
delete: clear(Y) & holding(X)
add: armempty & on(X,Y)

unstack(X,Y) - preconds: on(X,Y) & clear(X) & armempty
delete: on(X,Y) & armempty
add: holding(X) & clear(Y)

pickup(X) - preconds: clear(X) & ontable(X) & armempty
delete: ontable(X) & armempty
add: holding(X)

putdown(X) - preconds: holding(X)
delete: holding(X)
add: ontable(X) & armempty

plan:

pickup(a)
stack(a,b)

start:

ontable(b)
clear(a)

goals:

then do what the delete list says to do...

Executing the plan

stack(X,Y) - preconds: clear(Y) & holding(X)
delete: clear(Y) & holding(X)
add: armempty & on(X,Y)

unstack(X,Y) - preconds: on(X,Y) & clear(X) & armempty
delete: on(X,Y) & armempty
add: holding(X) & clear(Y)

pickup(X) - preconds: clear(X) & ontable(X) & armempty
delete: ontable(X) & armempty
add: holding(X)

putdown(X) - preconds: holding(X)
delete: holding(X)
add: ontable(X) & armempty

plan:

pickup(a)
stack(a,b)

start:

ontable(b)
clear(a)

goals:

and then do what the add list says to do...

Executing the plan

stack(X,Y) - preconds: clear(Y) & holding(X)
delete: clear(Y) & holding(X)
add: armempty & on(X,Y)

unstack(X,Y) - preconds: on(X,Y) & clear(X) & armempty
delete: on(X,Y) & armempty
add: holding(X) & clear(Y)

pickup(X) - preconds: clear(X) & ontable(X) & armempty
delete: ontable(X) & armempty
add: holding(X)

putdown(X) - preconds: holding(X)
delete: holding(X)
add: ontable(X) & armempty

plan:

pickup(a)
stack(a,b)

start: armempty
ontable(b)
clear(a)

goals:

and then do what the add list says to do...

Executing the plan

stack(X,Y) - preconds: clear(Y) & holding(X)
delete: clear(Y) & holding(X)
add: armempty & on(X,Y)

unstack(X,Y) - preconds: on(X,Y) & clear(X) & armempty
delete: on(X,Y) & armempty
add: holding(X) & clear(Y)

pickup(X) - preconds: clear(X) & ontable(X) & armempty
delete: ontable(X) & armempty
add: holding(X)

putdown(X) - preconds: holding(X)
delete: holding(X)
add: ontable(X) & armempty

plan:

pickup(a)
stack(a,b)

start: armempty
ontable(b)
clear(a)
on(a,b)

goals:

and then do what the add list says to do...

Executing the plan

stack(X,Y) - preconds: clear(Y) & holding(X)
delete: clear(Y) & holding(X)
add: armempty & on(X,Y)

unstack(X,Y) - preconds: on(X,Y) & clear(X) & armempty
delete: on(X,Y) & armempty
add: holding(X) & clear(Y)

pickup(X) - preconds: clear(X) & ontable(X) & armempty
delete: ontable(X) & armempty
add: holding(X)

putdown(X) - preconds: holding(X)
delete: holding(X)
add: ontable(X) & armempty

plan:

pickup(a)
stack(a,b)

start: armempty
ontable(b)
clear(a)
on(a,b)

goals:

the plan is finished...how does the resulting state compare to original goal?

Executing the plan

stack(X,Y) - preconds: clear(Y) & holding(X)
delete: clear(Y) & holding(X)
add: armempty & on(X,Y)

unstack(X,Y) - preconds: on(X,Y) & clear(X) & armempty
delete: on(X,Y) & armempty
add: holding(X) & clear(Y)

pickup(X) - preconds: clear(X) & ontable(X) & armempty
delete: ontable(X) & armempty
add: holding(X)

putdown(X) - preconds: holding(X)
delete: holding(X)
add: ontable(X) & armempty

plan:

pickup(a)
stack(a,b)

start: armempty
ontable(b)
clear(a)
on(a,b)

goals:

ontable(b)
clear(a)
armempty
on(a,b)

the plan is finished...how does the resulting state compare to original goal?

Executing the plan

stack(X,Y) - preconds: clear(Y) & holding(X)
delete: clear(Y) & holding(X)
add: armempty & on(X,Y)

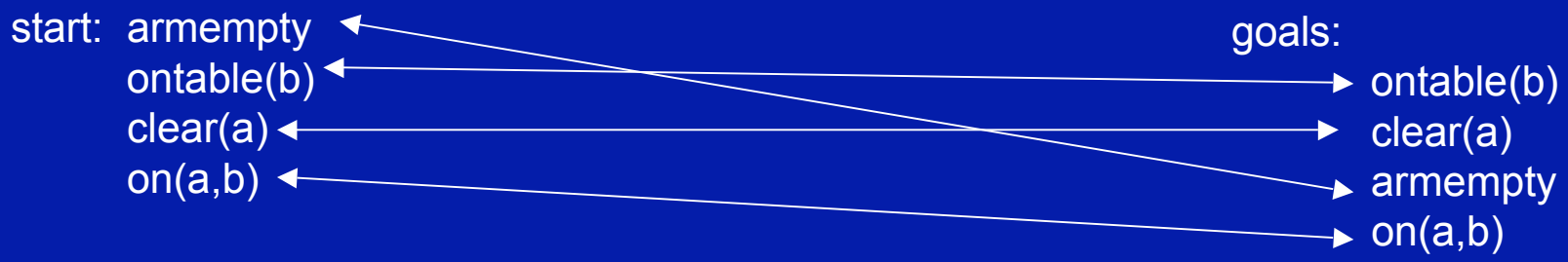
unstack(X,Y) - preconds: on(X,Y) & clear(X) & armempty
delete: on(X,Y) & armempty
add: holding(X) & clear(Y)

pickup(X) - preconds: clear(X) & ontable(X) & armempty
delete: ontable(X) & armempty
add: holding(X)

putdown(X) - preconds: holding(X)
delete: holding(X)
add: ontable(X) & armempty

plan:

pickup(a)
stack(a,b)



give your robot a cookie

How about some tasty CILOG code?

```
/*
This is an implementation of the simple STRIPS planner
shown on page 302 of the Computational Intelligence text.

launch it with the query:
cilog: ask goals(G) & achieve_all(G,init,Plan).

Note that the add list is denoted here by "achieves" instead of
"add", and that the add and delete lists aren't exactly lists.
*/

/* stack action */
preconditions(stack(X,Y),[cleartop(Y),holding(X)]).
achieves(stack(X,Y),armempty).
achieves(stack(X,Y),on(X,Y)).
deletes(stack(X,Y),cleartop(Y)).
deletes(stack(X,Y),holding(X)).
```

How about some tasty CILOG code?

```
/* pickup action */
preconditions(pickup(X),[cleartop(X),ontable(X),armempty]).
achieves(pickup(X),holding(X)).
deletes(pickup(X),ontable(X)).
deletes(pickup(X),armempty).


/* initial situation */

holds(ontable(a),init).
holds(ontable(b),init).
holds(cleartop(a),init).
holds(cleartop(b),init).
holds(armempty,init).

achieves(init,X) <- holds(X,init).

goals([ontable(b),cleartop(a),armempty,on(a,b)]).
```

How about some tasty CILOG code?

```
/* the simple STRIPS planner */

remove(X,[X|Y],Y).

achieve_all([],W0,W0).
achieve_all(Goals,W0,W2) <-
     remove(G,Goals,Rem_Gs) &
     achieve(G,W0,W1) &
     achieve_all(Rem_Gs,W1,W2).

achieve(G,W,W) <- holds(G,W).
achieve(G,W0,do(Action,W1)) <-
     achieves(Action,G) &
     preconditions(Action,Pre) &
     achieve_all(Pre,W0,W1).
```