CPSC 322
Introduction to Artificial Intelligence

November 22, 2004

Things...

Term project is due one week from today

The final exam will be at noon on Friday, December 10, in MCML 166

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?

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

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

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

How about some tasty CILOG code?

```
/* 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).
```

Desirable attributes of a knowledge representation approach

- capture generalities in the world being modeled
- easily modifiable to reflect changes so that new knowledge can be derived from old knowledge
- transparent - understandable by people who provide the knowledge as well as those who look at it later
- usable even if not entirely accurate or complete
- explicitly represent important objects and relationships
- natural constraints on how one object or relation influences another should be obvious
- irrelevant detail should be suppressed (abstracted away)
- complete -- everything that needs to be represented can be represented
- concise -- what needs to be said can be said efficiently
- fast -- you can store and retrieve information quickly
- computable -- enables reasoning to proceed easily with known procedures (doesn't rely on bizarre coding tricks)

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?

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

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

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

How about some tasty CILOG code?

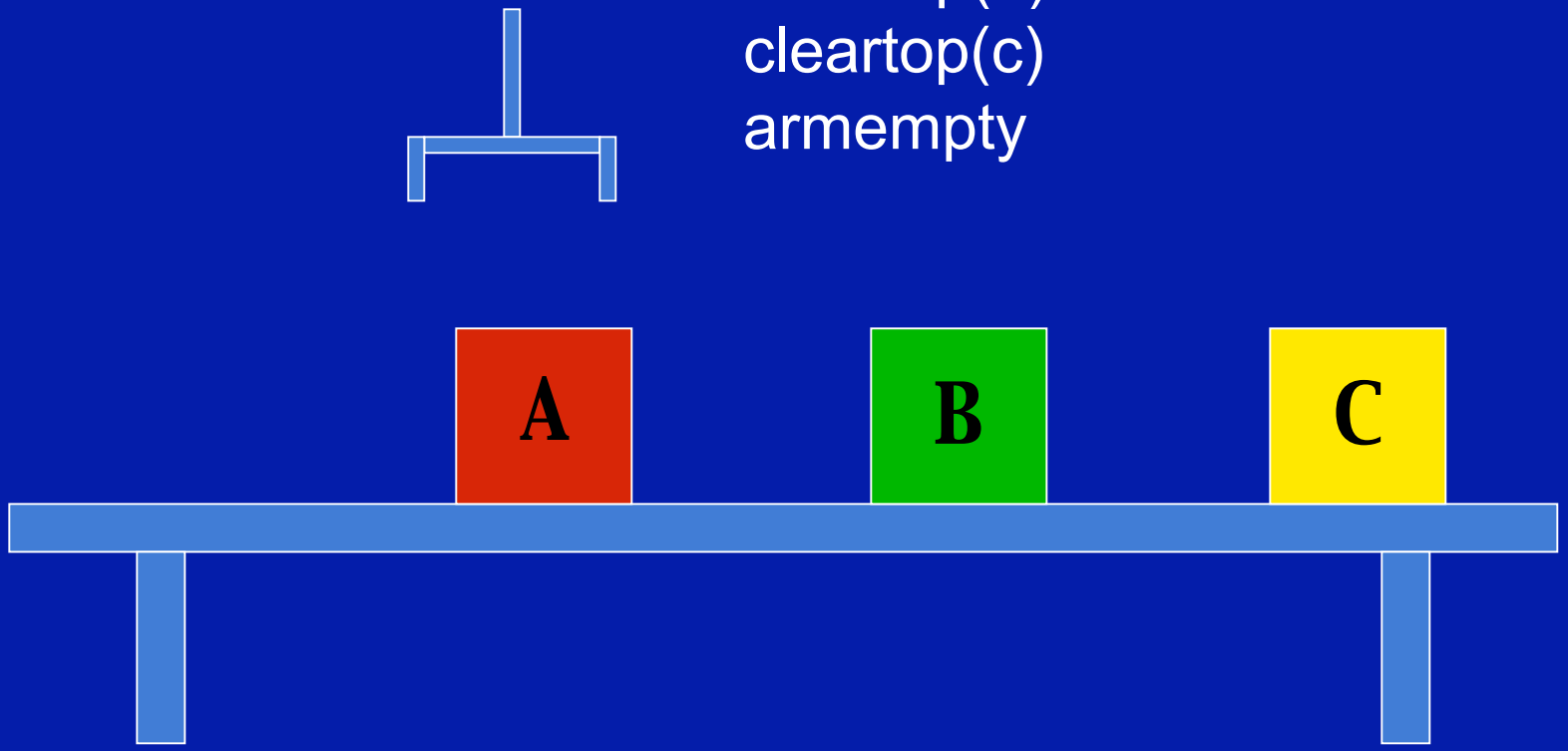
```
/* initial situation */

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)]).
```

Did choosing a good KR approach make this problem easier?

```
/* 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).
```

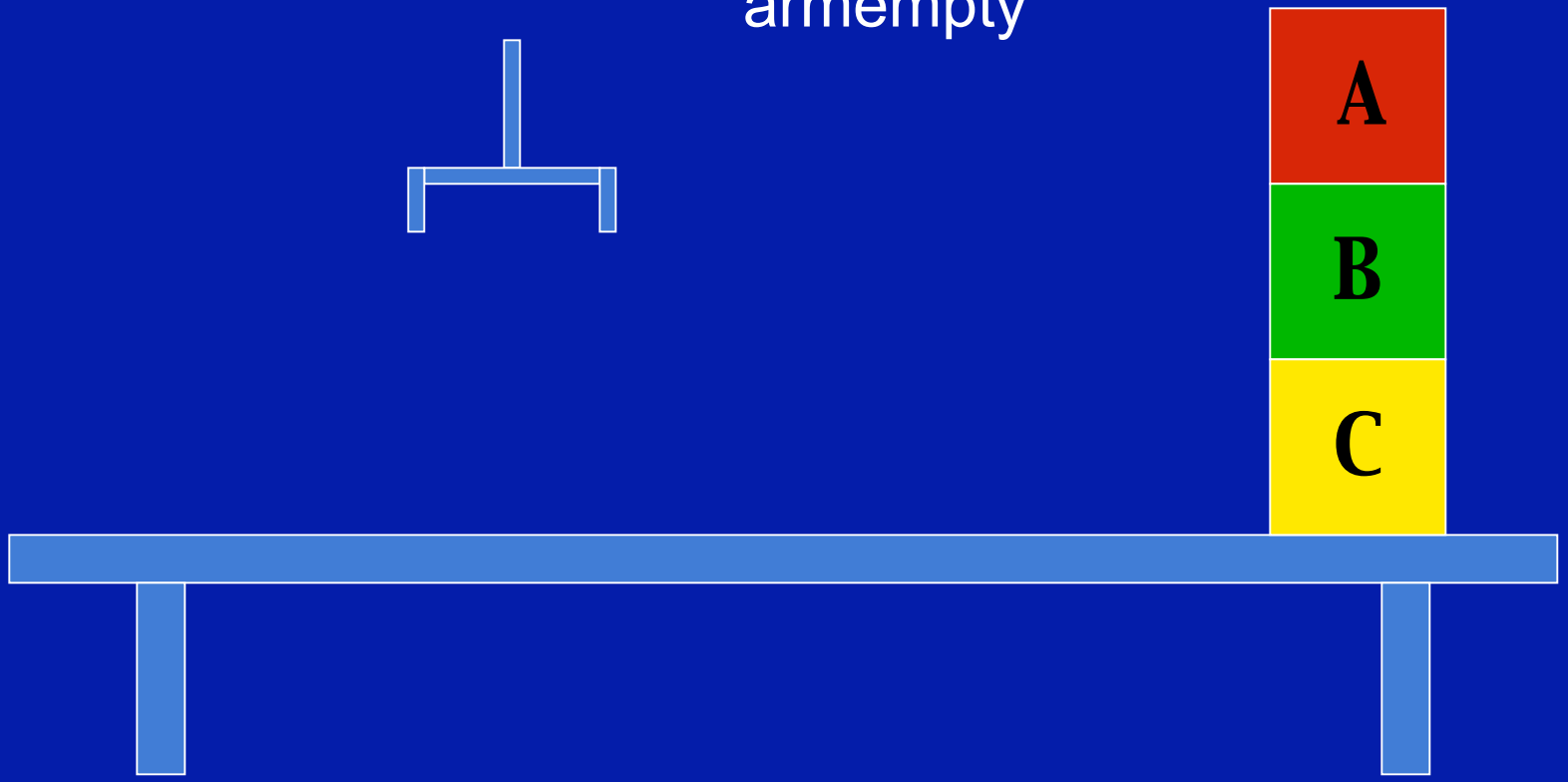
A new start state

ontable(a)
ontable(b)
ontable(c)
cleartop(a)
cleartop(b)
cleartop(c)
armempty



A new goal state

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

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

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

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

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

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

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

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

goals:
clear(c)
holding(b)
on(a,b)

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

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

plan:

stack(b,c)

goals:

clear(c)
holding(b)
on(a,b)

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

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

plan:

stack(b,c)

goals:

clear(c)
holding(b)
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

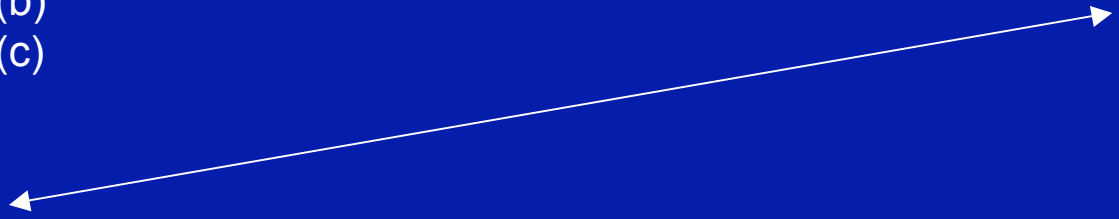
plan:

stack(b,c)

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

goals:

clear(c)
holding(b)
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

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

plan:

stack(b,c)

goals:

holding(b)
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

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

plan:

stack(b,c)

goals:

holding(b)
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

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

plan:

stack(b,c)

goals:

holding(b)
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

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

plan:

stack(b,c)

goals:

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

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

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

plan:

pickup(b)
stack(b,c)

goals:

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

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

Do you see a problem developing?

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

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

plan:

pickup(b)
stack(b,c)

goals:

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

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