




# BLG435E

## Artificial Intelligence



### Lecture 2: Intelligent Agents



1

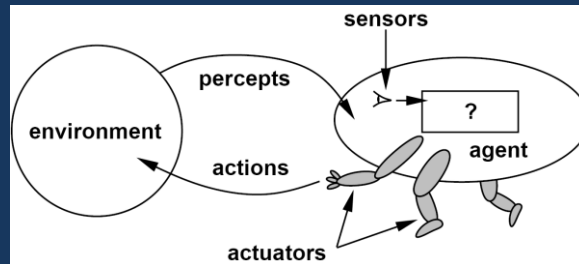
## Outline

- Agents and Environments
- Rationality
- PEAS
- Environment Types
- Agent Types

Artificial Intelligence (BLG435E) @ ITU :: Computer Engineering Department, Dr. Sanem Sanel

2

## Agents

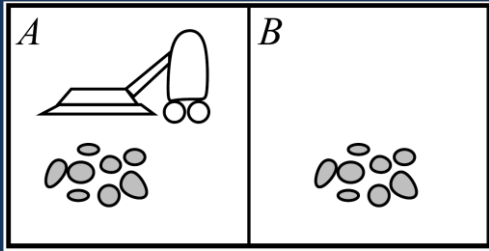


## Agents

- Agents perceive their own actions
  - Effects?
- Percept: the agent's perceptual input
- Percept sequence: the complete history
- Action choices depend on the percept sequence
- Agent function, abstract mathematical description (agent's behavior)
- Agent program implements the function



# Vacuum-Cleaner World



Percepts: location and contents, e.g.,  $[A, Dirty]$

Actions: *Left*, *Right*, *Suck*, *NoOp*

Artificial Intelligence (BLG435E) @ ITU :: Computer Engineering Department, Dr. Sanem Sariel

5

# A Vacuum-Cleaner Agent Function

Percept sequence	Action
$[A, Clean]$	<i>Right</i>
$[A, Dirty]$	<i>Suck</i>
$[B, Clean]$	<i>Left</i>
$[B, Dirty]$	<i>Suck</i>
$[A, Clean], [A, Clean]$	<i>Right</i>
$[A, Clean], [A, Dirty]$	<i>Suck</i>
$\vdots$	$\vdots$

What is the right way to fill out the table?

What makes an agent good, bad or stupid?

Artificial Intelligence (BLG435E) @ ITU :: Computer Engineering Department, Dr. Sanem Sariel

6

## A Vacuum-Cleaner Agent Function

Percept sequence	Action
<i>[A, Clean]</i>	<i>Right</i>
<i>[A, Dirty]</i>	<i>Suck</i>
<i>[B, Clean]</i>	<i>Left</i>
<i>[B, Dirty]</i>	<i>Suck</i>
<i>[A, Clean], [A, Clean]</i>	<i>Right</i>
<i>[A, Clean], [A, Dirty]</i>	<i>Suck</i>
⋮	⋮

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*

Artificial Intelligence (BLG435E) @ ITU :: Computer Engineering Department, Dr. Sanem Sariel

7

## Rationality

- A rational agent does the right thing
- What is rational at any given time depends on:
  - The performance measure that defines the criterion of success
  - The agent’s prior knowledge of the environment
  - The actions that the agent can perform
  - The agent’s percept sequence to date

Artificial Intelligence (BLG435E) @ ITU :: Computer Engineering Department, Dr. Sanem Sariel

8

## Vacuum Cleaner Agent - PM



- The amount of dirt cleaned up in a single eight-hour shift.
- Rewarding agent for having a clean floor.
- Factoring amount of electricity consumed and the amount of noise generated
- Design PM according to what you want



## 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.
- With a rational agent, what you ask is what you get



## Rationality vs. Perfection



- Omniscience is impossible in reality
- Agents don't estimate the actual outcome of actions
- Rationality maximizes expected outcome, while perfection maximizes actual performance



## Further Dimensions in Rationality



- Information gathering
  - Exploration
  - Helps maximize the expected outcome
- Learning
- Autonomy
- With or without initial knowledge



## Agents <> Environments



- Task environment forms the problem
  - Rational agents are the solutions
- The task environment affects the appropriate design of the agent



## The Nature of Environments



- PEAS for task environments:
  - Performance measure
  - Environment
  - Actuators
  - Sensors
- PEAS for automated taxi driver



## Properties of Task Environments



- Fully observable vs. Partially observable
- Deterministic vs. Stochastic
  - Strategic
- Episodic vs. Sequential
- Static vs. Dynamic
  - semidynamic



## Properties of Task Environments




- Discrete vs. Continuous
- Single agent vs. Multiagent
  - competitive
  - cooperative






### Examples of Task Environments




Task Environment	Observable	Deterministic	Episodic	Static	Discrete	Agents
Crossword puzzle						
Chess with a clock						
Poker						
Backgammon						
Taxi driving						




Artificial Intelligence (BLG435E) @ ITU :: Computer Engineering Department, Dr. Sanem Sariel

17

### Examples of Task Environments




Task Environment	Observable	Deterministic	Episodic	Static	Discrete	Agents
Crossword puzzle	Fully	Deterministic	Sequential	Static	Discrete	Single
Chess with a clock						
Poker						
Backgammon						
Taxi driving						




Artificial Intelligence (BLG435E) @ ITU :: Computer Engineering Department, Dr. Sanem Sariel

18

### Examples of Task Environments




Task Environment	Observable	Deterministic	Episodic	Static	Discrete	Agents
Crossword puzzle	Fully	Deterministic	Sequential	Static	Discrete	Single
Chess with a clock	Fully	Strategic	Sequential	Semi	Discrete	Multi
Poker						
Backgammon						
Taxi driving						




Artificial Intelligence (BLG435E) @ ITU :: Computer Engineering Department, Dr. Sanem Sariel

19

### Examples of Task Environments




Task Environment	Observable	Deterministic	Episodic	Static	Discrete	Agents
Crossword puzzle	Fully	Deterministic	Sequential	Static	Discrete	Single
Chess with a clock	Fully	Strategic	Sequential	Semi	Discrete	Multi
Poker	Partially	Stochastic	Sequential	Static	Discrete	Multi
Backgammon						
Taxi driving						




Artificial Intelligence (BLG435E) @ ITU :: Computer Engineering Department, Dr. Sanem Sariel

20

### Examples of Task Environments




Task Environment	Observable	Deterministic	Episodic	Static	Discrete	Agents
Crossword puzzle	Fully	Deterministic	Sequential	Static	Discrete	Single
Chess with a clock	Fully	Strategic	Sequential	Semi	Discrete	Multi
Poker	Partially	Stochastic	Sequential	Static	Discrete	Multi
Backgammon	Fully	Stochastic	Sequential	Static	Discrete	Multi
Taxi driving						




Artificial Intelligence (BLG435E) @ ITU :: Computer Engineering Department, Dr. Sanem Sariel

21

### Examples of Task Environments



Task Environment	Observable	Deterministic	Episodic	Static	Discrete	Agents
Crossword puzzle	Fully	Deterministic	Sequential	Static	Discrete	Single
Chess with a clock	Fully	Strategic	Sequential	Semi	Discrete	Multi
Poker	Partially	Stochastic	Sequential	Static	Discrete	Multi
Backgammon	Fully	Stochastic	Sequential	Static	Discrete	Multi
Taxi driving	Partially	Stochastic	Sequential	Dynamic	Continuous	Multi



Artificial Intelligence (BLG435E) @ ITU :: Computer Engineering Department, Dr. Sanem Sariel

22

## The Structure of Agents



- The job of AI is to design the agent program
- Agent architecture
- Agent = Architecture + Program



## Agent Types



- Simple reflex agents
- Model-based reflex agents
- Goal-based reflex agents
- Utility-based agents
- All these agents can be converted into learning agents



# Simple Reflex Agents

```
graph LR
    subgraph Agent
        Sensors --> W1[What the world is like now]
        W1 --> W2[What action I should do now]
        CAR([Condition-action rules]) --> W2
        W2 --> Actuators
    end
    subgraph Environment
    end
    Actuators --> Environment
    Environment --> Sensors
```

Artificial Intelligence (BLG435E) @ ITU :: Computer Engineering Department, Dr. Sanem Sariel

25

# Reflex Vacuum Agent Program

```
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

(setq joe (make-agent :name 'joe :body (make-agent-body)
                     :program (make-reflex-vacuum-agent-program)))

(defun make-reflex-vacuum-agent-program ()
  #'(lambda (percept)
      (let ((location (first percept)) (status (second percept)))
        (cond ((eq status 'dirty) 'Suck)
              ((eq location 'A) 'Right)
              ((eq location 'B) 'Left))))))
```

A	B

Artificial Intelligence (BLG435E) @ ITU :: Computer Engineering Department, Dr. Sanem Sariel

26

# Simple-Reflex Agent Program

```
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
```

Artificial Intelligence (BLG435E) @ ITU :: Computer Engineering Department, Dr. Sanem Sariel

27

# Model-based Reflex Agents

The diagram illustrates the internal components of a Model-based Reflex Agent. It is divided into two main sections: the **Agent** (left) and the **Environment** (right). Within the Agent, there are several interconnected components: 

- Sensors**: Receives input from the Environment and feeds into the 'What the world is like now' box.
- State**: A memory component that receives input from 'How the world evolves' and 'What my actions do', and feeds into the 'What the world is like now' box.
- What the world is like now**: A central box that receives input from Sensors and State, and feeds into the 'What action I should do now' box.
- Condition-action rules**: A box that receives input from 'What the world is like now' and feeds into the 'What action I should do now' box.
- What action I should do now**: A box that receives input from 'What the world is like now' and 'Condition-action rules', and feeds into the **Actuators**.
- Actuators**: Send output back to the Environment.


Artificial Intelligence (BLG435E) @ ITU :: Computer Engineering Department, Dr. Sanem Sariel

28

# Model-based Agent Program

```
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

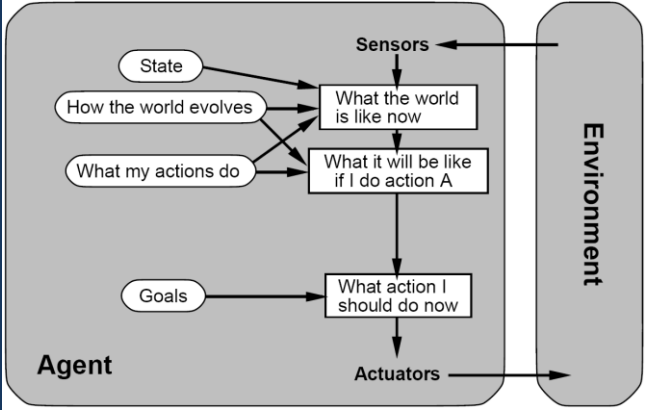
  state ← UPDATE-STATE(state, action, percept)
  rule ← RULE-MATCH(state, rules)
  action ← RULE-ACTION[rule]
  return action
```




Artificial Intelligence (BLG435E) @ ITU :: Computer Engineering Department, Dr. Sanem Sariel

29

# Goal-Based Agents



The diagram illustrates the internal components of a Goal-Based Agent and its interaction with the Environment. The Agent's internal state is represented by a large rounded rectangle containing several boxes and ovals. On the left, ovals represent 'State', 'How the world evolves', 'What my actions do', and 'Goals'. In the center, there are three stacked boxes: 'What the world is like now', 'What it will be like if I do action A', and 'What action I should do now'. On the right, an oval represents 'Actuators'. Arrows show the flow of information: 'Sensors' (from the Environment) point to 'What the world is like now'. 'State', 'How the world evolves', and 'What my actions do' all point to 'What the world is like now'. 'What the world is like now' points to 'What it will be like if I do action A'. 'Goals' points to 'What action I should do now'. 'What it will be like if I do action A' points to 'What action I should do now'. Finally, 'What action I should do now' points to 'Actuators', which then interacts with the 'Environment'.



Artificial Intelligence (BLG435E) @ ITU :: Computer Engineering Department, Dr. Sanem Sariel

30

# Utility-based Agents

```
graph TD
    subgraph Agent
        S[Sensors] --> W1[What the world is like now]
        W1 --> W2[What it will be like if I do action A]
        W2 --> H[How happy I will be in such a state]
        H --> A[What action I should do now]
        A --> Act[Actuators]
        S --> S2[State]
        S --> S3[How the world evolves]
        S --> S4[What my actions do]
        S --> S5[Utility]
    end
    subgraph Environment
        Act --> Env[Environment]
        Env --> S
    end
```

Artificial Intelligence (BLG435E) @ ITU :: Computer Engineering Department, Dr. Sanem Sariel

31

# Learning Agents

```
graph TD
    subgraph Agent
        S[Sensors] --> Critic
        Critic -- feedback --> LE[Learning element]
        LE -- changes --> PE[Performance element]
        PE -- knowledge --> LE
        LE -- learning goals --> PG[Problem generator]
        PG -- knowledge --> PE
        PE --> Act[Actuators]
    end
    subgraph Environment
        Act --> Env[Environment]
        Env --> S
    end
    PS[Performance standard] --> Critic
```

Artificial Intelligence (BLG435E) @ ITU :: Computer Engineering Department, Dr. Sanem Sariel

32