Regression

- Idea: don't solve one subgoal by itself, but keep track of all subgoals that must be achieved.
- Given a set of goals:
  - If they all hold in the initial state, return the empty plan
  - Otherwise, choose an action *A* that achieves one of the subgoals. This will be the last action in the plan.
  - Determine what must be true immediately before *A* so that all of the goals will be true immediately after. Recursively solve these new goals.

Regression as Path Finding

- The nodes are sets of goals. Arcs correspond to actions.
- A node labeled with goal set G has a neighbor for each action A that achieves one of the goals in G.
- The neighbor corresponding to action A is the node with the goals G_A that must be true immediately before the action A so that all of the goals in G are true immediately after A. G_A is the weakest precondition for action A and goal set G.
- Search can stop when you have a node where all the goals are true in the initial state.

Weakest preconditions

$wp(A, GL, WP)$ is true if WP is the weakest precondition that must occur immediately before action A so every element of goal list GL is true immediately after A.

For the STRIPS representation (with all predicates primitive):

- $wp(A, GL, WP)$ is *false* if any element of GL is on delete list of action A.
- Otherwise WP is

  $$preconds(A) \cup \{G \in GL : G \notin add\_list(A)\}.$$

  where $preconds(A)$ is the list of preconditions of action A and $add\_list(A)$ is the add list of action A.

Weakest Precondition Example

The weakest precondition for

$$[sitting\_at(rob, lab2), carrying(rob, parcel)]$$

to be true after $move(rob, Pos, lab2)$ is that

$$\begin{aligned}&[autonomous(rob),\\&\qquad adjacent(Pos, lab2),\\&\qquad sitting\_at(rob, Pos),\\&\qquad carrying(rob, parcel)]\end{aligned}$$

is true immediately before the action.

⇐

⇑ ⇒

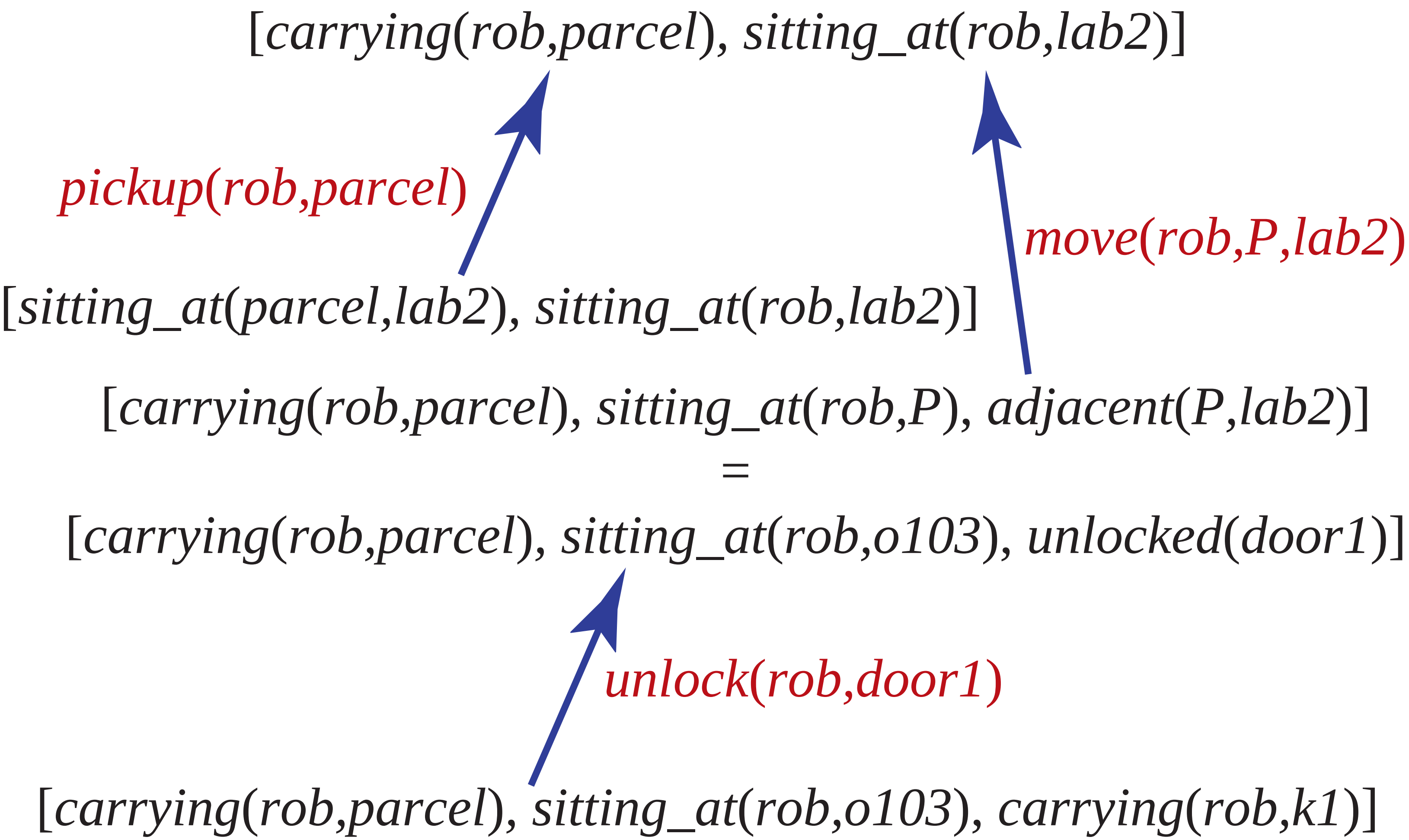
A Regression Planner

% $solve(GL, W)$ is true if every element of goal list GL is true

% in world W.

```
$solve(GoalSet, init) \leftarrow$
    $holdsall(GoalSet, init).$
$solve(GoalSet, do(Action, W)) \leftarrow$
    $consistent(GoalSet) \wedge$
    $choose\_goal(Goal, GoalSet) \wedge$
    $choose\_action(Action, Goal) \wedge$
    $wp(Action, GoalSet, NewGoalSet) \wedge$
    $solve(NewGoalSet, W).$
```

Regression Search Space Example



[carrying(rob,parcel), sitting_at(rob,lab2)]

pickup(rob,parcel)

move(rob,P,lab2)

[sitting_at(parcel,lab2), sitting_at(rob,lab2)]

[carrying(rob,parcel), sitting_at(rob,P), adjacent(P,lab2)]

=

[carrying(rob,parcel), sitting_at(rob,o103), unlocked(door1)]

unlock(rob,door1)

[carrying(rob,parcel), sitting_at(rob,o103), carrying(rob,k1)]