



NANYANG  
TECHNOLOGICAL  
UNIVERSITY  
SINGAPORE

# Introduction to Data Science and Artificial Intelligence **Intelligent Agents**

Assoc Prof Bo AN

*Research area:* artificial intelligence,  
computational game theory, optimization  
[www.ntu.edu.sg/home/boan](http://www.ntu.edu.sg/home/boan)  
*Email:* boan@ntu.edu.sg  
*Office:* N4-02b-55





# Lesson Outline

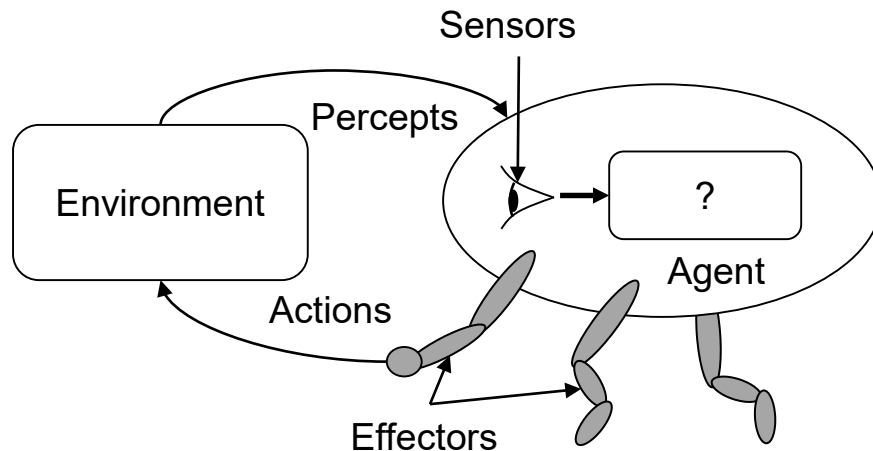
- What is an agent?
- How can one describe the task/problem for the agent?
- What are the properties of the task environment for the agent?
- What are the different basic kinds of agents for intelligent systems?
- Problem formulation



# Agent

An **agent** is an entity that

- **Perceives** through sensors (e.g. eyes, ears, cameras, infrared range sensors)
- **Acts** through effectors (e.g. hands, legs, motors)





# Rational Agents

- A rational agent is one that does the **right** thing
- **Rational action**: action that maximises the expected value of an objective **performance** measure given the percept sequence to date
- Rationality depends on
  - performance measure
  - everything that the agent has perceived so far
  - built-in knowledge about the environment
  - actions that can be performed



# Example: Google X2: Driverless Taxi

- **Percepts:** video, speed, acceleration, engine status, GPS, radar, ...
- **Actions:** steer, accelerate, brake, horn, display, ...
- **Goals:** safety, reach destination, maximise profits, obey laws, passenger comfort,...
- **Environment:** Singapore urban streets, highways, traffic, pedestrians, weather, customers, ...



Image source: [https://en.wikipedia.org/wiki/Waymo#/media/File:Waymo\\_Chrysler\\_Pacifica\\_in\\_Los\\_Altos,\\_2017.jpg](https://en.wikipedia.org/wiki/Waymo#/media/File:Waymo_Chrysler_Pacifica_in_Los_Altos,_2017.jpg)





# Example: Medical Diagnosis System

- **Percepts:** symptoms, findings, patient's answers, ...
- **Actions:** questions, medical tests, treatments, ...
- **Goals:** healthy patient, faster recovery, minimise costs, ...
- **Environment:** Patient, hospital, clinic, ...

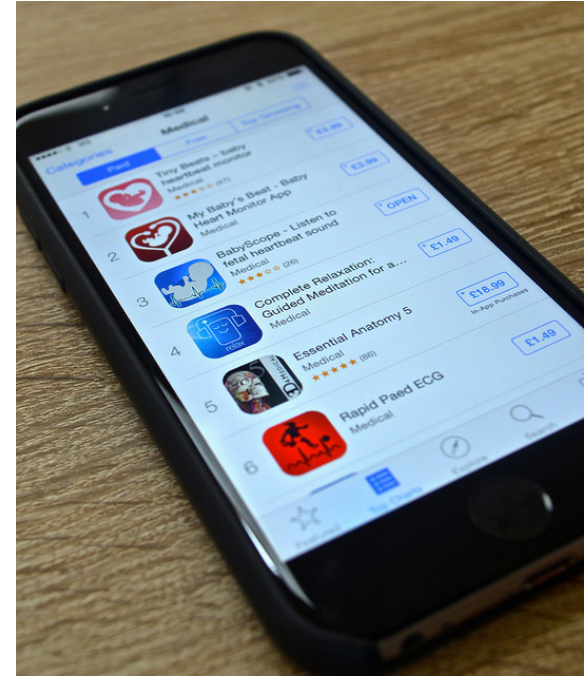


Image source: <https://www.flickr.com/photos/134647712@N07/20008817459>



# Autonomous Agents

- Do **not** rely entirely on built-in knowledge about the environment (i.e. not entirely pre-programmed)
- Otherwise,
  - The agent will **only** operates successfully when the built-in knowledge are all correct
- Adapt to the environments through experience

## Example: Driverless Car

- ❑ Learn to drive in driving center
- ❑ Drive at NTU
- ❑ Drive on public roads
- ❑ Drive in highways
- ❑ Drive in City Hall

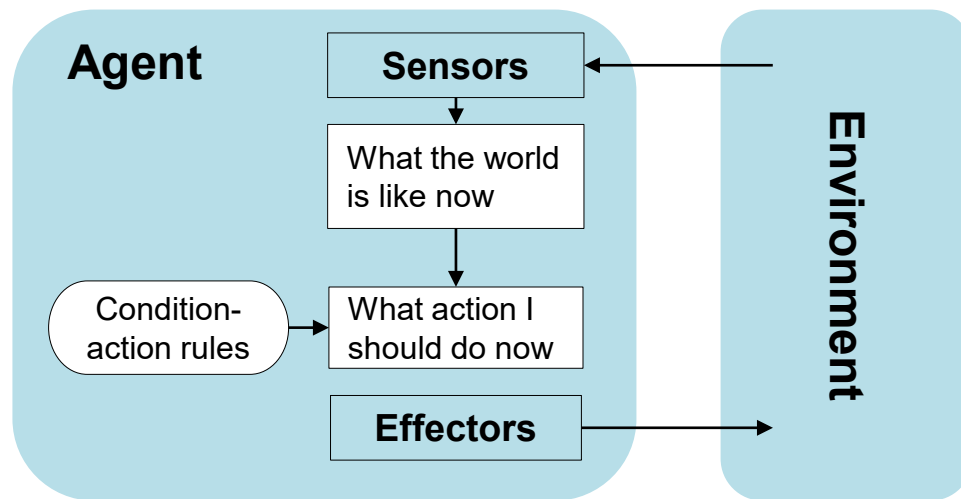


# Simple Reflex Agents

## Example

**If** car-in-front-is-braking **then** initiate-braking

1. Find the **rule** whose condition matches the current situation (as defined by the percept)
2. Perform the action associated with that rule





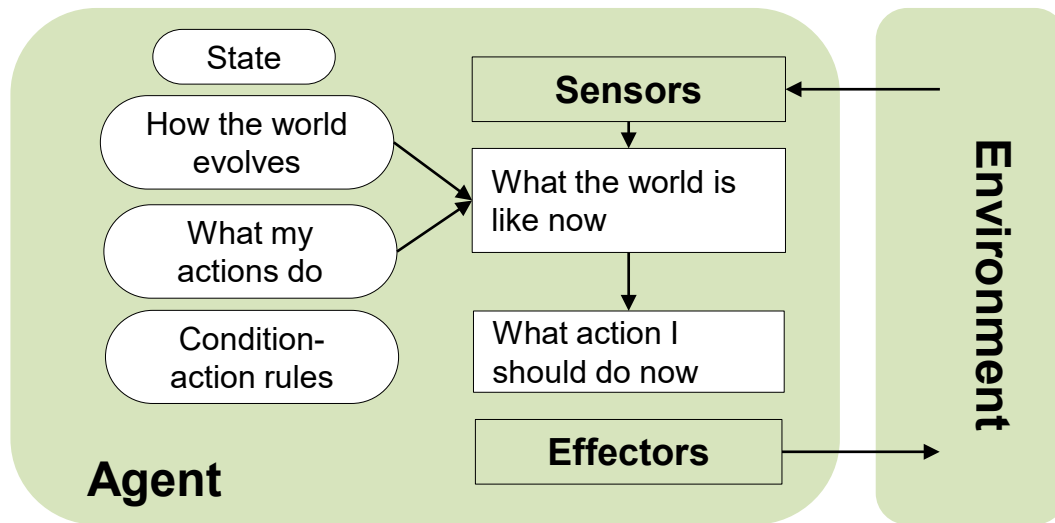


# Reflex Agents with State

## Example

If yesterday-at-NTU **and** no-traffic-jam-now **then** go-Orchard

1. Find the rule whose condition matches the current situation (as defined by the percept and the stored internal **state**)
2. Perform the action associated with that rule



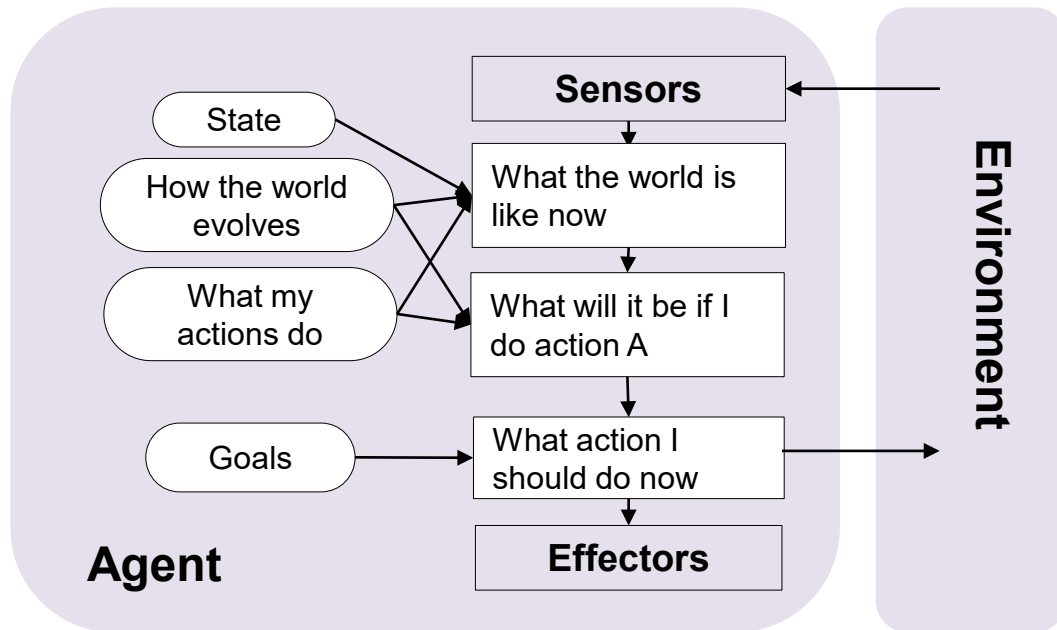


# Goal-Based Agents

Needs some sort of **goal** information

## Example: Driverless Taxi

- At a junction (known state), should I go left, right, or straight on?
- Reach Orchard (Destination)?



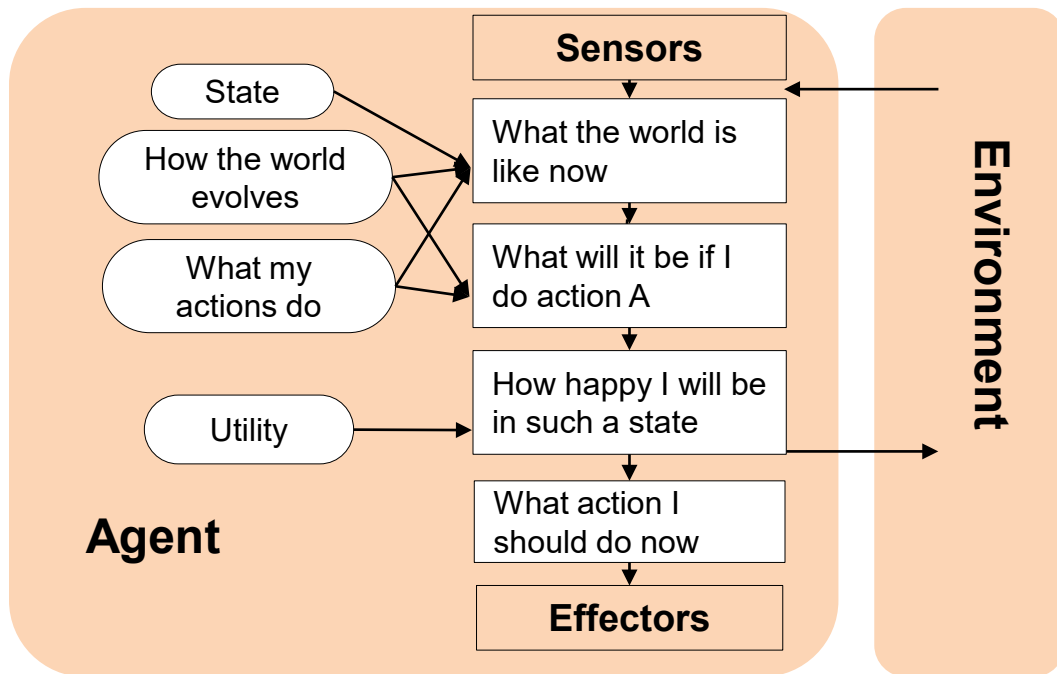


# Utility-Based Agents

- There may be **many** action sequences that can achieve the same goal, which action sequence should it take?
- How happy will agent be if it attains a certain state? → Utility

## Example: Driverless Taxi

- Go to Orchard (Destination) via PIE? AYE?
- Which one charges lower fare?



Question to think:

- Why is problem formulation (e.g., characterize percept, goals etc) important?





# Types of Environment

Accessible (vs  
inaccessible)

Agent's sensory apparatus gives it access to the complete state of the environment

Deterministic (vs  
nondeterministic)

The next state of the environment is completely determined by the current state and the actions selected by the agent

Episodic (vs  
Sequential)

Each episode is not affected by the previous taken actions

Static (vs  
dynamic)

Environment does not change while an agent is deliberating

Discrete (vs  
continuous)

A limited number of distinct percepts and actions





# Example: Driverless Taxi

Accessible?	No. Some traffic information on road is missing
Deterministic?	No. Some cars in front may turn right suddenly
Episodic?	No. The current action is based on previous driving actions
Static?	No. When the taxi moves, Other cars are moving as well
Discrete?	No. Speed, Distance, Fuel consumption are in real domains



# Example: Chess

Accessible?	Yes. All positions in chessboard can be observed
Deterministic?	Yes. The outcome of each movement can be determined
Episodic?	No. The action depends on previous movements
Static?	Yes. When there is no clock, when are you considering the next step, the opponent can't move; <b>Semi</b> . When there is a clock, and time is up, you will give up the movement
Discrete?	Yes. All positions and movements are in discrete domains



# Example: Minesweeper

Accessible?	No. Mines are hidden
Deterministic?	No. Mines are randomly assigned in different positions
Episodic?	No. The action is based on previous outcomes
Static?	Yes. When are you considering the next step, no changes in environment
Discrete?	Yes. All positions and movements are in discrete domains



# Slot machines - One-Armed Bandit

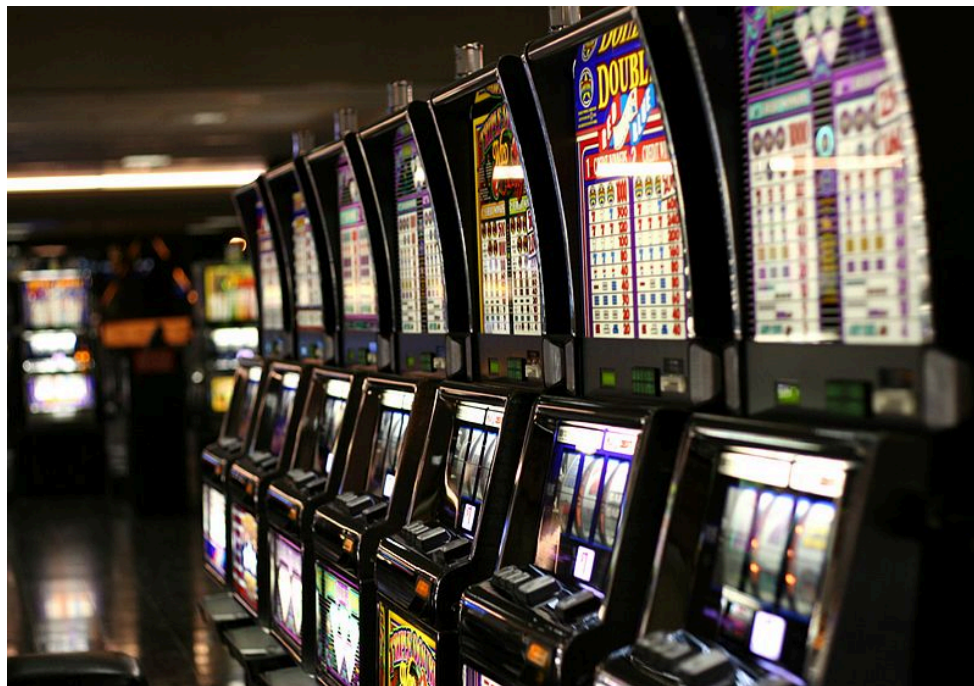


Image source: [https://commons.wikimedia.org/wiki/File:Las\\_Vegas\\_slot\\_machines.jpg](https://commons.wikimedia.org/wiki/File:Las_Vegas_slot_machines.jpg)

Accessible?	No
Deterministic?	No
Episodic?	Yes
Static?	Yes

Question to think:

- Can you find another scenario whose environment is episodic?







# Design of Problem-Solving Agent

## Idea

- Systematically considers the **expected outcomes** of different possible sequences of actions that lead to states of known value
- Choose the best one
  - shortest journey from A to B?
  - most cost effective journey from A to B?



# Design of Problem-Solving Agent

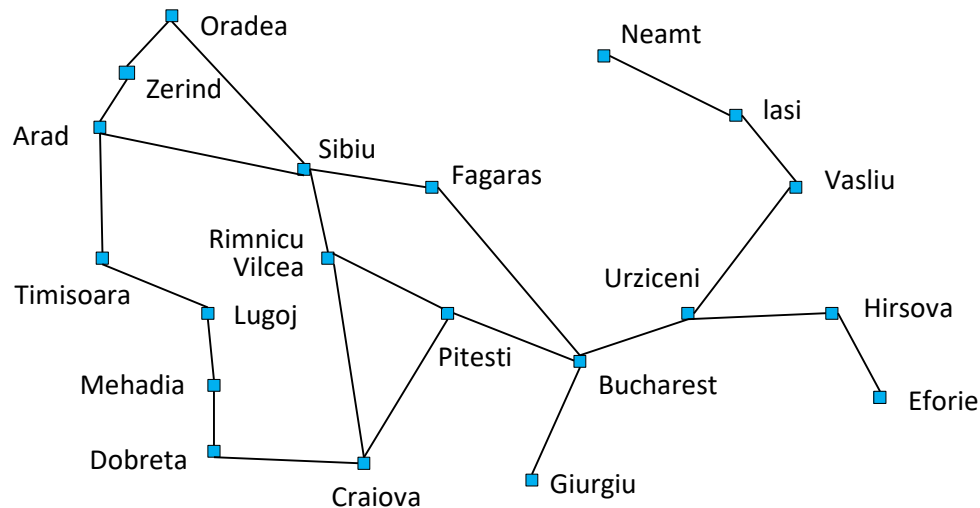
## Steps

1. Goal formulation
2. Problem formulation
3. Search process
  - No knowledge → uninformed search
  - Knowledge → informed search
4. Action execution (follow the recommended route)



# Goal-based Agent: Example

On holiday in Romania



- Currently in Arad (**Initial state**). Flight leaves tomorrow from Bucharest.
- **Goal**: be in Bucharest (other factors: cost, time, most scenic route, etc.)
- **State**: be in a city (defined by the map)
- **Action**: transition between states (highways defined by the map)



# Example: Vacuum Cleaner Agent

- Robotic vacuum cleaners move autonomously
- Some can come back to a docking station to charge their batteries
- A few are able to empty their dust containers into the dock as well

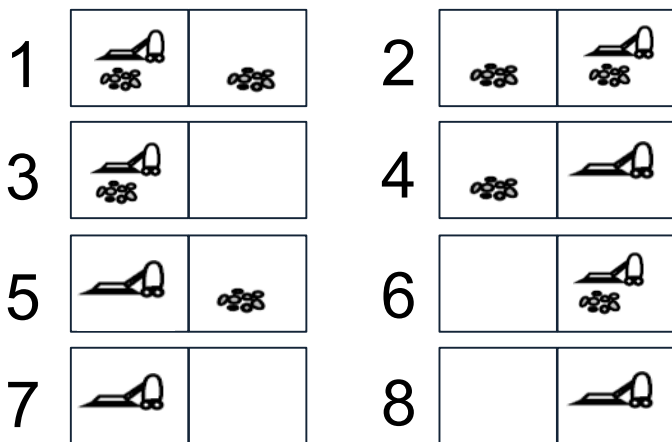


Image source: [https://commons.wikimedia.org/wiki/File:IRobot\\_Roomba\\_780.jpg](https://commons.wikimedia.org/wiki/File:IRobot_Roomba_780.jpg)



# Example: A Simple Vacuum World

Two locations, each location may or may not contain dirt, and the agent may be in one location or the other.



- 8 possible world states
- Possible actions: **left**, **right**, and **suck**
- Goal: clean up all dirt →  
Two goal states, i.e. {7, 8}



# Well-Defined Formulation

Definition of a problem	The information used by an agent to decide what to do
Specification	<ul style="list-style-type: none"><li>• Initial state</li><li>• Action set, i.e. available actions (successor functions)</li><li>• State space, i.e. states reachable from the initial state<ul style="list-style-type: none"><li>• Solution path: sequence of actions from one state to another</li></ul></li><li>• Goal test predicate<ul style="list-style-type: none"><li>• Single state, enumerated list of states, abstract properties</li></ul></li><li>• Cost function<ul style="list-style-type: none"><li>• Path cost <math>g(n)</math>, sum of all (action) step costs along the path</li></ul></li></ul>
Solution	A path (a sequence of operators leading) from the Initial-State to a state that satisfies the Goal-Test





# Measuring Problem-Solving Performance

## Search Cost

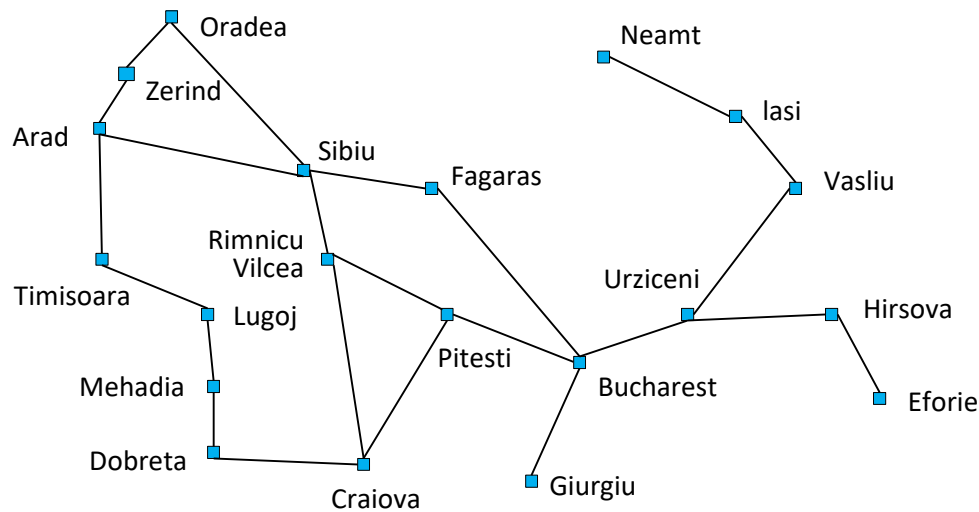
- What does it cost to find the solution?
  - e.g. How long (time)? How many resources used (memory)?

## Total cost of problem-solving

- Search cost ("offline") + Execution cost ("online")
- Trade-offs often required
  - Search a very long time for the optimal solution, or
  - Search a shorter time for a "good enough" solution



# Single-State Problem Example

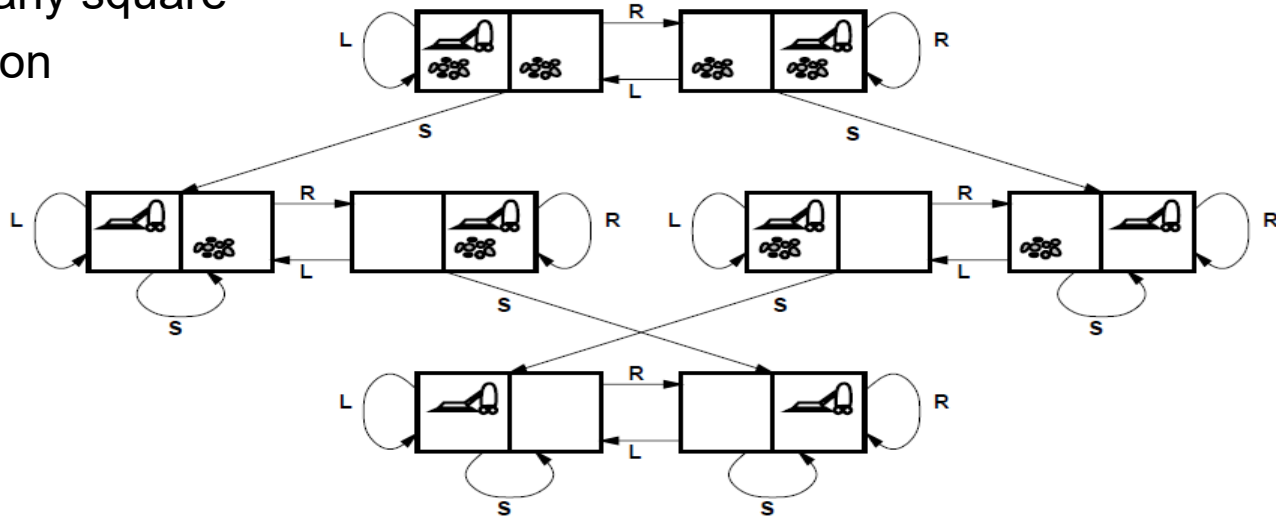


- **Initial state:** e.g., “at Arad”
- Set of possible **actions** and the corresponding next states
  - e.g., Arad  $\rightarrow$  Zerind
- **Goal test:**
  - explicit (e.g.,  $x =$  “at Bucharest”)
- **Path cost** function
  - e.g., sum of distances, number of operators executed solution: a sequence of operators leading from the initial state to a goal state

# Example: Vacuum World (Single-state Version)



- **Initial state:** one of the eight states shown previously
- **Actions:** left, right, suck
- **Goal test:** no dirt in any square
- **Path cost:** 1 per action





# Example: 8-puzzle

- **States**: integer locations of tiles
  - number of states =  $9!$
- **Actions**: move blank left, right, up, down
- **Goal test**: = goal state (given)
- **Path cost**: 1 per move

Start state

5	4	
6	1	8
7	3	2

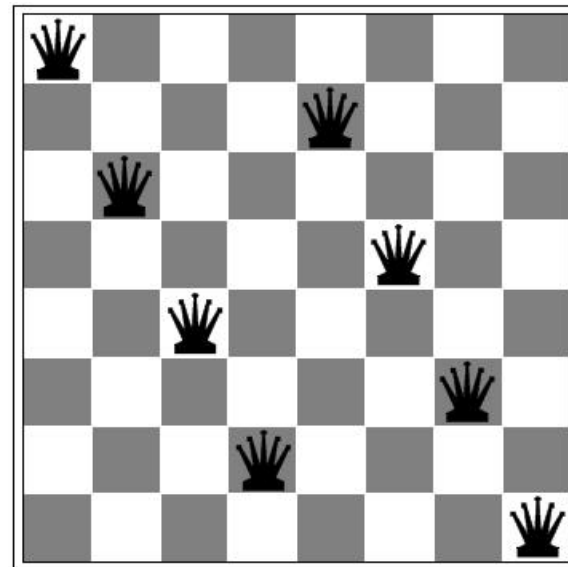
Goal state

1	2	3
8		4
7	6	5



# Example: 8-queens

- **States:** Any arrangement of 0 to 8 queens on the board
- **Actions:** Add a queen to any empty square
- **Goal test:** 8 queens are on the board, none attacked
- **Path cost:** Not necessary





# Real-World Problems

Route finding problems:

- Routing in computer networks
- Robot navigation
- Automated travel advisory
- Airline travel planning

Touring problems:

- Traveling Salesperson problem
- “Shortest tour”: visit every city exactly once



Image source: [https://commons.wikimedia.org/wiki/File:Font\\_Awesome\\_5\\_solid\\_route.svg](https://commons.wikimedia.org/wiki/File:Font_Awesome_5_solid_route.svg)