15Z701- ARTIFICIAL INTELLIGENCE

Course Faculty
Dr.D.Indumathi
Assistant Professor(SG)
CSE,PSGCT

Course Outcomes

Upon completion of this course, the students will be able to

- **CO1:** Apply the fundamental AI concepts to represent and solve real-world problems
- **CO2:** Analyze and solve AI problems using heuristic, non-heuristic and adversarial search strategies
- **CO3:** Employ knowledge representation, logical reasoning and statistical methods to solve AI problems
- **CO4:** Apply planning and learning concepts to solve AI problems
- **CO5:** Apply NLP concepts to analyze and solve real-world problems

Syllabus:15Z701-AI

INTRODUCTION: Artificial Intelligence -History - The State of Art - Intelligent Agents - Structure - Environment. (5)

SEARCH STRATEGIES: Breadth-First Search - Uniform Cost Search - Depth-First Search - Depth-Limited Search - Iterative Deepening Search - Bidirectional Search - Heuristic Search Techniques - A* Search - AO* Algorithm - **Adversarial Search:** Minimax Algorithm - Alphabeta Pruning. (12)

KNOWLEDGE AND REASONING: Representation - First Order Predicate Logic – Inference – Unification - Forward and Backward Chaining - Resolution - Reasoning with Default Information - Truth Maintenance Systems - Acting under Uncertainty - Statistical Reasoning - Probability and Bayes Theorem - Certainty Factors and Rule Based Systems - Dempster-Shafer Theory. (10)

PLANNING AND LEARNING:Planning with State Space Search: Partial Order Planning - Planning Graphs - Examples. **Forms of Learning**: Inductive Learning - Explanation Based Learning - Statistical Learning - Learning With Complete Data. (10)

NATURAL LANGUAGE PROCESSING: Phases - Syntactic Processing - Semantic Analysis - Discourse and Pragmatic Processing. (8)

TEXT BOOKS

1. Stuart J Russell and Peter Norvig, "Artificial Intelligence - A Modern Approach", Third Edition, Prentice Hall of India/ Pearson Education, New Delhi, 2015.
2. Elaine Rich and Kevin Knight, "Artificial Intelligence", Tata McGraw Hill Publishing Company, New Delhi, 2014.

ARTIFICIAL INTELLIGENCE

INTRODUCTION

Introduction

- Artificial Intelligence is the study of how to make computers do things which, at the moment, people do better
- AI-it attempts not just to understand but also to build intelligent entities
- Artificial Intelligence is an area of computer science that emphasizes the creation of intelligent machines that work and react like humans

Introduction (Contd..)

- There are three kinds of intelligence: one kind understands things for itself.
- The other appreciates what others can understand,
- The third understands neither for itself nor through others
- This first kind is excellent, the second good, and the third kind useless

-Niccolo Machiavelli

(1467-1527), Italian diplomat

Introduction (Contd..)

- Intelligence requires knowledge
- Knowledge possesses some less desirable properties, including
    - It is voluminous
    - It is hard to characterize accurately
    - It is constantly changing
    - It differs from data by being organised in a way that corresponds to the ways it will be used

AI technique

- Knowledge should be represented in such a way that
  - It captures generalizations
  - It can be understood by people
  - It can easily be modified to correct errors and to reflect changes in the world and in our world view
  - It can be used in a great many situations even if it is not totally accurate or complete
  - It can be used to help its own sheer bulk by narrowing the range of possibilities

Applications

- Autonomous planning and scheduling – NASA's remote agent program
- Game playing –IBM Deep blue vs Gary kasparov
- Autonomous control-Self driving vehicles
- Diagnosis-medical expert systems
- Logistics planning-automated logistics planning and scheduling for transportation
- Robotics – Robot assistants in microsurgery
- Language understanding and problem solving

To solve an AI problem

- Define the problem precisely
- Analyze the problem
- Isolate and represent the task knowledge
- Choose the best problem solving technique and apply it

Problem Characteristics

1. Is the problem decomposable?
2. Can solution steps be ignored or atleast undone if they prove unwise?
   - Ignorable (theorem proving)
   - Recoverable (8-puzzle)
   - Irrecoverable (chess)
3. Is the problems universe predictable?
   - 8 puzzle problem
   - Lawyer defending his client against a murder charge
4. Is a good solution to the problem obvious without comparison to all other problem solutions?
   - Absolute or relative

Problem Characteristics (Contd..)

- Is the desired solution a state of the world or a path to a state?
  - finding a consistent interpretation for the sentence
    - **The bank president ate a dish of pasta salad with the fork**
  - problems whose solutions is a path
    - **Eg: Water jug problem**
- Is a large amount of knowledge absolutely required to solve the problem?
- Will the solution of the problem require interaction between the computer and a person?
  - solitary or conversational

Water jug problem

- A Water Jug Problem: You are given two jugs, a 4-gallon one and a 3-gallon one, a pump which has unlimited water which you can use to fill the jug, and the ground on which water may be poured. Neither jug has any measuring markings on it. How can you get exactly 2 gallons of water in the 4-gallon jug?

- State Representation and Initial State – we will represent a state of the problem as a tuple (x, y) where x represents the amount of water in the 4-gallon jug and y represents the amount of water in the 3-gallon jug. Note $0 \leq x \leq 4$, and $0 \leq y \leq 3$. Our initial state: (0,0)

Goal Predicate – state = (2,y) where $0 \leq y \leq 3$.

Operators – we must define a set of operators that will take us from one state to another:

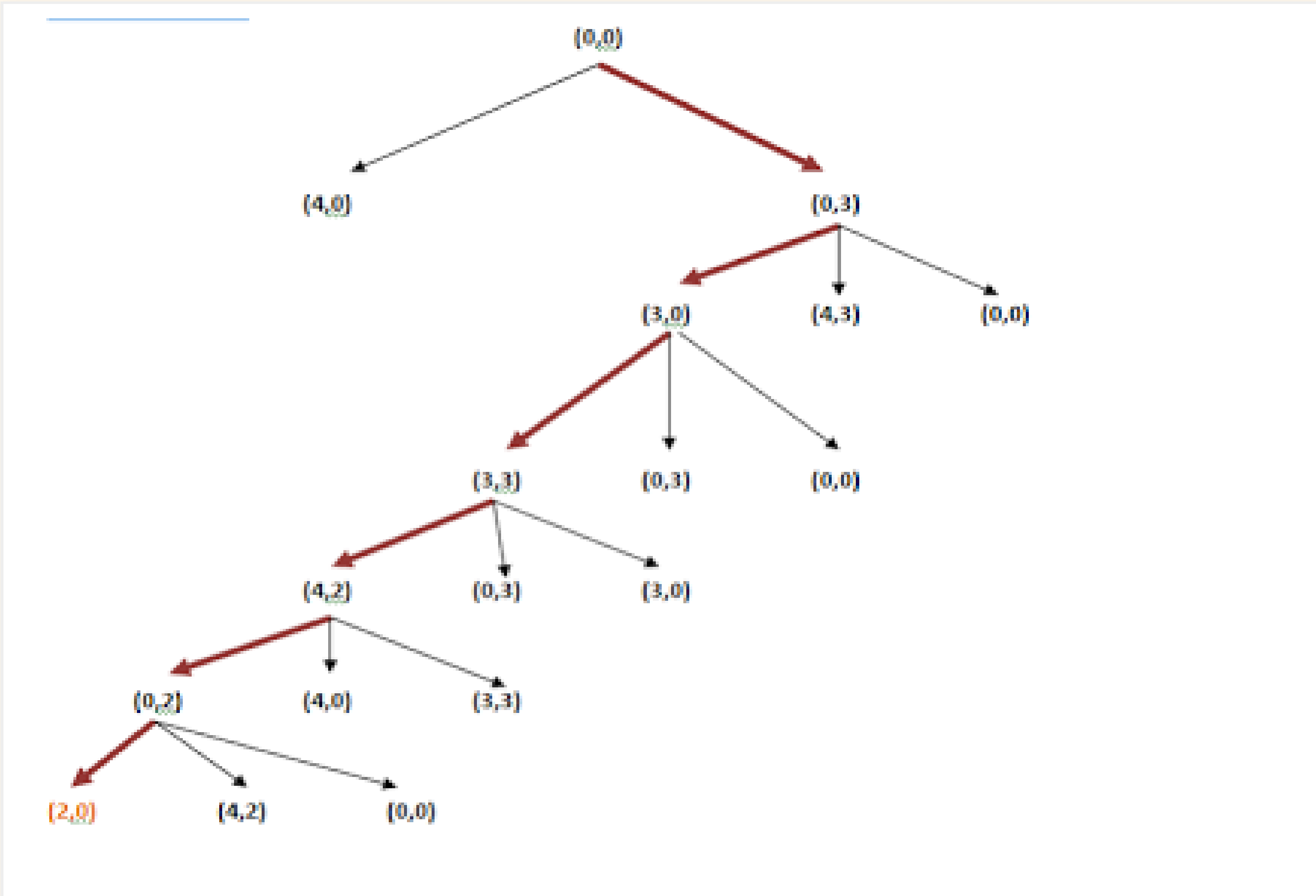
1. Fill 4-gal jug	(x,y) $x < 4$	$\rightarrow$	(4,y)
2. Fill 3-gal jug	(x,y) $y < 3$	$\rightarrow$	(x,3)
3. Empty 4-gal jug on ground	(x,y) $x > 0$	$\rightarrow$	(0,y)
4. Empty 3-gal jug on ground	(x,y) $y > 0$	$\rightarrow$	(x,0)
5. Pour water from 3-gal jug to fill 4-gal jug	(x,y) $0 < x+y \geq 4$ and $y > 0$	$\rightarrow$	(4, y - (4 - x))
6. Pour water from 4-gal jug to fill 3-gal-jug	(x,y) $0 < x+y \geq 3$ and $x > 0$	$\rightarrow$	(x - (3-y), 3)
7. Pour all of water from 3-gal jug into 4-gal jug	(x,y) $0 < x+y \leq 4$ and $y \geq 0$	$\rightarrow$	(x+y, 0)
8. Pour all of water from 4-gal jug into 3-gal jug	(x,y) $0 < x+y \leq 3$ and $x \geq 0$	$\rightarrow$	(0, x+y)

Possible solution

Gals in 4-gal jug	Gals in 3-gal jug	Rule Applied
0	0	
		1. Fill 4
4	0	
		6. Pour 4 into 3 to fill
1	3	
		4. Empty 3
1	0	
		8. Pour all of 4 into 3
0	1	
		1. Fill 4
4	1	
		6. Pour into 3
2	3	

Another possible solution

State Space Tree:



Definitions of AI

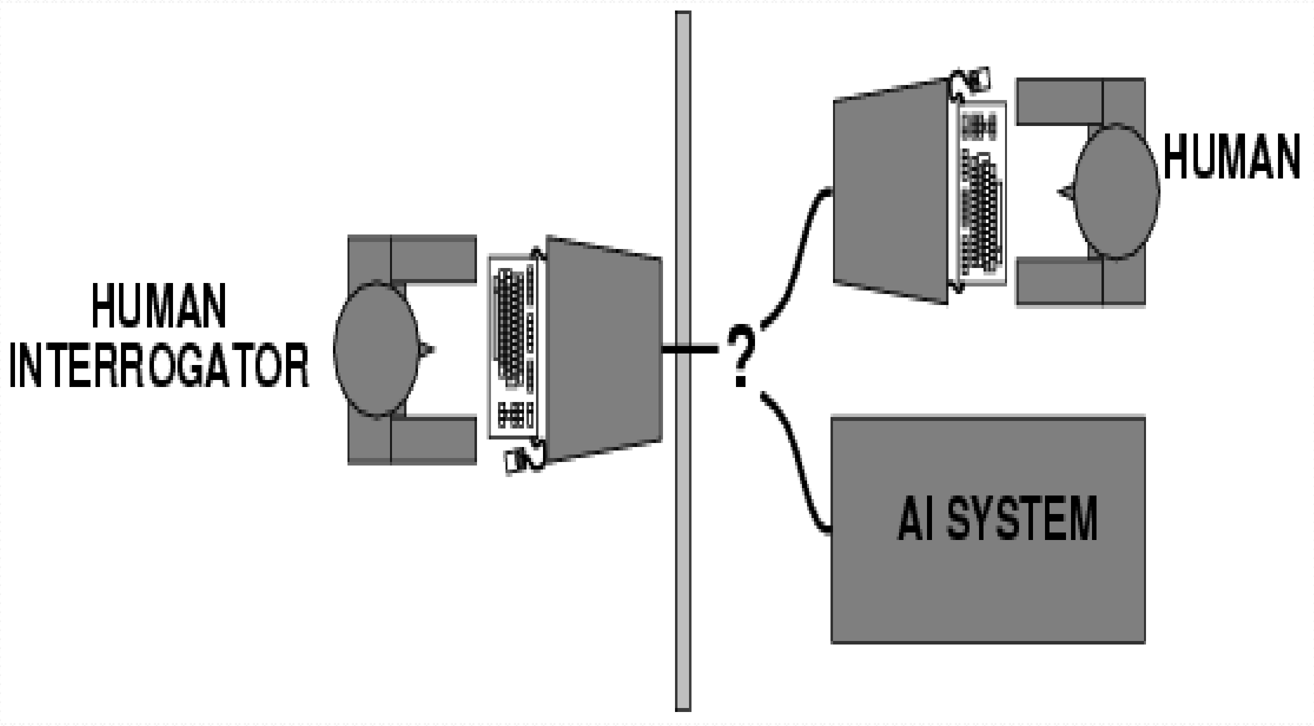
Thinking Humanly "The exciting new effort to make computers think . . . *machines with minds*, in the full and literal sense." (Haugeland, 1985) "[The automation of] activities that we associate with human thinking, activities such as decision-making, problem solving, learning . . ." (Bellman, 1978)	**Thinking Rationally** "The study of mental faculties through the use of computational models." (Charniak and McDermott, 1985) "The study of the computations that make it possible to perceive, reason, and act." (Winston, 1992)
Acting Humanly "The art of creating machines that perform functions that require intelligence when performed by people." (Kurzweil, 1990) "The study of how to make computers do things at which, at the moment, people are better." (Rich and Knight, 1991)	**Acting Rationally** "Computational Intelligence is the study of the design of intelligent agents." (Poole *et al.*, 1998) "AI . . . is concerned with intelligent behavior in artifacts." (Nilsson, 1998)

Acting humanly: The Turing Test approach

The computer would need to possess the following capabilities:

- **natural language processing** to enable it to communicate successfully in English;
- **knowledge representation** to store what it knows or hears;
- **automated reasoning** to use the stored information to answer questions and to draw new conclusions
- **machine learning** to adapt to new circumstances and to detect and extrapolate patterns.

TURING TEST



Total Turing Test

To pass the total Turing Test, the computer will need

- **computer vision** to perceive objects, and
- **robotics** to manipulate objects and move about.

Thinking humanly: The cognitive modeling approach

- Through introspection—trying to catch our own thoughts as they go by;
- Through psychological experiments—observing a person in action;
- Through brain imaging—observing the brain in action.
- Eg: GPS(Newell and Simon 1961)
- AI and Cognitive science in Computer Vision- incorporates neurophysiological evidence into computational models

Thinking rationally: The “laws of thought” approach

Syllogisms provided patterns for argument structures that always yielded correct conclusions when given correct premises -Aristotle

Logic

- “Socrates is a man; all men are mortal; therefore, Socrates is mortal.”
- **logicist** tradition within AI build on programs to create intelligent systems
- Laws of thought approach-emphasize on correct inferences

Acting rationally: The rational agent approach

- An **Agent** is just something that acts
- A **Rational agent** is one that acts so as to achieve the best outcome or, when there is uncertainty, the best expected outcome.
- Limited rationality
- The rational-agent approach has two advantages over the other approaches.
  - It is more general than the "laws of thought" approach because correct inference is just one of several possible mechanisms for achieving rationality.
  - It is more amenable to scientific development than approaches based on human behavior or human thought.

Agents

- An **Agent** is anything that can be viewed as perceiving its **environment** through **sensors** and acting upon that environment through **actuators**.
- A **human agent** has eyes, ears, and other organs for sensors and hands, legs, vocal tract, and so on for actuators.
- A **robotic agent** might have cameras and infrared range finders for sensors and various motors for actuators.
- A **software agent** receives keystrokes, file contents, and network packets as sensory inputs and acts on the environment by displaying on the screen, writing files, and sending network packets.

Agent Schematic Diagram

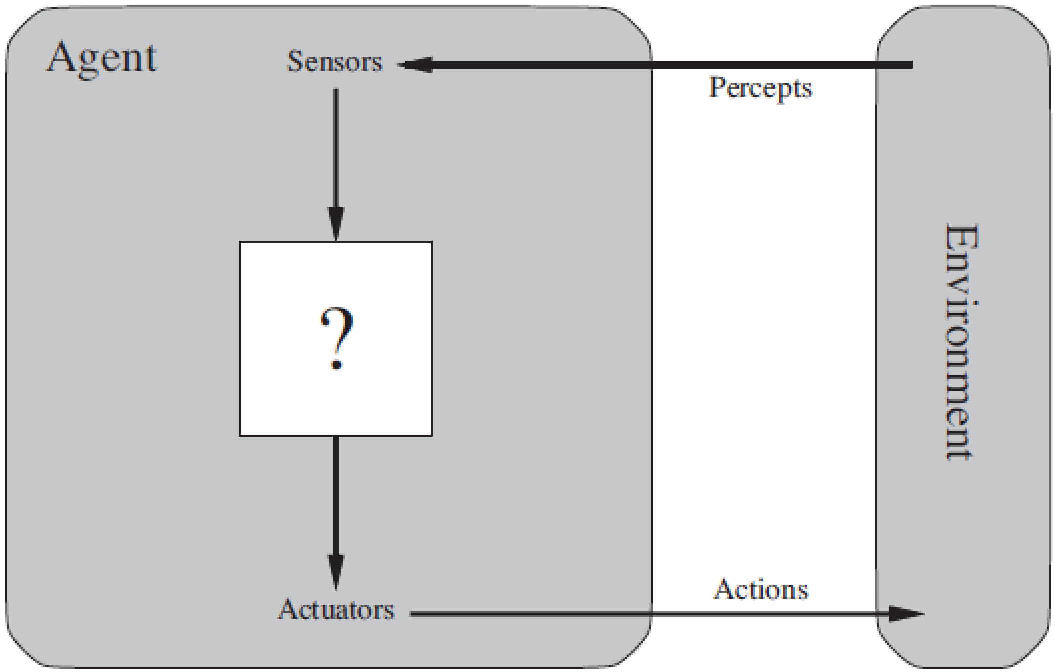


Figure 2.1 Agents interact with environments through sensors and actuators.

Agents and environments

- The agent function maps from percept histories to actions:

$$[f: \mathcal{P}^{*} \rightarrow \mathcal{A}]$$

- The agent program runs on the physical architecture to produce f

agent = architecture + program

- The agent function is an abstract mathematical description; the agent program is a concrete implementation, running within some physical system.

Vacuum-Cleaner World

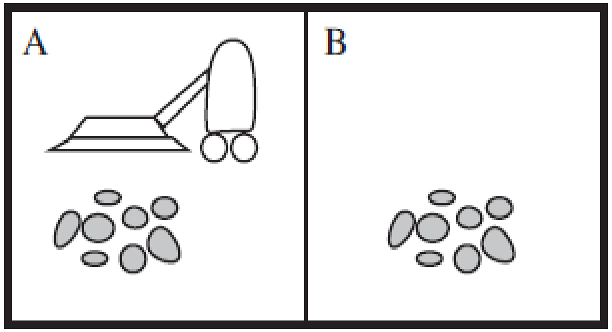


Figure 2.2 A vacuum-cleaner world with just two locations.

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*
⋮	⋮
[*A*, *Clean*], [*A*, *Clean*], [*A*, *Clean*]	*Right*
[*A*, *Clean*], [*A*, *Clean*], [*A*, *Dirty*]	*Suck*
⋮	⋮

Figure 2.3 Partial tabulation of a simple agent function for the vacuum-cleaner world shown in Figure 2.2.

Rational agents

- An agent should strive to "do the right thing", based on what it can perceive and the actions it can perform. The right action is the one that will cause the agent to be most successful
- Performance measure: An objective criterion for success of an agent's behavior
- E.g., performance measure of a vacuum-cleaner agent could be amount of dirt cleaned up, amount of time taken, amount of electricity consumed, amount of noise generated, etc.

What is Rational?

- 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.

- 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.

PEAS

- PEAS: Performance measure, Environment, Actuators, Sensors
- Must first specify the setting for intelligent agent design
- E.g., the task of designing an automated taxi driver, Medical diagnosis system, Interactive Tutor etc:
  - Performance measure
  - Environment
  - Actuators
  - Sensors

PEAS

- E.g., the task of designing an automated taxi driver:
  - Performance measure: Safe, fast, legal, comfortable trip, maximize profits
  - Environment: Roads, other traffic, pedestrians, customers
  - Actuators: Steering wheel, accelerator, brake, signal, horn
  - Sensors: Cameras, sonar, speedometer, GPS, odometer, engine sensors, keyboard

PEAS

- Agent: Medical diagnosis system
- Performance measure: Healthy patient, minimize costs, lawsuits
- Environment: Patient, hospital, staff
- Actuators: Screen display (questions, tests, diagnoses, treatments, referrals)
- Sensors: Keyboard (entry of symptoms, findings, patient's answers)

PEAS

- Agent: Part-picking robot
- Performance measure: Percentage of parts in correct bins
- Environment: Conveyor belt with parts, bins
- Actuators: Jointed arm and hand
- Sensors: Camera, joint angle sensors

PEAS

- Agent: Interactive English tutor
- Performance measure: Maximize student's score on test
- Environment: Set of students
- Actuators: Screen display (exercises, suggestions, corrections)
- Sensors: Keyboard

Assignment

- For each of the following activities, give a PEAS description of the task environment and characterize it in terms of the properties

1. Playing Cricket
2. Part-picking Robot
3. Knitting a Sweater
4. Bidding on an item at an auction

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.

Eg:, a vacuum agent with only a local dirt sensor cannot tell whether there is dirt in other squares, and an automated taxi cannot see what other

drivers are thinking. If the agent has no sensors at all then the environment is **unobservable**.

- Single agent (vs. multiagent): An agent operating by itself in an environment.

Eg: an agent solving a crossword puzzle by itself is clearly in a single-agent environment, whereas an agent playing chess is in a two agent environment

Environment types

- Deterministic (vs. stochastic): The next state of the environment is completely determined by the current state and the action executed by the agent.

Eg: Taxi driving is clearly stochastic because one can never predict the behavior of traffic exactly;

The vacuum world is deterministic

Environment types

- Episodic (vs. sequential): The agent's experience is divided into atomic "episodes" (each episode consists of the agent perceiving and then performing a single action), and the choice of action in each episode depends only on the episode itself.

Eg: Chess and taxi driving are sequential:

Chess tournament: episodic

Environment types

- 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)

Eg: Chess: Static

Taxi driving: Dynamic

- Discrete (vs. continuous):The 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.

Eg: Chess: Discrete

Taxi driving: Continuous

Table Driven Agent

```
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
```

Figure 2.7 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.

Table Driven Agent

- Drawbacks:
  - Huge table
  - Take a long time to build the table
  - No autonomy
  - Even with learning, need a long time to learn the table entries

Agent types

- Four basic types in order of increasing generality:
  - Simple reflex agents
  - Model-based reflex agents
  - Goal-based agents
  - Utility-based agents

Simple Reflex 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
```

Figure 2.8 The agent program for a simple reflex agent in the two-state vacuum environment. This program implements the agent function tabulated in Figure 2.3.

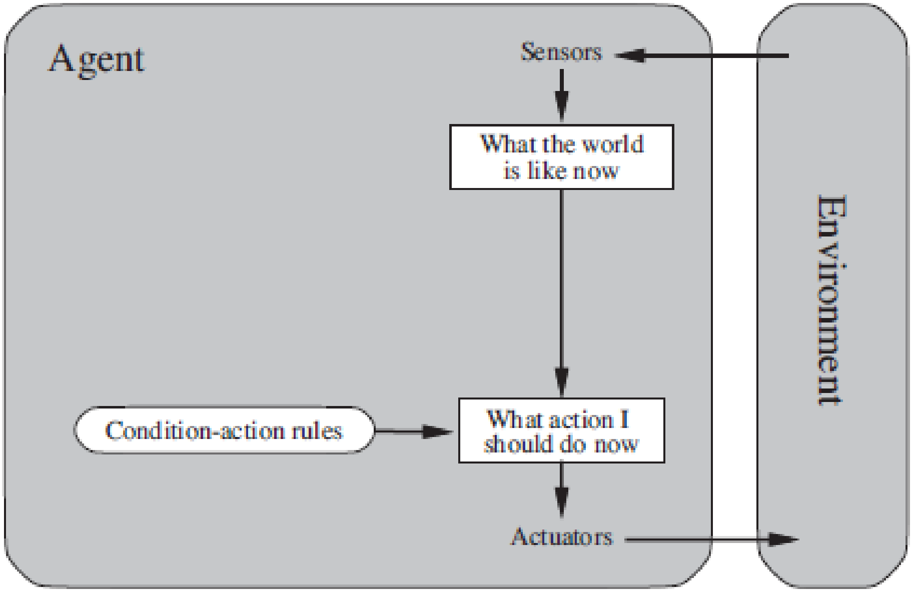


Figure 2.9 Schematic diagram of a simple reflex agent.

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*

Figure 2.10 A simple reflex agent. It acts according to a rule whose condition matches the current state, as defined by the percept.

Simple Reflex Agent

- Correct decision is made on the basis of only the current percept—only if the environment is fully observable.

  Eg: **if** *car-in-front-is-braking* **then** *initiate-braking*.

- Infinite loops are possible in simple reflex agents operating in partially observable environments.
- RANDOMIZATION is one possible solution- if the agent can **randomize** its actions

  Eg: flip a coin and decide

Vacuum-Cleaner World

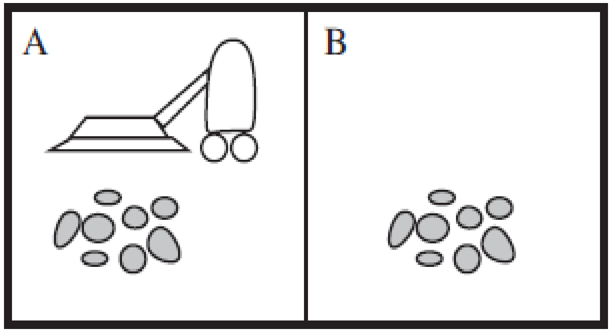


Figure 2.2 A vacuum-cleaner world with just two locations.

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*
⋮	⋮
[*A*, *Clean*], [*A*, *Clean*], [*A*, *Clean*]	*Right*
[*A*, *Clean*], [*A*, *Clean*], [*A*, *Dirty*]	*Suck*
⋮	⋮

Figure 2.3 Partial tabulation of a simple agent function for the vacuum-cleaner world shown in Figure 2.2.

Model-based reflex agents

- Handles partial observability- keep track of the part of the world it can't see now.
- Agent maintains some sort of **internal state** that depends on the percept history and thereby reflects at least some of the unobserved aspects of the current state.
- This knowledge about "how the world works"—whether implemented in simple Boolean circuits or in complete scientific theories—is called a **model** of the world. An agent that uses such a model is called a **model-based agent**.
- Eg: braking problem-previous frame and current frame

Model-based reflex agents

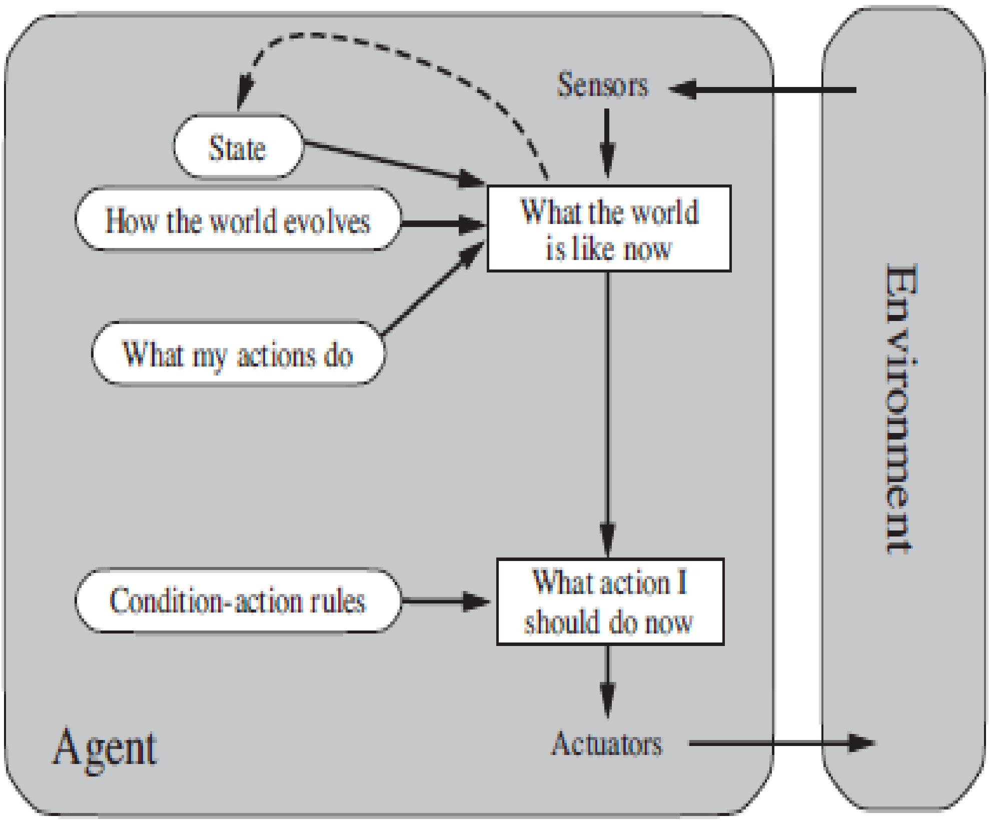


Figure 2.11 A model-based reflex agent.

Model-based reflex agents

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.

Model-based reflex agents

- Eg: an automated taxi may not be able to see around the large truck that has stopped in front of it – it has to make a guess
- Uncertainty about the current state may be unavoidable, but the agent still has to make a decision.

Goal-based agents

- Along with current state description, the agent needs some sort of **goal** information that describes situations that are desirable

  Eg: being at the passenger's destination.
- The goal-based agent's behavior can easily be changed to go to a different destination, simply by specifying that destination as the goal.
- The agent program can combine this with the model based reflex agent to choose actions that achieve the goal.
- Goal action can be single or multiple actions

Model-based,goal based agent

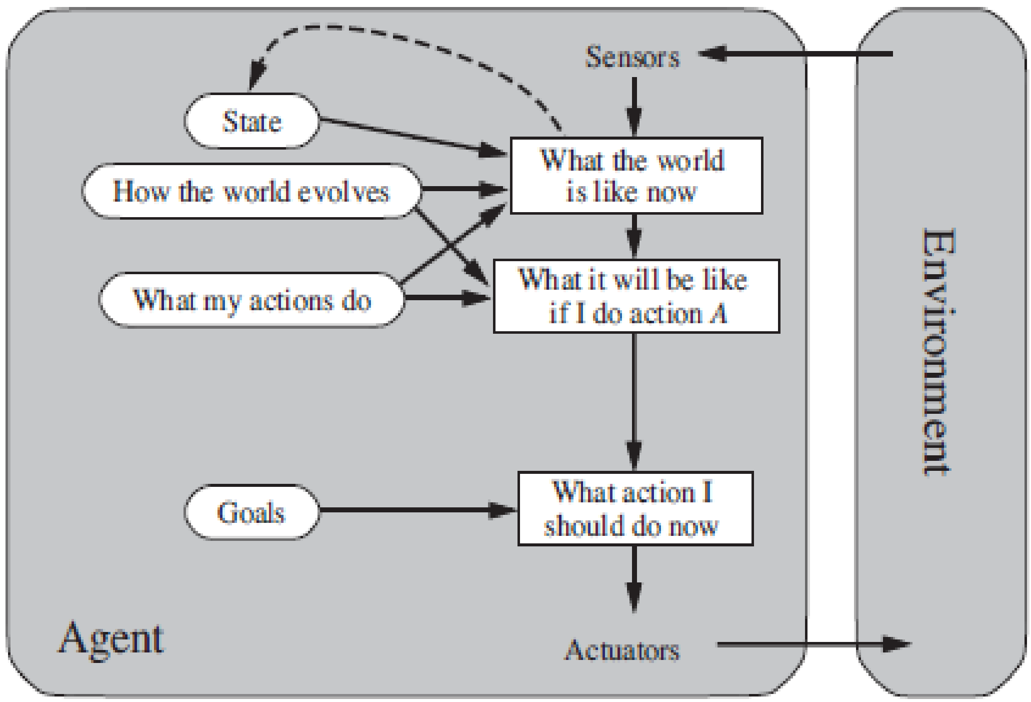


Figure 2.13 A model-based, goal-based agent. It keeps track of the world state as well as a set of goals it is trying to achieve, and chooses an action that will (eventually) lead to the achievement of its goals.

Utility-based agents

- Goals just provide a crude binary distinction between "happy" and "unhappy" states. **(UTILITY)**
- A 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 a 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.

Model-based,utility based agent

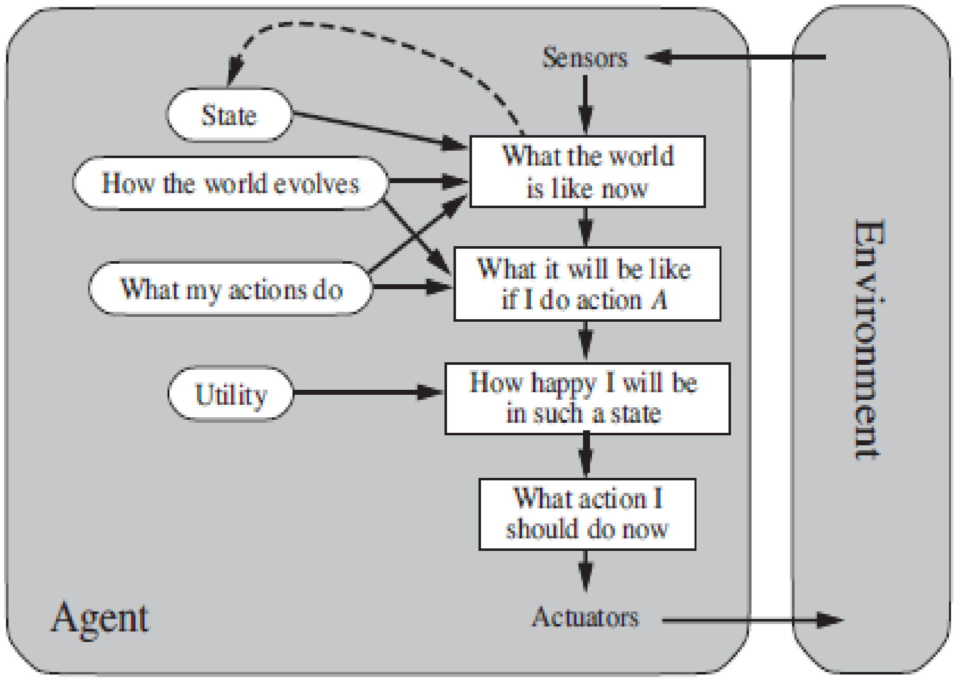


Figure 2.14 A model-based, utility-based agent. It uses a model of the world, along with a utility function that measures its preferences among states of the world. Then it chooses the action that leads to the best expected utility, where expected utility is computed by averaging over all possible outcome states, weighted by the probability of the outcome.

Utility-based agents(Contd..)

- For conflicting goals,(for example, speed and safety), the utility function specifies the appropriate tradeoff.
- For more than one goal- utility provides a way in which the likelihood of success can be weighed against the importance of the goals
- A utility-based agent has to model and keep track of its environment, tasks that 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

Learning Agents

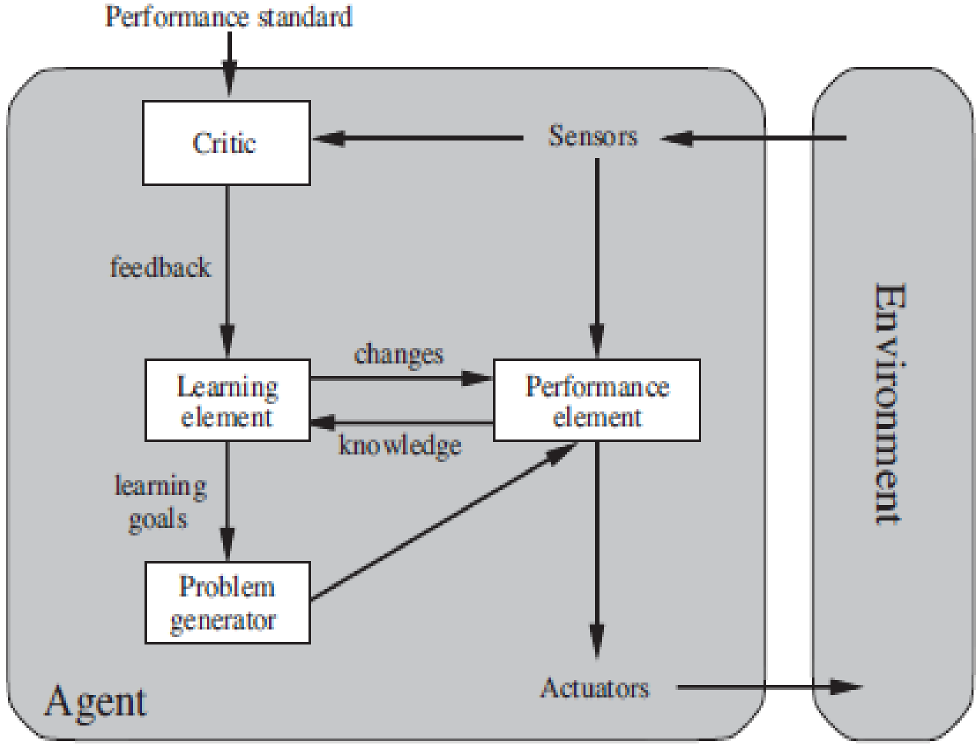


Figure 2.15 A general learning agent.

Learning Agents

Four conceptual components,

- **Learning element**, which is responsible for making improvements,
- **Performance element**, which is responsible for selecting external actions., it takes in percepts and decides on actions.
- The learning element uses feedback from the **Critic** on how the agent is doing and determines how the performance element should be modified to do better in the future.

Learning Agents(Contd..)

- **Problem generator -** responsible for suggesting actions that will lead to new and informative experiences
- Suggests exploratory actions(suboptimal actions)
- Learning in intelligent agents can be summarized as a process of modification of each component of the agent to bring the components into closer agreement with the available feedback information, thereby improving the overall performance of the agent.

Automated Taxi

- **Performance Element (PE):** collection of knowledge and procedures the taxi has for selecting its driving actions. The taxi goes out on the road and drives, using this PE
- The **Critic** observes the world and passes information along to the learning element.
- Eg: after the taxi makes a quick left turn across three lanes of traffic, the critic observes the shocking language used by other drivers.
- From this experience, the **Learning element** is able to formulate a rule saying this was a bad action, and the PE element is modified by installation of the new rule.
- The **Problem generator** might identify certain areas of behavior in need of improvement and suggest experiments, such as trying out the brakes on different road surfaces under different conditions

Exercise

- Consider an e-shopping agent
  - Create a learning agent based architecture for this agent
  - List the Components responsibilities

THANK YOU