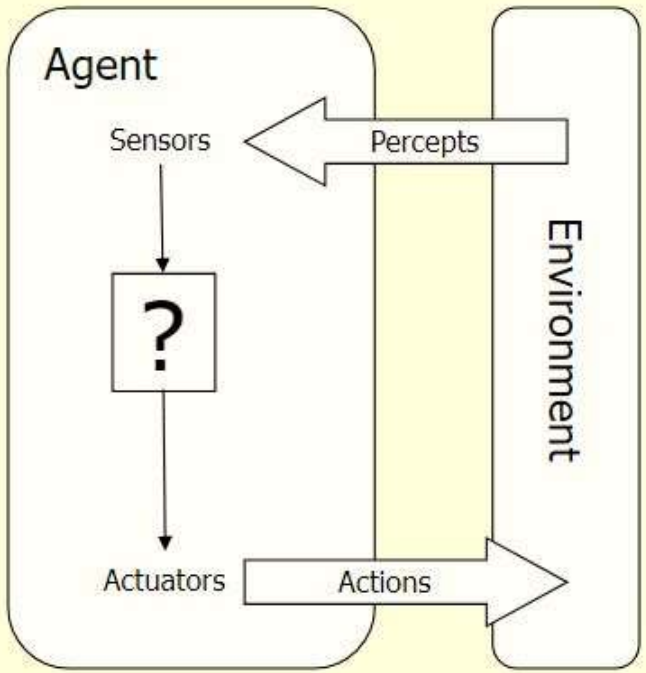
Artificial Intelligence
Chapter 2: Intelligent Agents

Agents and Environments

- An **Agent** is anything that can be viewed as perceiving its **environment** through **sensors** and acting upon that environment through **actuators**

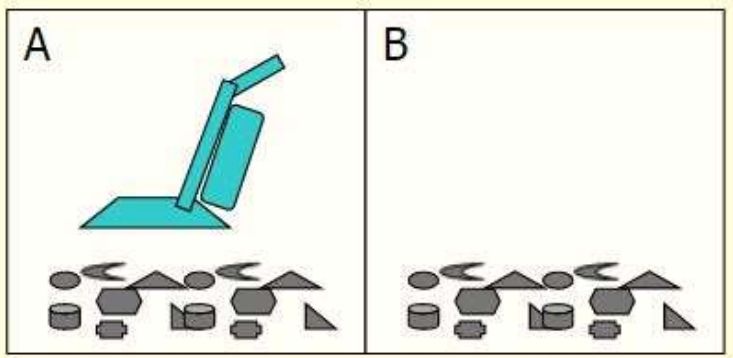


Agents and Environments

- Percept – the agent's perceptual inputs
  - percept sequence is a sequence of everything the agent has ever perceived
- Agent Function – describes the agent's behavior
  - Maps any given percept sequence to an action
  - f : P* -> A
- Agent Program – an implementation of an agent function for an artificial agent

Agents and Environments

- Example: Vacuum Cleaner World
  - Two locations: squares A and B
  - Perceives what square it is in
  - Perceives if there is dirt in the current square
  - Actions
    - move left
    - move right
    - suck up the dirt
    - do nothing



Agents and Environments

- Agent Function: Vacuum Cleaner World
  - If the current square is dirty, then suck, otherwise move to the other square

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

Agents and Environments

- But what is the right way to fill out the table?
  - is the agent
    - good or bad
    - intelligent or stupid
  - can it be implemented in a small program?

```
Function Reflex-Vacuum-Agent([location, status]) return an action
    if status == Dirty then return Suck
    else if location = A then return Right
    else if location = B then return Left
```

Good Behavior and Rationality

- Rational Agent – an agent that does the "right" thing
  - Every entry in the table for the agent function is filled out correctly
  - Doing the right thing is better than doing the wrong thing
    - What does it mean to do the right thing?

Good Behavior and Rationality

- Performance Measure
  - A scoring function for evaluating the environment space

- ***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 what ever built-in knowledge the agent has.

Good Behavior and Rationality

- Rational != omniscient
- Rational != clairvoyant
- Rational != successful

- Rational -> exploration, learning, autonomy

The Nature of Environments

- Task environments
  - The "problems" to which a rational agent is the "solution"

- PEAS
  - Performance
  - Environment
  - Actuators
  - Sensors

The Nature of Environments

- Properties of task environments
  - Fully Observable vs. Partially Observable
  - Deterministic vs. Stochastic
  - Episodic vs. Sequential
  - Static vs. Dynamic
  - Discrete vs. Continuous
  - Single agent vs. Multi-agent

- The real world is partially observable, stochastic, sequential, dynamic, continuous, multi-agent

The Nature of Environments

- Examples
  - Solitaire
  - Backgammon
  - Automated Taxi
  - Mars Rover

The Structure of Agents

- Agent = Architecture + Program
- Basic algorithm for a rational agent
  - While (true) do
    - Get percept from sensors into memory
    - Determine best action based on memory
    - Record action in memory
    - Perform action
- Most AI programs are a variation of this theme

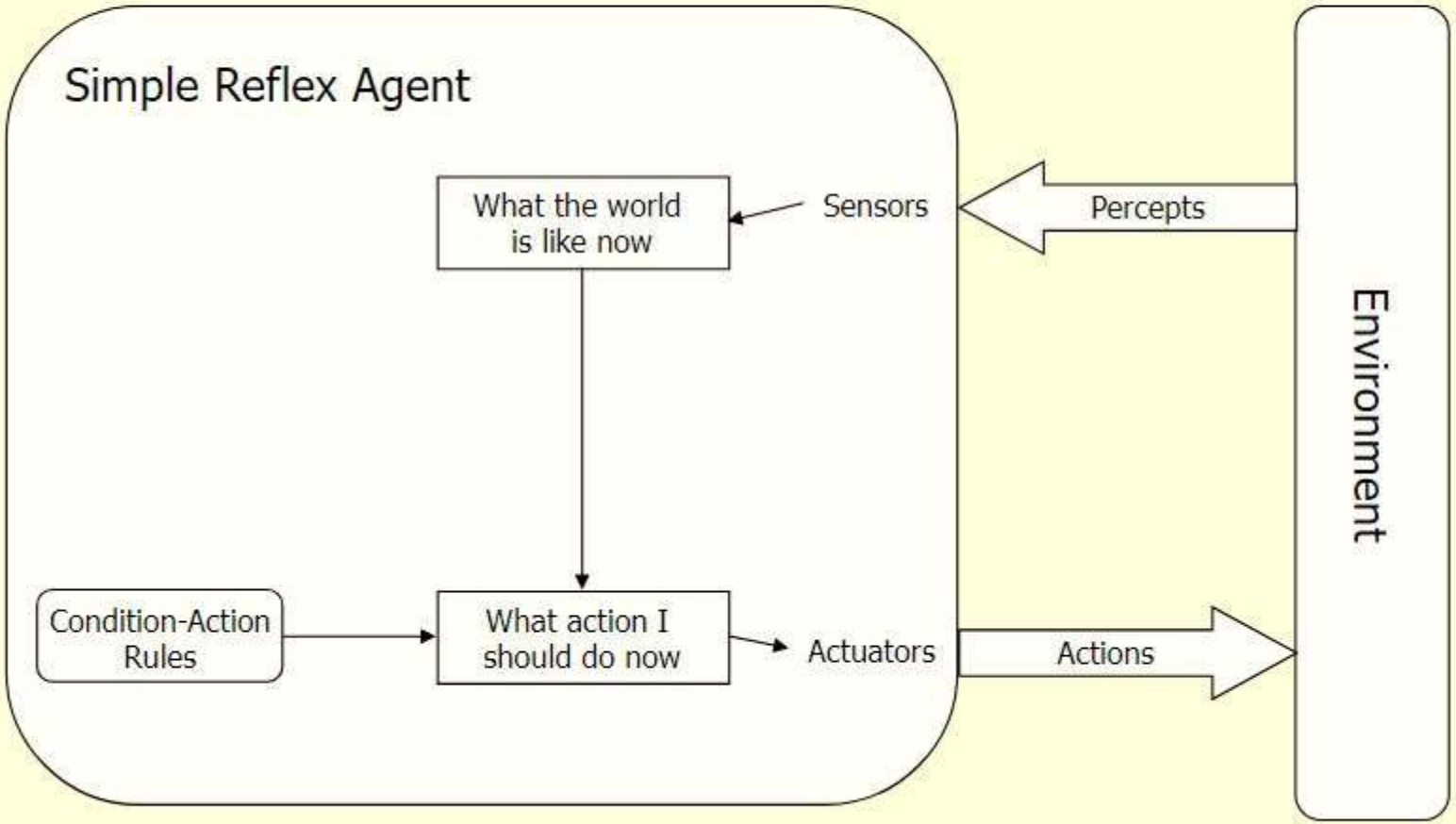
The Structure of Agents

- Table Driven Agent

```
function Table-Driven-Agent (percept) return action
static:     percepts, a sequence, initially empty
            table, a table of actions, indexed by
            percept sequences, initially fully
            specified

  append percept to the end of the table
  action <- LOOKUP( percept, table )
  return action
```

The Structure of Agents



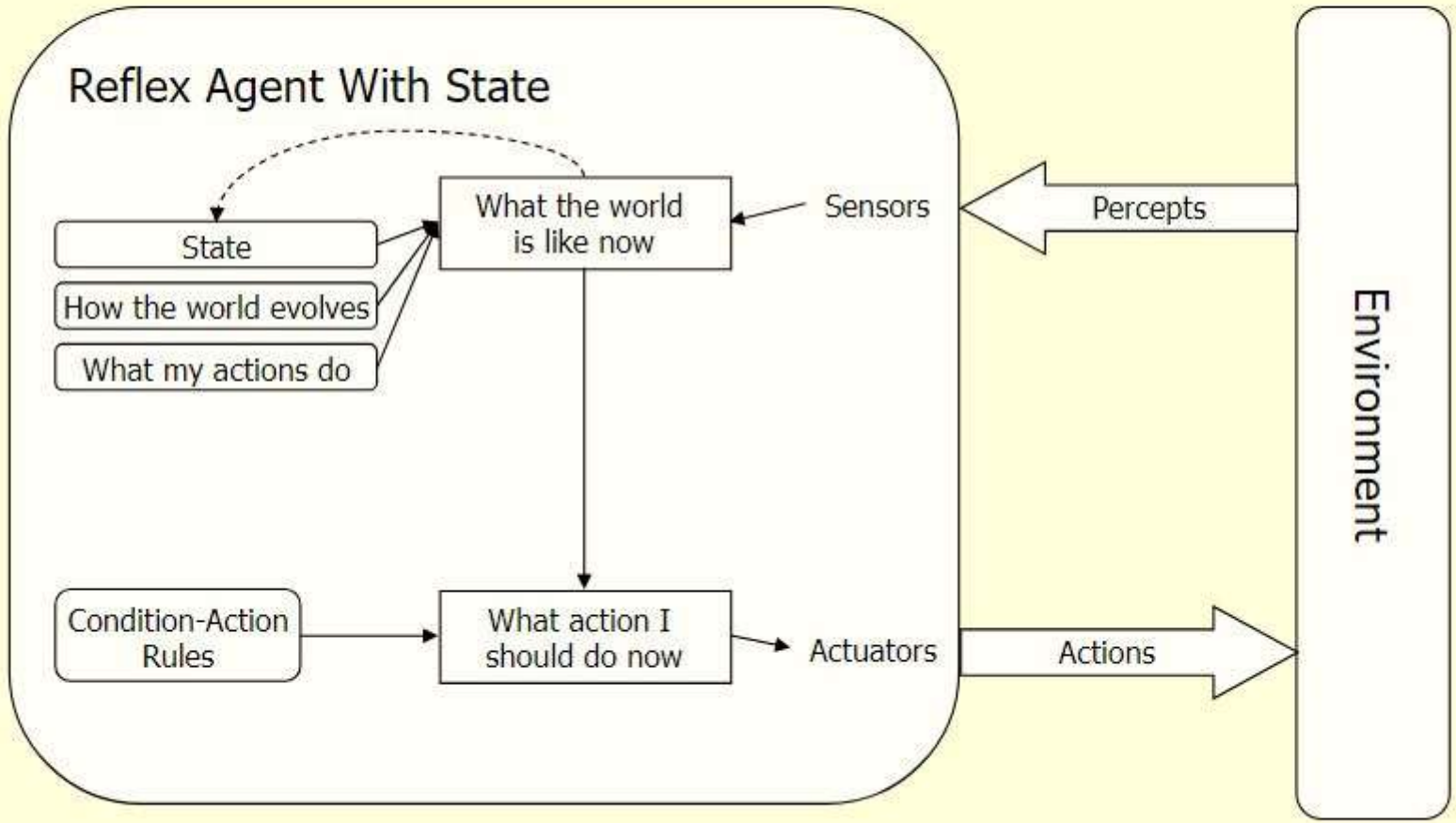
The Structure of Agents

- Simple Reflex Agent

```
function Simple-Reflex-Agent (percept) return action
static:     rules, a set of condition-action rules

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

The Structure of Agents



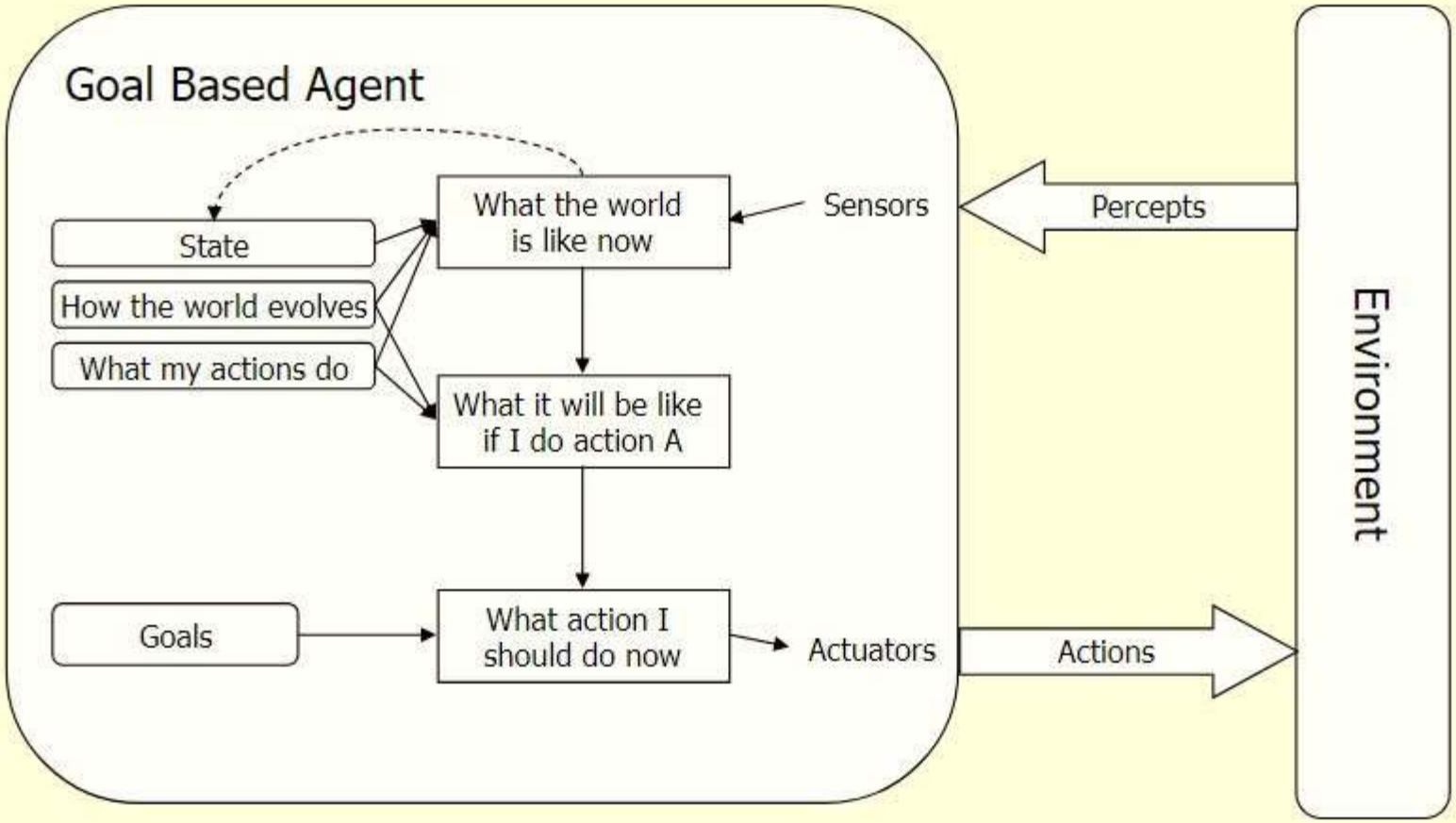
The Structure of Agents

- Reflex Agent With State

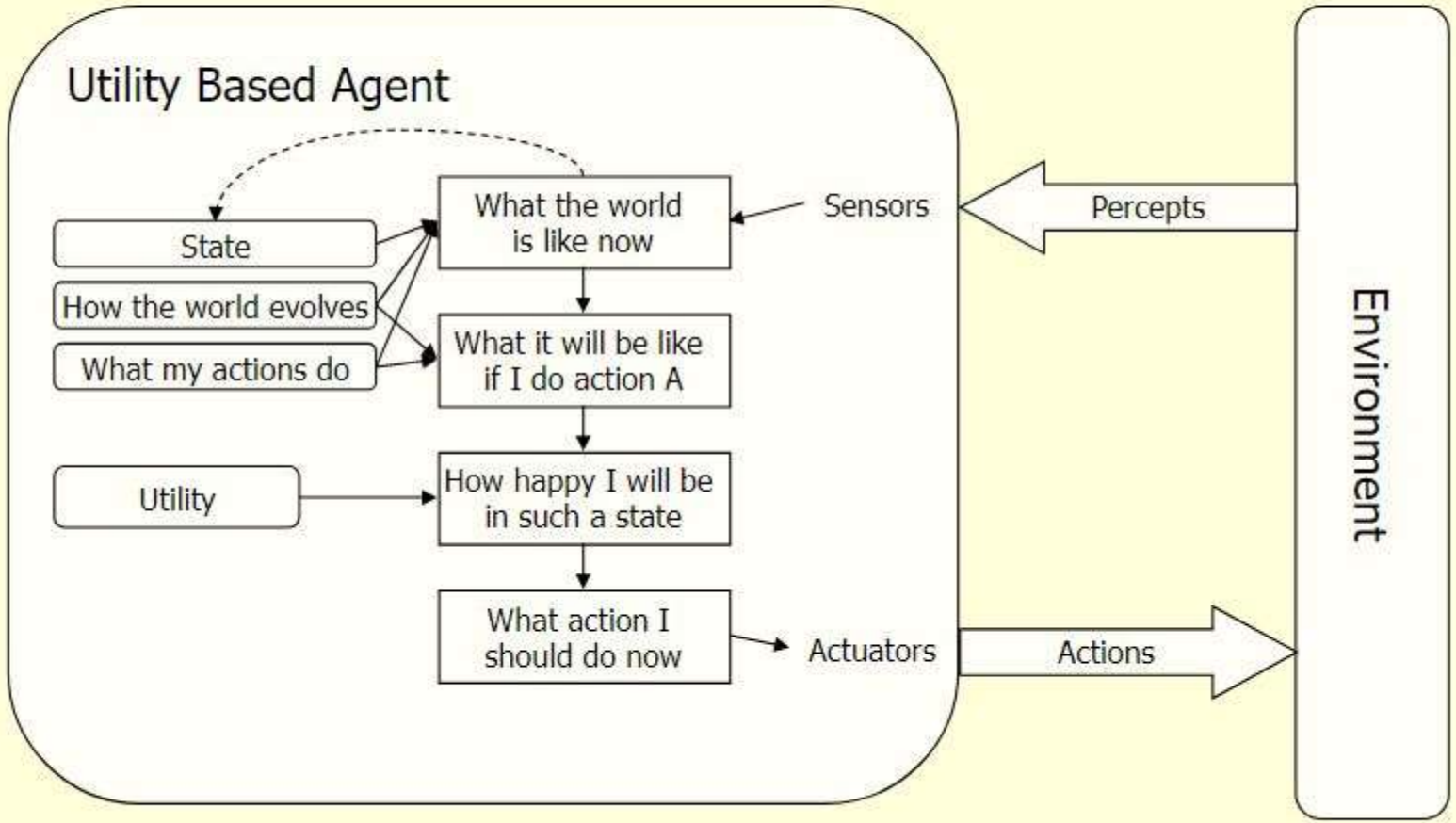
```
function Reflex-Agent-With-State (percept) return action
static: state, a description of the current world state
        rules, a set of condition-action rules
        action, the most recent action, initially none

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

The Structure of Agents



The Structure of Agents



The Structure of Agents