Planning

Given

- an initial world description
- a description of available actions
- a goal

a **plan** is a sequence of actions that will achieve the goal.

Example Planning

If you want a plan to achieve Rob holding the key $k1$ and being at $o103$, you can issue the query

$$?carrying(rob, k1, S) \wedge at(rob, o103, S).$$

This has an answer

$$
\begin{aligned}
S = do(&move(rob, mail, o103),\\
&do(pickup(rob, k1),\\
&\quad do(move(rob, o103, mail),\\
&\qquad do(move(rob, o109, o103), init)))).
\end{aligned}
$$

Forward Planner

- Search in the state-space graph, where the nodes represent states and the arcs represent actions.
- Search from initial state to a state that satisfies the goal.
- A complete search strategy (e.g., A^* or iterative deepening) is guaranteed to find a solution.
- Branching factor is the number of legal actions. Path length is the number of actions to achieve the goal.
- You usually can't do backward planning in the state space, as the goal doesn't uniquely specify a state.

Planning as Resolution

- Idea: backward chain on the situation calculus rules or the situation calculus axiomatization of STRIPS.
- A complete search strategy (e.g., A^* or iterative deepening) is guaranteed to find a solution.
- When there is a solution to the query with situation $S = do(A, S_1)$, action A is the last action in the plan.
- You can virtually always use a frame axiom so that the search space is largely unconstrained by the goal.

Goal-directed searching

- Given a goal, you would like to consider only those actions that actually achieve it.
- **Example:**

$$?carrying(rob, parcel, S) \wedge in(rob, lab2, S).$$

the last action needed is irrelevant to the left subgoal.