Artificial Intelligence
ADC503

Prof. Smita S. Mane

Asst. Prof.

Dept. of AI & DS,

VESIT,Chembur

Curriculum

Solving Problems by Searching

Module 3

3		**Solving Problems by Searching**	12
	3.1	Definition, State space representation, Problem as a state space search, Problem formulation, Well-defined problems	
	3.2	Solving Problems by Searching, Performance evaluation of search strategies, Time Complexity, Space Complexity, Completeness, Optimality	
	3.3	Uninformed Search: Depth First Search, Breadth First Search, Depth Limited Search, Iterative Deepening Search, Uniform Cost Search, Bidirectional Search	
	3.4	Informed Search: Heuristic Function, Admissible Heuristic, Informed Search Technique, Greedy Best First Search, A* Search, Local Search: Hill Climbing Search, Simulated Annealing Search, Optimization: Genetic Algorithm	
	3.5	Game Playing, Adversarial Search Techniques, Mini-max Search, Alpha-Beta Pruning	

Agents

- An **agent** is anything that can be viewed as **perceiving** its environment through **sensors** and **acting** upon that environment through **effectors**.
- A human agent has eyes, ears, and other organs for sensors, and hands, legs, mouth, and other body parts for effectors.
- A robotic agent substitutes cameras and infrared range finders for the sensors and various motors for the effectors.

Problem-Solving Agent

Four general steps in problem solving:

- **Goal formulation**
  - What are the successful world states
- **Problem formulation**
  - What actions and states to consider given the goal
- **Search**
  - Examine different possible sequences of actions that lead to states of known value and then choose the best sequence
- **Execute**
  - Perform actions on the basis of the solution

- Problem solving We want: –To automatically solve a problem
- We need: –A representation of the problem –Algorithms that use some strategy to solve the problem defined in that representation.

Problem representation

- State space: a problem is divided into a set of steps from the initial state to the goal state.
- A problem is defined by its elements and their relations.
- A state is a representation of those elements in a given moment.
- Two special states are defined: –
- Initial state (starting point)
- Final state (goal state)

State modification: successor function

- A successor function is needed to move between different states.
- A successor function is a description of possible actions, a set of operators. It is a transformation function on a state representation, which move it into another state.
- The successor function defines a relation of accessibility among states.

State space

- The **state space** is the set of all states reachable from the initial state.
- It forms a graph (or map) in which the nodes are states and the arcs between nodes are actions.
- A **path** in the state space is a sequence of states connected by a sequence of actions.
- The solution of the problem is part of the map formed by the state space.

Problem solution

- A **solution** in the state space is a path from the initial state to a goal state.

- **Path/solution cost** : function that assigns a numeric cost to each path, the cost of applying the operators to the states
- Solution quality is measured by the path cost function, and an **optimal solution** has the lowest path cost among all solutions.

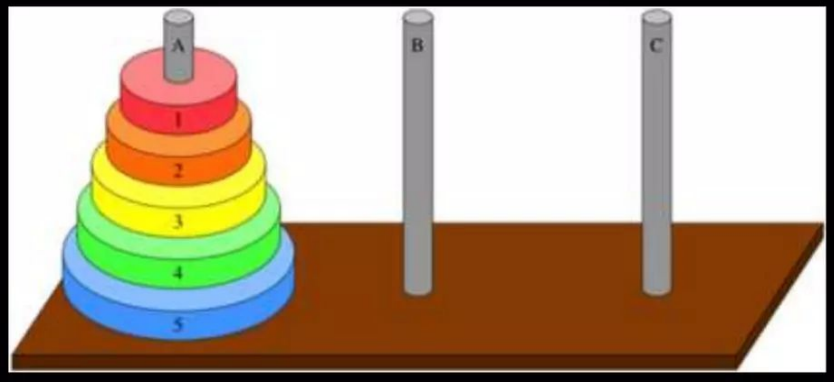
Problem description

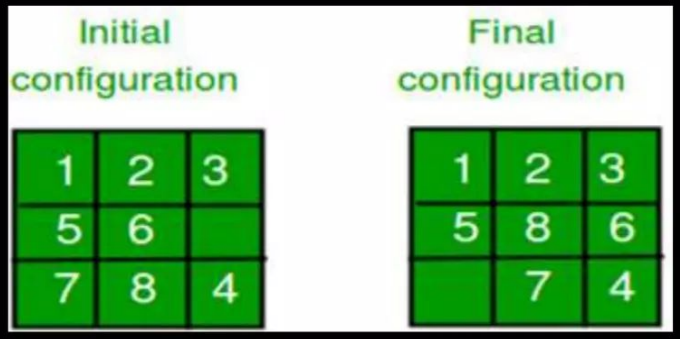
- Components:
  - State space
  - Initial state
  - Goal state (or the conditions it has to fulfill)
  - Available actions (operators to change state)
  - Restrictions (e.g., cost)

Problem Space

Define the problem in detail is known as problem space. There are 4 things concerned to solve a particular problem.

- **Define the problem precisely.**
- **Analyze** the problem.
- Isolate and represent the **task knowledge** i.e. required to solve the problem.
- Selection of the **best problem** solving technique and apply it. Like Tower of Hanoi, 8 puzzle games.





3		**Solving Problems by Searching**	12
	3.1	Definition, State space representation, Problem as a state space search, Problem formulation, Well-defined problems	
	3.2	Solving Problems by Searching, Performance evaluation of search strategies, Time Complexity, Space Complexity, Completeness, Optimality	
	3.3	Uninformed Search: Depth First Search, Breadth First Search, Depth Limited Search, Iterative Deepening Search, Uniform Cost Search, Bidirectional Search	
	3.4	Informed Search: Heuristic Function, Admissible Heuristic, Informed Search Technique, Greedy Best First Search, A* Search, Local Search: Hill Climbing Search, Simulated Annealing Search, Optimization: Genetic Algorithm	
	3.5	Game Playing, Adversarial Search Techniques, Mini-max Search, Alpha-Beta Pruning	

State space search

- State space search is a process used in the field of computer science, including artificial intelligence (AI), in which successive configurations or states of an instance are considered, with the intention of finding a goal state with the desired property.

Important Points

- **State:** A specific configuration of the problem.
- **Initial State:** The starting point of the search.
- **Goal State:** The desired end configuration.
- **Transition:** An action that changes one state to another.
- **Path:** A sequence of states connected by transitions.
- **Search Strategy:** The method used to explore the state space.

Principles and Features of State Space Search

- **Expansiveness:** The number of successors that each state can generate. This impacts how many new states are explored from a given state.
- **Branching Factor:** The average number of successors in each state. It influences the width of the search tree and the overall complexity of the search.
- **Depth:** The length from the initial state to the goal state in the search tree. Deeper search trees can increase the time required to find a solution.

- **Completeness:** A search strategy is complete if it guarantees finding a solution, assuming one exists.
- **Optimality:** A search strategy is optimal if it guarantees finding the best solution according to a specified criterion.
- **Time Complexity:** The duration of the state space exploration. It is influenced by the branching factor and the depth of the search tree.
- **Space Complexity:** The amount of memory required to carry out the search. This depends on the number of states that need to be stored in memory simultaneously.

Steps in State Space Search

Step 1: Define the State Space

Determine the collection of all potential states and their interchanging states. To do this, the problem must be modelled in a fashion that encompasses all relevant configurations and actions.

Step 2: Pick a Search Strategy

- Breadth-First Search (BFS)
- Uniform Cost Search (UCS)
- Greedy Best-First Search
- A* Search Algorithm

Steps in State Space Search

Step 3: Start the Search

Add the initial state to the frontier (the collection of states to be investigated) by starting there.

Step 4: Extend the Nodes

- Using the selected search technique, iteratively expands nodes from the frontier, producing successor states and appending them to the frontier. After each node has been expanded, determine whether it now corresponds to the desired state. If so, retrace your route to the objective and call off the hunt.

Steps in State Space Search

Step 5: Address State Repetition

Put in place safeguards to prevent revisiting the same state, including keeping track of the states you've been to.

Step 6: End the Search

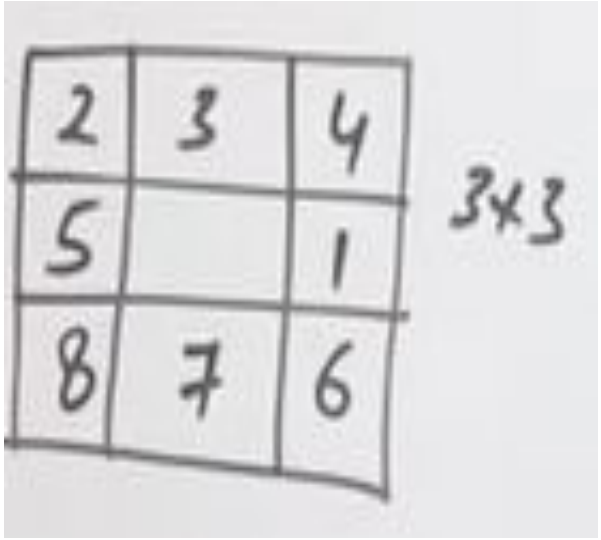
- The search comes to an end when the desired state is discovered or, in the event that no viable solution is identified, when every state has been investigated.
- AI systems are able to tackle complicated issues in an organized and methodical manner by employing these methods to systematically explore the state space.

State Space Representation

- **States:** Different arrangements of the issue, frequently shown as graph nodes.
- **Initial State:** The initial setting that the search starts with.
- **Goal State(s):** The ideal configuration(s) denoting a resolution.
- **Actions:** The processes via which a system changes states.
- **Transition Model:** Explains what happens when states are subjected to actions.
- **Path cost:** The expense of moving from an initial state to a certain state, expressed as a numerical value linked to each path.

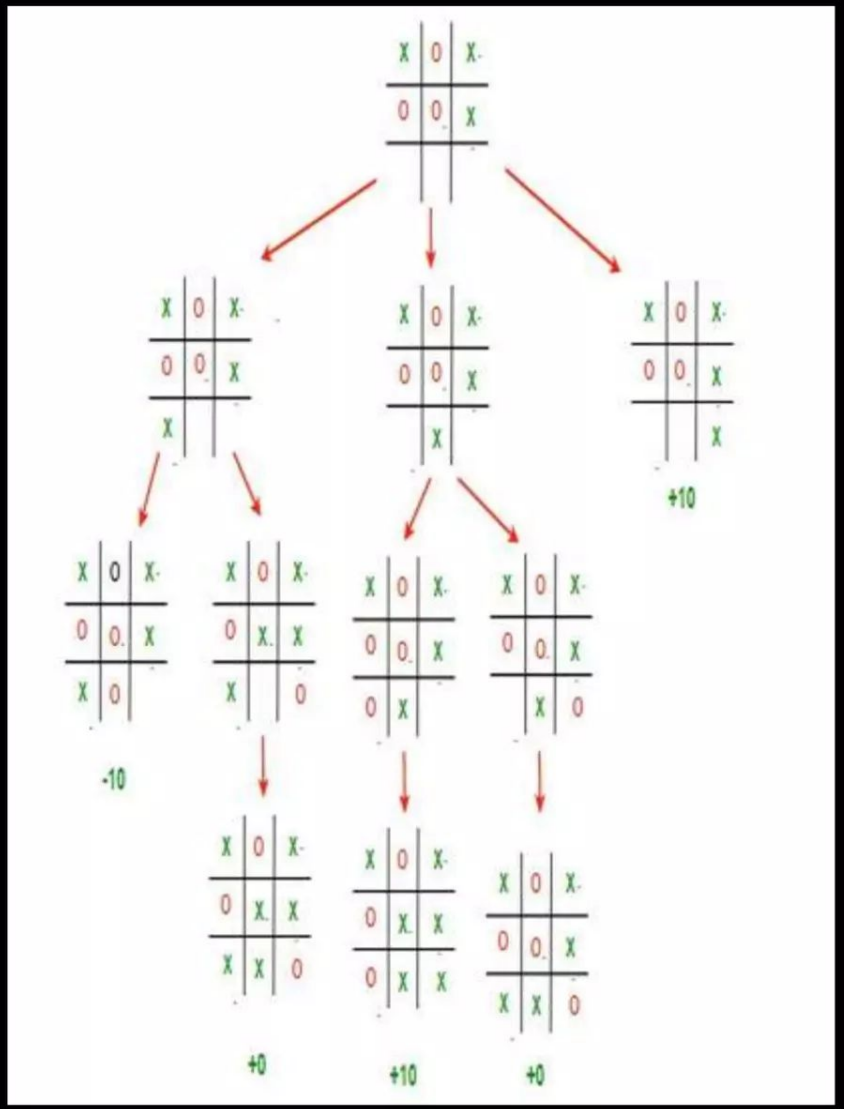
'State Space Search'

$S: \{ S, A, \text{Action}(s), \text{Result}(s,a), \text{Cost}(s,a) \}$



State Space

- **Set of states and the connections** between them to represent the problem.
- **Graph** is used represent problem.
- **States** are represented by **nodes** of the graph, and the **operators by edges** between nodes.
- For simplicity it is represented as **trees**, where the initial state is the root of the tree.



Problem solving agent-

Goal Based Agent

- Decide what to do by finding sequence of actions that lead to desirable states (GOAL)
- **A state** is a situation that an agent can find itself in.

Problem Solving Agent

- Rational agents need to perform sequences of actions in order to achieve goals.
- Intelligent behavior can be generated by having a look-up table or reactive policy that tells the agent what to do in every circumstance, but:
  - Such a table or policy is difficult to build
  - All contingencies must be anticipated

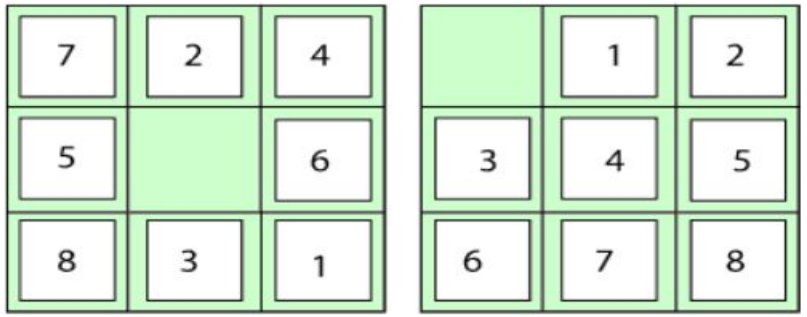
Types of States

- **World states :** the actual concrete situations in the real world

- **Representational states :** the abstract descriptions of the real world that are used by the agent in deliberating about what to do

Problem Solving Agent

- A more general approach is for the agent to have knowledge of the world and how its actions affect it and be able to simulate execution of actions in an internal model of the world in order to determine a sequence of actions that will accomplish its goals.
- This is the general task of problem solving and is typically performed by searching through an internally modelled space of world states.



Problem Solving task

- **Given:**
  - An initial state of the world
  - A set of possible actions or operators that can be performed.
  - A goal test that can be applied to a single state of the world to determine if it is a goal state.
- **Find:**
  - A solution stated as a path of states and operators that shows how to transform the initial state into one that satisfies the goal test.
  - The initial state and set of operators implicitly define a state space of states of the world and operator transitions between them. May be infinite.

Problem-solving agent

Four general steps in problem solving:

- **Goal formulation**
  - What are the successful world states
- **Problem formulation**
  - What actions and states to consider give the goal
- **Search**
  - Determine the possible sequence of actions that lead to the states of known values and then choosing the best sequence.
- **Execute**
  - Give the solution perform the actions.

Steps Performed by Problem-solving agent

- **Problem Formulation:** Most important step of problem-solving which decides what actions should be taken to achieve the formulated goal.
- Five components involved in problem formulation:
  - **Initial State:** It is the starting state or initial step of the agent towards its goal.
  - **Actions:** It is the description of the possible actions available to the agent.
  - **Transition Model:** It describes what each action does.
  - **Goal Test:** It determines if the given state is a goal state.
  - **Path cost:** It assigns a numeric cost to each path that follows the goal. The problem-solving agent selects a cost function, which reflects its performance measure. Remember, **an optimal solution has the lowest path cost among all the solutions.**

Start state

Searching for a solution to the 8-puzzle.

Aside: in this tree, immediate duplicates are removed.

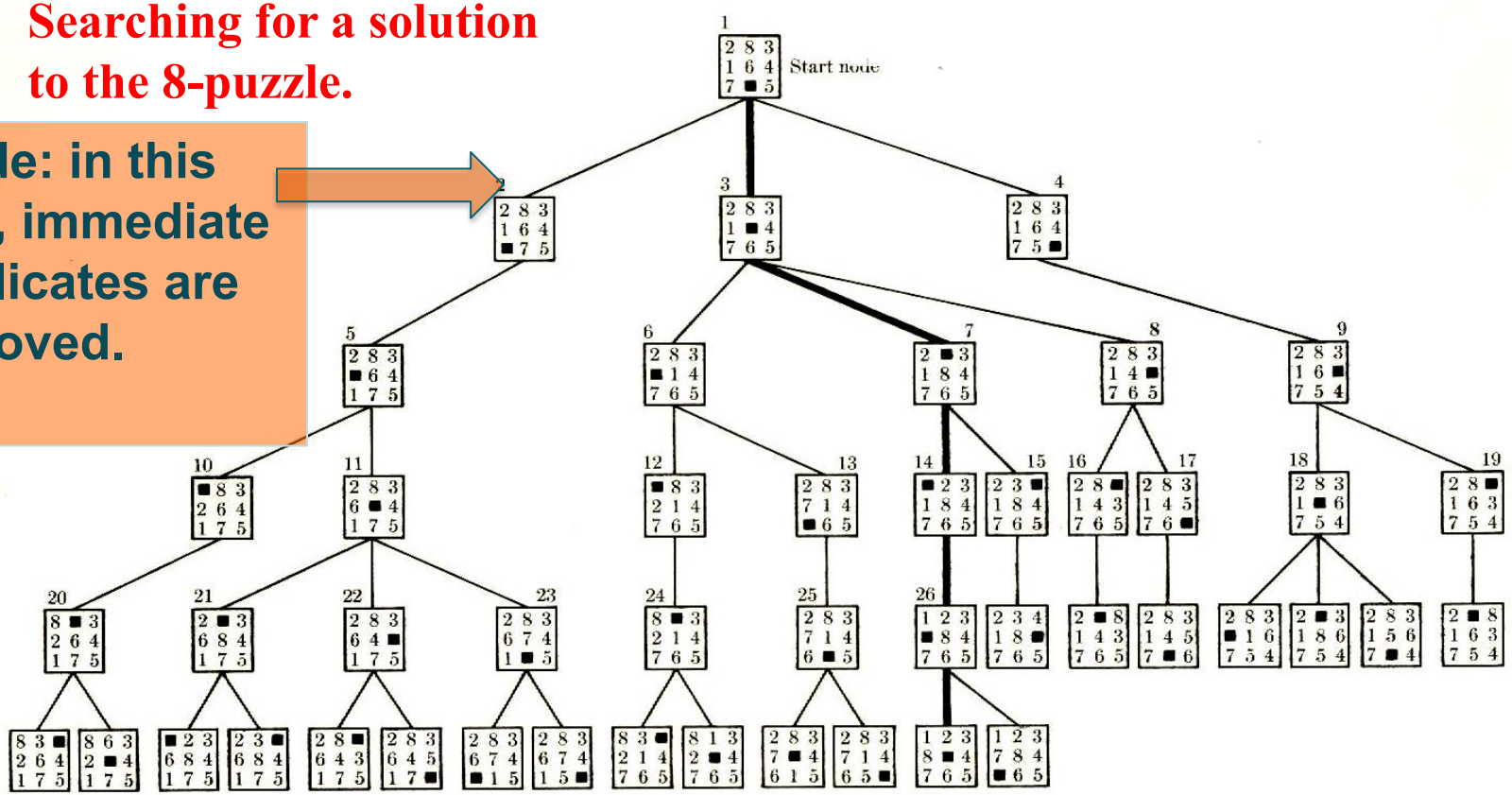


FIG. 3-2 *The tree produced by a bre*adth-first search.

A breadth-first search tree. (More detail soon.)

Branching factor 1, 2, or 3 (max). So, approx. 2 --- # nodes roughly doubles at each level. Number states of explored nodes grows exponentially with depth.

Example Problems : Activity

- Vacuum Cleaner Agent
- Travel Salesman Problem – Pathfinding to Romania
- 8 Queens problem
- 8 Puzzle Problem
- Robot Block World
- Missionaries and Cannibals

State Space and Search Tree

- A **state space :** a graph whose **nodes are the set of all states**, and whose **links are actions** that transform one state into another.

- **A search tree :** a tree (a graph with no undirected loops) in which the root node is the start state and the set of children for each node consists of the states reachable by taking any action.

Defining Search Problems

- **A statement of a Search problem has 4 components**
  1. A set of states
  2. A set of "operators" which allow one to get from one state to another
  3. A start state S
  4. A set of possible goal states, or ways to test for goal states
  5 Cost path

- **A solution consists of**
  - a sequence of operators which transform **S** into a goal state **G**

- **Representing real problems in a State-Space search framework**
  - may be many ways to represent states and operators
  - key idea: represent only the relevant aspects of the problem (**abstraction**)

Problem Formulation

- Process of deciding what actions and states to consider , given a goal.
- E.g. travel within Maharashtra– map can be given as input , Mumbai to Pune.
- Agent can decide what to do by – -
  - Examining different possible sequence of actions that lead to states of known values.
  - Then choosing the best sequence.
- This process of looking for such sequence is called as "**search**"

Problem-Solving Agents

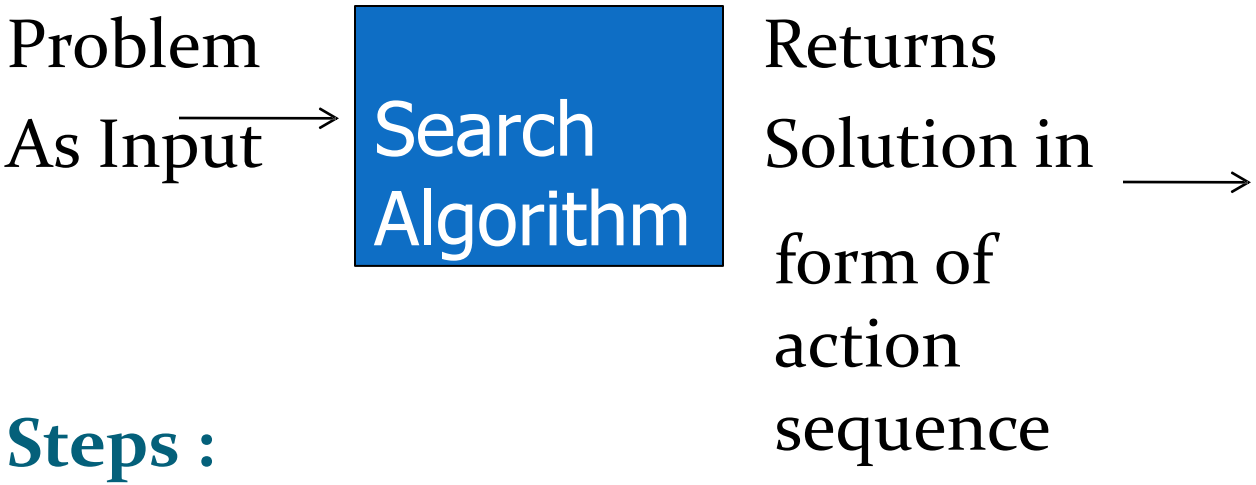
- Intelligent agents can solve problems by searching a state-space

- **State-space Model**
  - The agent's model of the world
  - Usually a set of discrete states
  - E.g., in driving, the states in the model could be towns/cities

- **Goal State(s)**
  - A goal is defined as a desirable state for an agent
  - There may be many states which satisfy the goal test
    - e.g., drive to a town with a ski-resort
  - OR just one state which satisfies the goal
    - e.g., drive to Pune

- **Operators (actions, successor function)**
  - operators are legal actions which the agent can take to move from one state to another

Search

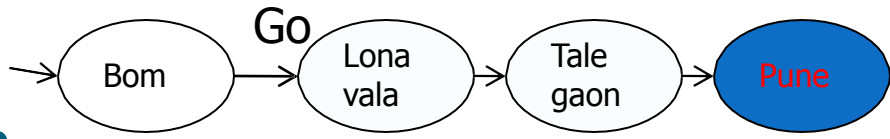


Steps :

1. Formulate Problem
2. Search
3. Execute Solution

Initial Simplifying Assumptions

- Environment is **static**
  - no changes in environment while problem is being solved
- Environment is **observable**
- Environment and actions are **discrete**
- Environment is **deterministic**



1. Problem Formulation

- Problem can be defined by four elements:

  - **Initial State** : In(Bombay)
  - **Goal State** : In (Pune)
  - **Description of possible actions** available to agent
    - SUCCESSOR_FN(x) returns (action, successor)
    - E.g. {<Go(Lonavala), In (Bombay)>,
      <Go(Talagaon),In (Lonavala)>,
      <Go(pune), In(pune)> }

State Space – **Initial state** and successor function implicitly **define state space** of problem. It can represented using Graph where nodes are states and arcs are actions.

The state-space graph

- **Graphs:**
  - nodes, arcs, directed arcs, paths
- **Search graphs:**
  - States are nodes
  - operators are directed arcs
  - solution is a path from start to goal
- **Problem formulation:**
  - Give an abstract description of states, operators, initial state and goal state.
- **Problem solving activity:**
  - Generate a **part of the search space** that **contains a solution**

Problem Formulation

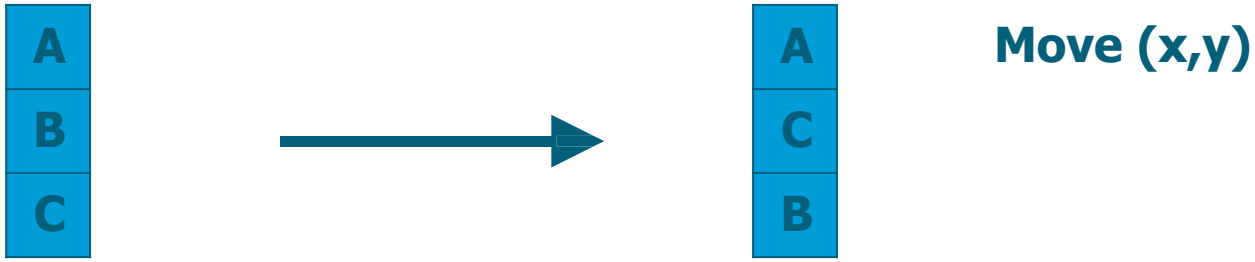
- **Path :** in state space is sequence of states connected by sequence of actions
- **Goal Test :** Determines whether given state is goal state.
  - Goal state can be **Explicit / Implicit e.g.** in chess – 'checkmate"
- **Step Cost :** It is cost of taking action a to go from state x to y
  - **Cost(x, a, y)**
- **Solution :** Solution to problem is a path from initial state to goal state.
- **Solution Quality :** Measured by Path Cost function and optimal solution
- **Abstraction :** Removing details from representation e.g. weather condition, rains, etc.

Example: Traveling from Mumbai to Pune

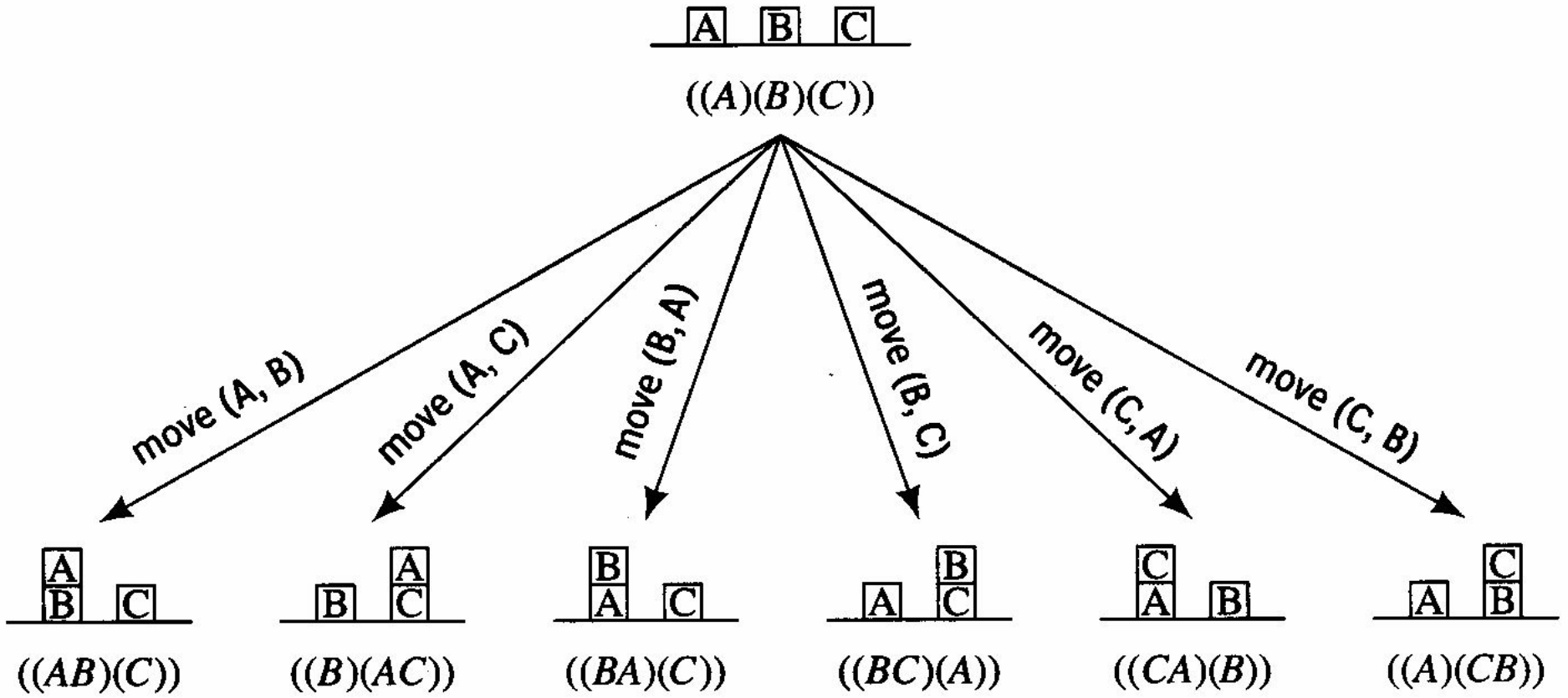
- **Initial State:** currently in Mumbai
- **Formulate goal:**
  - be in PUNE
- **Formulate problem:**
  - states: various cities
  - actions/operators: drive between cities
- **Find solution**
  - By searching through states to find a goal
  - sequence of cities, e.g., Lonavala, Talegaon, Baner, Pune
- **Execute states** that lead to a solution

Robot block world

- Given a set of blocks in a certain configuration,
- Move the blocks into a goal configuration.
- Example :
  - (c,b,a) (b,c,a)

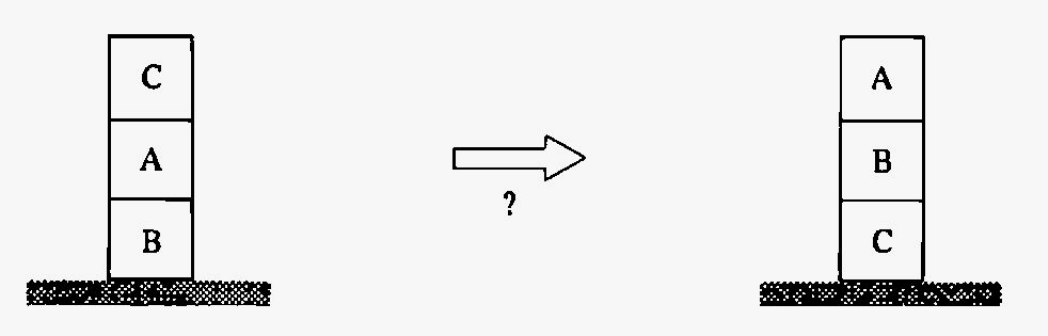


Operator Description



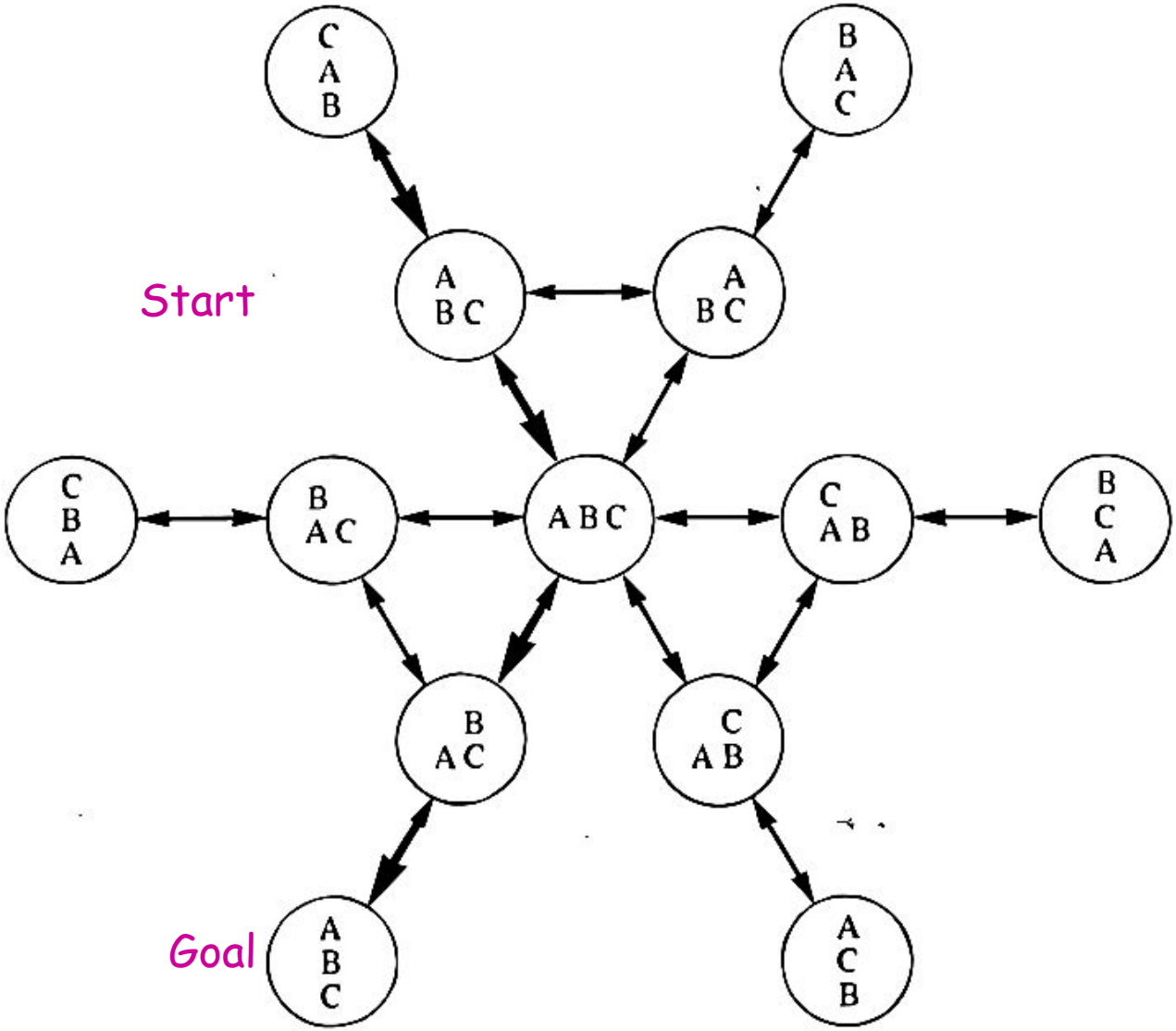
Effects of Moving a Block

A problem from blocks world



Find a sequence of robot moves to re-arrange blocks

Blocks world state space



State Space

- State space = Directed graph
- Nodes ~ Problem situations
- Arcs ~ Actions, legal moves

- Problem = (State space, Start, Goal condition)
- Note: several nodes may satisfy goal condition

- Solving a problem ~ Finding a path
- Problem solving ~ Graph search
- Problem solution ~ **Path from start to a goal node**

Vacuum Cleaner Agent

Initial State : Any state that is designated as initial state

Goal State / Test : check whether all squares are clean

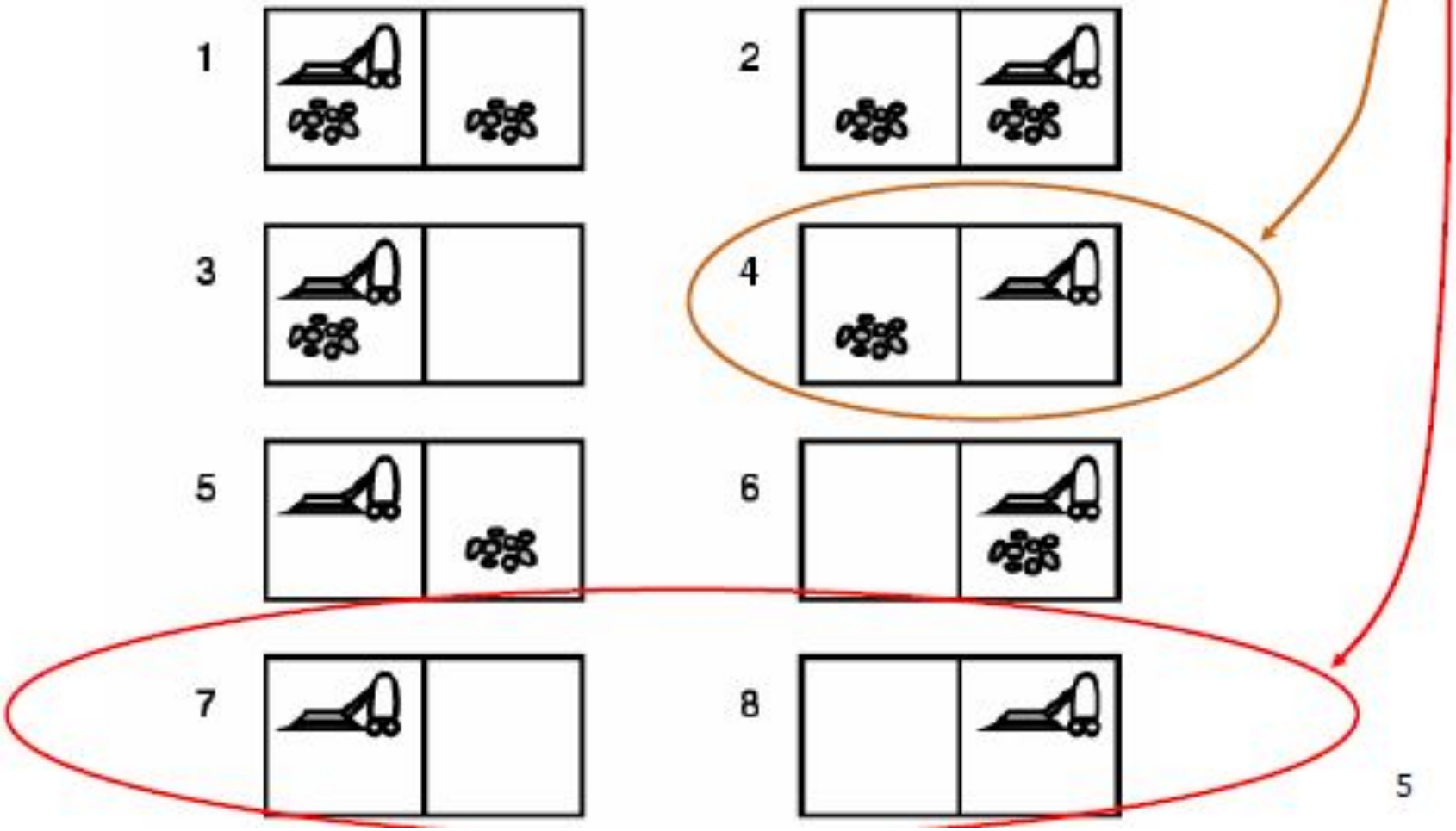
States : Agent in one of two locations each of which might / or might not contain dirt

Successor Function : actions (left, right, suck)

Path Cost : each step cost 1 – number of steps in the path.

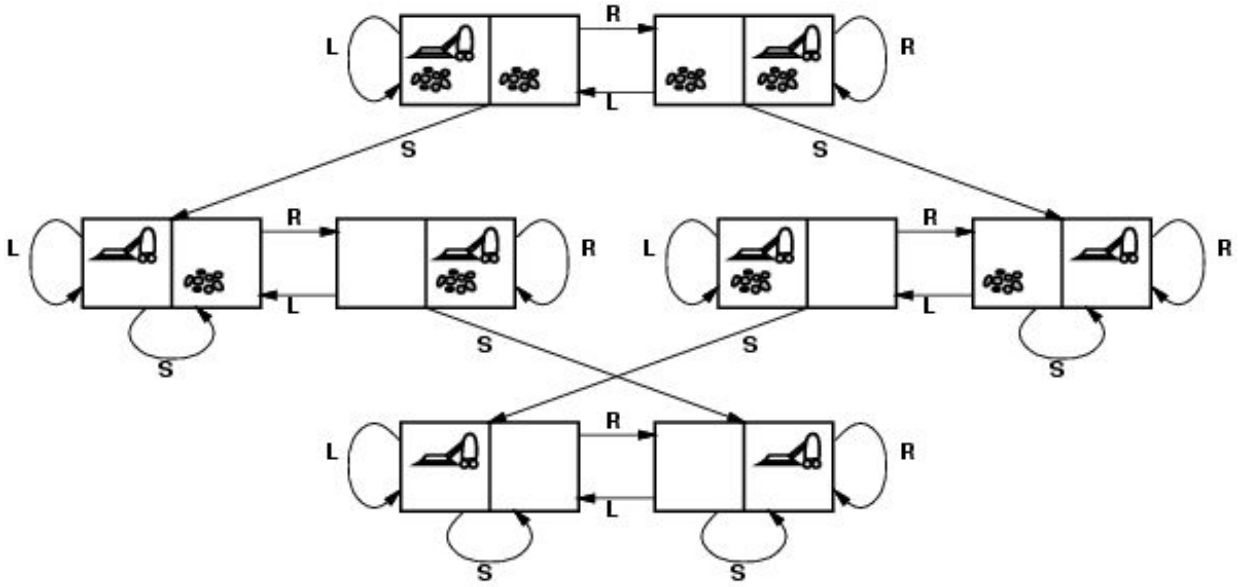
Clean House Task

- Want to clean "house"
  ⇒ be in State#7 or State#8
- Initial world: State#4
- Actions: { Left, Right, Suck }



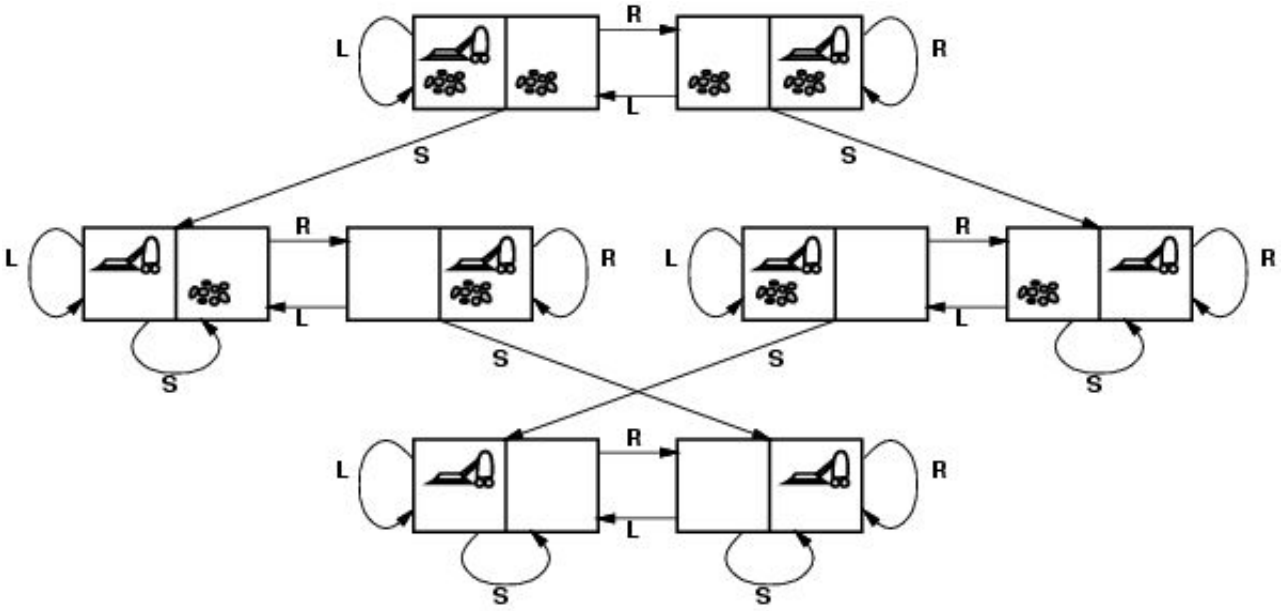
5

Vacuum world state space graph



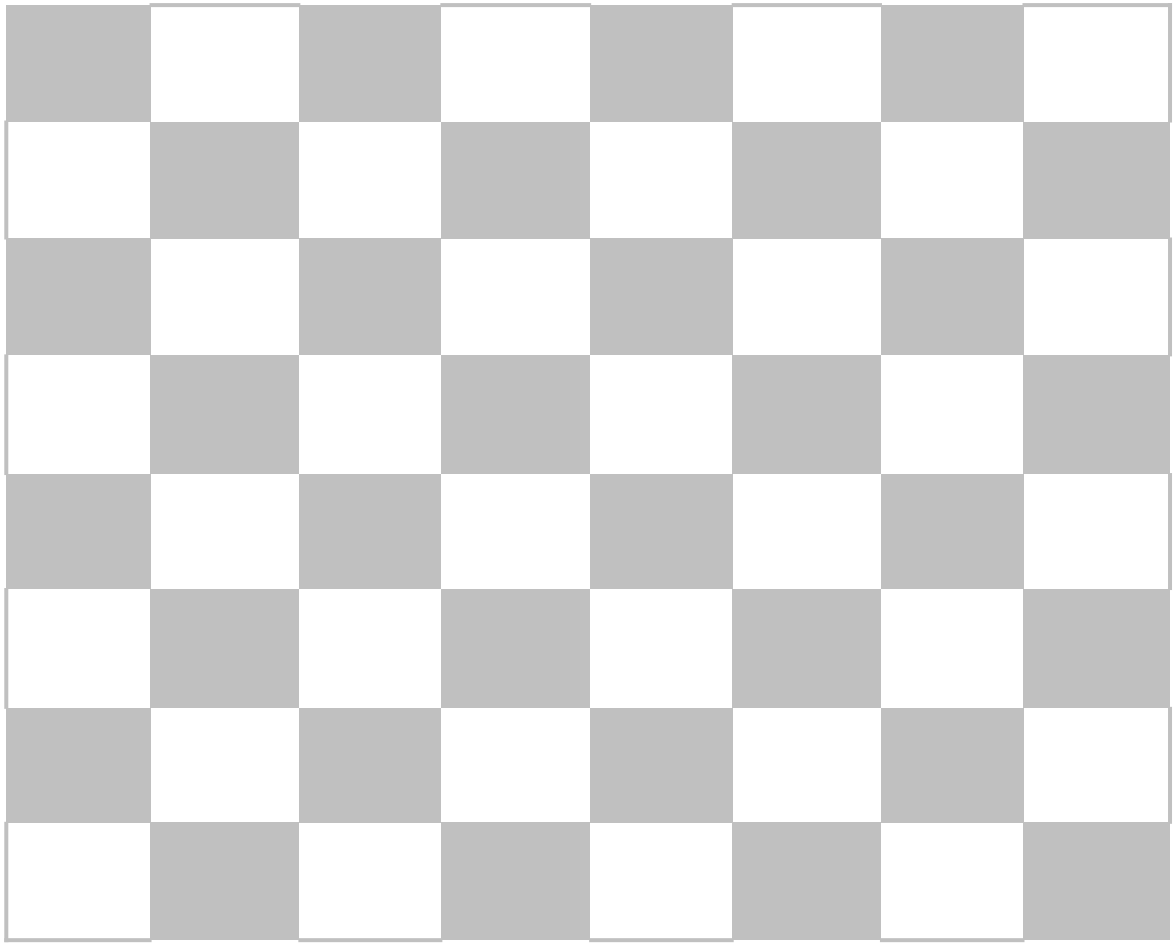
- states?
- actions?
- goal test?
- path cost?

Vacuum world state space graph

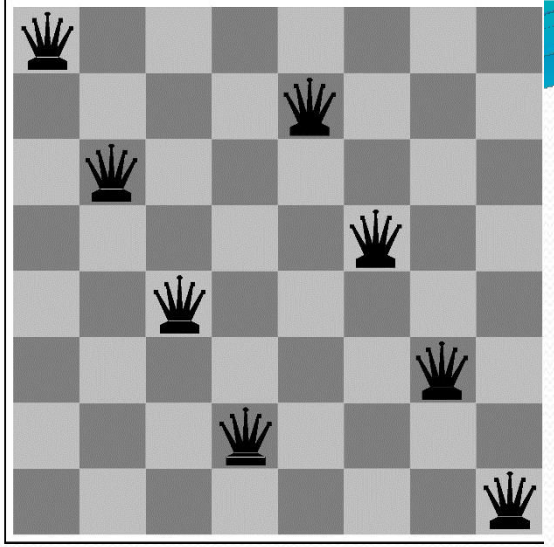


- states? integer dirt and robot location
- actions? *Left*, *Right*, *Suck*
- goal test? no dirt at all locations
- path cost? 1 per action

Example: 8-queen problem



Example: 8-Queens



- states? -any arrangement of n<=8 queens

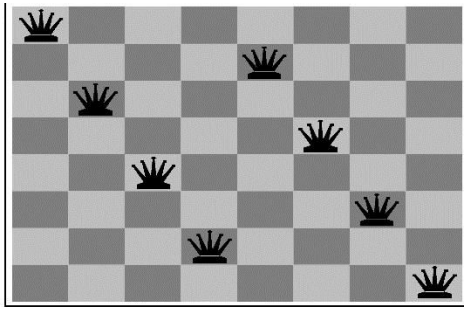
  -*or* arrangements of n<=8 queens in leftmost n columns, 1 per column, such that no queen attacks any other.

- initial state? no queens on the board
- actions? -add queen to any empty square

  -*or* add queen to leftmost empty square such that it is not attacked by other queens.

- goal test? 8 queens on the board, none attacked.
- path cost? 1 per move

8 queens – sol 2



- State : any arrangement of 8 Q on board
- Initial state : all Q at column 1
- Successor : change position of one Q
- Operator : move

8 queens – sol 3

- State : any arrangement of 8 Q on board
- Initial state : all Q at row 1
- Successor : change position of one Q
- Operator : move

The sliding tile problem

Start and Goal Configurations for the Eight–Puzzle

8-puzzle: 181,440 states
15-puzzle: 1.3 trilion
24-puzzle: 10^25

The Sliding Tile

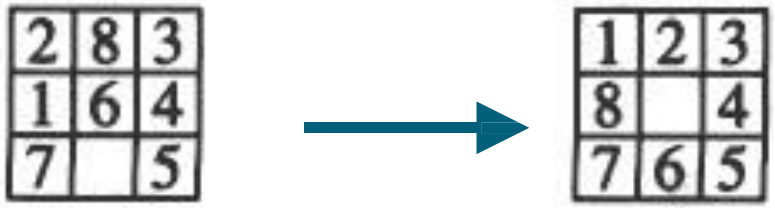


Figure 8.1

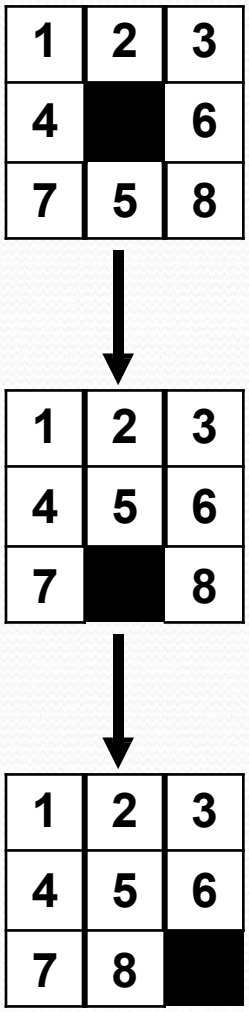
Start and Goal Configurations for the Eight-Puzzle

$move(x,loc\ y,loc\ z)$

Up
Down
Left
Right

The "8-Puzzle" Problem

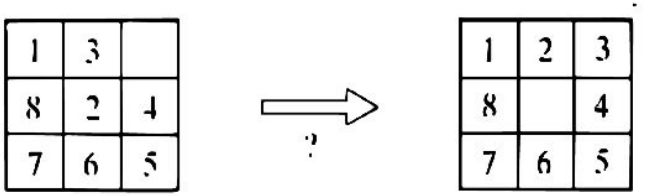
Start State



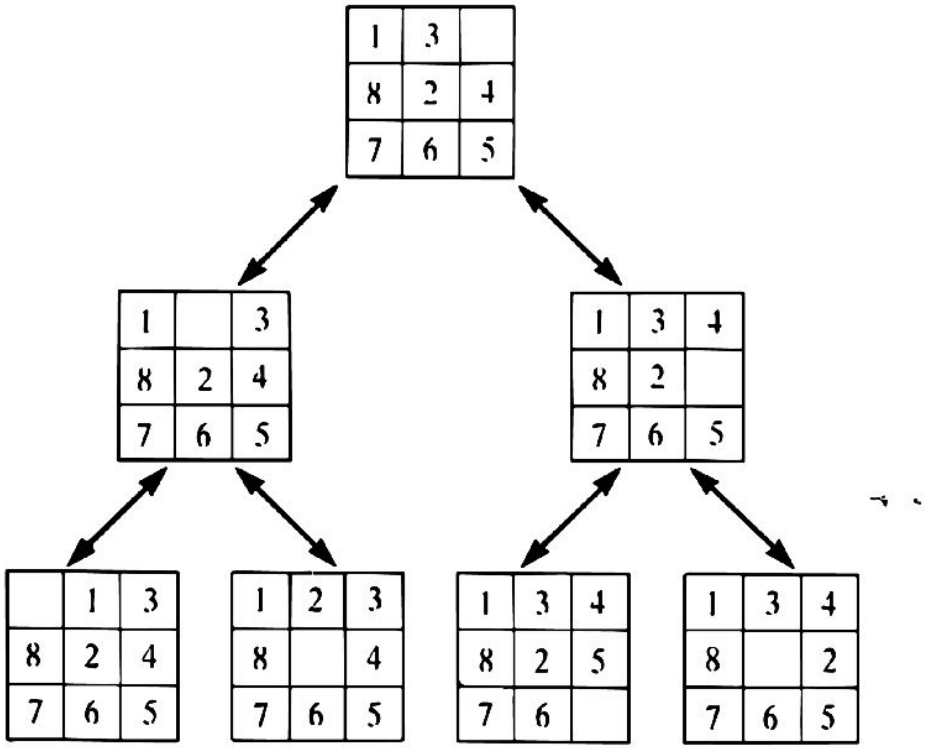
Goal State

8 puzzle

Initial State

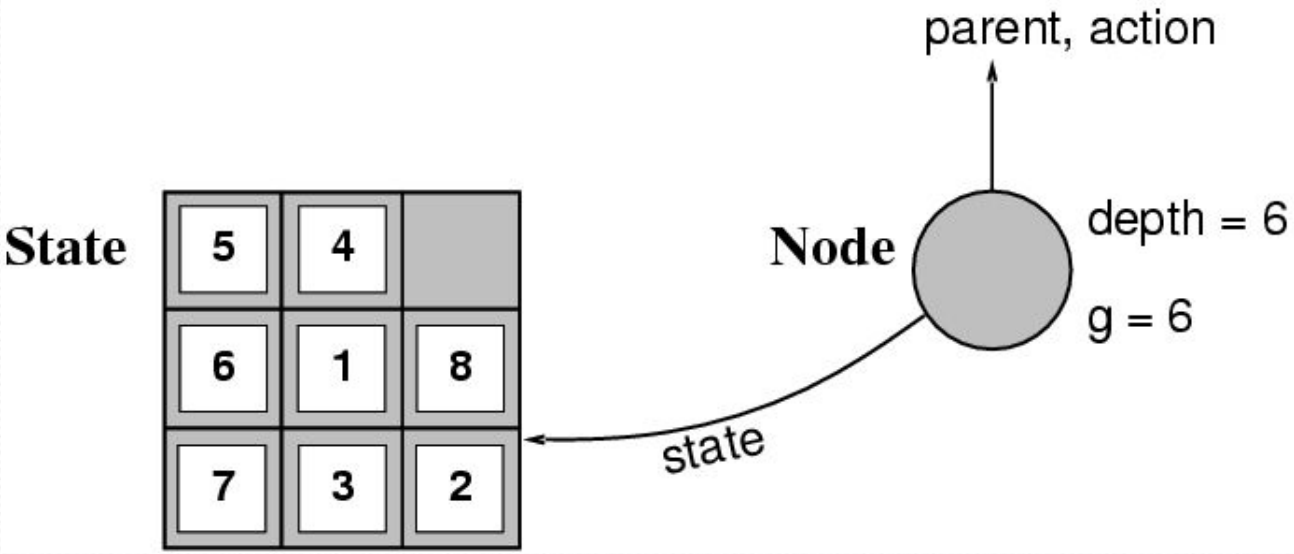


Final State



States versus Nodes

- A state is a (representation of) a physical configuration
- A node is a data structure constituting part of a search tree contains info such as: state, parent node, action, path cost $g(x)$, depth



- The Expand function creates new nodes, filling in the various fields and using the SuccessorFn of the problem to create the corresponding states.

State Spaces versus Search Trees

- State Space
  - Set of valid states for a problem
  - Linked by operators
  - e.g., 20 valid states (cities) in the Romanian travel problem
- **Search Tree**
  - Root node = initial state
  - Child nodes = states that can be visited from parent
  - Note that the depth of the tree can be infinite
    - E.g., via repeated states
  - Partial search tree
    - Portion of tree that has been expanded so far
- Fringe - Leaves of partial search tree, candidates for expansion

Search trees = data structure to search state-space

Search Tree for the 8 puzzle problem

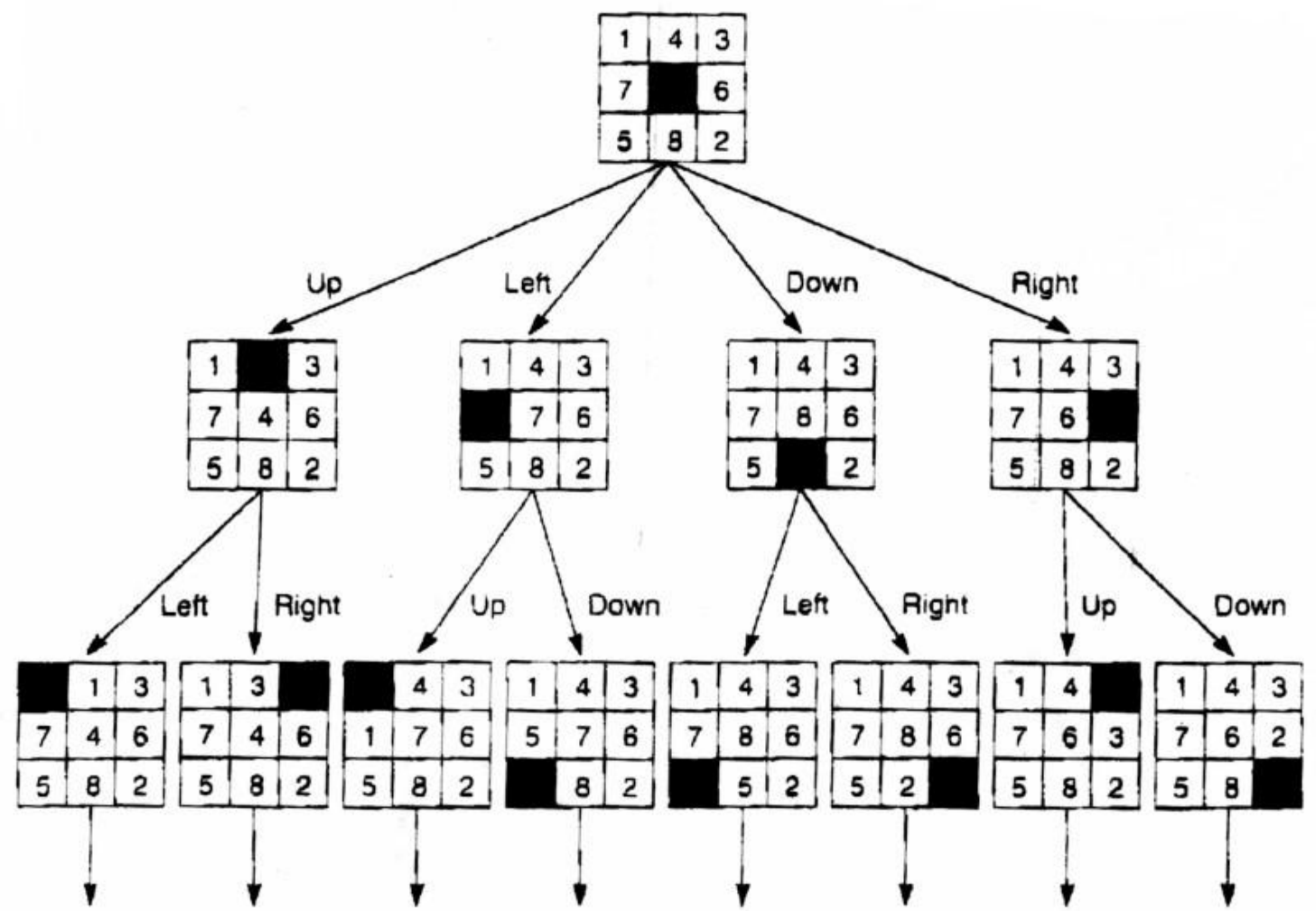


Figure 3.6 State space of the 8-puzzle generated by "move blank" operations.

Summary

- **Intelligent agents can often be viewed as searching for problem solutions in a discrete state-space**
- Space obtained through a problem abstraction process.
- Search consists of
  - state space
  - operators
  - start state
  - goal states

- **A Search Tree is an efficient way to represent a search**

- **There are a variety of general search techniques, including**
  - Depth-First Search
  - Breadth-First Search