Reinforcement Learning (RL)

Review Chapter:

Quick Introduction to Intelligent Agents

Review (Artificial Intelligence)

Saeed Saeedvand, Ph.D.

Contents

In this Chapter:

- ✓ Agents and environments
- ✓ Being rational
- ✓ Performance metric, environment, actuators, sensors
- ✓ Types of environments
- ✓ Types of agents

Objective:

- ✓ In this introductionary chapter we aim to understand related basic concepts of AI and underlying relations between AI and RL.

Agents

What is an Agent?

- ✓ Anything that can control its surroundings through sensors, understand and act in that environment through influencers.

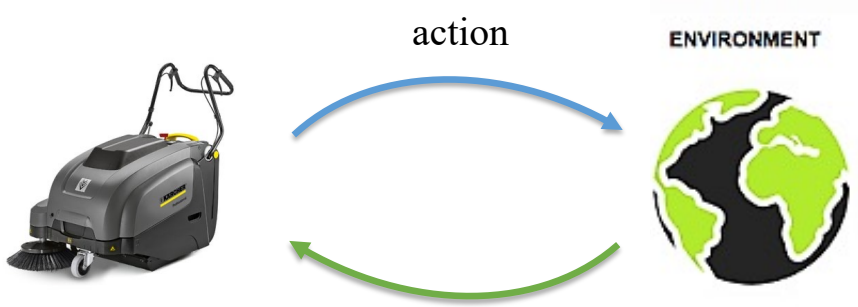
Human as an agent:

- ✓ Eyes, ears and other sensory organs as sensors
- ✓ Hands, feet, mouth and other body parts as effectors

Robot as an agent:

- ✓ Cameras, laser scanner, and infrared distance meter as sensors
- ✓ Types of engines as actuators

Agents and environment



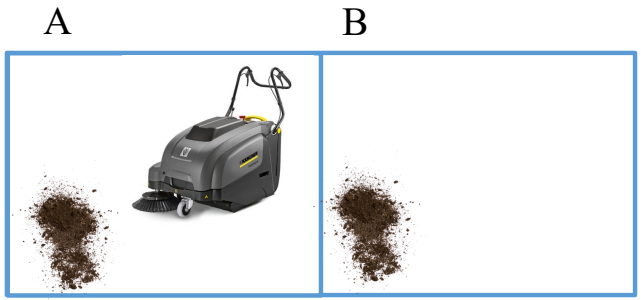
- The agent function maps the perceptual history to the actions:

$$[f: P^* \rightarrow A]$$

- Operating program runs on physical architecture to generate f

Agent = architecture + application

Sweeper world!



Perceptions:

- ✓ The location and contents, e.g. [A, dirty]

Actions:

- ✓ Move left and right, suck and NoP

Logical agents

✓ An agent must be based on what it can understand and the actions it can take as the right thing to do to be the most successful agent "**do the right thing**"

Efficiency measure:

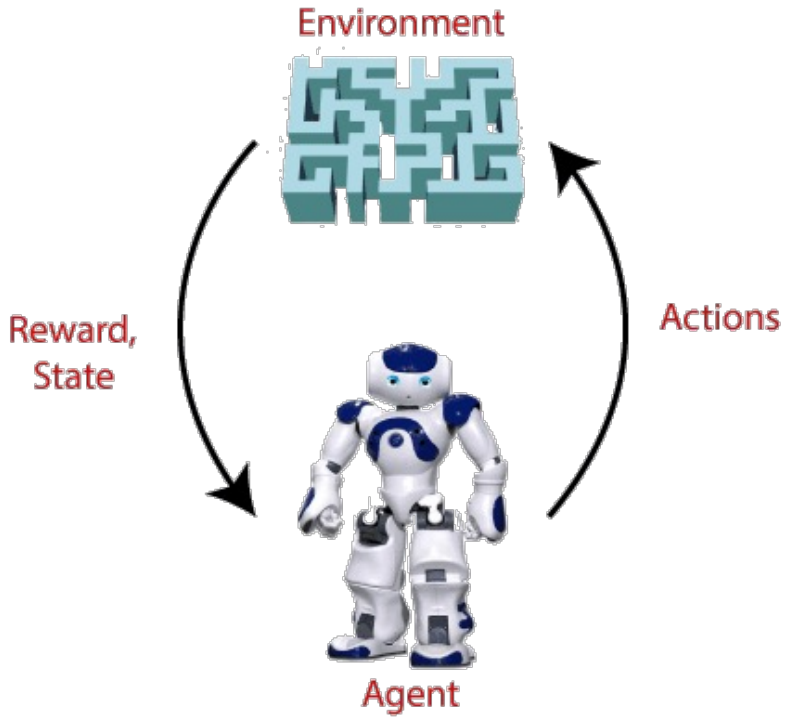
A target for measuring the success rate of a agent's behavior

Example: Success measure of the operating world of sweeper (vacuum cleaner):

- The amount of dust cleaned
- The amount of time consumed
- The amount of electricity consumed
- The amount of noise produced
- ... ?

Logical agents

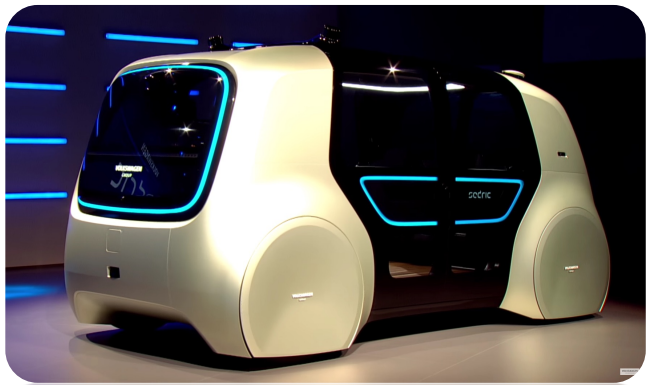
- ✓ For each possible perceptual sequence, a rational agent must select an action based on the evidence received from the perceptual sequence and internal knowledge that is expected to **maximize its efficiency**.



Rational Agents

- ✓ To be **Rational** with **Comprehensive knowledge** (to know everything with an Unlimited knowledge) **is different**.
- ✓ The **agent can perform actions that provide useful** information **through changes in future perceptions** (knowledge gathering, exploration)
- ✓ An agent is **autonomous** if its behavior is determined by its experience (along with learning and adaptability)

PEAS
(Performance measure, Environment, Actuators, Sensors)



✓ In designing an agent, the PEAS must first be determined.

Example: Self-driving Taxi (Unmanned vehicle)

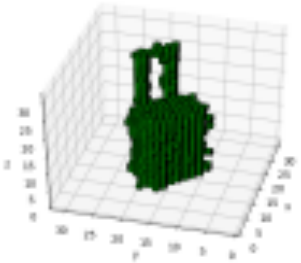
- **Performance measure:** security, speed, convenience, profit, etc.
- **Environment:** streets, pedestrians, customers, etc.
- **Actuators:** steering wheel, accelerator, brakes, horn, lights, etc.
- **Sensors:** speedometer, cameras, odometry sensors, engine sensors, keys, microphone, etc.

PEAS

Example: Medical diagnosis system

- **Performance measure:** patient health, cost minimization, etc.
- **Environment:** patient, hospital, staff, etc.
- **Actuators:** Screen (questions, tests, diagnoses, treatment), etc.
- **Sensors:** Cameras, Scanner, keyboard (Receive patient symptoms, findings and responses), etc.

PEAS

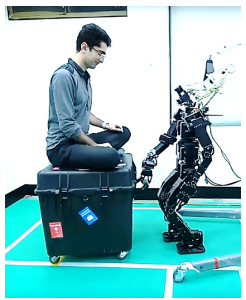


Example: An object-dragging robot

- **Performance measure:** Distance, Stability, Success rate, time, etc.
- **Environment:** Floor and objects, etc.
- **Actuators:** Arms and feet, etc.
- **Sensors:** Camera, joint angles, LIDAR, pressure sensors, etc.







PEAS

Example: Conversational language teacher

- **Performance measure:** maximizing the student's score in the exam, etc.
- **Environment:** a collection of students.
- **Actuators:** Screen (exercises, suggestions and corrections), speakers, arms, etc.
- **Sensors:** Touch screen, keyboard, voice commands, etc.

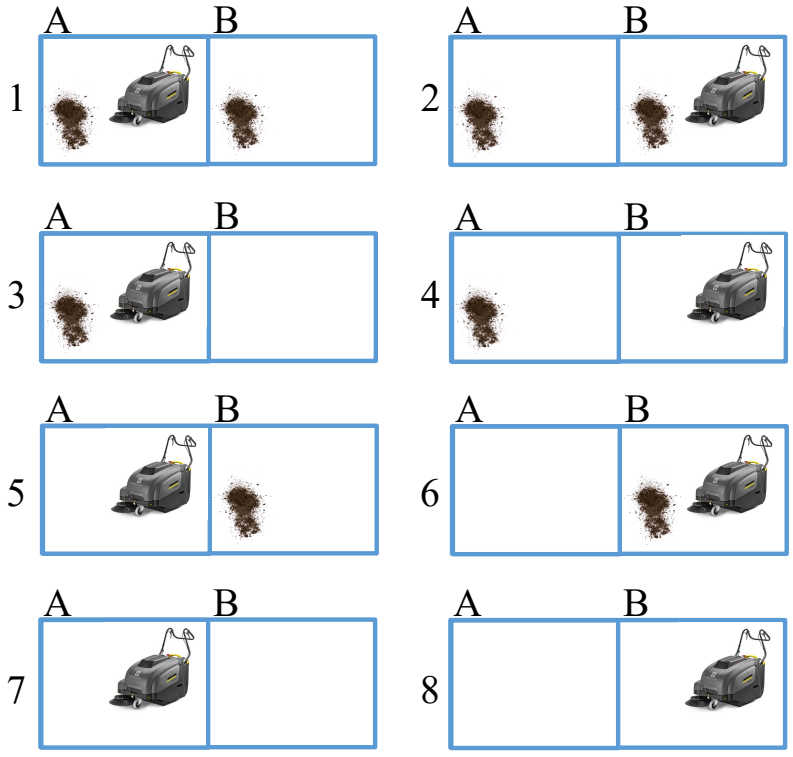
Environment

✓ Each environment has a set of **states**:

- The environment is only in one of these **states at any given time**.

Example: The world of vacuum cleaner.

S = {1, 2, 3, 4, 5, 6, 7, 8}



Environment and agent

✓ At the starting moment, the environment is in one of the possible states.

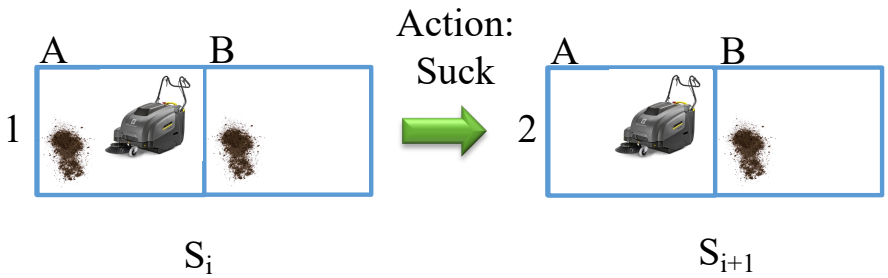
- The action of the agent in the environment, changes the state of the environment

  • Current state: S_i

  • Action: A

  • Next state: S_{i+1}

✓ Example: The world of vacuum cleaner.



Types of Environment

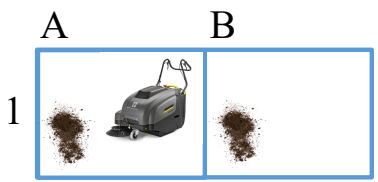
Fully observable:

- ✓ As opposed to partially observable
- ✓ An environment in which operating sensors allow the **full state of the environment** to be accessed at any **point in time**.

Example: The world of vacuum cleaner.

- ✓ Sensors: [location, status]
  - Location detection: left or right
  - Status detection: dirty or clean

S_i= [LEFT, DIRTY]



S_i

S_i= [[LEFT, RIGHT], [CLEAN, DIRTY]]

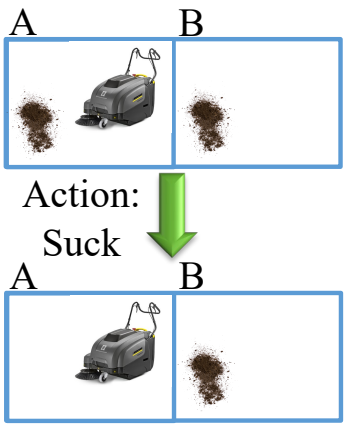
Types of Environment

Deterministic:

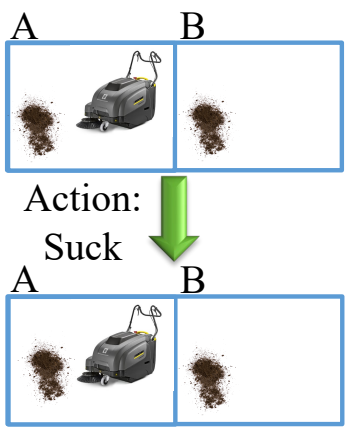
- ✓ The next state of the environment can be completely determined by the current state and the action performed by the agent (It is opposed to Stochastic).

Note: If the environment is **deterministic** except for the action of other agents, then the environment is strategic.

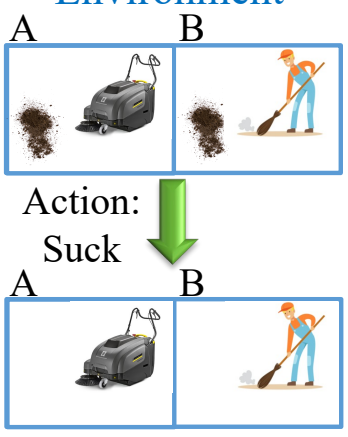
Deterministic Environment



Stochastic Environment



Strategic Environment

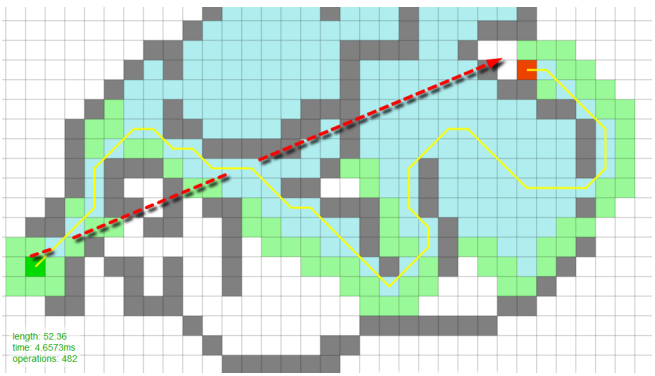


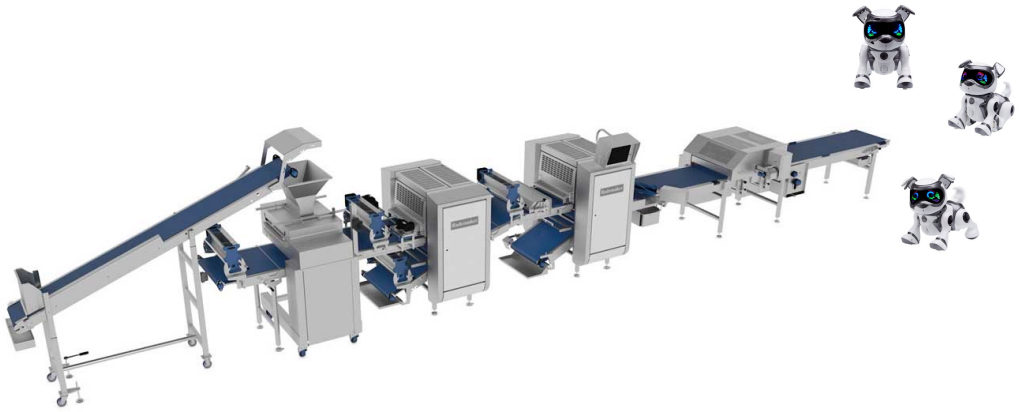
Types of Environment

Episodic:

- ✓ **Non-decomposable "episode":** The experience of the agent is divided into episode (the perception of the agent and then the performance of an action)
- ✓ The choice of action in each episode depends only on the period itself.

Example: Path planning, or Quality control robot.





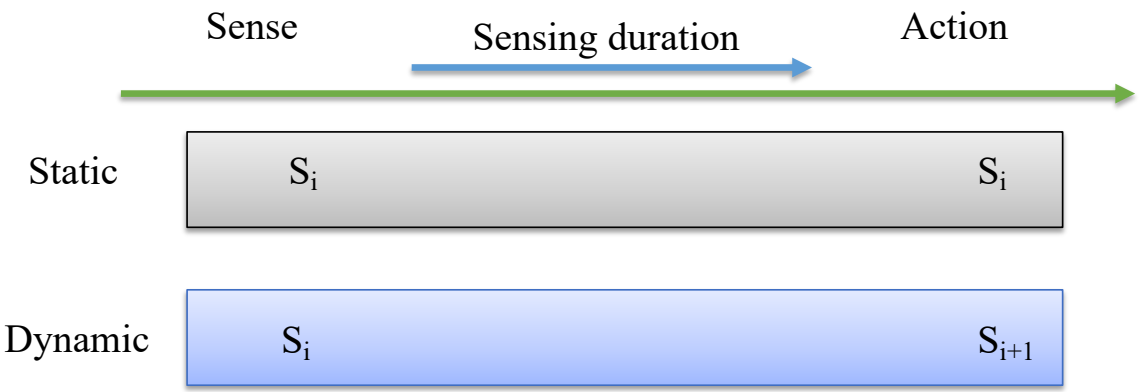
Types of Environment

Static:

- ✓ **Static (as opposed to dynamic):** The environment does not change while measuring the agent (to select the action).
- ✓ Note: here we only mean state of the environment (excluding agent itself), but in RL we consider both.

Semi-dynamic:

- ✓ If the environment itself does not change over time but the performance measure changes, then the environment is semi-dynamic.

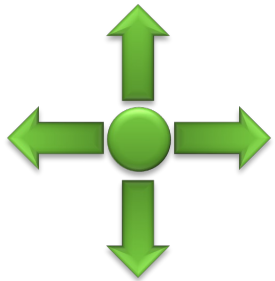


Types of Environment

Discrete:

- ✓ Discrete (as opposed to **continuous**): An environment in which a limited and distinct number of perceptions and actions are clearly defined.
- ✓ In a discrete environment, the set of state states is a discrete set and the states are easily distinguishable.

Example: The world of vacuum cleaner.



– State = {1, 2, …, 8}
– Action = {Left, Right, Suck, NoP}
– Percept = {[Left, Clean], [Left, Dirty], [Right, Clean], …}

What is the example for continuous action?

Types of Environment

Single-agent (versus multi-agent):

✓ A single agent operates alone in the environment.

Example:

- Vacuum cleaner
- Crossword Solver

Multi-agent:

✓ A number of agents that interact with each other.

Example:

- Chess (competitive)
- RoboCup (between members of one team and between members of two teams competing)
- Self-driving Taxi environment (Partial team)

Types of Environment

✓ The type of environment largely determines the design of the agent.

Real World: Partially observable, Stochastic, Dynamic, Continuous and Multi-agent

	Chess with time	Chess without time	Self-driving Taxi
Fully observable	Yes	Yes	No
Deterministic	Strategic	Strategic	No
Episodic	No	No	No
Static	Partially Dynamic	Yes	No
Discrete	Yes	Yes	No
Single agent	No	No	No

Agents functions and programs

Key points:

- ✓ An agent is determined entirely by the agent's function.

Note: The agent function maps the perceptual sequence to action.

- ✓ An agent function (or a small equivalence class) is logical (rational).

Objective:

- ✓ To find a way to implement the logical fitness function in a concise and useful way

An agent based on the search tree

- ✓ A method to describe the operating function.
- ✓ Indicates the appropriate activity for each possible perceptual sequence.

Percept Sequence	**Action**
[A, Clean]	Right
[A, Dirty]	Suck
[B, Clean]	Left
[B, Dirty]	Suck
[A, Clean], [A, Clean]	Right
[A, Clean], [A, Dirty]	Suck
…	

Search table-based agent's program

function TABLE-DRIVEN-AGENT (percept) **returns** an action

static: a sequence of percepts: initially empty table,

a table of actions indexed by percept sequence: initially fully specified

append percept to the end of percepts

action ← LOOKUP (percepts, table)

return action

Search table-based agent

Disadvantages:

- Very large table (for example in chess 1,000,000 rows)
- Every time we need to search from each state
- Too much time to create and fill the table and high probability of error
- Lack of autonomy

Depth	Nodes	Time	Memory
2	1100	.11 seconds	1 megabyte
4	111,100	11 seconds	106 megabytes
6	10^7	19 minutes	10 gigabytes
8	10^9	31 hours	1 terabytes
10	10^{11}	129 days	101 terabytes
12	10^{13}	35 years	10 petabytes
14	10^{15}	3,523 years	1 exabyte

Figure 3.11 Time and memory requirements for breadth-first search. The numbers shown assume branching factor $b = 10$; 10,000 nodes/second; 1000 bytes/node.

Different types of agents

The four main popular types are:

1) Simple reflex agent
2) Model-based reflex agent
3) Goal-based agent
4) Utility-based agent

1) Simple reflex agent

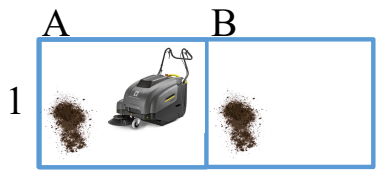
✓ The simplest type of agent

✓ At any given moment, the action is selected solely based on current perception.

Example:

Function REFLEX-VACCUM-AGENT([location, status]) **returns** an action

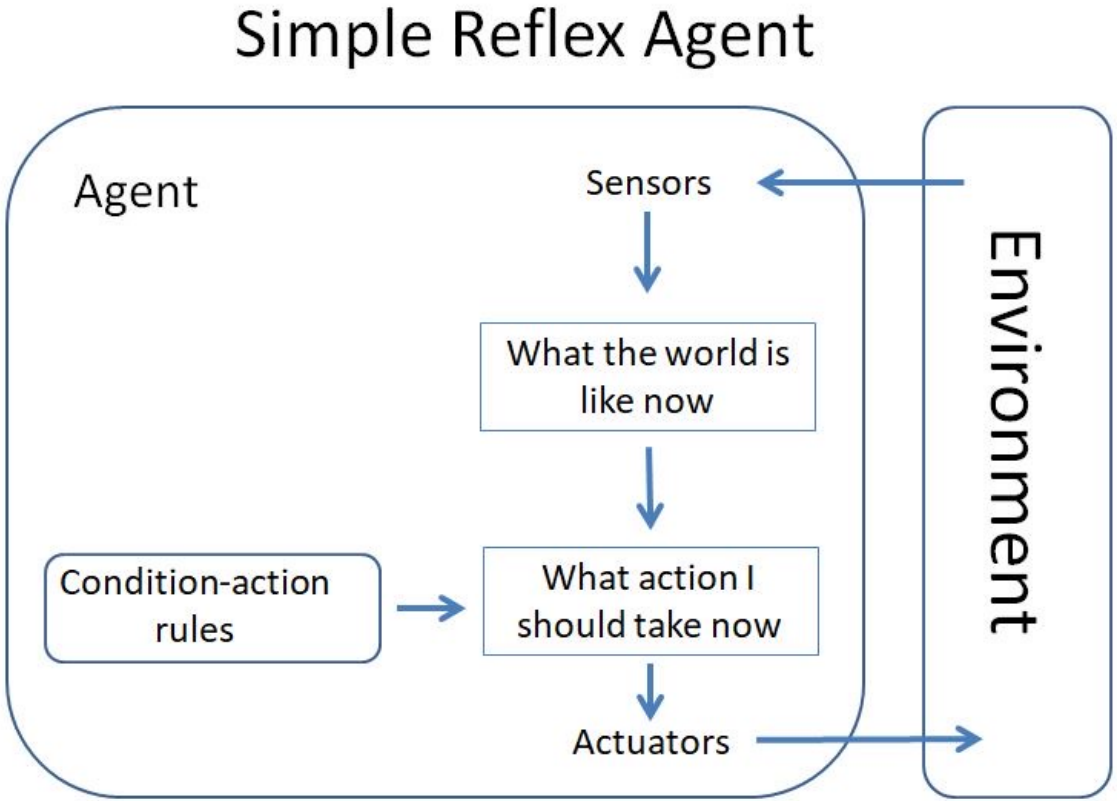
if status =Dirty **then return** Suck
else if location =A **then return** Right
else if location =B **then return** Left



S_i

❑ Includes condition-action rules such as:
"If the front car brake light comes on, then brake"

1) Simple reflex agent



1) Simple reflex agent

function SIMPLE-REFLEX-AGENT(percept) **returns** an action

static: rules, a set of condition-action rules

state ← INTERPRET-INPUT (percept)

rule ← RULE-MATCH (state, rules)

action ← RULE-ACTION [rule]

return action

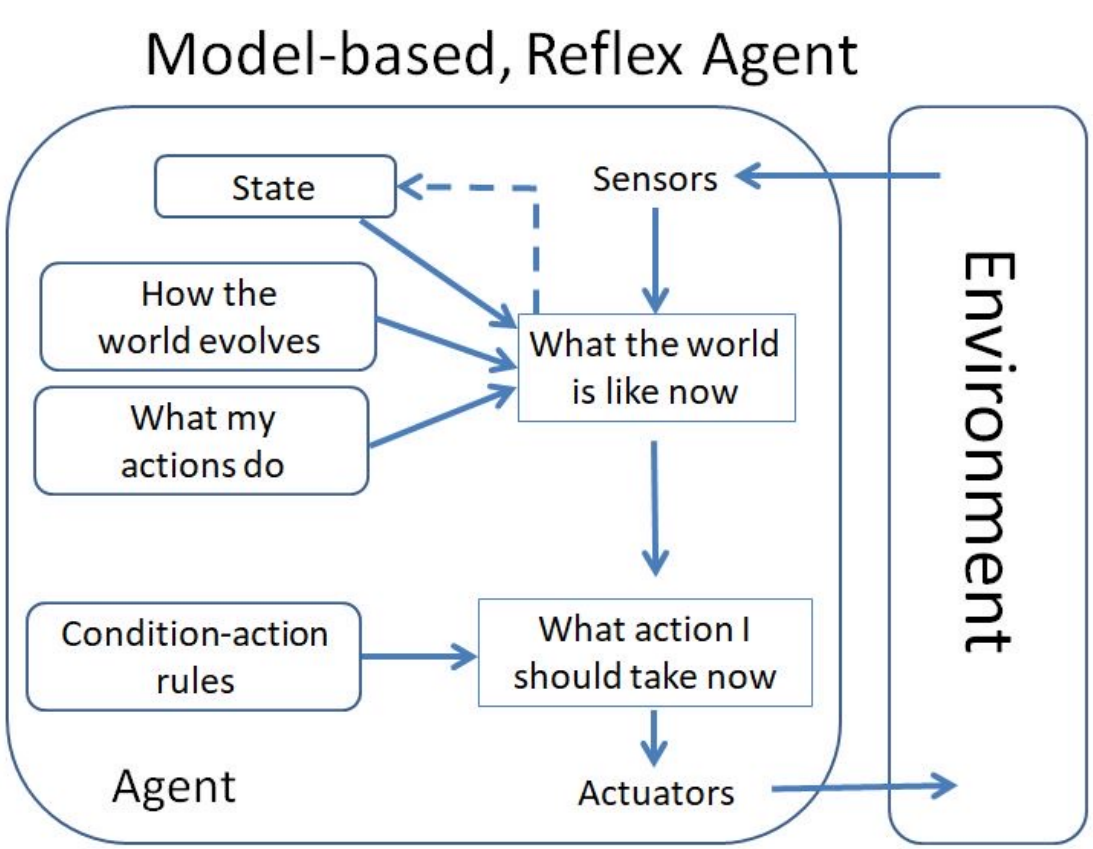
2) Model-based reflex agent

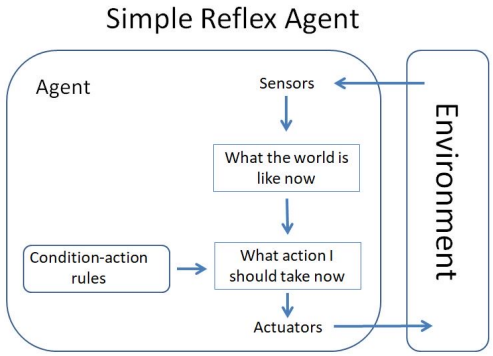
- ✓ A simple reflex agent only works if the environment is fully observable.
- ✓ So if the environment is partially observable, it is necessary to follow the changes in the world.

Example: Self-driving Taxi:

- Requires two types of knowledge
  - How to change the world
  - The effect of acting on the world

2) Model-based reflex agent





2) Model-based reflex agent

function REFLEX-AGENT-WITH-STATE(percept) **returns** an action

static: state, a description of the current world state

rules, a set of condition-action rules

action, the most recent action, initially none

state ← UPDATE-STATE (state, action, percept)

rule ← RULE-MATCH (state, rules)

action ← RULE-ACTION [rule]

return action

3) Goal-based agent

Information needed to decide what to do:

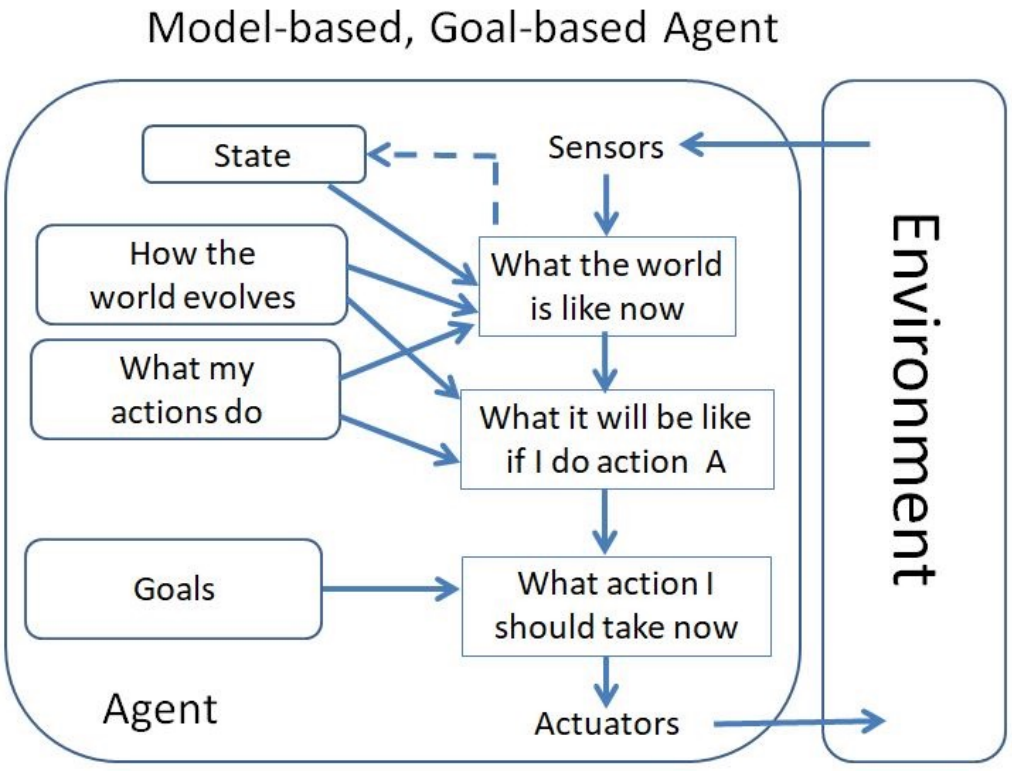
- Information related to the current situation
- Target information (description of the desired situation)

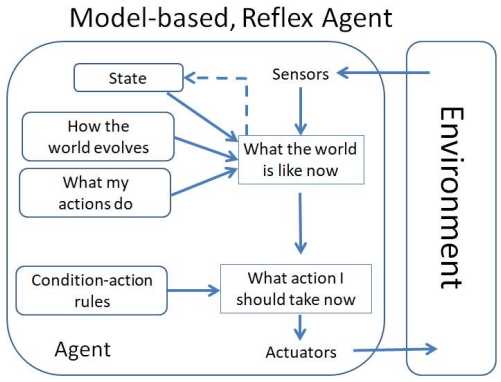
Example: What is the appropriate action for an Self-driving Taxi at a crossroads? (top down view)

(up, down, left, right)

- If several actions are required to reach the goal
  - ✓ Search
  - ✓ Planning

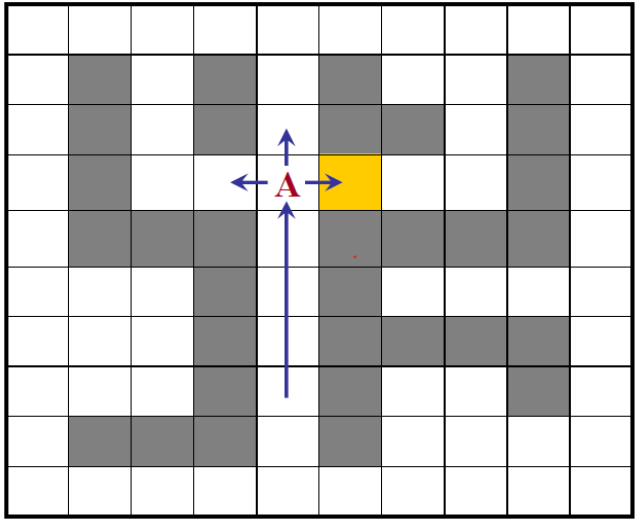
3) Goal-based agent

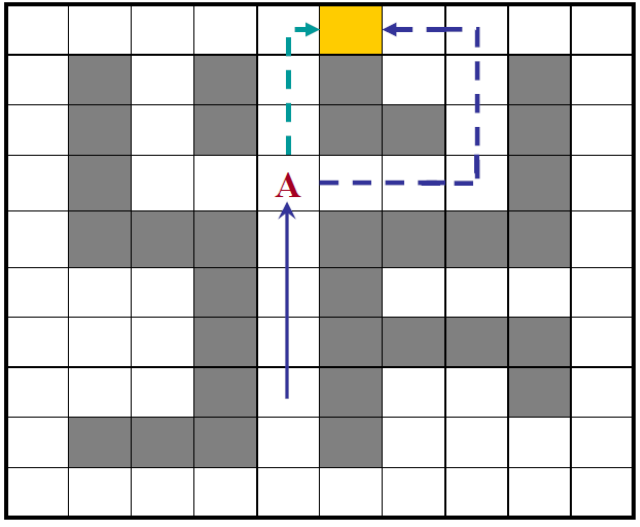




3) Goal-based agent

[UP, UP, UP, RIGHT]
[RIGHT, RIGHT, RIGHT, UP, UP, UP, LEFT, LEFT]





4) Utility-based agent

✓ In many environments, “goals only” are not appropriate for producing high quality behavior

Example: Self-driving Taxi

- ➢ There may be several ways to get to your destination, but **some are faster, safer, or cheaper** than others.

✓ Objectives are to describe situations (desired and undesired)

Reward function:

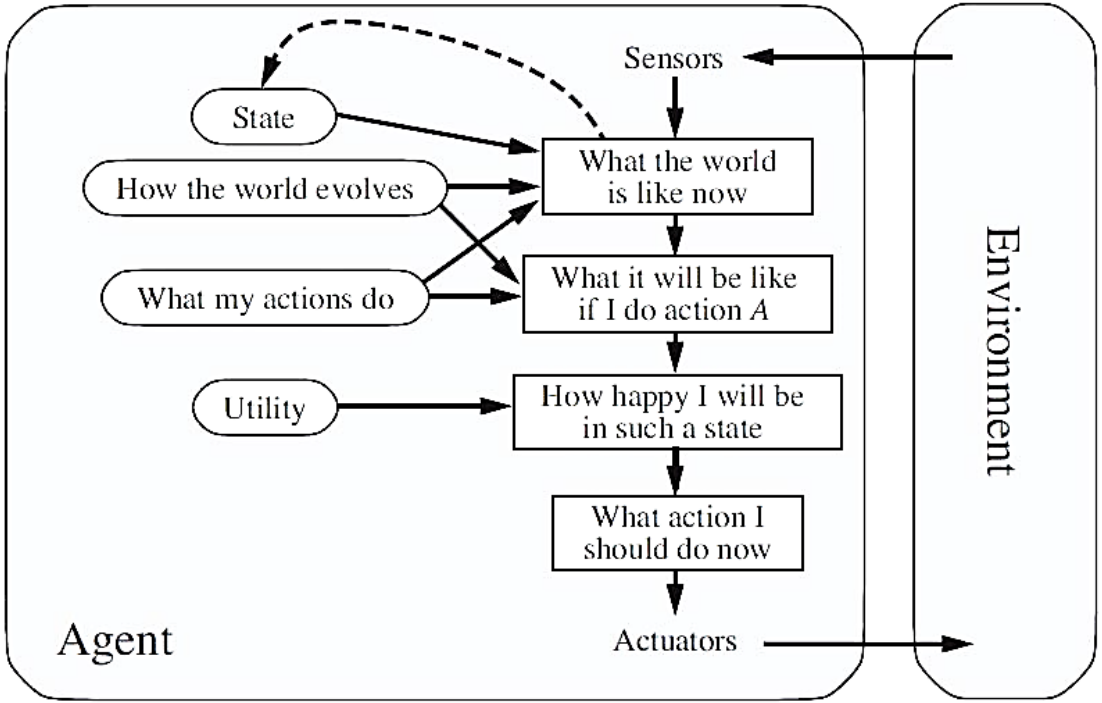
- ✓ Maps a state (or sequence of states) to a real number that describes its degree of usefulness.

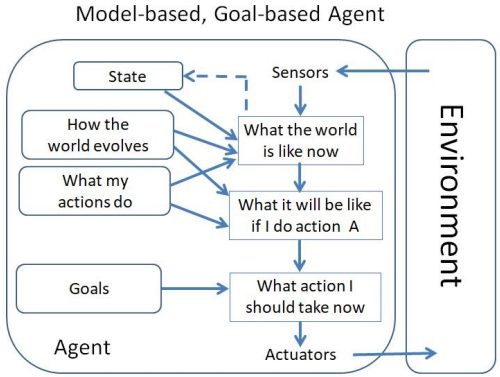
Ability to decide in cases where:

- ➢ The goals are conflicting
- ➢ There are several goals, but achieving any of them is not certain

4) Utility-based agent

- ✓ A **utility-based agent** is an **agent** that acts **based** not only on what the goal is, but the best way to reach that goal





Learning agent

Learning factor components

- Learning element: to make improvement
- Efficiency element (Reward): selection of external activities
- Critic: Generate feedback according to the efficiency standard for the learning element
- Problem generator: Proposing exploration activities

Learning agent

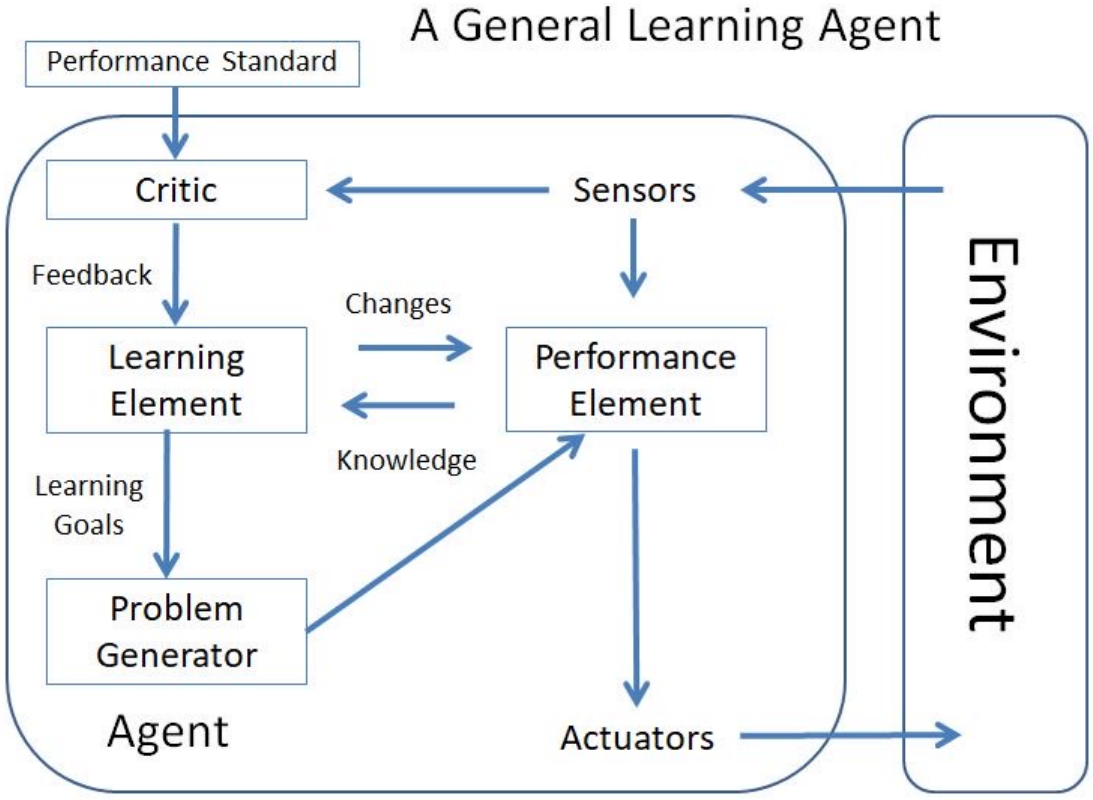
Types of knowledge that the learner agent can learn:

- Direct learning from perceptual sequence
- Learn how to change the world: See two consecutive states
- Learning about the effect of agent action: Observing the results of agent actions

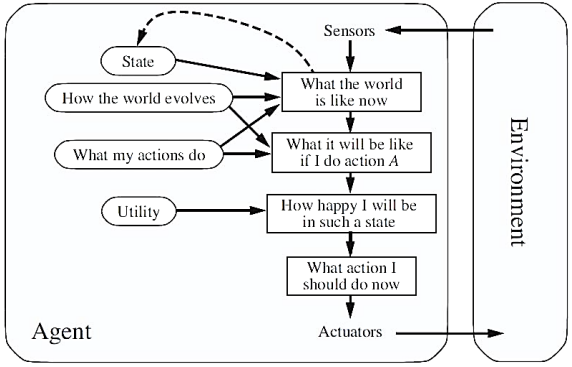
Example: How to brake on a wet roads?

✓ Rewards and punishment

Learning agent



Utility-based agent



Summery

- ✓ We introduced agents and environment.
- ✓ We discussed different types of environments,
  - Fully observable, etc..
- ✓ We discussed different types of agents
- ✓ Finally **introduced learning agent**