

## State-Space Search

To solve a problem via search, we need to formulate:

- What information is stored in each state? This must be a complete snapshot of the partially-solved problem. Phrase this in terms of explicit variables and data types, like designing a class.
- What is the initial state?
- What is the goal state? Is there more than one? Can we list them all explicitly (small number of them), or do we need to write a function that determines if a state is a goal state? Think about a function IS-GOAL( $s$ ) that returns true if and only if  $s$  is a goal state.
- What actions are possible from a state? Are they the same for every state, or does the possible set of actions in a state depend upon the information in the state?
  - You should be able to construct a function ACTIONS( $s$ ) that returns the set of all possible actions from a state  $s$ .
  - You should be able to construct a function RESULT( $s, a$ ) that returns the state that you end up in if you take action  $a$  from state  $s$ .
- What is my cost function? This is a function that calculates how good/bad a sequence of states is (a path) from the initial state to whatever the current state is. Normally you do this by defining a function COST( $s, a, s'$ ) that defines the cost of being in state  $s$ , taking action  $a$ , and ending in state  $s'$ . This function, sometimes called the step cost, should normally always return a number  $\geq 0$ .

States vs nodes

- A **state** corresponds to a configuration about the world, and is the concept of a “vertex” in the search space.
- A **node** is a bookkeeping data structure used in the search algorithms, and corresponds to a “vertex” in the search tree that the algorithm generates.