Artificial Intelligence

Lecture 2

Supta Richard Philip

Department of CSE
City University

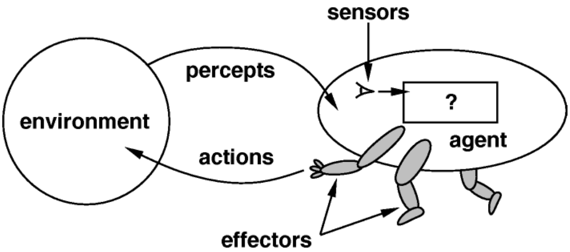
City University, April 2019

Table of Contents

Table of Contents

Intelligent Agents

- An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors.
- Our aim is to design agents.
- A rational agent is one that performs the actions that cause the agent to be most successful.



Structure of Intelligent Agents

- The job of AI is to design the **agent program:** a function that implements the agent mapping from percepts to actions. We assume this program will run on some sort of computing device call the architecture.
- The **architecture** makes the percepts from the sensors available to the agent program. runs the program and feeds the program's action choices to the effectors as they are generated.
- **agent=architecture + program**

Agent Programs

- All agent programs have roughly the same skeleton; they accept percepts from the environment and generate actions.

```
function SKELETON-AGENT(percept) returns action
   static: memory, the agent's memory of the world

   memory ← UPDATE-MEMORY(memory, percept)
   action ← CHOOSE-BEST-ACTION(memory)
   memory ← UPDATE-MEMORY(memory, action)
   return action
```

PAGE (Percepts, Actions, Goals, Environment)

Must first specify the setting for intelligent agent design.
Consider, e.g., the task of designing **an automated taxi:**

- **Percepts:** video, accelerometers, gauges, engine sensors, keyboard, GPS, . . .
- **Actions:** steer, accelerate, brake, horn, speak/display, . . .
- **Goals:** safety, reach destination, maximize profits, obey laws, passenger comfort, . . .
- **Environment:** US urban streets, freeways, traffic, pedestrians, weather, customers, . . .

More Example

Agent Type	Percepts	Actions	Goals	Environment
Medical diagnosis system	Symptoms, findings, patient's answers	Questions, tests, treatments	Healthy patient, minimize costs	Patient, hospital
Satellite image analysis system	Pixels of varying intensity, color	Print a categorization of scene	Correct categorization	Images from orbiting satellite
Part-picking robot	Pixels of varying intensity	Pick up parts and sort into bins	Place parts in correct bins	Conveyor belt with parts
Refinery controller	Temperature, pressure readings	Open, close valves; adjust temperature	Maximize purity, yield, safety	Refinery
Interactive English tutor	Typed words	Print exercises, suggestions, corrections	Maximize student's score on test	Set of students

Environment types

- Fully observable (vs. partially observable): An agent's sensors give it access to the complete state of the environment at each point in time.
- Deterministic (vs. stochastic): The next state of the environment is completely determined by the current state and the action executed by the agent. (If the environment is deterministic except for the actions of other agents, then the environment is strategic)
- Episodic (vs. sequential): An agent's action is divided into atomic episodes. Decisions do not depend on previous decisions/actions.

Environment types - cont.

- Static (vs. dynamic): The environment is unchanged while an agent is deliberating. (The environment is semidynamic if the environment itself does not change with the passage of time but the agent's performance score does)
- Discrete (vs. continuous): A limited number of distinct, clearly defined percepts and actions. How do we represent or abstract or model the world?
- Single agent (vs. multi-agent): An agent operating by itself in an environment. Does the other agent interfere with my performance measure?

Task and Environment

task environm.	observable	determ./ stochastic	episodic/ sequential	static/ dynamic	discrete/ continuous	agents
crossword puzzle	fully	determ.	sequential	static	discrete	single
chess with clock	fully	strategic	sequential	semi	discrete	multi
poker	partial	stochastic	sequential	static	discrete	multi
back gammon	fully	stochastic	sequential	static	discrete	multi
taxi driving	partial	stochastic	sequential	dynamic	continuous	multi
medical diagnosis	partial	stochastic	sequential	dynamic	continuous	single
image analysis	fully	determ.	episodic	semi	continuous	single
partpicking robot	partial	stochastic	episodic	dynamic	continuous	single
refinery controller	partial	stochastic	sequential	dynamic	continuous	single
interact. Eng. tutor	partial	stochastic	sequential	dynamic	discrete	multi

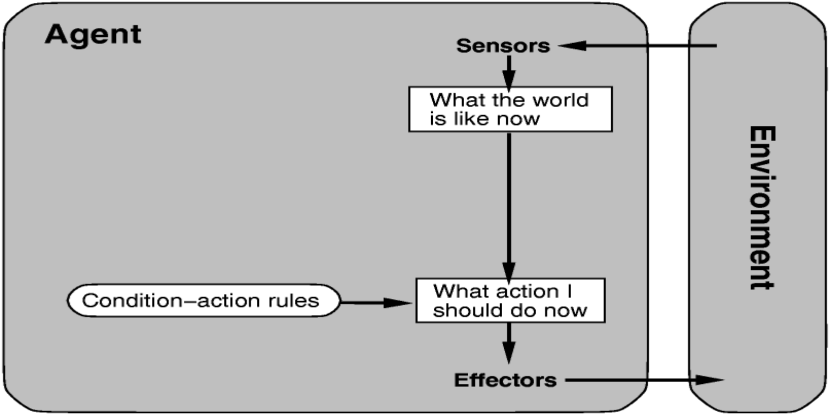
ICS-171: 16

y

Agent types

- Simple reflex agents
- Agents that keep track of the world
- Goal-based agents
- Utility agents

Simple reflex agents



Simple reflex agents Algorithm

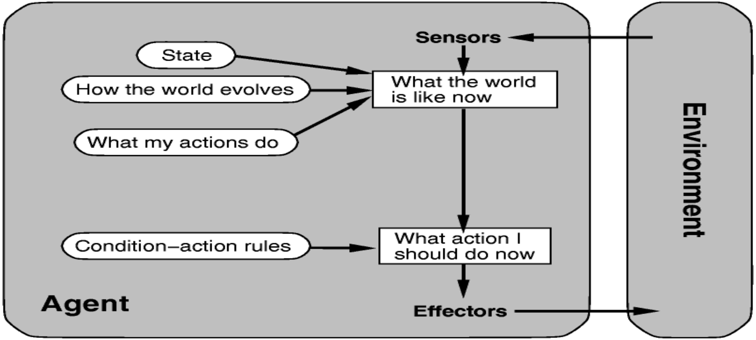
- If **condition** then **action**

```
function SIMPLE-REFLEX-AGENT(percept) returns action
   static: rules, a set of condition-action rules

   state ← INTERPRET-INPUT(percept)
   rule ← RULE-MATCH(state, rules)
   action ← RULE-ACTION[rule]
   return action
```

Reflex agents with state

- Agents that keep track of the world(state).



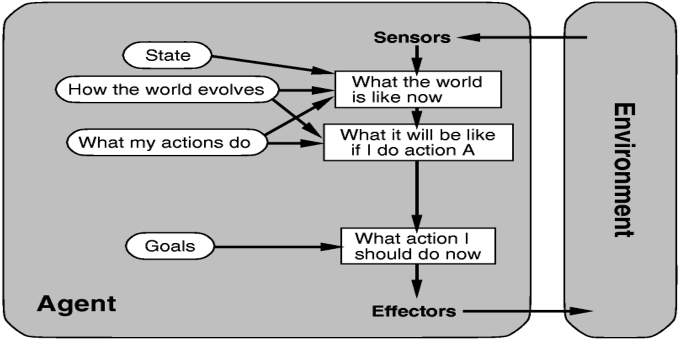
Reflex agents with state Algorithm

```
function REFLEX-AGENT-WITH-STATE(percept) returns action
   static: state, a description of the current world state
           rules, a set of condition-action rules

   state ← UPDATE-STATE(state, percept)
   rule ← RULE-MATCH(state, rules)
   action ← RULE-ACTION[rule]
   state ← UPDATE-STATE(state, action)
   return action
```

Goal-based agents

- **Search** and **Planning** are the sub fields of AI devoted to finding action sequences that achieve goals.



Utility-based agents

- Goals alone are not enough to generate high-quality behaviour. For example, there are many action sequences that will get the taxi to its destination, but some are quicker, safer, more reliable, cheaper, etc.

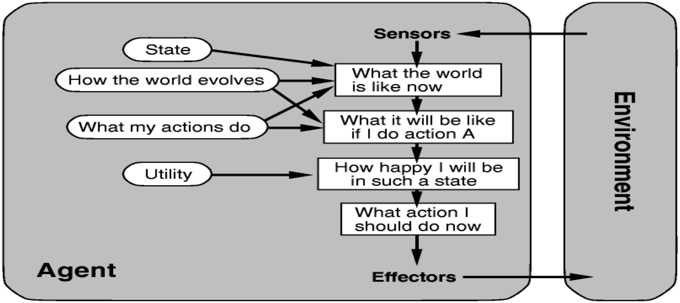


Table of Contents

Problem formulation

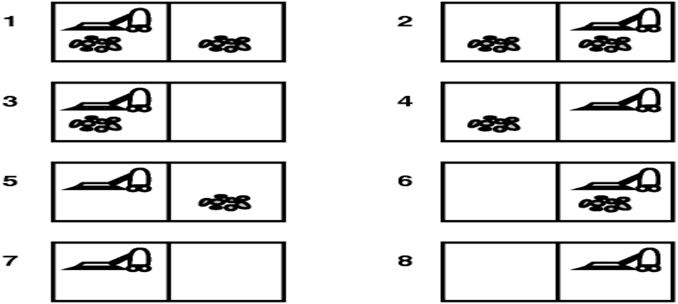
- Intelligent agents are supposed to act in such a way that the environment goes through a sequence of states that maximizes the performance measure.
- We will consider a goal to be a set of states - just those states in which the goal is satisfied.
- Actions can be viewed as causing transitions between states.
- Problem formulation is the process of deciding what actions and states to consider.

Search Problem

A search problem consists of:

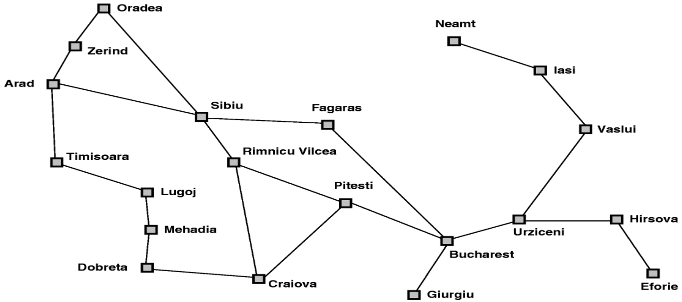
- A state space
- A successor function(with action, cost)
- A start state and a goal test
- A solution is a sequence of actions (a plan) which transforms the start state to a goal state
- Example - Pacman game 3x3 board

Example- Vacuum World



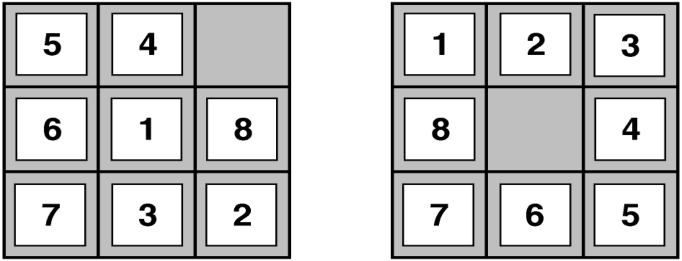
- In this case there are eight possible world states.
- There are three possible actions: left, right, and suck.
- The goal is to clean up all the dirt, i.e., the goal is equivalent to the set of states 7,8.

Example - Romania



- initial states: Arad
- goal state: Bucharest
- operators: successor function $S(x)$ set of possible actions
- path cost: a function that assigns a cost to a path.

Example - The 8-puzzle problem

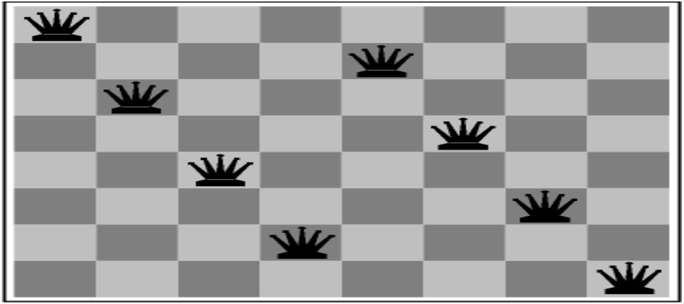


Start State **Goal State**

- states: a state description specifies the location of each of the eight tiles in one of the nine squares. For efficiency it is also useful to include a location for the blank.
- operators: blank moves left, right, up or down.
- goal test: as in figure
- path cost: length of path

Example - The 8-queens problem

The goal of this problem is to place 8 queens on the board so that none can attack the others. The following is not a solution!



- goal test: 8 queens on board, none attacked
- path cost: irrelevant
- states: any arrangement of 0-8 queens on the board
- operators: add or remove a queen to/from any square

Table of Contents

References

- Stuart Russell and Peter Norvig. 2009. Artificial Intelligence: A Modern Approach (3rd ed.). Prentice Hall Press, Upper Saddle River, NJ, USA.
- http://www.massey.ac.nz/ mjjohnso/notes/59302/all.html