Artificial Intelligence: 6.034
Massachusetts Institute of Technology — March 16, 2012
Spring 2012 — Recitation 6

Planning and Search

- We can model the world using a **state transition system** $\Sigma = (S, A, E, \gamma)$, where
  - S is the set of possible **states** (assignments of values to random variables),
  - A is the set of **actions** we can take to change the state of the world,
  - E is the set of **events** that can happen to change the state of the world, and
  - γ is a **transition function**, where $\gamma(s, a)$ is the set of possible states that could result from action (or event) a in state s.

  An action a is **applicable** in state s if $\gamma(s, a) \neq \emptyset$.

- In the **STRIPS** representation, the state of the world is a set of statements in first-order logic, and each action consists of some precondition statements and some statements to be added and/or removed.

- A **factored** expression splits up each state into a fixed set of variables, each of which can have a value. A **lifted** expression has variables in it, and a **ground** expression does not. An **operator** is a lifted expression, which, when values are plugged in for the variables, becomes a specific action.

- A **plan** is a sequence of actions, generally for the purpose of ending up at one of a set of **goal states**. To find plans, we can use **state-space search**.

- When we have a limited amount of time, we can use **iterative deepening**, in which we search completely to a given depth, and if we still have time, we continue to the next depth.

- If two heuristics h_1 and h_2 are both admissible, h_2 **dominates** h_1 if $h_2(n) \geq h_1(n)$ for all n. If h_2 dominates h_1, h_2 is at least as good as h_1 for search.

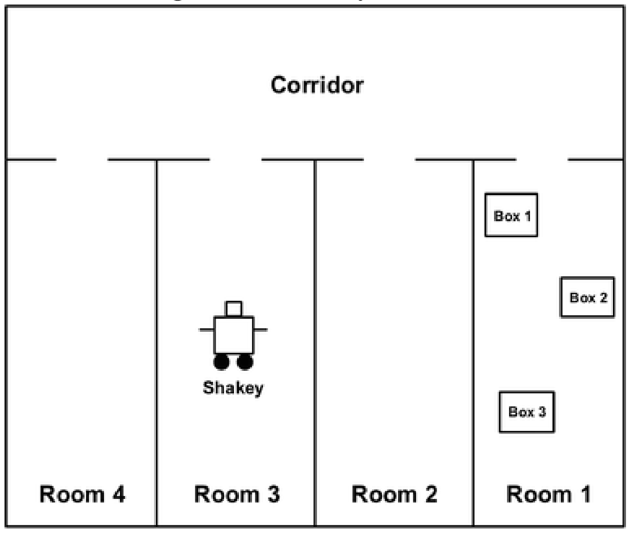
- We can make an admissible heuristic by using a **relaxed** version of the original problem, in which the optimal solution is easy to compute (or to estimate) and costs no more than the optimal solution to our original problem.

- Sometimes a heuristic function is not provided and must be computed. The **HSP** planner computes a function $g_s(p)$ that is an estimate of the minimum number of steps to get from state s to a state in which atom p is true. This function is computed by ignoring the delete lists in the STRIPS action representations (so actions are approximated as only being able to add assertions, not delete them).

Exercises

1. (AIMA 3rd 10.4) The original STRIPS planner was designed to control Shakey the robot. Figure 10.14 shows a version of Shakey's world consisting of four rooms lined up along a corridor, where each room has a door and a light switch. The actions in Shakey's world include moving from place to place, pushing movable objects (such as boxes), climbing onto and down from rigid objects (such as boxes), and turning light switches on and off. The robot itself could not climb on a box or toggle a switch, but the planner was capable of finding and printing out plans that were beyond the robot's abilities. Shakey's six actions are the following:

   Figure 1: Shakey's world.

   - $Go(x, y, r)$, which requires that Shakey be At x and that x and y are locations In the same room r. By convention a door between two rooms is in both of them.
   - Push a box b from location x to location y within the same room: $Push(b, x, y, r)$. You will need the predicate Box and constants for the boxes.
   - Climb onto a box from position x: $ClimbUp(x, b)$; climb down from a box to position x: $ClimbDown(b, x)$. We will need the predicate On and the constant $Floor$.
   - Turn a light switch on or off: $TurnOn(s, b)$; $TurnOff(s, b)$. To turn a light on or off, Shakey must be on top of a box at the light switch's location.

   Write PDDL sentences for Shakey's six actions and the initial state from Figure 1. Construct a plan for Shakey to get Box_2 into $Room_2$.

2. (AIMA 3rd 10.6) Explain why dropping negative effects from every action schema in a planning problem results in a relaxed problem.