Intelligent Agent

Dr. Subarna Chatterjee

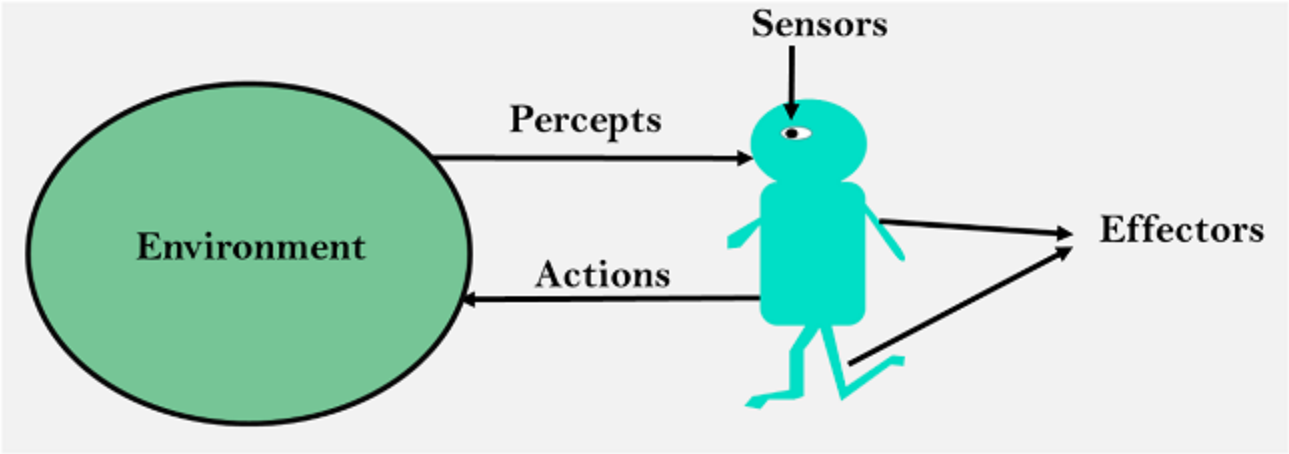
subarna.cs.et@msruas.ac.in

Outline

- Introduction to Agents
- Types of Agents
- Intelligent Agent
- Agent Environment
- Turing Test in AI

Introduction to Agents

- An agent acts in an environment.



- An agent perceives(see, hear, take i/p) its environment through sensors. The complete set of inputs at a given time is called a **percept**. The current **percept**, or a sequence of percepts can influence the actions of an agent.

- The agent can change the environment through actuators or effectors. An operation involving an effector is called an **action**. **Actions** can be grouped into action sequences.

Cont..

- The agent can do the following things:-
  - Operate in an environment
  - Perceives the environment through its sensors
  - Acts upon the environment through actuators or effectors
  - Have goals(objectives that the agent has to satisfy)

Cont..

- The agent perceives its environment through sensors.
  - The complete set of inputs at a given time is called a percept
  - The current percept, or a sequence of percepts can influence the actions of an agent.

- It can change the environment through actuators or effectors.
  - An operation involving an actuator or effector is called an **action**.
  - **Actions** can be grouped into action sequences.

Cont..

1. Have actuators and sensors
2. The agent can have goals which it tries to achieve.
3. Thus, an agent can be looked upon as a system that
   - Implements a mapping from percept sequences to actions.
4. A performance measure has to be used to evaluate an agent.
   - An **autonomous agent** decides autonomously which action to take in the current situation to maximize progress towards its goals.

Cont..

- Behavior and performance of IA in terms of agent function
  - **Performance history** (sequence) to action mapping
  - **Ideal Mapping**: specifies which actions an agent O/P to take at any point in time
- **Performance measure:** a subjective measure to characterize how successful an agent is (e.g. speed, power usage, accuracy, money etc).

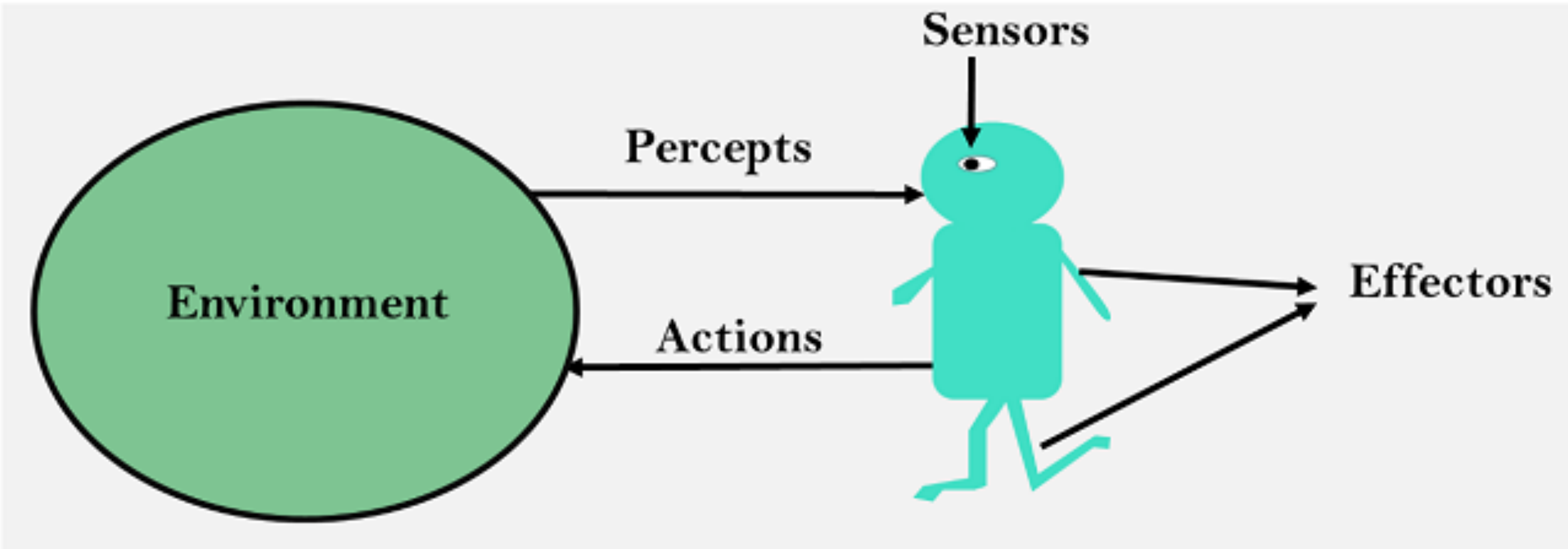
Intelligent Agent

- **Agents in AI:** An AI system can be defined as the study of the rational agent and its environment. The agents sense the environment through sensors and act on their environment through actuators. An AI agent can have mental properties such as knowledge, belief, intention, etc.
- **What is an Agent?**
  - An agent can be anything that perceiveits environment through sensors and act upon that environment through actuators. An Agent runs in the cycle of **perceiving**, **thinking**, and **acting**. An agent can be:
  - **Human-Agent:** A human agent has eyes, ears, and other organs which work for sensors and hand, legs, vocal tract work for actuators.
  - **Robotic Agent:** A robotic agent can have cameras, infrared range finder, NLP for sensors and various motors for actuators.
  - **Software Agent:** Software agent can have keystrokes, file contents as sensory input and act on those inputs and display output on the screen.
  - Hence the world around us is full of agents such as thermostat, cellphone, camera, and even we are also agents.

Cont..

- Before moving forward, we should first know about sensors, effectors, and actuators.
- **Sensor:** Sensor is a device which detects the change in the environment and sends the information to other electronic devices. An agent observes its environment through sensors.
- **Actuators:** Actuators are the component of machines that converts energy into motion. The actuators are only responsible for moving and controlling a system. An actuator can be an electric motor, gears, rails, etc.
- **Effectors:** Effectors are the devices which affect the environment. Effectors can be legs, wheels, arms, fingers, wings, fins, and display screen.

Cont..



Cont..

- **Intelligent Agents:**
  - An intelligent agent is an autonomous entity which act upon an environment using sensors and actuators for achieving goals. An intelligent agent may learn from the environment to achieve their goals. A thermostat is an example of an intelligent agent.
- Following are the main four rules for an AI agent:
  - **Rule 1:** An AI agent must have the ability to perceive the environment.
  - **Rule 2:** The observation must be used to make decisions.
  - **Rule 3:** Decision should result in an action.
  - **Rule 4:** The action taken by an AI agent must be a rational action.

Cont..

- **Rational Agent:**
- A rational agent is an agent which has clear preference, models uncertainty, and acts in a way to maximize its performance measure with all possible actions.
- A rational agent is said to perform the right things. AI is about creating rational agents to use for game theory and decision theory for various real-world scenarios.
- For an AI agent, the rational action is most important because in AI reinforcement learning algorithm, for each best possible action, agent gets the positive reward and for each wrong action, an agent gets a negative reward.
- Note: Rational agents in AI are very similar to intelligent agents.

Cont..

- **Rationality:**
- The rationality of an agent is measured by its performance measure. Rationality can be judged on the basis of following points:
- Performance measure which defines the success criterion.
- Agent prior knowledge of its environment.
- Best possible actions that an agent can perform.
- The sequence of percepts.

- Note: Rationality differs from Omniscience because an Omniscient agent knows the actual outcome of its action and act accordingly, which is not possible in reality.

Cont..

- **Structure of an AI Agent**
  - The task of AI is to design an agent program which implements the agent function. The structure of an intelligent agent is a combination of architecture and agent program. It can be viewed as:

$$Agent \ = \ Architecture \ + \ Agent\ program$$

- Following are the main three terms involved in the structure of an AI agent:
  - **Architecture:** Architecture is machinery that an AI agent executes on.
  - **Agent Function:** Agent function is used to map a percept to an action.

$$f: P* \rightarrow \ A$$

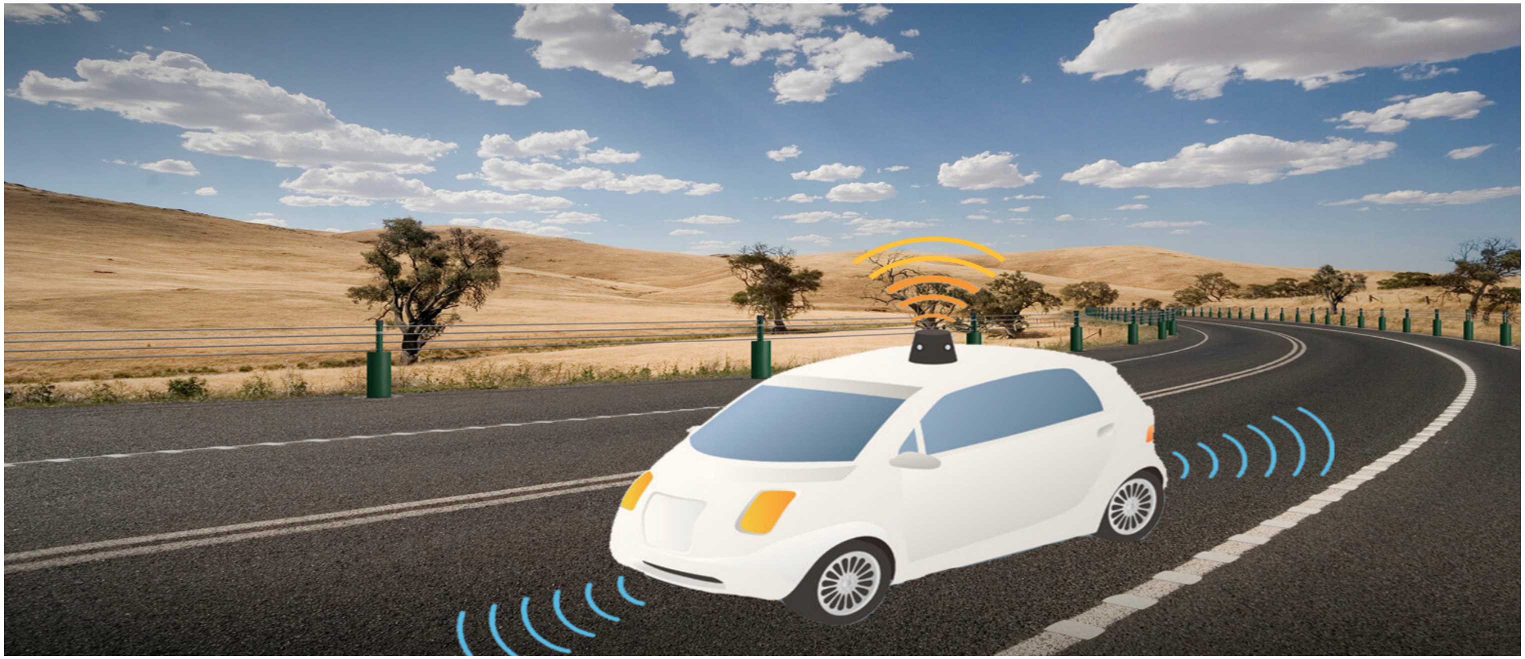
- **Agent program:** Agent program is an implementation of agent function. An agent program executes on the physical architecture to produce function f.

Cont..

- **PEAS Representation**
- PEAS is a type of model on which an AI agent works upon. When we define an AI agent or rational agent, then we can group its properties under PEAS representation model. It is made up of four words:
  - **P:** Performance measure
  - **E:** Environment
  - **A:** Actuators
  - **S:** Sensors
- Here performance measure is the objective for the success of an agent's behavior.

Cont..

- **PEAS for self-driving cars:** Let's suppose a self-driving car then PEAS representation will be:
  - **Performance:** Safety, time, legal drive, comfort
  - **Environment:** Roads, other vehicles, road signs, pedestrian
  - **Actuators:** Steering, accelerator, brake, signal, horn
  - **Sensors:** Camera, GPS, speedometer, odometer, accelerometer, sonar.



Cont..

- **PEAS for Taxi-driver :** Let's suppose a Taxi-driver then PEAS representation will be:
  - **Performance:** safe, fast, comfortable (maximize profits)
  - **Environment:** roads, other traffic, pedestrians, customers
  - **Actuators:** steering, accelerator, brake, signal, horn
  - **Sensors:** cameras, sonar, speedometer, GPS, odometer, accelerometer, engine sensors

Environment Properties

- Fully observable vs. partially observable
- Deterministic vs. stochastic / strategic
- Episodic vs. sequential
- Static vs. dynamic
- Discrete vs. continuous
- Single agent vs. multiagent

Environment Examples

Environment	Obser vable	Determi nistic	Episodic	Static	Discrete	Agents
Chess with a clock						
Chess without a clock						

Fully observable vs. partially observable
Deterministic vs. stochastic / strategic
Episodic vs. sequential
Static vs. dynamic
Discrete vs. continuous
Single agent vs. multiagent

Environment Examples

Environment	Observable	Deterministic	Episodic	Static	Discrete	Agents
Chess with a clock	Fully	Strategic	Sequential	Semi	Discrete	Multi
Chess without a clock	Fully	Strategic	Sequential	Static	Discrete	Multi

Fully observable vs. partially observable
Deterministic vs. stochastic / strategic
Episodic vs. sequential
Static vs. dynamic
Discrete vs. continuous
Single agent vs. multiagent

Environment Examples

Environment	Observable	Deterministic	Episodic	Static	Discrete	Agents
Chess with a clock	Fully	Strategic	Sequential	Semi	Discrete	Multi
Chess without a clock	Fully	Strategic	Sequential	Static	Discrete	Multi
Poker						

Fully observable vs. partially observable
Deterministic vs. stochastic / strategic
Episodic vs. sequential
Static vs. dynamic
Discrete vs. continuous
Single agent vs. multiagent

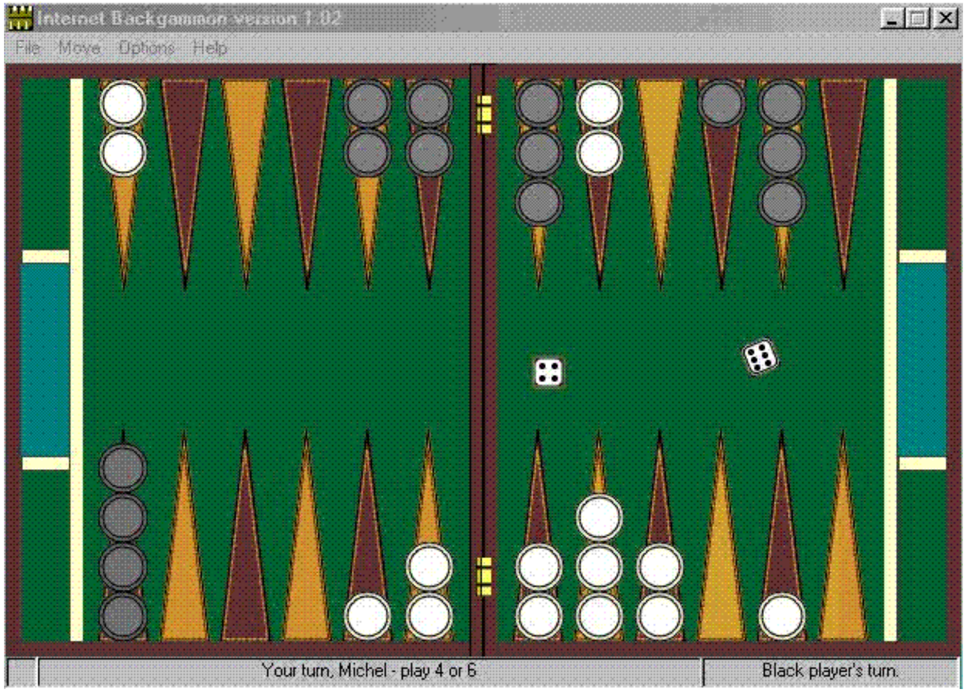
Environment Examples

Environment	Observable	Deterministic	Episodic	Static	Discrete	Agents
Chess with a clock	Fully	Strategic	Sequential	Semi	Discrete	Multi
Chess without a clock	Fully	Strategic	Sequential	Static	Discrete	Multi
Poker	Partial	Strategic	Sequential	Static	Discrete	Multi

Fully observable vs. partially observable
Deterministic vs. stochastic / strategic
Episodic vs. sequential
Static vs. dynamic
Discrete vs. continuous
Single agent vs. multiagent

Environment Examples

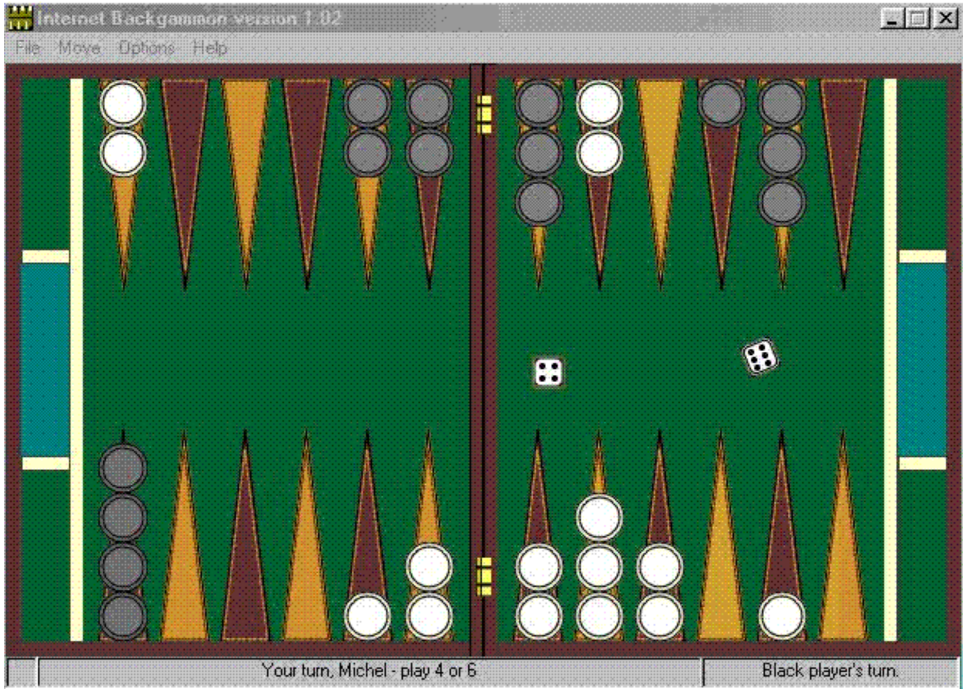
Environment	Observable	Deterministic	Episodic	Static	Discrete	Agents
Chess with a clock	Fully	Strategic	Sequential	Semi	Discrete	Multi
Chess without a clock	Fully	Strategic	Sequential	Static	Discrete	Multi
Poker	Partial	Strategic	Sequential	Static	Discrete	Multi
Backgammon						



Fully observable vs. partially observable
Deterministic vs. stochastic / strategic
Episodic vs. sequential
Static vs. dynamic
Discrete vs. continuous
Single agent vs. multiagent

Environment Examples

Environment	Observable	Deterministic	Episodic	Static	Discrete	Agents
Chess with a clock	Fully	Strategic	Sequential	Semi	Discrete	Multi
Chess without a clock	Fully	Strategic	Sequential	Static	Discrete	Multi
Poker	Partial	Strategic	Sequential	Static	Discrete	Multi
Backgammon	Fully	Stochastic	Sequential	Static	Discrete	Multi



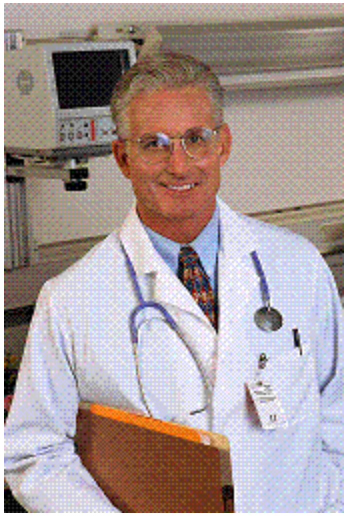
Fully observable vs. partially observable
Deterministic vs. stochastic / strategic
Episodic vs. sequential
Static vs. dynamic
Discrete vs. continuous
Single agent vs. multiagent

Environment Examples

Environment	Observable	Deterministic	Episodic	Static	Discrete	Agents
Chess with a clock	Fully	Strategic	Sequential	Semi	Discrete	Multi
Chess without a clock	Fully	Strategic	Sequential	Static	Discrete	Multi
Poker	Partial	Strategic	Sequential	Static	Discrete	Multi
Backgammon	Fully	Stochastic	Sequential	Static	Discrete	Multi
Taxi driving	Partial	Stochastic	Sequential	Dynamic	Continuous	Multi

Fully observable vs. partially observable
Deterministic vs. stochastic / strategic
Episodic vs. sequential
Static vs. dynamic
Discrete vs. continuous
Single agent vs. multiagent

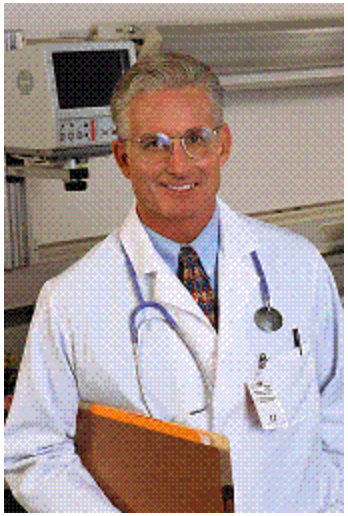
Environment Examples



Environment	Observable	Deterministic	Episodic	Static	Discrete	Agents
Chess with a clock	Fully	Strategic	Sequential	Semi	Discrete	Multi
Chess without a clock	Fully	Strategic	Sequential	Static	Discrete	Multi
Poker	Partial	Strategic	Sequential	Static	Discrete	Multi
Backgammon	Fully	Stochastic	Sequential	Static	Discrete	Multi
Taxi driving	Partial	Stochastic	Sequential	Dynamic	Continuous	Multi
Medical diagnosis						

Fully observable vs. partially observable
Deterministic vs. stochastic / strategic
Episodic vs. sequential
Static vs. dynamic
Discrete vs. continuous
Single agent vs. multiagent

Environment Examples



Environment	Observable	Deterministic	Episodic	Static	Discrete	Agents
Chess with a clock	Fully	Strategic	Sequential	Semi	Discrete	Multi
Chess without a clock	Fully	Strategic	Sequential	Static	Discrete	Multi
Poker	Partial	Strategic	Sequential	Static	Discrete	Multi
Backgammon	Fully	Stochastic	Sequential	Static	Discrete	Multi
Taxi driving	Partial	Stochastic	Sequential	Dynamic	Continuous	Multi
Medical diagnosis	Partial	Stochastic	Episodic	Static	Continuous	Single

Fully observable vs. partially observable
Deterministic vs. stochastic / strategic
Episodic vs. sequential
Static vs. dynamic
Discrete vs. continuous
Single agent vs. multiagent

Environment Examples

Environment	Observable	Deterministic	Episodic	Static	Discrete	Agents
Chess with a clock	Fully	Strategic	Sequential	Semi	Discrete	Multi
Chess without a clock	Fully	Strategic	Sequential	Static	Discrete	Multi
Poker	Partial	Strategic	Sequential	Static	Discrete	Multi
Backgammon	Fully	Stochastic	Sequential	Static	Discrete	Multi
Taxi driving	Partial	Stochastic	Sequential	Dynamic	Continuous	Multi
Medical diagnosis	Partial	Stochastic	Episodic	Static	Continuous	Single
Image analysis						

Fully observable vs. partially observable
Deterministic vs. stochastic / strategic
Episodic vs. sequential
Static vs. dynamic
Discrete vs. continuous
Single agent vs. multiagent

Environment Examples

Environment	Observable	Deterministic	Episodic	Static	Discrete	Agents
Chess with a clock	Fully	Strategic	Sequential	Semi	Discrete	Multi
Chess without a clock	Fully	Strategic	Sequential	Static	Discrete	Multi
Poker	Partial	Strategic	Sequential	Static	Discrete	Multi
Backgammon	Fully	Stochastic	Sequential	Static	Discrete	Multi
Taxi driving	Partial	Stochastic	Sequential	Dynamic	Continuous	Multi
Medical diagnosis	Partial	Stochastic	Episodic	Static	Continuous	Single
Image analysis	Fully	Deterministic	Episodic	Semi	Discrete	Single

Fully observable vs. partially observable

Deterministic vs. stochastic / strategic

Episodic vs. sequential

Static vs. dynamic

Discrete vs. continuous

Single agent vs. multiagent

Environment Examples

Environment	Observable	Deterministic	Episodic	Static	Discrete	Agents
Chess with a clock	Fully	Strategic	Sequential	Semi	Discrete	Multi
Chess without a clock	Fully	Strategic	Sequential	Static	Discrete	Multi
Poker	Partial	Strategic	Sequential	Static	Discrete	Multi
Backgammon	Fully	Stochastic	Sequential	Static	Discrete	Multi
Taxi driving	Partial	Stochastic	Sequential	Dynamic	Continuous	Multi
Medical diagnosis	Partial	Stochastic	Episodic	Static	Continuous	Single
Image analysis	Fully	Deterministic	Episodic	Semi	Discrete	Single
Robot part picking						

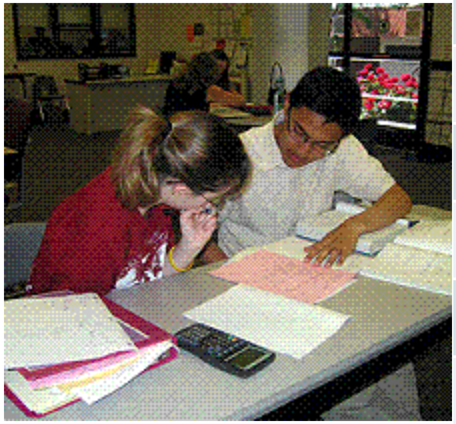
Fully observable vs. partially observable
Deterministic vs. stochastic / strategic
Episodic vs. sequential
Static vs. dynamic
Discrete vs. continuous
Single agent vs. multiagent

Environment Examples

Environment	Observable	Deterministic	Episodic	Static	Discrete	Agents
Chess with a clock	Fully	Strategic	Sequential	Semi	Discrete	Multi
Chess without a clock	Fully	Strategic	Sequential	Static	Discrete	Multi
Poker	Partial	Strategic	Sequential	Static	Discrete	Multi
Backgammon	Fully	Stochastic	Sequential	Static	Discrete	Multi
Taxi driving	Partial	Stochastic	Sequential	Dynamic	Continuous	Multi
Medical diagnosis	Partial	Stochastic	Episodic	Static	Continuous	Single
Image analysis	Fully	Deterministic	Episodic	Semi	Discrete	Single
Robot part picking	Fully	Deterministic	Episodic	Semi	Discrete	Single

Fully observable vs. partially observable
Deterministic vs. stochastic / strategic
Episodic vs. sequential
Static vs. dynamic
Discrete vs. continuous
Single agent vs. multiagent

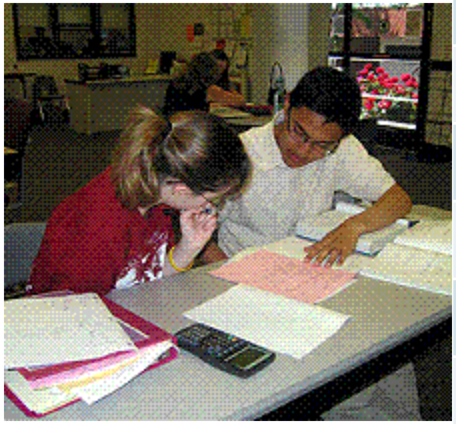
Environment Examples



Environment	Observable	Deterministic	Episodic	Static	Discrete	Agents
Chess with a clock	Fully	Strategic	Sequential	Semi	Discrete	Multi
Chess without a clock	Fully	Strategic	Sequential	Static	Discrete	Multi
Poker	Partial	Strategic	Sequential	Static	Discrete	Multi
Backgammon	Fully	Stochastic	Sequential	Static	Discrete	Multi
Taxi driving	Partial	Stochastic	Sequential	Dynamic	Continuous	Multi
Medical diagnosis	Partial	Stochastic	Episodic	Static	Continuous	Single
Image analysis	Fully	Deterministic	Episodic	Semi	Discrete	Single
Robot part picking	Fully	Deterministic	Episodic	Semi	Discrete	Single
Interactive English tutor						

Fully observable vs. partially observable
Deterministic vs. stochastic / strategic
Episodic vs. sequential
Static vs. dynamic
Discrete vs. continuous
Single agent vs. multiagent

Environment Examples



Environment	Observable	Deterministic	Episodic	Static	Discrete	Agents
Chess with a clock	Fully	Strategic	Sequential	Semi	Discrete	Multi
Chess without a clock	Fully	Strategic	Sequential	Static	Discrete	Multi
Poker	Partial	Strategic	Sequential	Static	Discrete	Multi
Backgammon	Fully	Stochastic	Sequential	Static	Discrete	Multi
Taxi driving	Partial	Stochastic	Sequential	Dynamic	Continuous	Multi
Medical diagnosis	Partial	Stochastic	Episodic	Static	Continuous	Single
Image analysis	Fully	Deterministic	Episodic	Semi	Discrete	Single
Robot part picking	Fully	Deterministic	Episodic	Semi	Discrete	Single
Interactive English tutor	Partial	Stochastic	Sequential	Dynamic	Discrete	Multi

Fully observable vs. partially observable
Deterministic vs. stochastic / strategic
Episodic vs. sequential
Static vs. dynamic
Discrete vs. continuous
Single agent vs. multiagent

Example of Agents with their PEAS representation

Agent	Performance measure	Environment	Actuators	Sensors
Medical Diagnose	•Healthy patient •Minimized cost	•Patient •Hospital •Staff	•Tests •Treatments	Keyboard (Entry of symptoms)
Vacuum Cleaner	•Cleanness •Efficiency •Battery life •Security	•Room •Table •Wood floor •Carpet •Various obstacles	•Wheels •Brushes •Vacuum Extractor	•Camera •Dirt detection sensor •Cliff sensor •Bump Sensor •Infrared Wall Sensor
Part -picking Robot	•Percentage of parts in correct bins.	•Conveyor belt with parts, •Bins	•Jointed Arms •Hand	•Camera •Joint angle sensors.

Agent Environment

- An environment is everything in the world which surrounds the agent, but it is not a part of an agent itself.
- It can be described as a situation in which an agent is present.
- It is where agent lives, operate and provide the agent with something to sense and act upon it.
- An environment is mostly said to be non-feministic.

Cont..

- **Features of Environment**
- As per Russell and Norvig, an environment can have various features from the point of view of an agent:
  1. Fully observable vs Partially Observable
  2. Static vs Dynamic
  3. Discrete vs Continuous
  4. Deterministic vs Stochastic
  5. Single-agent vs Multi-agent
  6. Episodic vs sequential
  7. Known vs Unknown
  8. Accessible vs Inaccessible

Cont..

1. Fully observable vs Partially Observable:

- If an agent sensor can sense or access the complete state of an environment at each point of time then it is **a fully observable** environment, else it is **partially observable**.
- A fully observable environment is easy as there is no need to maintain the internal state to keep track history of the world.
- An agent with no sensors in all environments then such an environment is called as **unobservable**.

2. Deterministic vs Stochastic:

- If an agent's current state and selected action can completely determine the next state of the environment, then such environment is called a deterministic environment.
- A stochastic environment is random in nature and cannot be determined completely by an agent.
- In a deterministic, fully observable environment, agent does not need to worry about uncertainty.

Cont..

3. Episodic vs Sequential:

- In an episodic environment, there is a series of one-shot actions, and only the current percept is required for the action.
- However, in Sequential environment, an agent requires memory of past actions to determine the next best actions.

4. Single-agent vs Multi-agent

- If only one agent is involved in an environment, and operating by itself then such an environment is called single agent environment.
- However, if multiple agents are operating in an environment, then such an environment is called a multi-agent environment.
- The agent design problems in the multi-agent environment are different from single agent environment.

Cont..

5. Static vs Dynamic:

- If the environment can change itself while an agent is deliberating then such environment is called a dynamic environment else it is called a static environment.
- Static environments are easy to deal because an agent does not need to continue looking at the world while deciding for an action.
- However for dynamic environment, agents need to keep looking at the world at each action.
- Taxi driving is an example of a dynamic environment whereas Crossword puzzles are an example of a static environment.

6. Discrete vs Continuous:

- If in an environment there are a finite number of percepts and actions that can be performed within it, then such an environment is called a discrete environment else it is called continuous environment.
- A chess game comes under discrete environment as there is a finite number of moves that can be performed.
- A self-driving car is an example of a continuous environment.

Cont..

7. Known vs Unknown

- Known and unknown are not actually a feature of an environment, but it is an agent's state of knowledge to perform an action.
- In a known environment, the results for all actions are known to the agent. While in unknown environment, agent needs to learn how it works in order to perform an action.
- It is quite possible that a known environment to be partially observable and an Unknown environment to be fully observable.

8. Accessible vs Inaccessible

- If an agent can obtain complete and accurate information about the state's environment, then such an environment is called an Accessible environment else it is called inaccessible.
- An empty room whose state can be defined by its temperature is an example of an accessible environment.
- Information about an event on earth is an example of Inaccessible environment.

Types of AI Agents

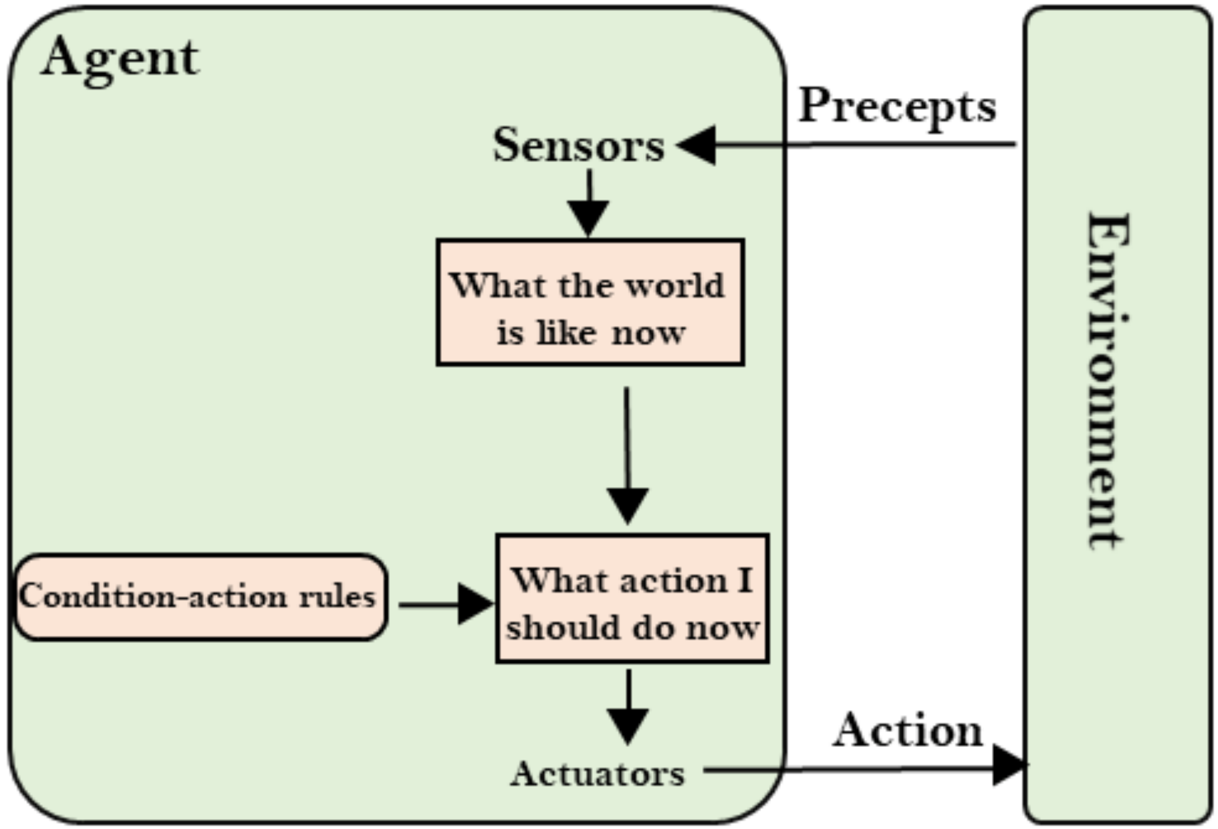
- Agents can be grouped into five classes based on their degree of perceived intelligence and capability. All these agents can improve their performance and generate better action over the time.
- These are given below:
  - Simple Reflex Agent
  - Model-based reflex agent
  - Goal-based agents
  - Utility-based agent
  - Learning agent

1. Simple Reflex agent

- The Simple reflex agents are the simplest agents. These agents take decisions on the basis of the current percepts and ignore the rest of the percept history.
- These agents only succeed in the fully observable environment.
- The Simple reflex agent does not consider any part of percepts history during their decision and action process.
- The Simple reflex agent works on Condition-action rule, which means it maps the current state to action. Such as a Room Cleaner agent, it works only if there is dirt in the room.
- Problems for the simple reflex agent design approach:
  - They have very limited intelligence
  - They do not have knowledge of non-perceptual parts of the current state
  - Mostly too big to generate and to store.
  - Not adaptive to changes in the environment.

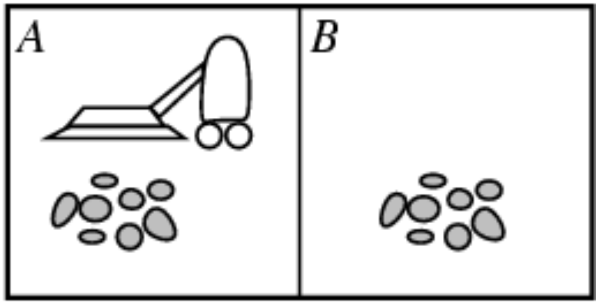
Cont..

- Use simple “if then” rules
- Can be short sighted




```
SimpleReflexAgent(percept)
  state   = InterpretInput(percept)
  rule     = RuleMatch(state, rules)
  action = RuleAction(rule)
  Return action
```

Example: Vacuum Agent



- Performance?
  - 1 point for each square cleaned in time T?
  - #clean squares per time step - #moves per time step?
- Environment: vacuum, dirt, multiple areas defined by square regions
- Actions: left, right, suck, idle
- Sensors: location and contents
  - [A, dirty]

- Rational is not omniscient
  - Environment may be partially observable
- Rational is not clairvoyant
  - Environment may be stochastic
- Thus Rational is not always successful

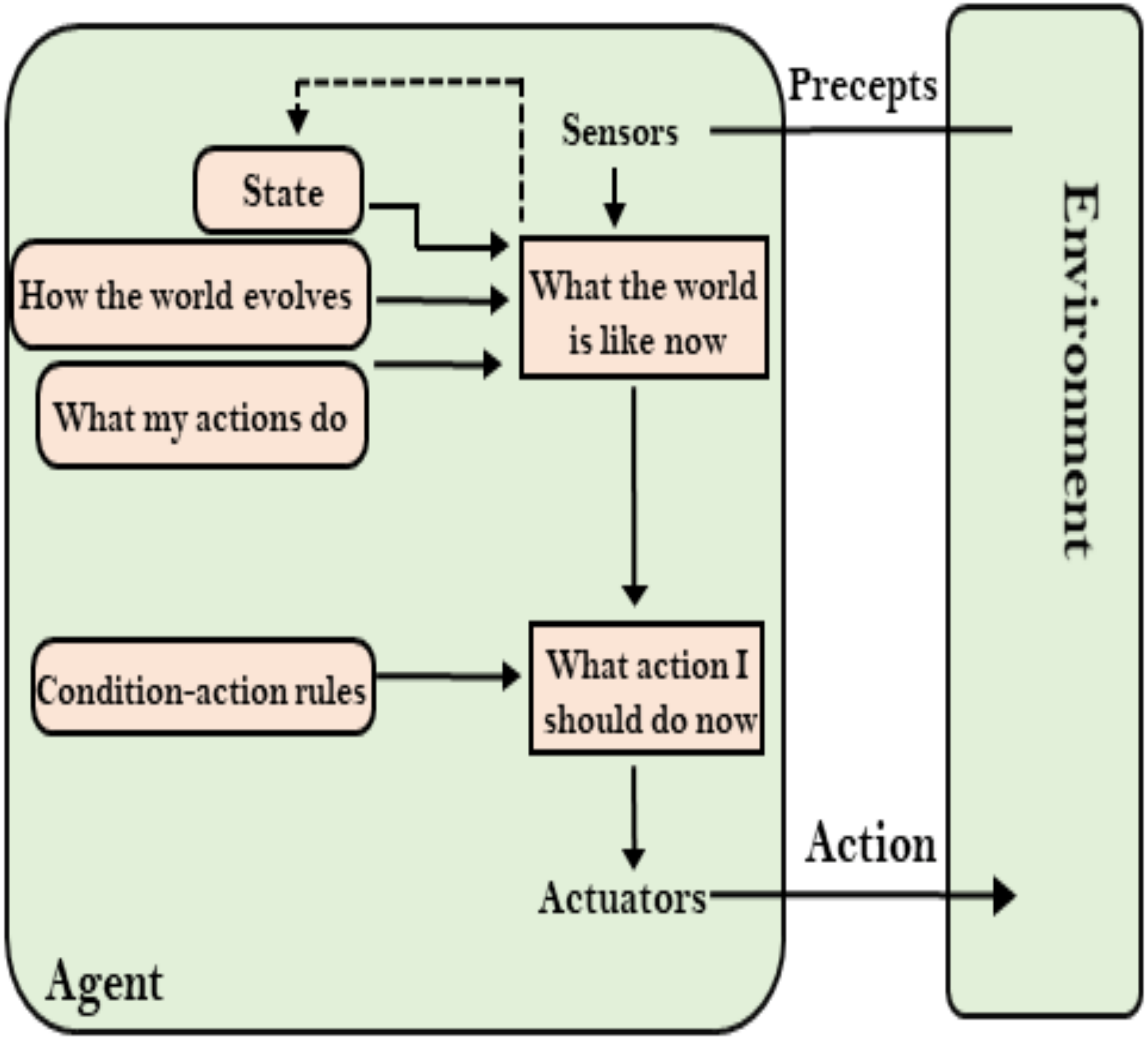
2. Model-based reflex agent

- The Model-based agent can work in a partially observable environment, and track the situation.
- A model-based agent has two important factors:
  - **Model:** It is knowledge about "how things happen in the world," so it is called a Model-based agent.
  - **Internal State:** It is a representation of the current state based on percept history.
- These agents have the model, "which is knowledge of the world" and based on the model they perform actions.
- Updating the agent state requires information about:
  - How the world evolves
  - How the agent's action affects the world.

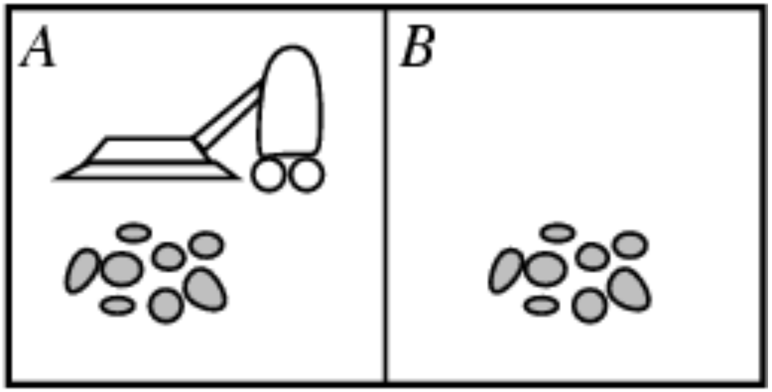
Cont..

- Store previously-observed information
- Can reason about unobserved aspects of current state

```
ReflexAgentWithState(percept)
  state   = UpdateDate(state,action,percept)
  rule    = RuleMatch(state, rules)
  action = RuleAction(rule)
  Return action
```



Reflex Vacuum Agent



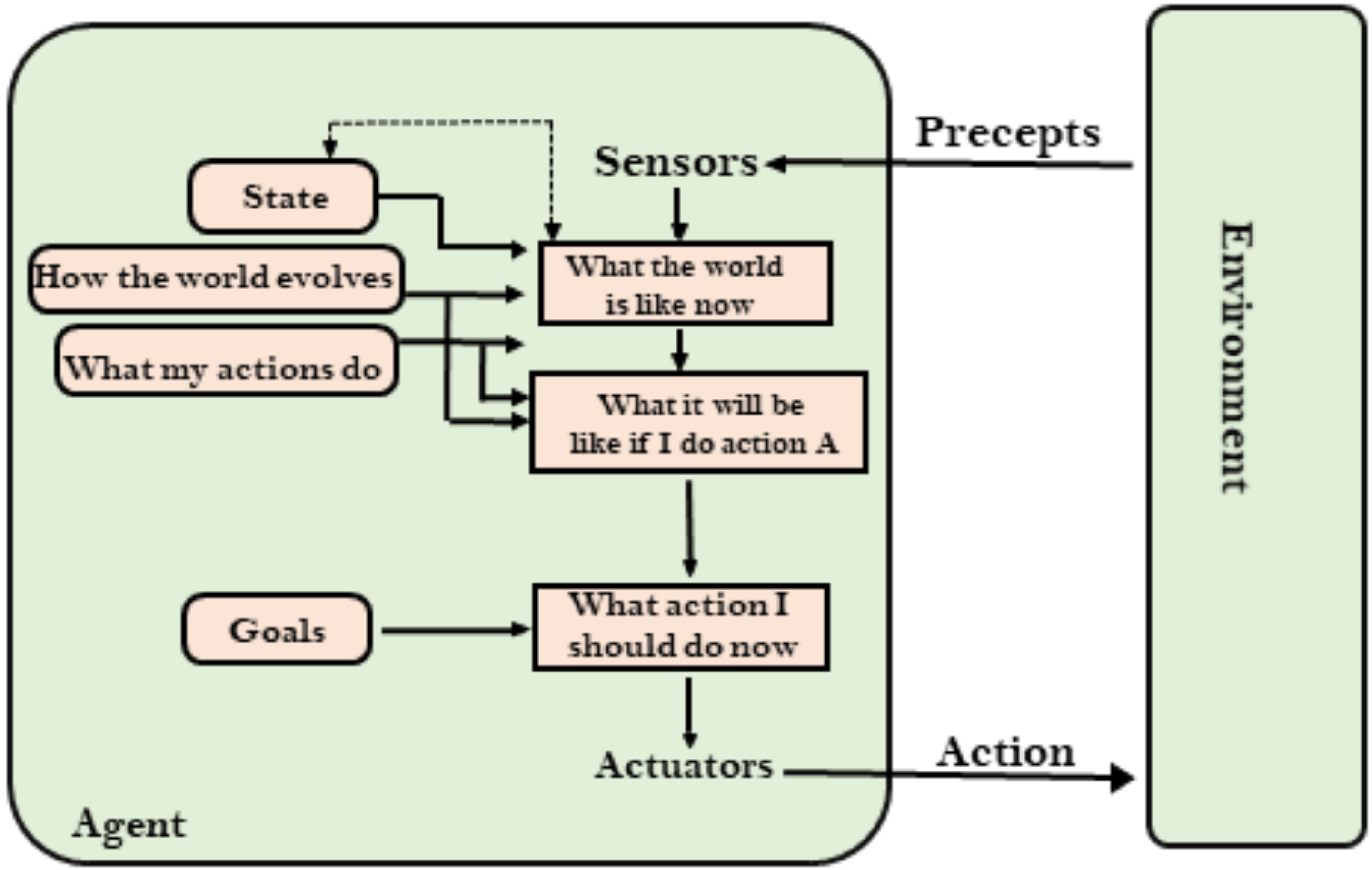
- If status=Dirty then return Suck
  else if location=A then return Right
  else if location=B then right Left
- If status=Dirty then Suck
  else if have not visited other square in >3 time units, go there

3. Goal-based agents

- The knowledge of the current state environment is not always sufficient to decide for an agent to what to do.
- The agent needs to know its goal which describes desirable situations.
- Goal-based agents expand the capabilities of the model-based agent by having the "goal" information.
- They choose an action, so that they can achieve the goal.
- These agents may have to consider a long sequence of possible actions before deciding whether the goal is achieved or not. Such considerations of different scenario are called searching and planning, which makes an agent proactive.

Cont..

- Goal reflects desires of agents
- May project actions to see if consistent with goals
- Takes time, world may change during reasoning

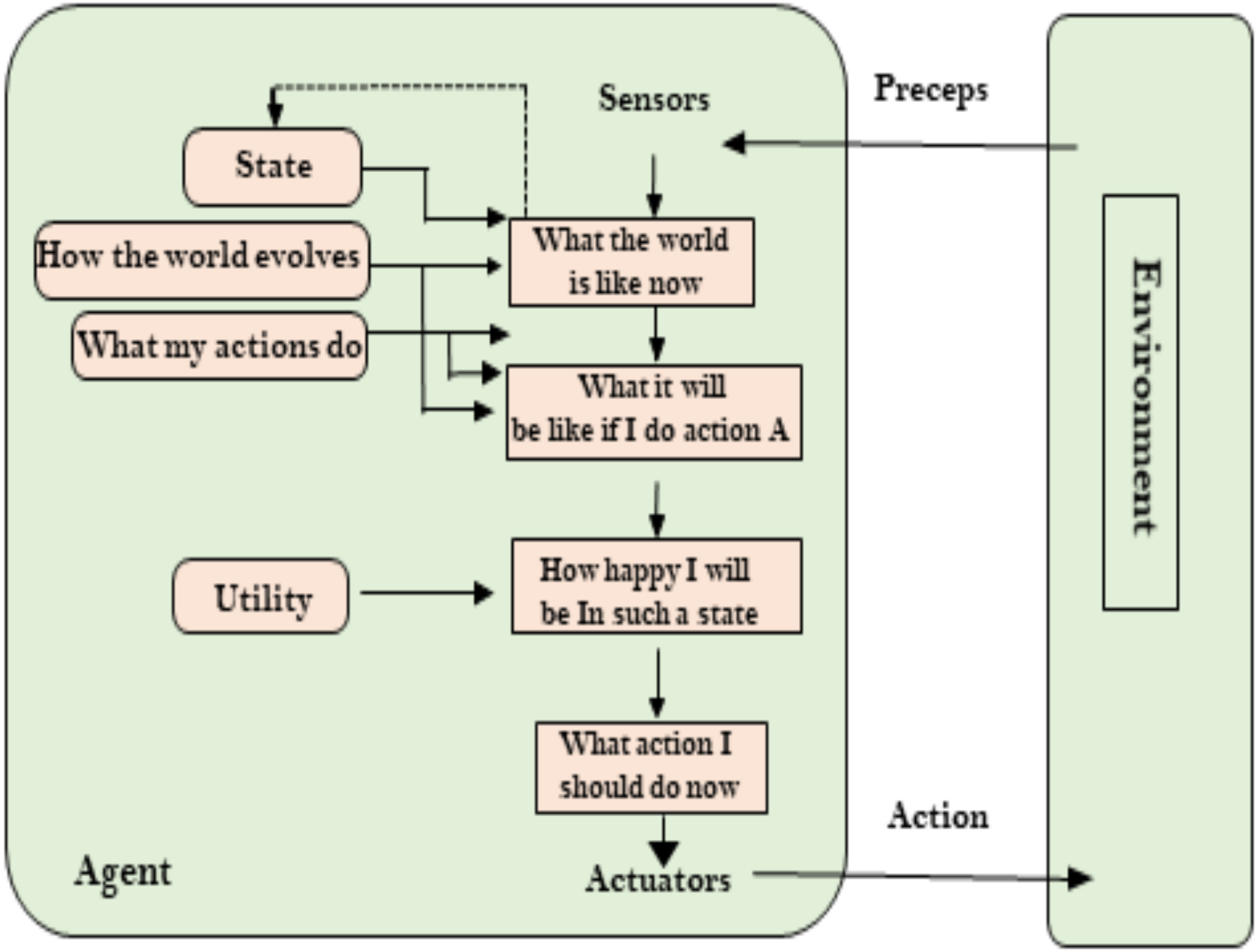


4. Utility-based agents

- These agents are similar to the goal-based agent but provide an extra component of utility measurement which makes them different by providing a measure of success at a given state.
- Utility-based agent act based not only goals but also the best way to achieve the goal.
- The Utility-based agent is useful when there are multiple possible alternatives, and an agent has to choose in order to perform the best action.
- The utility function maps each state to a real number to check how efficiently each action achieves the goals.

Cont..

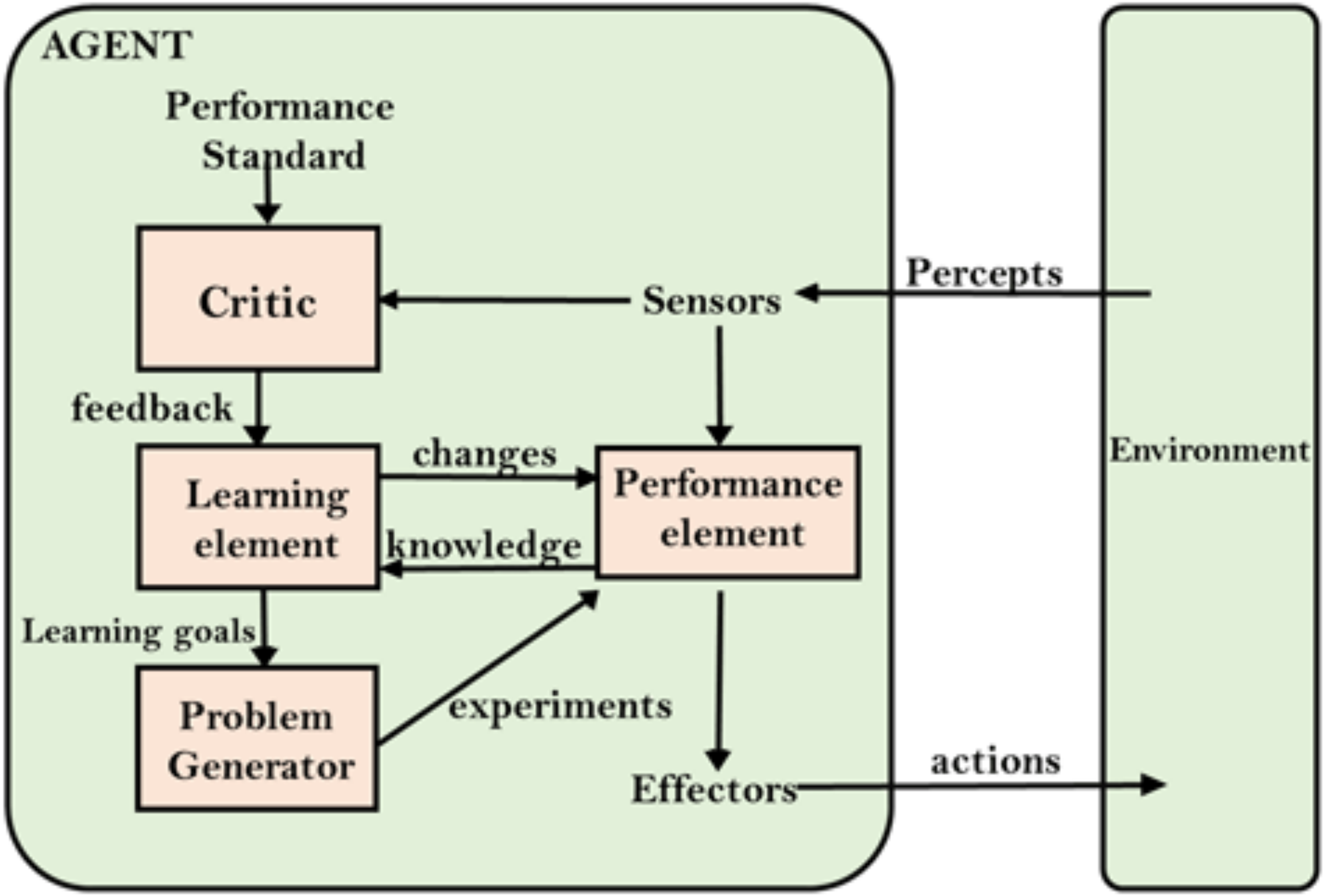
- Evaluation function to measure utility f(state) -> value
- Useful for evaluating competing goals



5. Learning Agents

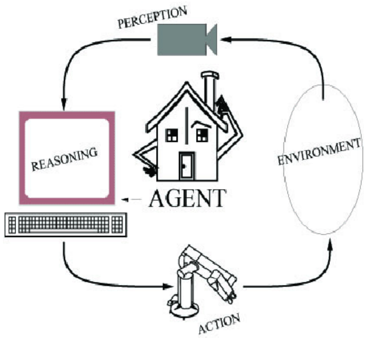
- A learning agent in AI is the type of agent which can learn from its past experiences, or it has learning capabilities.
- It starts to act with basic knowledge and then able to act and adapt automatically through learning.
- A learning agent has mainly four conceptual components, which are:
  - **Learning element:** It is responsible for making improvements by learning from environment
  - **Critic:** Learning element takes feedback from critic which describes that how well the agent is doing with respect to a fixed performance standard.
  - **Performance element:** It is responsible for selecting external action
  - **Problem generator:** This component is responsible for suggesting actions that will lead to new and informative experiences.
- Hence, learning agents are able to learn, analyze performance, and look for new ways to improve the performance.

Cont..


AGENT
Performance Standard
Critic
Sensors
Percepts
feedback
changes
Learning element
knowledge
Performance element
Environment
Learning goals
Problem Generator
experiments
actions
Effectors

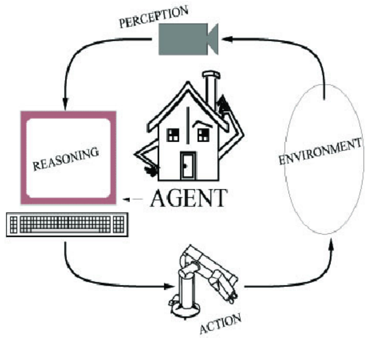
Xavier mail delivery robot

- Performance: Completed tasks
- Environment: [See for yourself]
- Actuators: Wheeled robot actuation
- Sensors: Vision, sonar, dead reckoning
- Reasoning: Markov model induction, A* search, Bayes classification



Pathfinder Medical Diagnosis System

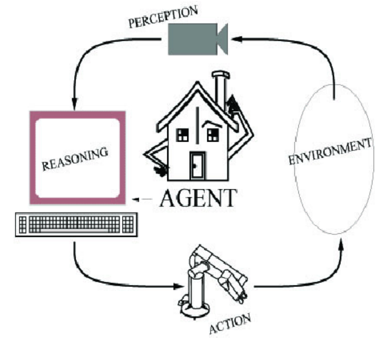
- Performance: Correct [Hematopathology diagnosis]
- Environment: Automate human diagnosis, partially observable, deterministic, episodic, static, continuous, single agent
- Actuators: Output diagnoses and further test suggestions
- Sensors: Input symptoms and test results
- Reasoning: Bayesian networks, Monte-Carlo simulations



TDGammon

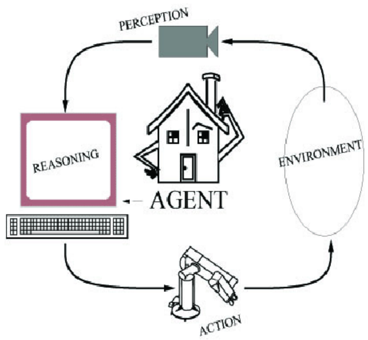
- Performance: Ratio of wins to losses
- Environment: Graphical output showing dice roll and piece movement, fully observable, stochastic, sequential, static, discrete, multiagent

  World Champion Backgammon Player
- Sensors: Keyboard input
- Actuator: Numbers representing moves of pieces
- Reasoning: Reinforcement learning, neural networks



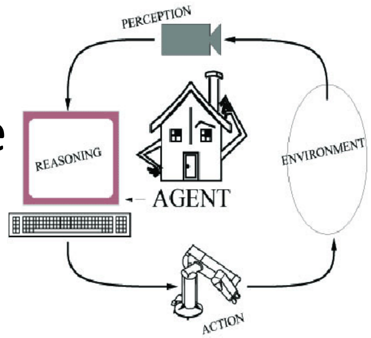
Alvinn

- Performance: Stay in lane, on road, maintain speed
- Environment: Driving Hummer on and off road without manual control (Partially observable, stochastic, episodic, dynamic, continuous, single agent), [Autonomous automobile]
- Actuators: Speed, Steer
- Sensors: Stereo camera input
- Reasoning: Neural networks



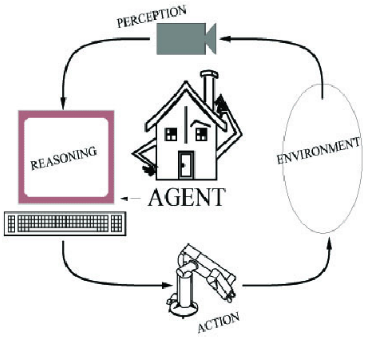
Talespin

- Performance: Entertainment value of generated story
- Environment: Generate text-based stories that are creative and understandable
  - One day Joe Bear was hungry. He asked his friend Irving Bird where some honey was. Irving told him there was a beehive in the oak tree. Joe threatened to hit Irving if he didn't tell him where some honey was.
  - Henry Squirrel was thirsty. He walked over to the river bank where his good friend Bill Bird was sitting. Henry slipped and fell in the river. Gravity drowned. Joe Bear was hungry. He asked Irving Bird where some honey was. Irving refused to tell him, so Joe offered to bring him a worm if he'd tell him where some honey was. Irving agreed. But Joe didn't know where any worms were, so he asked Irving, who refused to say. So Joe offered to bring him a worm if he'd tell him where a worm was. Irving agreed. But Joe didn't know where any worms were, so he asked Irving, who refused to say. So Joe offered to bring him a worm if he'd tell him where a worm was…
- Actuators: Add word/phrase, order parts of story
- Sensors: Dictionary, Facts and relationships stored in database
- Reasoning: Planning



Webcrawler Softbot

- Search web for items of interest
- Perception: Web pages
- Reasoning: Pattern matching
- Action: Select and traverse hyperlinks

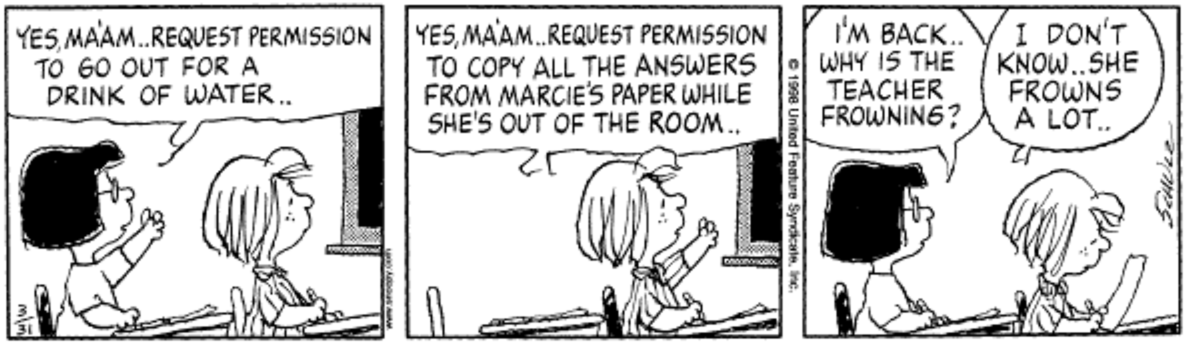


Other Example AI Systems

- Translation of Caterpillar truck manuals into 20 languages
- Shuttle packing
- Military planning (Desert Storm)
- Intelligent vehicle highway negotiation
- Credit card transaction monitoring
- Billiards robot
- Juggling robot
- Credit card fraud detection
- Lymphatic system diagnoses
- Mars rover
- Sky survey galaxy data analysis

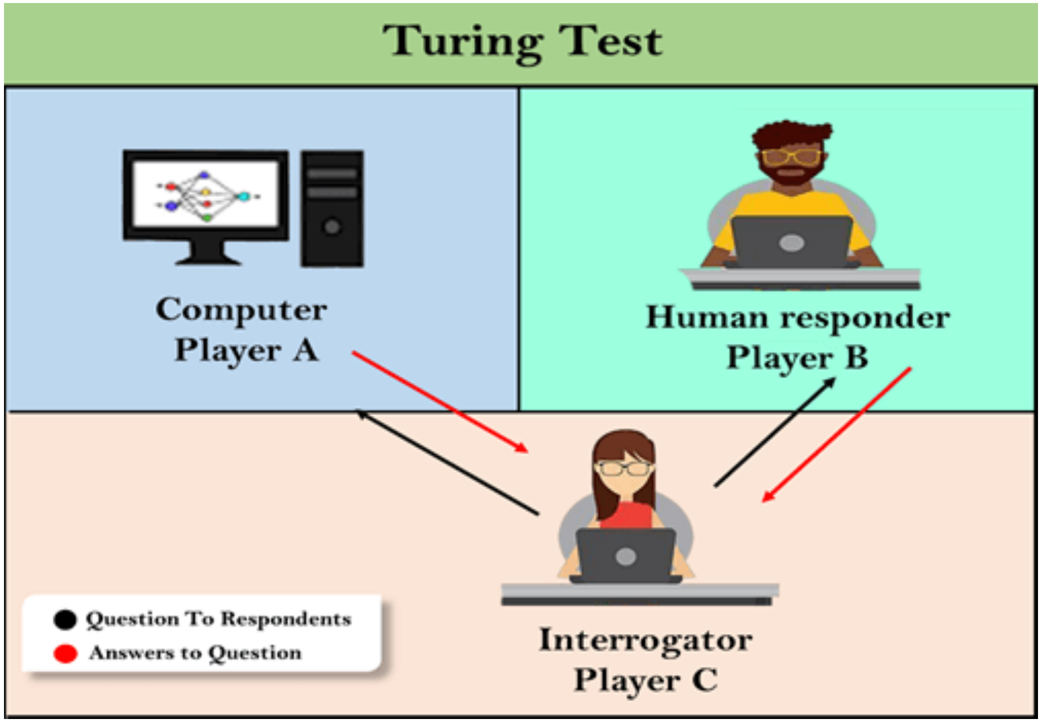
Other Example AI Systems

- Knowledge Representation
- Search
- Problem solving
- Planning
- Machine learning
- Natural language processing
- Uncertainty reasoning
- Computer Vision
- Robotics



Turing Test in AI

- In 1950, Alan Turing introduced a test to check whether a **machine can think like a human or not**, this test is known as the Turing Test. In this test, Turing proposed that the **computer can be said to be an intelligent if it can mimic human response under specific conditions**.
- Turing Test was introduced by Turing in his 1950 paper, "Computing Machinery and Intelligence," which considered the question, "**Can Machine think?**"



Cont..

- The Turing test is based on a party game "**Imitation game**," with some modifications. This game involves three players in which one player is Computer, another player is human responder, and the third player is a human Interrogator, who is isolated from other two players and his job is to find that which player is machine among two of them.
- Consider, Player A is a computer, Player B is human, and Player C is an interrogator. Interrogator is aware that one of them is machine, but he needs to identify this on the basis of questions and their responses.
- The conversation between all players is via keyboard and screen so the result would not depend on the machine's ability to convert words as speech.
- The test result does not depend on each correct answer, but only how closely its responses like a human answer. The computer is permitted to do everything possible to force a wrong identification by the interrogator.

Cont..

- The questions and answers can be like:
  - Interrogator: Are you a computer?
  - PlayerA (Computer): No
  - Interrogator: Multiply two large numbers such as (256896489*456725896)
  - Player A: Long pause and give the wrong answer.
- In this game, if an interrogator would not be able to identify which is a machine and which is human, then the computer passes the test successfully, and the machine is said to be intelligent and can think like a human.
- "In 1991, the New York businessman Hugh Loebner announces the prize competition, offering a $100,000 prize for the first computer to pass the Turing test. However, no AI program to till date, come close to passing an undiluted Turing test".

Chatbots to attempt the Turing test:

- **ELIZA:** ELIZA was a Natural language processing computer program created by Joseph Weizenbaum. It was created to demonstrate the ability of communication between machine and humans. It was one of the first chatterbots, which has attempted the Turing Test.
- **Parry:** Parry was a chatterbot created by Kenneth Colby in 1972. Parry was designed to simulate a person with **Paranoid schizophrenia**(most common chronic mental disorder). Parry was described as "ELIZA with attitude." Parry was tested using a variation of the Turing Test in the early 1970s.
- **Eugene Goostman:** Eugene Goostman was a chatbot developed in Saint Petersburg in 2001. This bot has competed in the various number of Turing Test. In June 2012, at an event, Goostman won the competition promoted as largest-ever Turing test content, in which it has convinced 29% of judges that it was a human.Goostman resembled as a 13-year old virtual boy.

The Chinese Room Argument:

- There were many philosophers who really disagreed with the complete concept of Artificial Intelligence. The most famous argument in this list was "**Chinese Room**."
- In the year **1980, John Searle** presented "**Chinese Room**" thought experiment, in his paper "**Mind, Brains, and Program**," which was against the validity of Turing's Test. According to his argument, "**Programming a computer may make it to understand a language, but it will not produce a real understanding of language or consciousness in a computer**."
- He argued that Machine such as ELIZA and Parry could easily pass the Turing test by manipulating keywords and symbol, but they had no real understanding of language. So it cannot be described as "thinking" capability of a machine such as a human.

Features required for a machine to pass the Turing test

- **Natural language processing:** NLP is required to communicate with Interrogator in general human language like English.
- **Knowledge representation:** To store and retrieve information during the test.
- **Automated reasoning:** To use the previously stored information for answering the questions.
- **Machine learning:** To adapt new changes and can detect generalized patterns.
- **Vision (For total Turing test):** To recognize the interrogator actions and other objects during a test.
- **Motor Control (For total Turing test):** To act upon objects if requested.

Thank You