In which we see how an agent can take advantage of problem structure to construct complex plans of action.

In this chapter, we introduce the basic ideas involved in planning systems. We begin by specifying a simple planning agent that is very similar to a problem-solving agent (Chapter 3) in that it constructs plans that achieve its goals, and then executes them. Section 11.2 explains the limitations of the problem-solving approach, and motivates the design of planning systems. The planning agent differs from a problem-solving agent in its representations of goals, states, and actions, as described in Section 11.4. The use of explicit, logical representations enables the planner to direct its deliberations much more sensibly. The planning agent also differs in the way it represents and searches for solutions. The remainder of the chapter describes in detail the basic partial-order planning algorithm, which searches through the space of plans to find one that is guaranteed to succeed. The additional flexibility gained from the partially ordered plan representation allows a planning agent to handle quite complicated domains.

A Simple Planning Agent

When the world state is accessible, an agent can use the percepts provided by the environment to build a complete and correct model of the current world state. Then, given a goal, it can call a suitable planning algorithm (which we will call IDEAL-PLANNER) to generate a plan of action. The agent can then execute the steps of the plan, one action at a time.

The algorithm for the simple planning agent is shown in Figure 11.1. This should be compared with the problem-solving agent shown in Figure 3.1. The planning algorithm IDEAL-PLANNER can be any of the planners described in this chapter or Chapter 12. We assume the existence of a function STATE-DESCRIPTION, which takes a percept as input and returns an initial state description in the format required by the planner, and a function MAKE-GOAL-QUERY, which is used to ask the knowledge base what the next goal should be. Note that the agent must deal with the case where the goal is infeasible (it just ignores it and tries another), and the case