COMP 401: Artificial Intelligence

Lecture 1

Introduction

23-9-2019

Course Information

- Course Code: Comp 401
- Course Title: Artificial Intelligence
- Number of Credit Hours: 3 (3 hours Lecture + 0 Lab.)
- Prerequisite : -
- Final Exam (duration): 3 hours
- Total Marks: 150
  - 105 Final Exam.
  - 37 Mid-Term.
  - 8 Oral Exam.

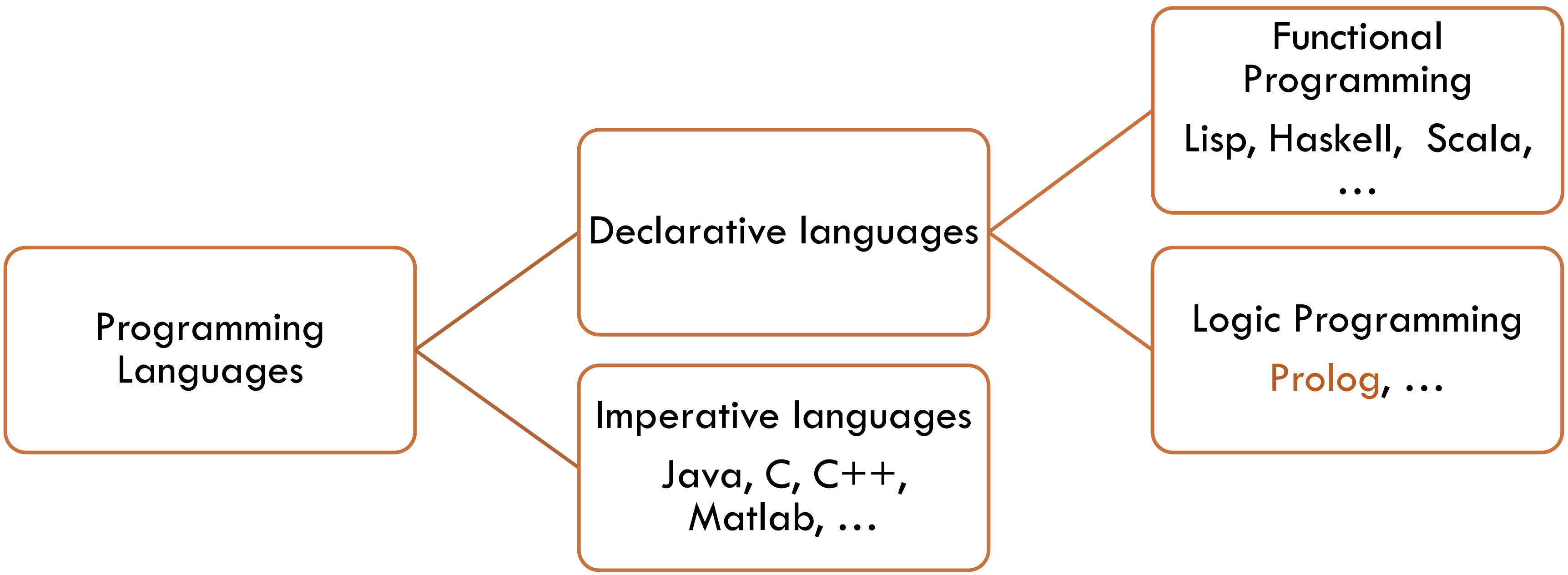
Text Book

- Stuart Russell and Peter Norvig (2010). Artificial Intelligence: A Modern Approach (3nd edition). Prentice Hall.

Course Outline

- Introduction.
- Intelligent agent; defining agents and distinguishing them from programs in general.
- Problem-solving by searching, knowledge representation; types of reasoning; logic languages; inferences in first-order logic.
- Applications

Implementation: Prolog Programming



Agenda of Lecture 1

- What is AI?
- Intelligent Agents
  - Agents and environments
  - Rationality
  - Task Environment: PEAS (Performance measure, Environment, Actuators, Sensors)
  - Environment types
  - Agent Structure
  - Kinds of Agent Programs

What is AI?

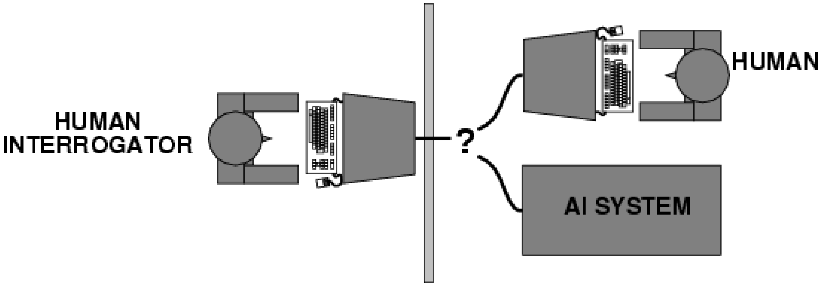
Views of AI fall into four categories:

	Human	Rational
Thought	Thinking Humanly	Thinking Rationally
Behavior	Acting Humanly	Acting Rationally

- The textbook advocates "acting rationally".
- A system is rational if it does the right thing, given what it knows.

Acting humanly: Turing Test

- Turing (1950) in his article "Computing machinery and intelligence":
  - "Can machines think?" → "Can machines behave intelligently?"
  - Operational test for intelligent behavior: the Imitation Game



- A Computer passes the test if a human interrogator, after posing some written questions, cannot tell whether the written response come from a person or a computer.
- Predicted that by 2000, a machine might have a 30% chance of fooling a lay person for 5 minutes.
- Anticipated all major arguments against AI in following 50 years
- Suggested major components of AI: knowledge representation, reasoning, language understanding, learning

Thinking humanly

- Bellman 1978, the automation of activities that we associate with human thinking, activities such as:
  - Decision-making
  - Problem solving
  - learning

Thinking rationally: "laws of thought"

- The Greek philosopher Aristotle provides patterns for argument structures that always yielded correct conclusions when given correct premises—for example, "Socrates is a man; all men are mortal; therefore, Socrates is mortal."
- These laws of thought were LOGIC supposed to govern the operation of the mind; their study initiated the field called **logic**.
- Direct line through mathematics and philosophy to modern AI

Problems of the "laws of thought"

1. It is not easy to take informal knowledge and state it in the formal terms required by logical notation, particularly when the knowledge is less than 100% certain.
2. There is a big difference between solving a problem “in principle” and solving it in practice. Even problems with just a few hundred facts can exhaust the computational resources of any computer unless it has some guidance as to which reasoning steps to try first.

Acting rationally: rational agent

- Agent: something that can act.
- Rational behavior: doing the right thing.

- The right thing: that which is expected to maximize goal achievement, given the available information
- A rational agent is one that acts so as to achieve the best outcome or, when there is uncertainty, the best expected outcome.
- Doesn't necessarily involve thinking – e.g., blinking reflex – but thinking should be in the service of rational action.

Programs and Agents

- Programs: executes some instructions to do something.
- Computer Agents: operate autonomously, perceive their environment, persist over a prolonged time period, adapt to change, and create and pursue goals.

Advantages of rational-agent approach

- It is more general than the "laws of thought" approach because correct inference is just one of several possible mechanisms for achieving rationality.
- Its is more amenable to scientific development than other approaches based on human behavior or human thought.

Intelligent Agents

- An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators.
- Examples

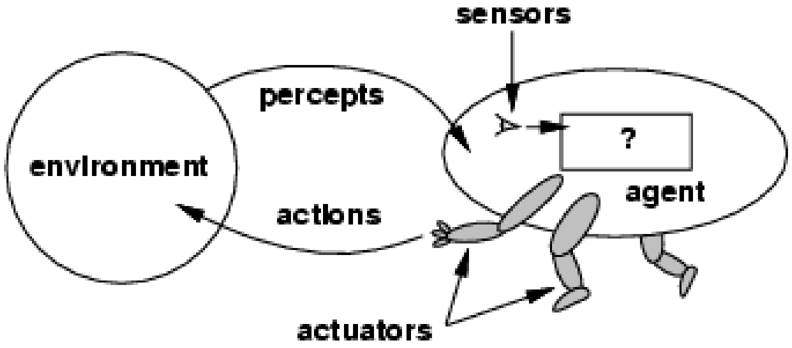
1. Human agent:
   - Sensors: eyes, ears, and other organs
   - Actuators: hands, legs, mouth, and other body parts
2. Robotic agent:
   - Sensors: cameras and infrared range finders
   - Actuators: various motors for actuators

Intelligent Agents (Cont.)

- Examples

3. Software agent:
   - receives keystrokes, file contents, and network packets as sensory inputs,
   - acts on the environment by displaying on the screen, writing files, and sending network packets.

Agents and environments

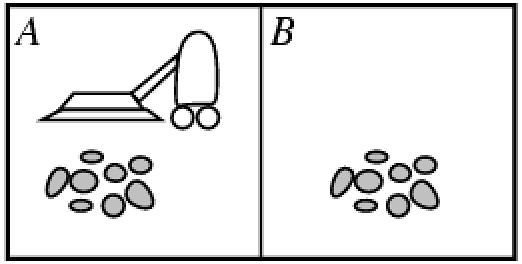


- An agent's behavior can be described by the agent function
- The agent function maps from percept histories to actions:

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

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

Example :Vacuum-cleaner World



- Percepts: location and contents, e.g., [A, Dirty]
- Actions: *Left*, *Right*, *Suck*, *NoOp*

Example (Cont.)

- One very simple agent function is:
  - if the current square is dirty, then suck; otherwise, move to the other square. A partial tabulation of this agent function is:

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

Rationality

- 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.
- Rationality depends on:
  - Performance measure: An objective criterion for success of an agent's behavior
  - The agent’s prior knowledge of the environment.
  - The actions that the agent can perform.
  - The agent’s percept sequence to date.

Example

- The 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.
- The environment: A, B
- Action: *Left*, *Right*, *Suck*, *NoOp*
- Percept sequence: [A, clean], [B, dirty]

Note

- 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.

Autonomous

- Intelligent agents should also be autonomous:
  - Agent's behavior not completely based on built-in knowledge, but also on its own experience (with ability to learn and adapt).
- Provide agent with enough built-in knowledge to get started, and a learning mechanism to allow it to derive knowledge from percepts (and other knowledge).
- For example, a vacuum-cleaning agent that learns to foresee where and when additional dirt will appear will do better than one that does not.

Task environment: PEAS

- PEAS: Performance measure, Environment, Actuators, Sensors
- Must first specify the setting for intelligent agent design
- In designing an agent, the first step must always be to specify the task environment as fully as possible.

- Consider, 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, accelerator, brake, signal, horn, display
  - Sensors: speedometer, GPS, odometer, accelerometer, engine sensors, keyboard

PEAS

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

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

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. A task environment is effectively fully observable if the sensors detect all aspects that are *relevant* to the choice of action; relevance, in turn, depends on the performance measure. For example an automated taxi is partially observable since it cannot see what other drivers are thinking. If the agent has no sensors at all then the environment is unobservable.

- Deterministic (vs. stochastic): The next state of the environment is completely determined by the current state and the action executed by the agent. (If the environment is deterministic except for the actions of other agents, then the environment is stochastic). For example, taxi driving is clearly stochastic in this sense, 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.
- The next episode does not depend on the actions taken in previous episodes. For example, chess and taxi driving are sequential.

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). For example, taxi driving is clearly dynamic.

- Discrete (vs. continuous): A limited number of distinct, clearly defined percepts and actions. 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.

- Single agent (vs. multiagent): An agent operating by itself in an environment. For example, an agent solving a crossword puzzle by itself is in a single-agent environment, whereas an agent playing chess is in a two agent environment.

Environment types

- **Known** vs. **unknown**: This distinction refers not to the environment itself but to the agent's (or designer's) of the environment. In a known environment, the outcomes (or outcome probabilities if the environment is stochastic) for all actions are given. If the environment is unknown, the agent will have to learn how it works in order to make good decisions.
- Note that the distinction between known and unknown environments is not the same as the one between fully and partially observable environments. It is quite possible for a *known* environment to be *partially* observable—for example, in solitaire card games, I know the rules but still unable to see the cards that have not yet been turned over.
- Conversely, an *unknown* environment can be *fully* observable—in a new video game, the screen may show the entire game state but I still don't know what the buttons do until I try them.

Environment types

Examples	Chess with a clock	Chess without a clock	Taxi driving
Fully observable	Yes	Yes	No
Deterministic	Strategic	Strategic	No
Episodic	No	No	No
Static	Semi	Yes	No
Discrete	Yes	Yes	No
Single agent	No	No	No

- The environment type largely determines the agent design
- The real world is (of course) partially observable, stochastic, sequential, dynamic, continuous, multi-agent

Agents Structures

- How the agents works
- The job of AI is to design an agent program that implements the agent function: the mapping from percepts to actions.

Agents Structures

- Agent = Agent program + Architecture.
- Agent program takes the current percept as input from the sensors and return an action to the actuators.
- Architecture is whatever the agent program runs on.
- The architecture has three functions:
  - Maps sensory input to data structures that are made available to the program.
  - Runs the program.
  - Maps program outputs onto signals to the effectors.

Kinds of Agent programs

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

Simple reflex agents

- These agents select actions on the basis of the *current* percept, ignoring the rest of the percept history.
- 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.
- An agent program for this agent is:

```
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
```

Goal-based agents

- Knowing something about the current state of the environment is not always enough to decide what to do.
- For example, at a road junction, the taxi can turn left, turn right, or go straight on. The correct decision depends on where the taxi is trying to get to (the goal).
- In other words, as well as a 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.

Summary

- Agent, agent function, agent program, rationality, autonomy, reflex agent, goal-based agent.
- Task environment: PEAS
- Environment types
- Kinds of Agent programs