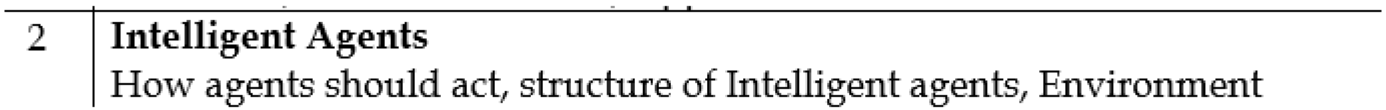
UNIT2
INTELLIGENT AGENTS

Ami Munshi

Syllabus and reference



2	**Intelligent Agents** How agents should act, structure of Intelligent agents, Environment

Reference:

Stuart Russel and Peter Norvig, *Artificial Intelligence: A Modern Approach*, 4th edition, Pearson Education, 2021.

Some terms

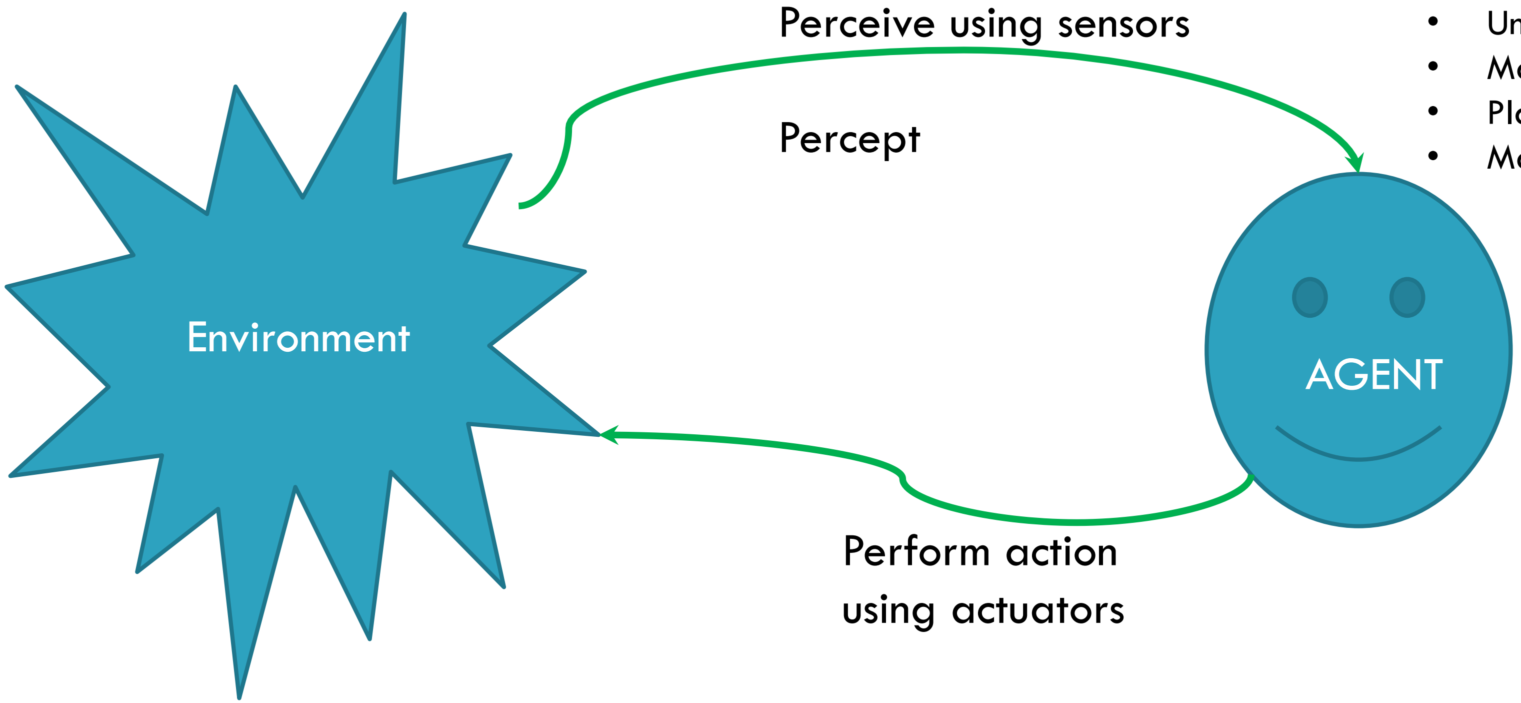
- Data
  - Facts and statistics collected for references or analysis
- Information
  - Data that is processed becomes information
- Knowledge
  - Facts, contains information, skills acquired through experience or education
- Intelligence
  - Ability to apply the acquired knowledge
- Representation:
  - Description or portrayal of someone or something in a particular way
  - Example: “White” for colour white
  - Example ✚ for doctor or medial help
- Reasoning:
  - Thinking about something in a logical and sensible way
  - Use of symbolic representations to derive new conclusions

Human Intelligence

- Intelligent behaviour is based on knowledge
- Decision made by human being depends upon what he knows

Intelligent Agent

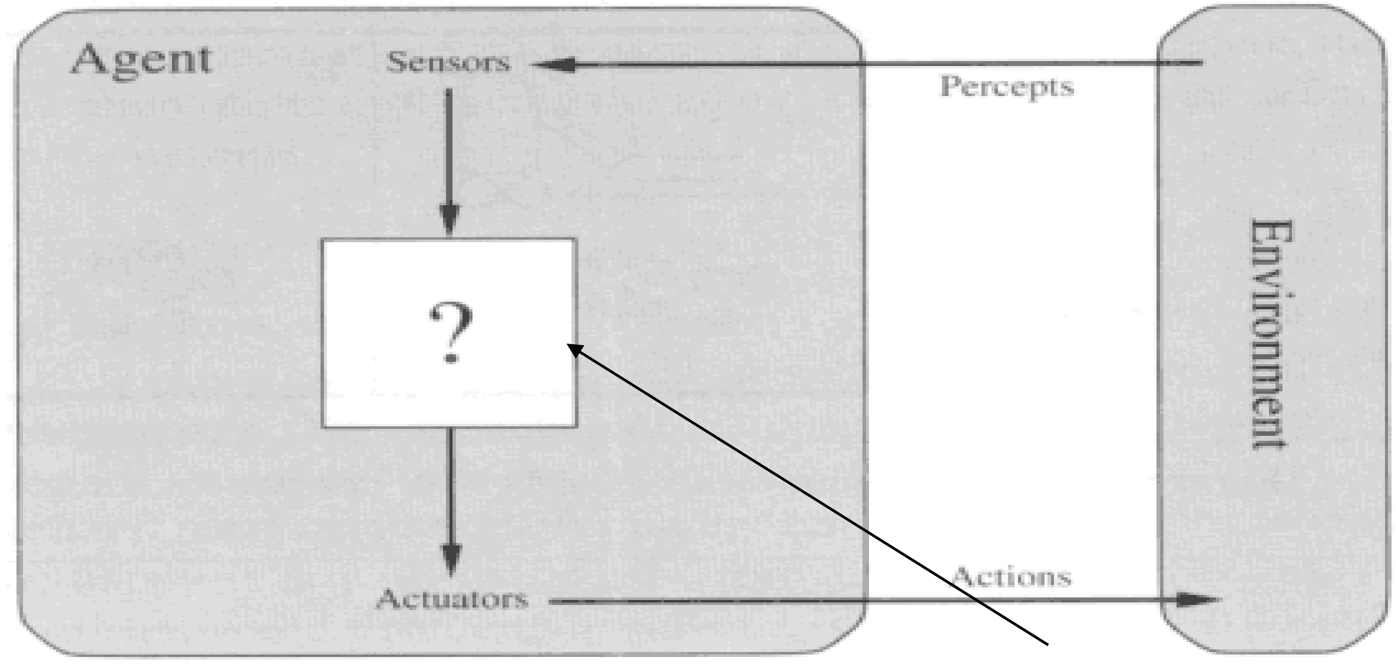
What is an agent?



- Understand
- Make reasons
- Plan
- Make Decision

Agent’s choice of action at any given instant can depend on the entire precept sequence observed to date

Diagram of an agent



What AI should fill

Stuart Russel and Peter Norvig, *Artificial Intelligence: A Modern Approach*, 4th edition, Pearson Education, 2021.

Agent

- An *agent*
  - It perceives the *environment* using *sensors*
  - It understands, thinks, makes reasons, makes plan, makes decisions
  - It performs action through *actuators*

- *Percept*- Agents perceptual input at any given instant
- *Percept sequence*- Complete history of anything that the agent has ever perceived
- *An agent's choice of action at any given instant can depend on the entire percept sequence observed to date, but not on anything it hasn't perceived*

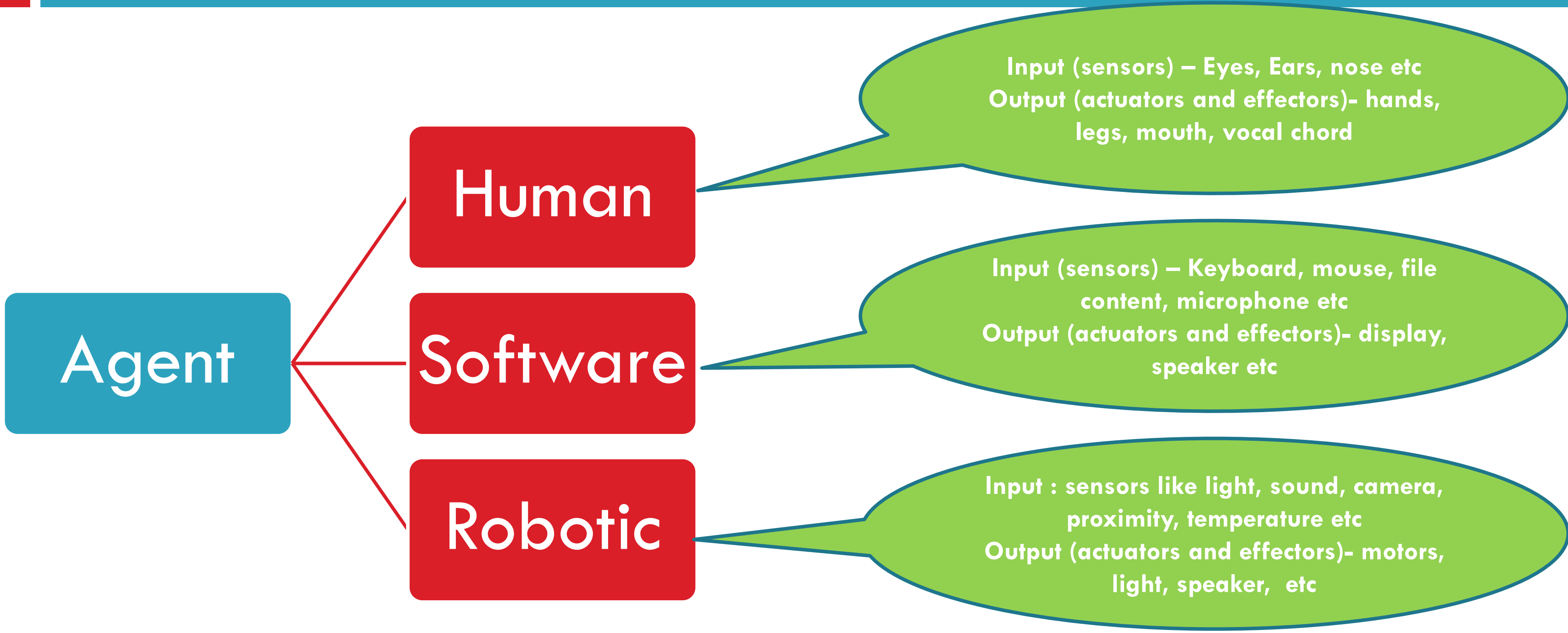
Agent

Perceiving
Agent
Acting
Thinking

Percept- Agents perceptual input at any given instant
Percept sequence- Complete history of anything that the agent has ever perceived
An agent's choice of action at any given instant can depend on the entire percept sequence observed to date, but not on anything it hasn't perceived

Agent Function- Maps any precept sequence to an action. It is an abstract mathematical description
Agent Program- Concrete implementation of agent function, running within some physical system

Agent Examples

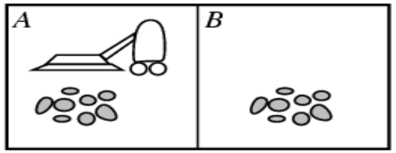


Agent
Human
Software
Robotic
Input (sensors) – Eyes, Ears, nose etc
Output (actuators and effectors)- hands, legs, mouth, vocal chord
Input (sensors) – Keyboard, mouse, file content, microphone etc
Output (actuators and effectors)- display, speaker etc
Input : sensors like light, sound, camera, proximity, temperature etc
Output (actuators and effectors)- motors, light, speaker, etc

Example: Vacuum-cleaner world

- Percepts: location and contents, e.g., [A,Dirty]
- Actions: *Left, Right, Suck, No operation*
- *Agent's function → look-up table*
  - *For many agents this is a very large table*

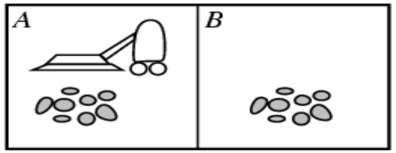
Vacuum cleaner world with just two locations



Example: Vacuum-cleaner world

- One simple agent function could be-
  - If square is dirty then
    - Suck
  - Else
    - Move to other square

Vacuum cleaner world with just two locations



Percept sequence	Action
$[A, Clean]$	$Right$
$[A, Dirty]$	$Suck$
$[B, Clean]$	$Left$
$[B, Dirty]$	$Suck$
$[A, Clean], [A, Clean]$	$Right$
$[A, Clean], [A, Dirty]$	$Suck$
⋮	⋮

Consider an example

- If it is proposed to measure performance by the amount of dirt cleaned up in a single eight-hour shift then
- A rational agent will try to maximize this performance measure by
  - cleaning up the dirt
  - then dumping it all on the floor
  - then cleaning it up again
  - and so on.
- Rather more suitable performance measure would reward the agent for having a clean floor
- For example
  - one point could be awarded for each clean square at each time step
  - And a penalty for electricity consumed and noise generated

Concept of rationality- Rational Agent

- A rational agent is expected
  - To perform rational actions based on its perception (current and past) to perform actions that maximizes the performance measure
  - To perform right thing
  - To make sensible decision
- Rationality of an agent depends on
  - Performance measure
  - Agents prior knowledge about its environment
  - Best action that the agent can perform
  - Percept sequence agent has acquired so far

***Performance measure** evaluates any given sequence of environment states*

It is better to design performance measures according to what one actually wants in the environment, rather than according to how one thinks the agent should behave

Rationality

- What is rational at any given time depends on four things-
  - Performance measure that defines the criterion of success.
  - Agent’s prior knowledge of the environment
  - Actions that the agent can perform
  - Agent’s percept sequence to date.
- Definition of rational agent
  - *For each possible percept sequence, a rational agent should*
    - *Select an action that is expected to maximize its performance measure, given the evidence provided by the percept sequence and whatever built-in knowledge the agent has*

Is vacuum cleaner agent rational??

- Consider the simple vacuum-cleaner agent that cleans a square if it is dirty and moves to the other square if not
- To claim that the agent is rational,
  - First we need to say what is the performance measure
  - what is known about the environment
  - what sensors and actuators the agent has
- Let us assume the following:
  - Performance measure awards one point for each clean square at each time step over a “lifetime” of 1000 time steps.
  - “geography” of the environment is known *a priori*
  - But the dirt distribution and the initial location of the agent are not known
  - Clean squares stay clean and sucking cleans the current square
  - The Left and Right actions move the agent left and right except when this would take the agent outside the environment- In which case the agent remains where it is.
  - Only available actions are Left , Right, and Suck
  - Agent correctly perceives its location and whether that location contains dirt
- *Under these circumstances we cab claim that* the agent is indeed rational

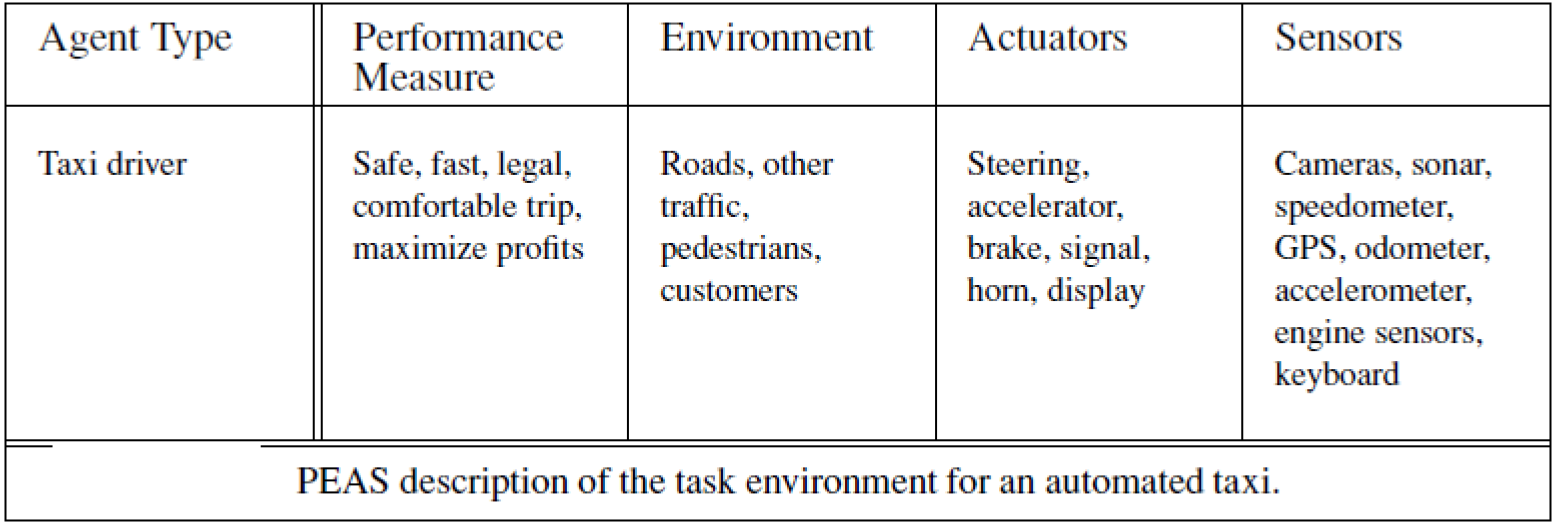
Can vacuum cleaner agent be irrational??

- One can see easily that the same agent would be irrational under different circumstances
- For example, once all the dirt is cleaned up, the agent will oscillate needlessly back
- and forth
- If the performance measure includes a penalty of one point for each movement left or right, the agent will fair poorly
- A better agent for this case would do nothing once it is sure that all the squares are clean
- If clean squares can become dirty again, the agent should occasionally check and re-clean them if needed
- If the geography of the environment is unknown the agent will need to explore it rather than stick to squares A and B

Task environment- PEAS

- Task undertaken by the agent is characterized by
  - P: Performance
    - Measure that defines success criterion
  - E: Environment
    - Agent's priori knowledge of environment
  - A: Actuators
    - Actions that agent can perform
  - S: Sensors
    - Agent's precept sequence to date

Example: Consider a task of designing automated taxi



Agent Type	Performance Measure	Environment	Actuators	Sensors
Taxi driver	Safe, fast, legal, comfortable trip, maximize profits	Roads, other traffic, pedestrians, customers	Steering, accelerator, brake, signal, horn, display	Cameras, sonar, speedometer, GPS, odometer, accelerometer, engine sensors, keyboard

PEAS description of the task environment for an automated taxi.

PEAS Activity- DIY

- Form PEAS table for the following Agent Types-
  - Medical diagnosis system
  - Satellite image analysis system
  - Part picking robot
  - Refinery controller
  - Interactive English Tutor

Task Environment classifications

- Full observable/Partially Observable/ Unobservable
- Single agent/Multi agent
- Competitive/Cooperative
- Deterministic/Stochastic
- Episodic/Sequential
- Static/Dynamic
- Discrete/Continuous

Task Environments-classification

- Fully observable
  - An agent's sensors give it access to thecomplete state of the environment at each point in time
  - Sensors detect all aspects that are *relevant* to the choice of action; relevance, in turn, depends on the performance measure
- Partially observable
  - Could be partially observable because of noisy and inaccurate sensors or because parts of the state are simply missing from the sensor data
    - For example, a vacuum agent with only a local dirt sensor cannot tell whether there is dirt in other squares
    - An automated taxi cannot see what other drivers are thinking
- Unobservable
  - Agent has no sensors at all then the environment

Task Environments-classification

- Single agent
  - Only one agent is involved in an environment, and operating by itself then such an environment is called single agent environment
  - Example- solving crossword puzzle by itself
- Multi agent
  - Multiple agents are operating in an environment, then such an environment is called a multi-agent environment
  - Example- Playing soccer match is a multi agent environment

Task Environments-classification

- Competitive
  - Example –chess
  - Entity B is trying to maximize the performance measure which minimizes the performance of B
- Cooperative
  - Example-
  - Taxi driving environment is partially cooperative- avoids collision and hence maximizes the performance
  - But it is partially competitive- why???
  - For instance like- only one car can occupy the parking space

Task Environments-classification

- Deterministic/Stochastic
  - If the next state of the environment is completely determined by the current state and the action executed by the agent, then we say the environment is deterministic otherwise it is stochastic
  - Taxi driver example is stochastic because one can never predict behaviour of traffic, tyres can blow, engine can fail

Task Environments-classification

- Episodic or sequential
  - In an episodic task environment, the agent's experience is divided into atomic episodes. In each episode the agent receives a percept and then performs a single action
  - Crucially, the next episode does not depend on the actions taken in previous episodes.
  - Example- Many classification tasks are episodic
    - Spotting defective part on an assembly line
  - In sequential environments, on the other hand, the current decision could affect all future decision
  - Example
    - Chess and taxi driving are sequential
    - In both cases, short-term actions can have long-term consequences.
- Episodic environments are much simpler than sequential environments because the agent does not need to think ahead

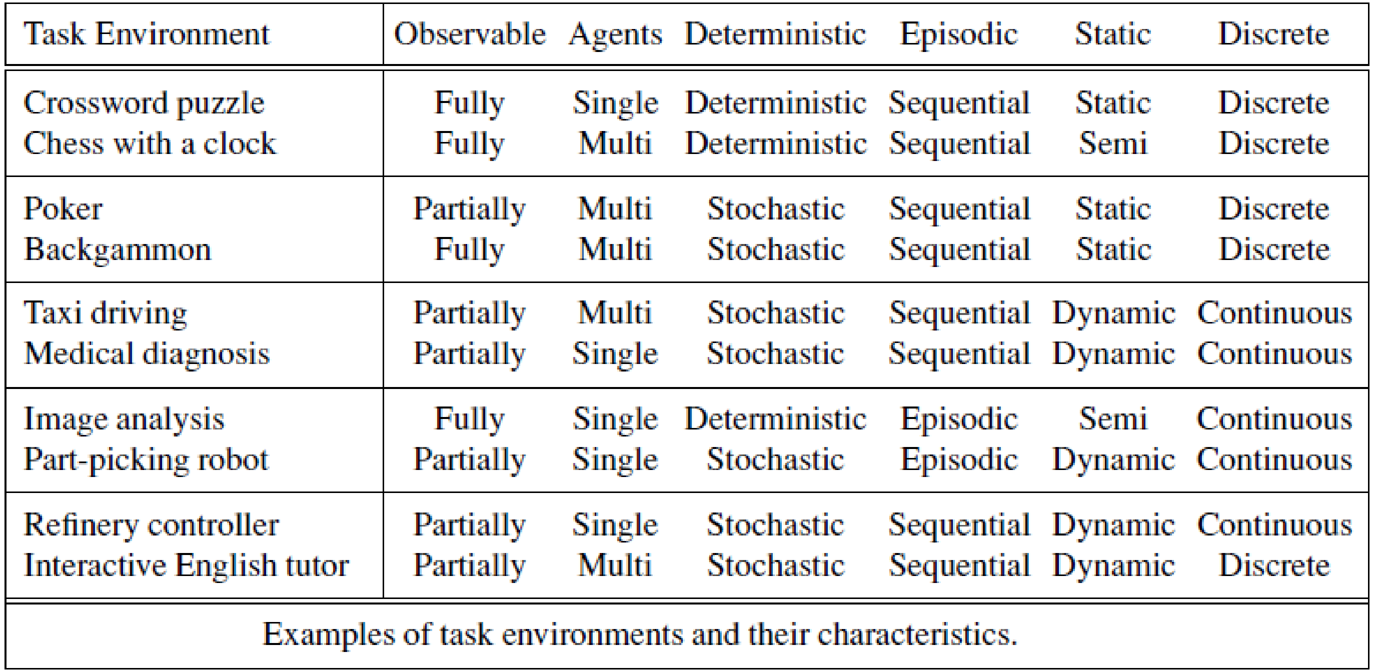
Task Environments-classification

- Static/Dynamic
  - If the environment can change while an agent is deliberating, then we say the environment is dynamic for that agent otherwise it is static.
  - Static environments are easy to deal with because the agent need not keep looking at the world while it is deciding on an action, nor need it worry about the passage of time
  - Dynamic environments are continuously asking the agent what it wants to do. If it hasn’t decided yet, that counts as deciding to do nothing
  - Taxi driving is dynamic
  - Crossword puzzles are static
- Find about semi-dynamic environment

Task Environments-classification

- Discrete/Continuous
  - Discrete/continuous distinction applies to the *state* of the environment, to the way *time* is handled, and to the *percepts* and *actions* of the agent
  - For example, the chess environment has a finite number of distinct states
  - Chess also has a discrete set of percepts and actions
  - Taxi driving is a continuous-state and continuous-time problem: the speed and location of the taxi and of the other vehicles sweep through a range of continuous values and do so smoothly over time
  - Taxi-driving actions are also continuous (steering angles, etc.)

Example of task environment characteristics



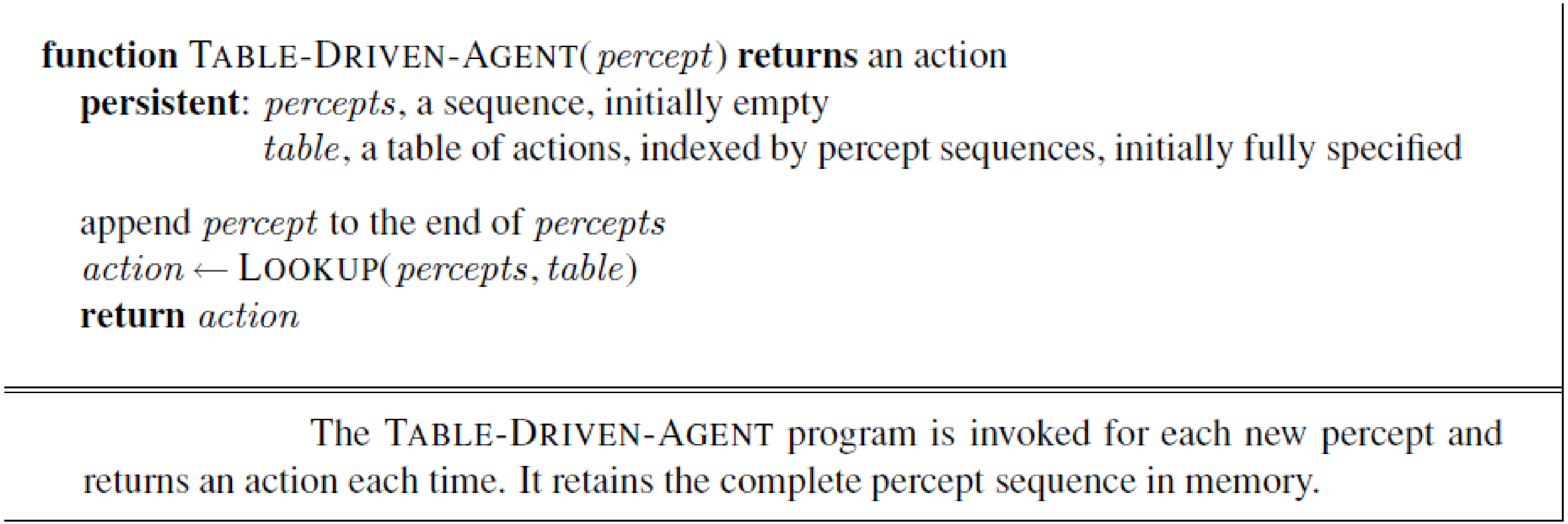
Task Environment	Observable	Agents	Deterministic	Episodic	Static	Discrete
Crossword puzzle	Fully	Single	Deterministic	Sequential	Static	Discrete
Chess with a clock	Fully	Multi	Deterministic	Sequential	Semi	Discrete
Poker	Partially	Multi	Stochastic	Sequential	Static	Discrete
Backgammon	Fully	Multi	Stochastic	Sequential	Static	Discrete
Taxi driving	Partially	Multi	Stochastic	Sequential	Dynamic	Continuous
Medical diagnosis	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Image analysis	Fully	Single	Deterministic	Episodic	Semi	Continuous
Part-picking robot	Partially	Single	Stochastic	Episodic	Dynamic	Continuous
Refinery controller	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Interactive English tutor	Partially	Multi	Stochastic	Sequential	Dynamic	Discrete

Examples of task environments and their characteristics.

Structure of Agent

- Job of AI is to design Agent Program that implements Agent Function
- Agent= Architecture + Program
  - Architecture- A sort of physical computing device with Physical sensors and actuators
    - A program should have appropriate architecture-
      - For example if program recommends walking action then agent architecture should have legs
  - Agent program implements Agent Function

Agent Program



```
function TABLE-DRIVEN-AGENT(percept) returns an action
    persistent: percepts, a sequence, initially empty
                table, a table of actions, indexed by percept sequences, initially fully specified

    append percept to the end of percepts
    action ← LOOKUP(percepts, table)
    return action
```

The TABLE-DRIVEN-AGENT program is invoked for each new percept and returns an action each time. It retains the complete percept sequence in memory.

Stuart Russel and Peter Norvig, *Artificial Intelligence: A Modern Approach*, 4th edition, Pearson Education, 2021.

Limitations of table driven agent approach

- Let P be the set of possible percepts and let T be the lifetime of the agent (the total number of percepts it will receive)
- The lookup table will contain $\sum_{t=1}^{T}|P|^{t}$ entries
- Example
  - Consider the automated taxi: the visual input from a single camera comes in at the rate of roughly 27 megabytes per second (30 frames per second, 640×480 pixels with 24bits of color information)
  - This gives a lookup table with over $10^{250,000,000,000}$ entries for an hour's driving.
- Large size of table means
  - No physical agent will have space to store the table
  - Designer would not have time to create the table
  - No agent could ever learn all the right table entries from its experience

Table Driven Agent approach

- In spite of the above limitations, Table Driven Agent does what we want- implements desired agent function
- It is important to write programs such that rational behaviour of the agent can be obtained from small amount of code rather that large table entry

Types of agent programs

- Simple reflex agents;
- Model-based reflex agents;
- Goal-based agents; and
- Utility-based agents.

Simple reflex Agent

- Simplest kind of agent
- These agents select actions on the basis
  - of the current percept, ignoring the rest of the percept history
- For example, the vacuum agent is a simple reflex agent, because its decision is based
  - only on the current location and
  - on whether that location contains dirt

Simple reflex agent- two state vacuum cleaner

```
function REFLEX-VACUUM-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
```

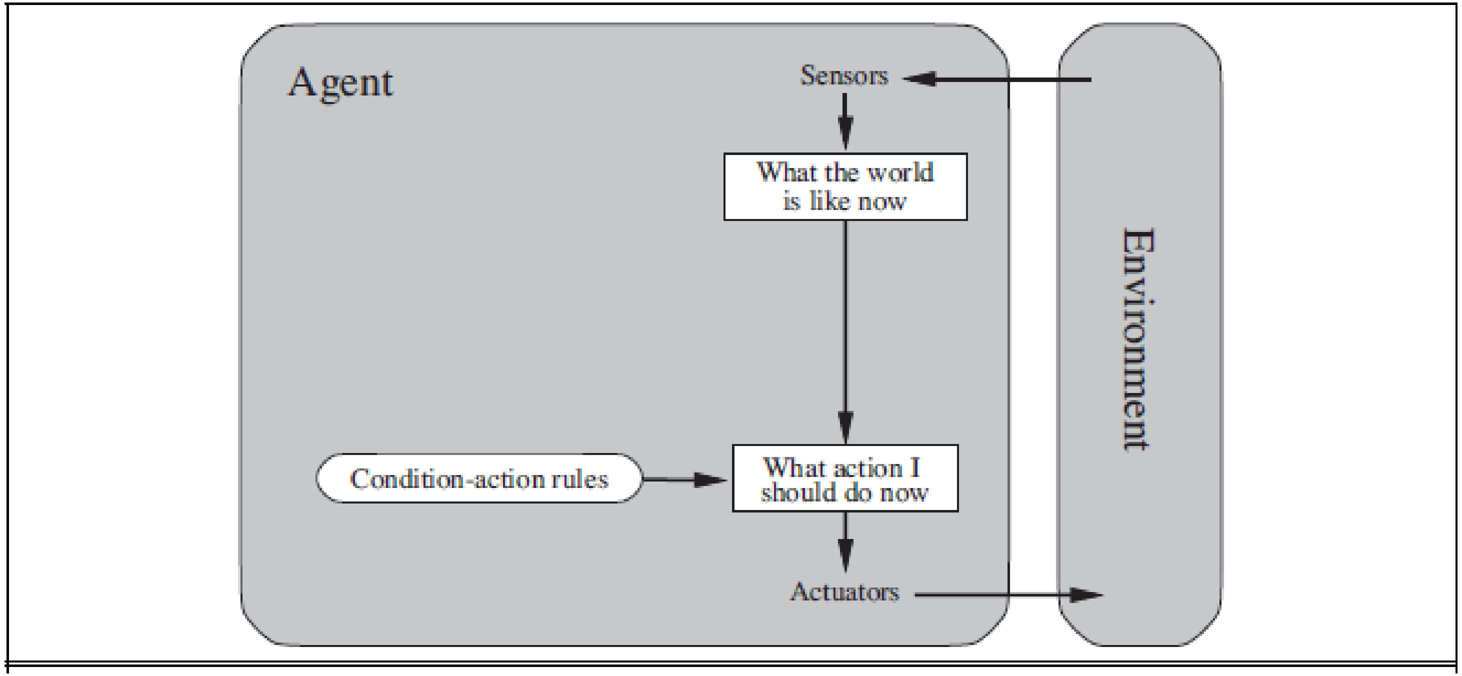
Notice that the vacuum agent program is very small indeed compared to the corresponding table

Obvious reduction comes from ignoring the percept history, which cuts down the number of possibilities

Condition Action Rule

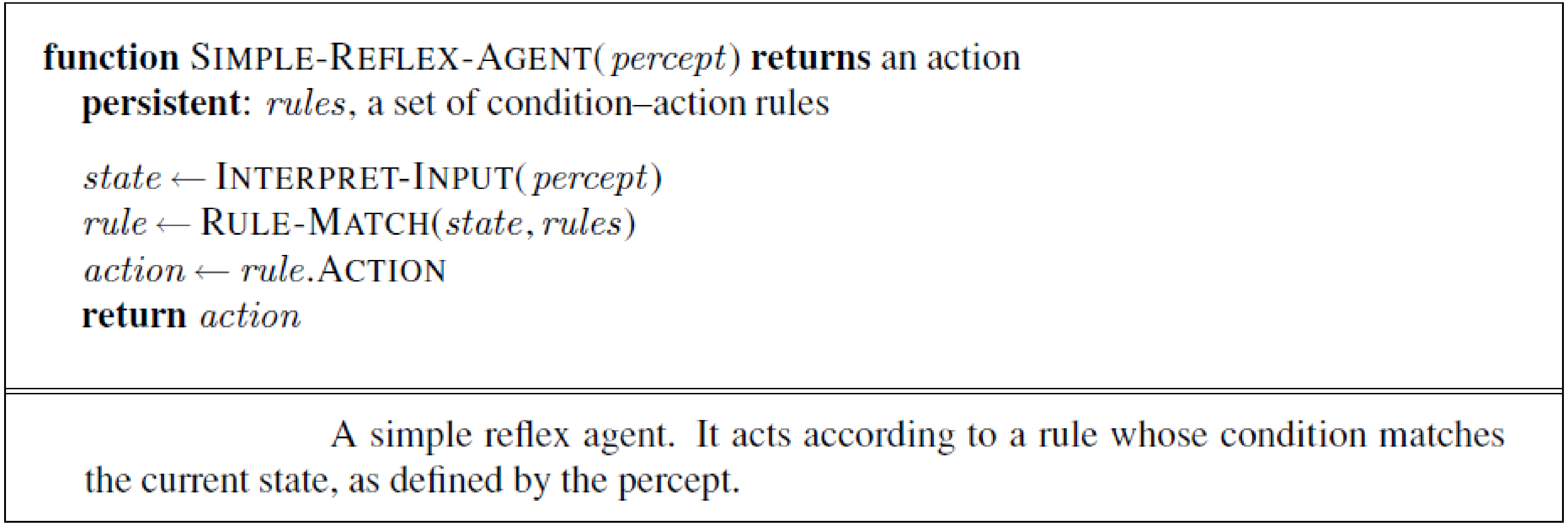
- Can Simple reflex behaviours occur even in more complex environments???- YES
  - Example-Imagine yourself as the driver of the automated taxi
    - If the car in front brakes and its brake lights come on, then
    - you should notice this and initiate braking
- In other words, some processing is done on the visual input to establish the condition
  - we call "The car in front is braking."
  - Then, this triggers some established connection in the agent program to the action "initiate braking"
- We call CONDITION–ACTION such a connection a **condition–action rule**, written as
  - **if** *car-in-front-is-braking* **then** *initiate-braking*
  - Also called if-else rules
- Does human have such rules??
  - Blinking when something approaches the eye

Schematic diagram of simple reflex agent



Stuart Russel and Peter Norvig, *Artificial Intelligence: A Modern Approach*, 4th edition, Pearson Education, 2021.

General program for simple reflex agent



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

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

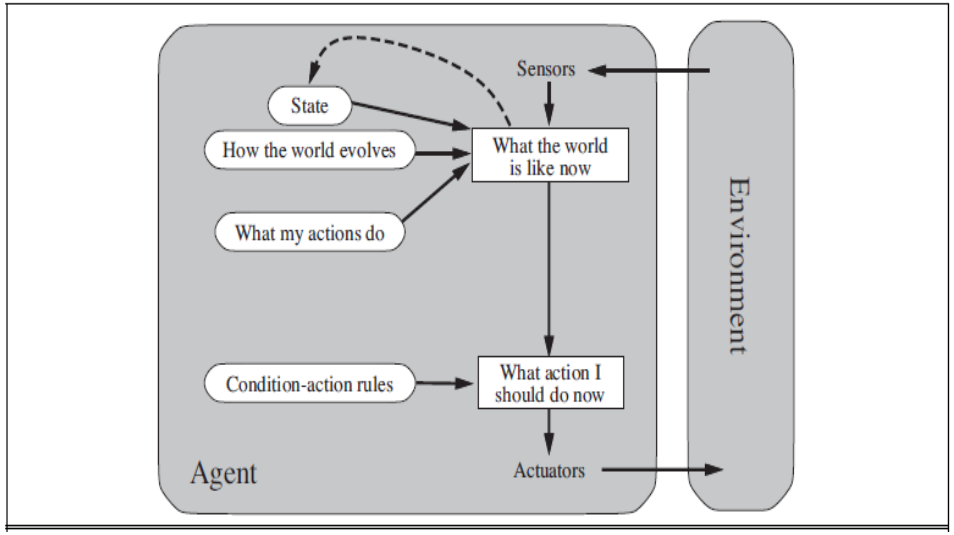
A simple reflex agent. It acts according to a rule whose condition matches the current state, as defined by the percept.

Note: Actual implementation can be done with the help of logic gates

Limitation of single reflex agents

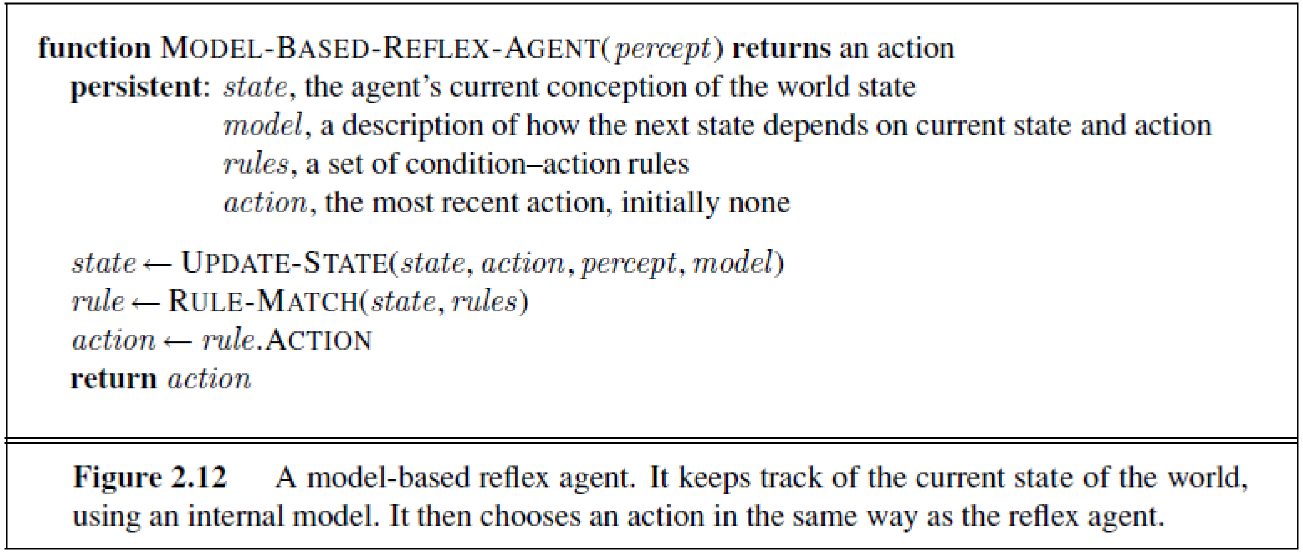
- Simple reflex agents have the admirable property of being simple, but they have limited intelligence
- The agent shown in program above will work *only if the correct decision can be made on the basis of only the current percept—that is, only if the environment is fully observable.*
- Even a little bit of un-observability can cause serious trouble.
- Example 1
  - Braking rule given earlier assumes that the condition *car-in-front-is-braking* can be determined from the current percept—a single frame of video
  - This works if the car in front has a centrally mounted brake light
  - What happens in case of older models where they have different configurations of tail lights, brake lights, and turn-signal lights, and it is not always possible to tell from a single image whether the car is braking
  - A simple reflex agent driving behind such a car would either brake continuously and unnecessarily, or, worse, never brake at all
- Example 2
  - We can see a similar problem arising in the vacuum world
  - Suppose that a simple reflex vacuum agent is deprived of its location sensor and has only a dirt sensor
  - Such an agent has just two possible percepts: [Dirty] and [Clean]
  - It can Suck in response to [Dirty]; what should it do in response to [Clean]?
  - Moving Left fails (forever) if it happens to start in square A, and moving Right fails (forever) if it happens to start in square B
  - Infinite loops are often unavoidable for simple reflex agents operating in partially observable environments
- One possible solution to escape from infinite loop is RANDOMIZATION– Read about this

Model based reflex agent



- Deals with partially observable environment
- It maintains some internal state
- Current percept is combined with internal state
- It updates the internal state based
  - How world evolves
  - What actions do
- Creates model of the world

Model based reflex agent



```
function MODEL-BASED-REFLEX-AGENT(percept) returns an action
  persistent: state, the agent's current conception of the world state
              model, a description of how the next state depends on current state and action
              rules, a set of condition–action rules
              action, the most recent action, initially none

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

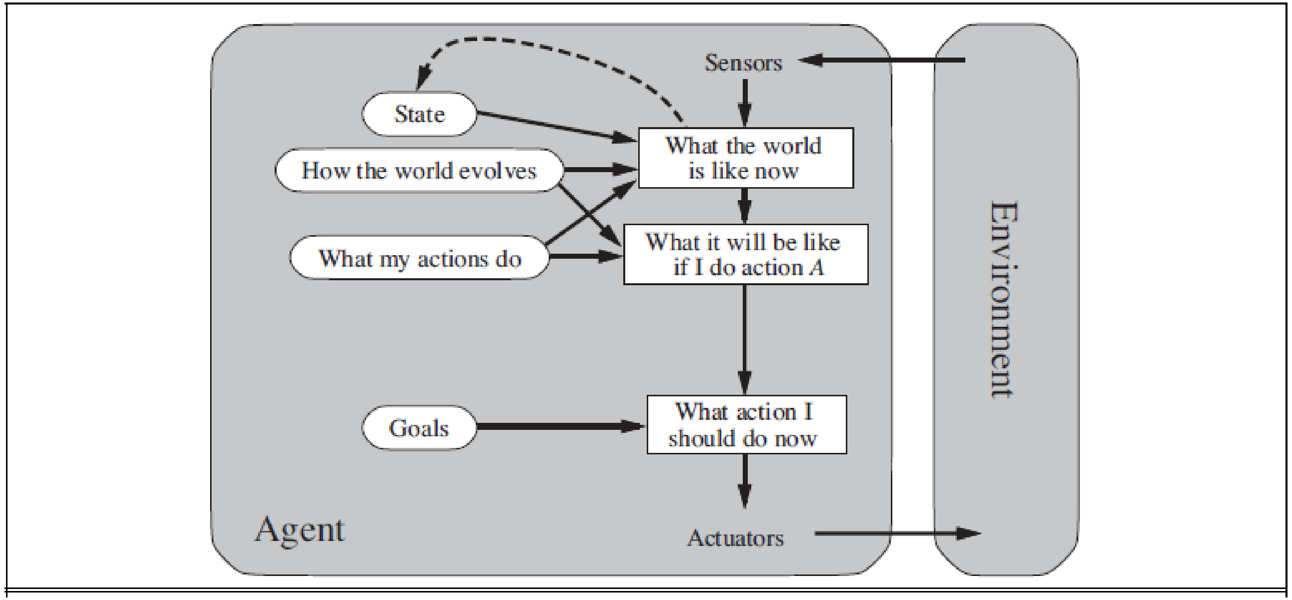
Figure 2.12 A model-based reflex agent. It keeps track of the current state of the world, using an internal model. It then chooses an action in the same way as the reflex agent.

Can you find some examples on model based reflex agents??

Goal based agents

- Knowing something about the current state of the environment and the history---Is it enough to decide what to do???
- For example
  - Taxi at a road junction can turn left, turn right, or go straight on
  - Correct decision depends on where the taxi is trying to get to
- So along with current state description, the agent needs some sort of **goal** information that describes situations that are desirable
- For example
  - being at the passenger’s destination
- Agent program can combine this with the model (the same information as was used in the model based reflex agent) to choose actions that achieve the goal

Goal based agents


Sensors
State
How the world evolves
What my actions do
What the world is like now
What it will be like if I do action A
Goals
What action I should do now
Agent
Actuators
Environment

Stuart Russel and Peter Norvig, *Artificial Intelligence: A Modern Approach*, 4th edition, Pearson Education, 2021.

Goal based agents

- Sometimes goal-based action selection is straightforward
  - For example
    - When goal satisfaction results immediately from a single action
- Sometimes it will be more tricky
  - For example
    - When agent has to consider long sequences of twists and turns in order to find a way to achieve the goal
  - **Search** and **planning** are the subfields of AI devoted to finding action sequences that achieve the agent's goals.

Compare goal based agent and reflex based agents

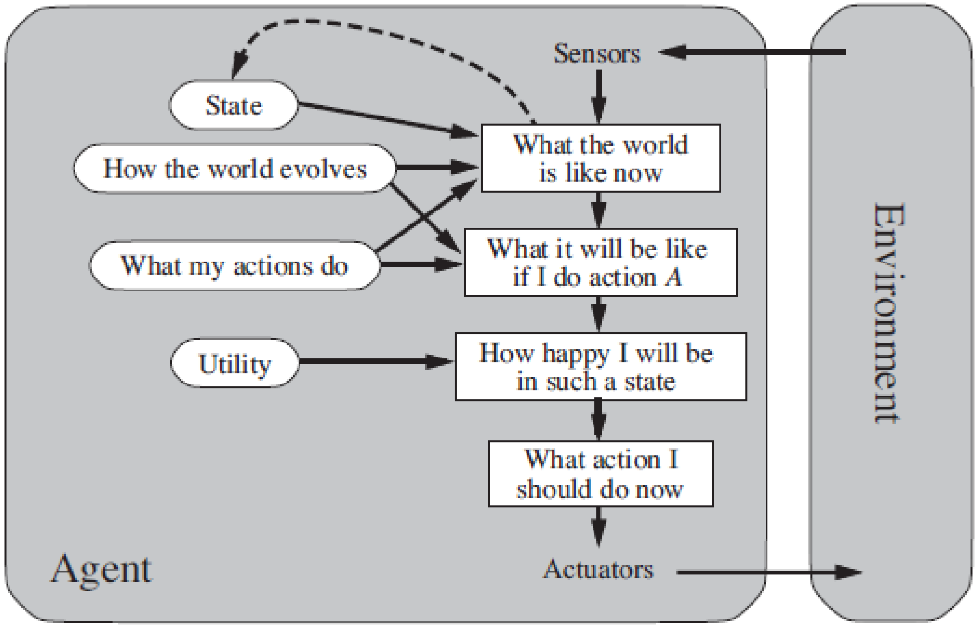
- DIY

Utility based reflex agents

- Are **goals** alone enough to generate high-quality behavior in most environments??
  - For example
    - Many action sequences will get the taxi to its destination (thereby achieving the goal)
    - But some are quicker, safer, more reliable, or cheaper than others
- Goals just provide a crude binary distinction between “happy” and “unhappy” states
- A more general performance measure should allow a comparison of different world states according to exactly how happy they would make the agent
- Because “happy” does not sound very scientific, economists and computer scientists use the term **utility** instead

Note: The word “utility” here refers to “the quality of being useful,” not to the electric company or waterworks

Utility based reflex agents


Sensors
State
How the world evolves
What my actions do
Utility
What the world is like now
What it will be like if I do action A
How happy I will be in such a state
What action I should do now
Actuators
Agent
Environment

Utility function

- We know that performance measure assigns a score to any given sequence of environment states
- So it can easily distinguish between more and less desirable ways of
- getting to the taxi’s destination
- An agent’s **utility function** is essentially an *internalization of the performance measure*
- If the internal utility function and the external performance measure are in agreement
- Then an agent that chooses actions to maximize its utility will be rational according to the external performance measure.

Utility based reflex agents

- In two kinds of cases, goals are inadequate but a utility-based agent can still make rational decisions
- First
  - when there are conflicting goals, only some of which can be achieved (for example, speed and safety)
  - utility function specifies the appropriate tradeoff
- Second
  - when there are several goals that the agent can aim for
  - none of which can be achieved with certainty
  - utility provides a way in which the likelihood of success can be weighed against the importance of the goals
- Partial observability and stochasticity are ubiquitous in the real world, therefore, is decision making under uncertainty
- Technically speaking, a rational utility-based agent chooses the action that maximizes the **expected utility** of the action outcomes
- That is, the utility the agent expects to derive, on average, given the probabilities and utilities of each outcome

Utility based reflex agents

- Utility-based agent
  - Has to model the environment
  - Keep track of its environment
- These tasks have involved a great deal of research on
  - perception, representation, reasoning, and learning
- Choosing the utility-maximizing course of action is also a difficult task, requiring ingenious algorithms

Summary

- An **agent** is something that perceives and acts in an environment
- The **agent function** for an agent specifies the action taken by the agent in response to any percept sequence
- The **performance measure** evaluates the behavior of the agent in an environment
- A **rational agent** acts so as to maximize the expected value of the performance measure given the percept sequence it has seen so far
- A **task environment** specification includes the performance measure, the external environment, the actuators, and the sensors
- In designing an agent, the first step must always be to specify the task environment as fully as possible
- Task environments vary along several significant dimensions
- They can be fully or partially observable, single-agent or multiagent, deterministic or stochastic, episodic or sequential, static or dynamic, discrete or continuous, and known or unknown.

Summary

- The **agent program** implements the agent function
- There exists a variety of basic agent-program designs reflecting the kind of information made explicit and used in the decision process
- The designs vary in efficiency, compactness, and flexibility
- The appropriate design of the agent program depends on the nature of the environment
- **Simple reflex agents** respond directly to percepts
- **Model-based reflex agents** maintain internal state to track aspects of the world that are not evident in the current percept
- **Goal-based agents** act to achieve their goals
- **Utility-based agents** try to maximize their own expected "happiness"
- All agents can improve their performance through **learning.**