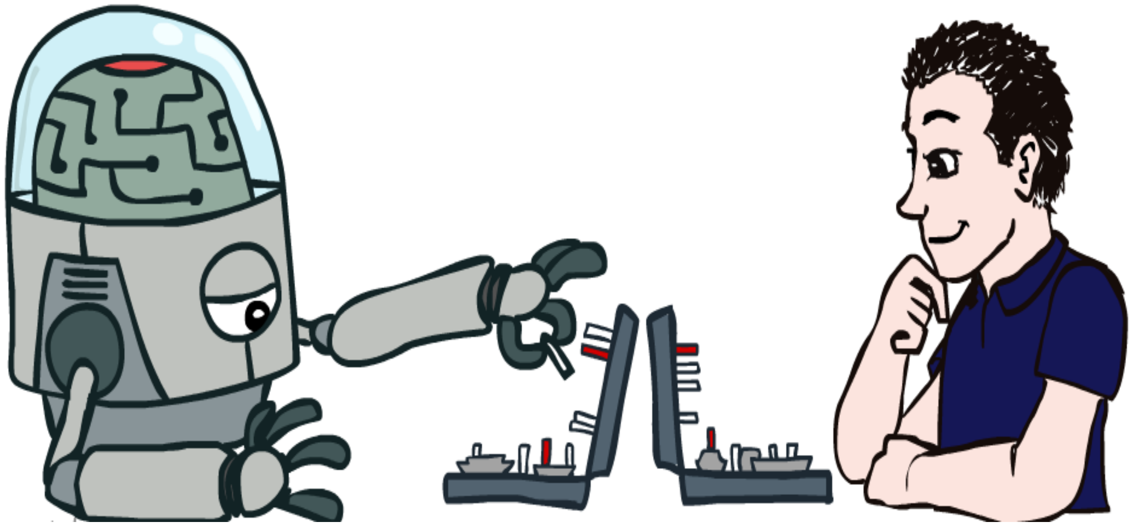
ARTIFICIAL INTELLIGENCE- CS411

Prof. Alaa Sagheer



2022-2023

Artificial Intelligence "CS 411"

- **Textbook**:

S. Russell and P. Norvig

Artificial Intelligence: A Modern Approach

Pearson, 2022, *4 th Edition*

- **Place:** Online Lectures
- **Grading:**

  Class Activity (5%),

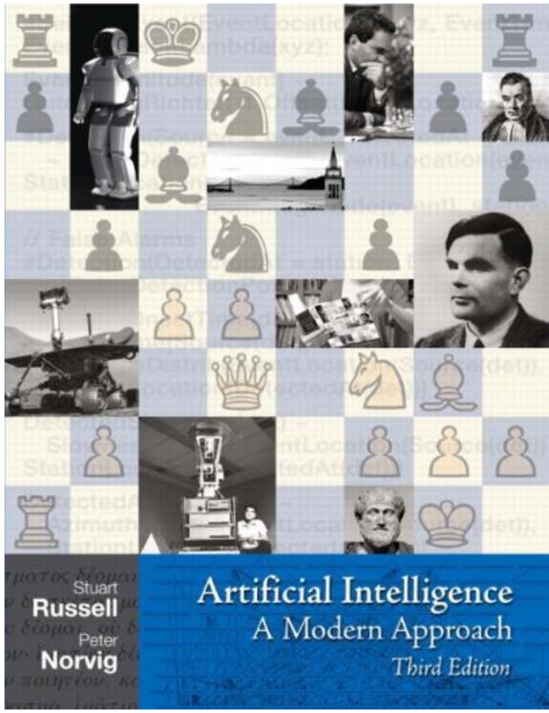
  Project @ Lab (10%),

  Quizzes @ Class (10%),

  Quizzes @ Lab (15 %),

  Mid-term exam (20%),

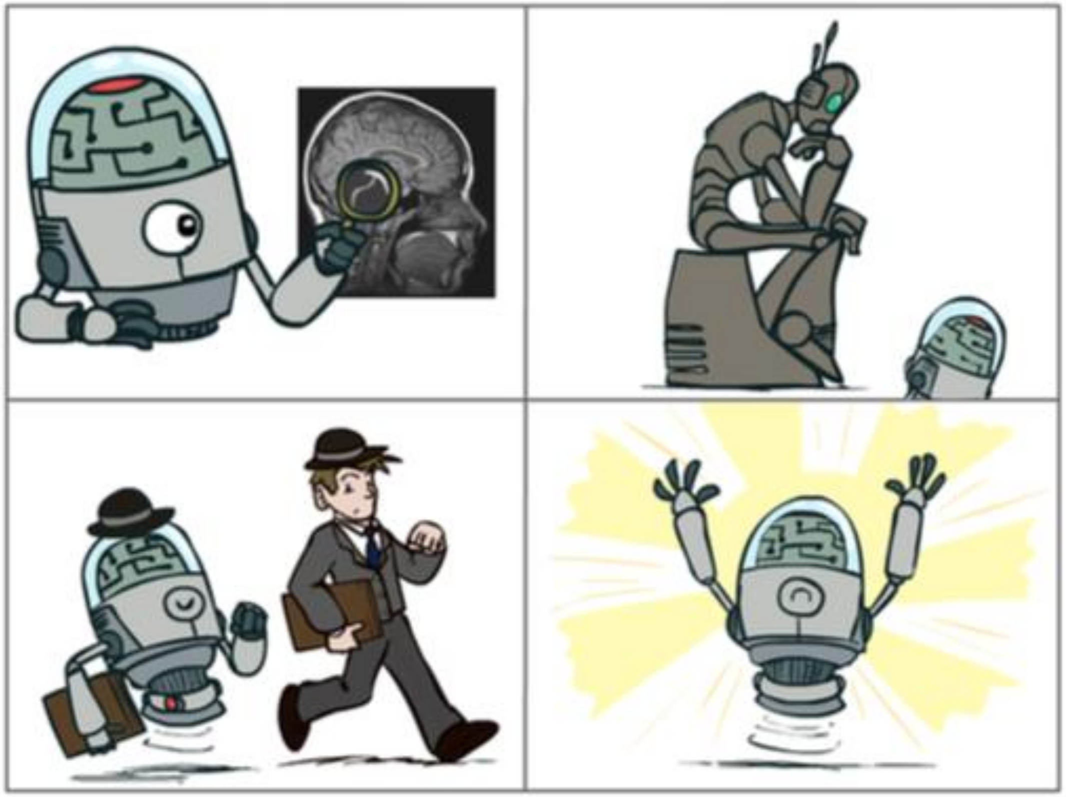
  Final exam (40%),



Review of Chapter 1

What is AI? **The science of making machines that:**

Thinking humanly



Thinking rationally

Acting humanly

Acting rationally

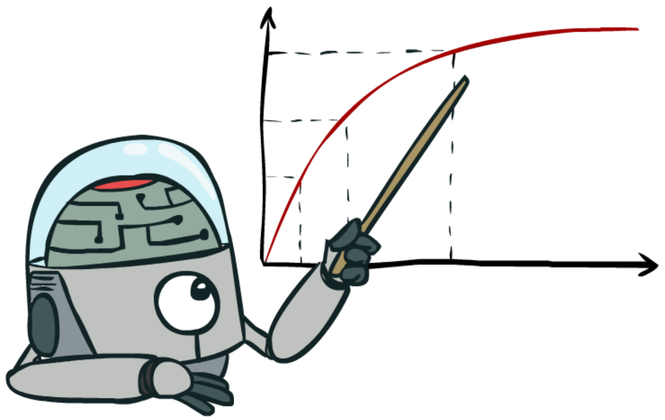
The textbook advocates *"acting rationally"*

Rational Decisions

- Here, we'll use the term **rational** in a very specific, technical way:
  - Rational: maximally achieving pre-defined goals
  - Rationality only concerns what decisions are made (not the thought process behind them)
  - Goals are expressed in terms of the **utility** of outcomes
  - Being rational means **maximizing your expected utility**

Rational Decisions

Maximize Your Expected Utility



What About the Brain?

- Brains (human minds) are very good at making rational decisions, but not perfect
- Brains aren't as modular as software, so hard to reverse engineer!

- "Brains are to intelligence as wings are to flight"
- Lessons learned from the brain: memory and simulation are key to decision making

Chapter 2

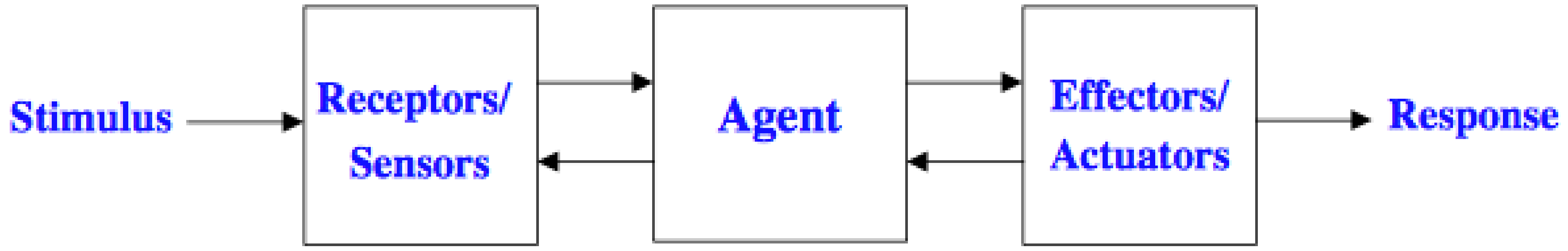
Intelligent Agents

Outline of ch2: Intelligent Agents

- Agents and environments
- Rationality
- PEAS (Performance measure, Environment, Actuators, Sensors)
- Environments types
- Agents types

AGENTS

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

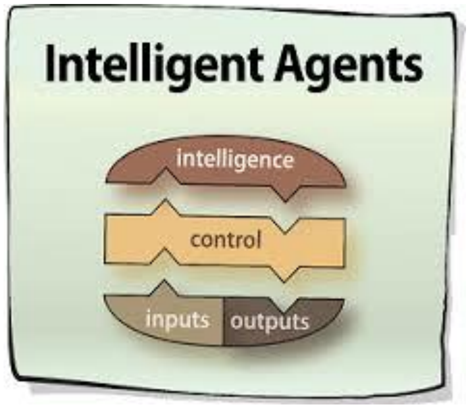


- If an agent's actions are based only on its built-in knowledge, and not on its own experience with its environment, then we say that the agent lacks autonomy.

- **An Autonomous Agent is an agent capable of making decisions about how it acts based on experience. And it is free to choose between different actions.**

AGENTS

- An **AGENT** is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators.
- Human **AGENT**: eyes, ears, and other organs for sensors; hands, legs, mouth, and other body parts for actuators,
-
- Intelligent **AGENT**: cameras and infrared range finders for sensors; various motors for actuators



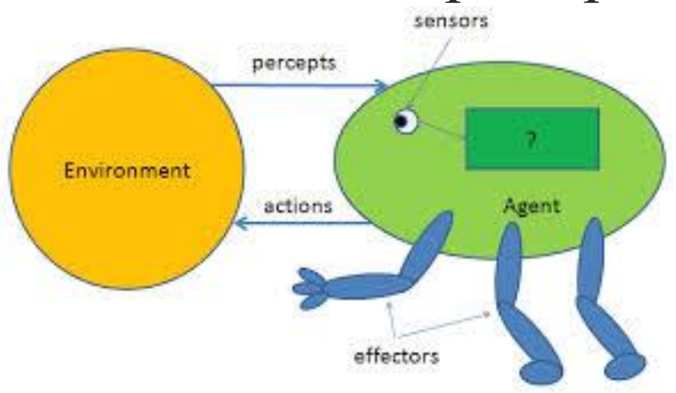
Agents and Environments

- The **AGENT** **function** maps from percept sequences* to actions:

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

- The **AGENT**'s choice of action depends on the entire percept sequence observed to date

- The **AGENT** **program** runs on the physical **architecture** to produce *f*:

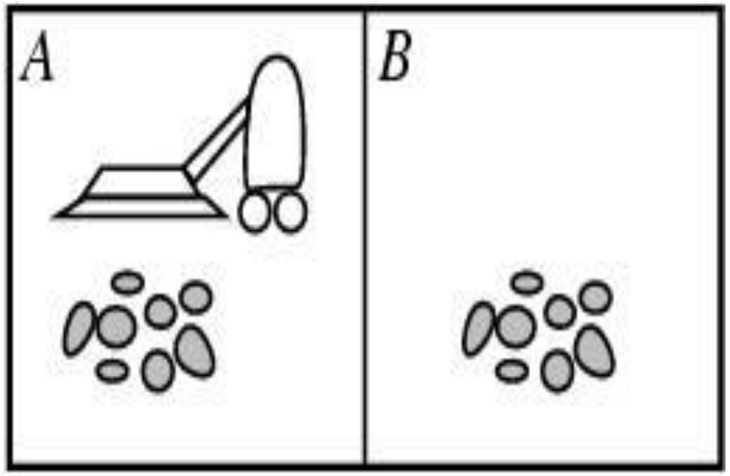


agent = architecture + program

The agent **function** is an abstract mathematical description; the agent **program** is a concrete implementation, running on the agent architecture.

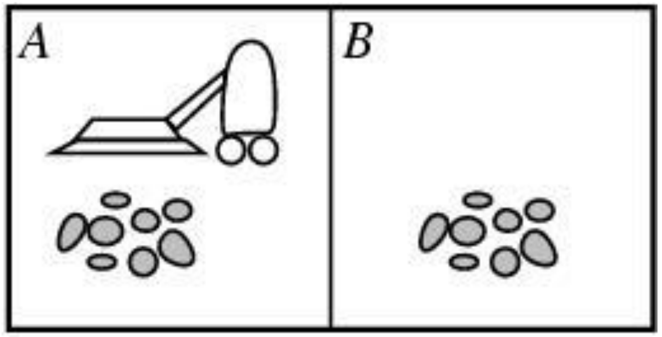
* Agent's **percept sequence** is the complete history of everything the agent has ever perceived.

The Vacuum-Cleaner World (1)



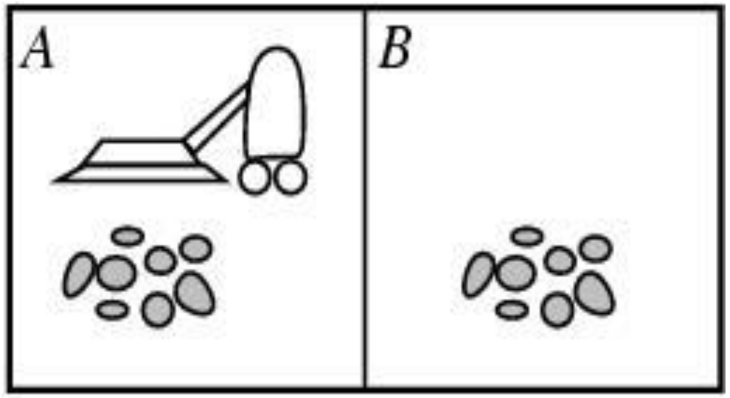
- **Environment**: square A and B
- **Percepts**: [location and content] e.g. *[A, Dirty]*
- **Actions**: left, right, suck, and no-op

The Vacuum-Cleaner World (2)



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
…………………	……….

The Vacuum-Cleaner World (3)



- Function REFLEX-VACUUM-AGENT ([location, status]) return an action

```
if status == Dirty then return Suck
else if location == A then return Right
else if location == B then return Left
```

- **What is the right function (way to fill out the table)? i.e, what makes an agent good/bad or intelligent/stupid?**

Rational Agents (1)

- 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,

  - There is NO fixed measure suitable for all agents!
  - 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.
  - As a general rule, it is better to design performance measures according to what one actually wants in the environment, rather than according to how one thinks the agent should behave.

Rational Agents (2)

What is rational at any given time depends on FOUR things:

- 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.

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.

Does the Vacuum-Cleaner Rational?

PEAS

- **P**erformance measure, **E**nvironment, **A**ctuators, **S**ensors
- **≈ Task environment**
  - In designing agent, the first step must always be specifying the task environment as fully as possible,
  - **a problem for which a rational agents is the solution**
- Consider, e.g., the task of designing an automated taxi:
  - Performance measure
  - Environment
  - Actuators
  - Sensors

PEAS for Automated Taxi

AGENT: 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 for a Medical Diagnosis System

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

PEAS for a Part-Picking Robot

- **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, jointed sensors

…Then, PEAS can be described as:

Performance Metrics: How does the AI know it's doing what it's supposed to be doing?

Environment: What environment does the agent interact with?

Actuators: How does the AI affect its environment?

Sensors: How does the AI get information from its environment?

Please check Table 2.5

Environment types

- ***Fully observable*** An agent's sensors give it access to the complete state of the environment at each point in time vs. ***partially observable*** because of noise or inaccurate sensors or missing parts **(VC, TD???)**

- ***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 partially observable, then it could appear to be stochastic **(VC, TD???)**

  . It is ***Uncertain environment*** if it is not fully observable or not deterministic.

- ***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 (Crucially, the next episode does not depend on the actions taken in previous episodes). i.e. Subsequent episodes do not depend on previous episodes, and so the agent can limit how far it needs to think ahead. In sequential environments, the current decision could affect all future decisions. Episodic are much simpler than sequential because the agent does not need to think. **(Part Picking, TD, Chess???)**

Environment types

- ***Static*** (vs. dynamic): If the environment can change while an agent is deliberating, then the environment is dynamic; otherwise, it is static. Static environment is easier. The environment is ***Semi Dynamic*** if the environment itself does not change with time but the agent's performance score does **(TD, Chess, Puzzles???) ***

- ***Discrete*** (vs. continuous): The distinction between both is related to the state of the environment, to the way time is handled, and to the percepts and actions of the agent. Chess has a limited no. of discrete states and discrete set of percept/action. Taxi is continuous state and continuous-time problem: the speed and location of the taxi and other vehicles sweep overtime. A ***discrete*** environment has a limited/finite number of distinct, clearly defined percepts and actions

- ***Single agent*** (vs. multiagent): An agent operating by itself in an environment (e.g. agent plays puzzles). Two agents playing chess is a multi-agent environment. Competitive Multiagent (Chess), cooperative Multiagent (Taxis).

Summary of Environment types

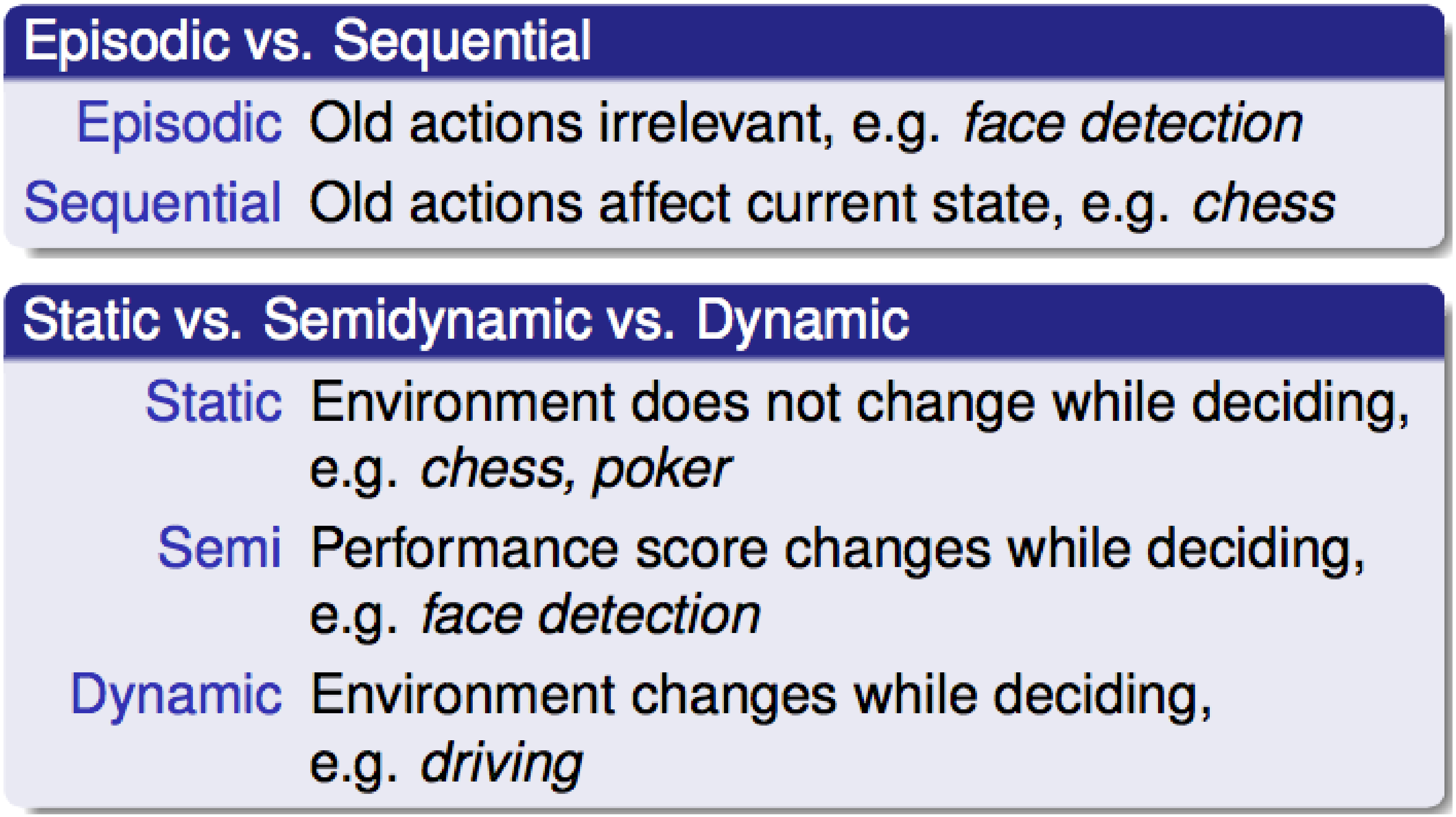
Fully vs. Partially Observable

Fully	All relevant to action is visible, e.g. *chess*
Partially	Part of environment unavailable, e.g. *poker*

Deterministic vs. Strategic vs. Stochastic

Determ	State + action determines next state, e.g. *crossword*
Strategic	State + action + other agent actions determines next state, e.g. *chess*
Stochastic	Next state not fully determined, e.g. *poker*

Summary of Environment types



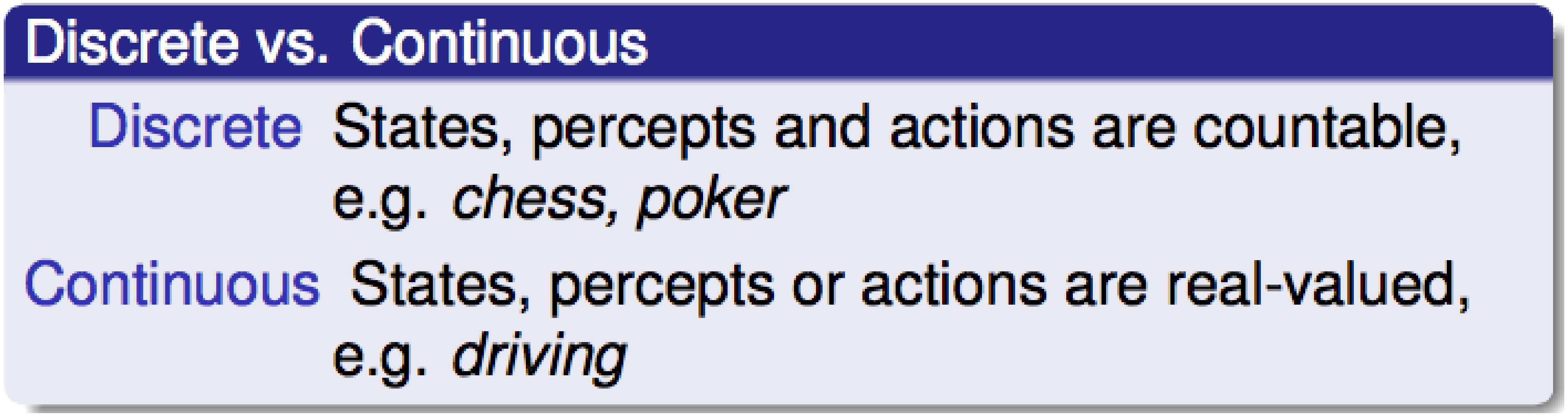
Episodic vs. Sequential

Episodic	Old actions irrelevant, e.g. *face detection*
Sequential	Old actions affect current state, e.g. *chess*

Static vs. Semidynamic vs. Dynamic

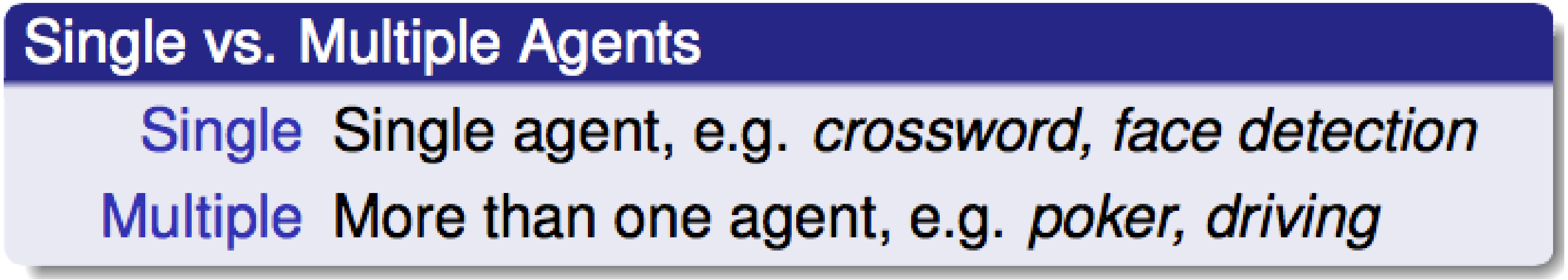
Static	Environment does not change while deciding, e.g. *chess, poker*
Semi	Performance score changes while deciding, e.g. *face detection*
Dynamic	Environment changes while deciding, e.g. *driving*

Summary of Environment types



Discrete vs. Continuous

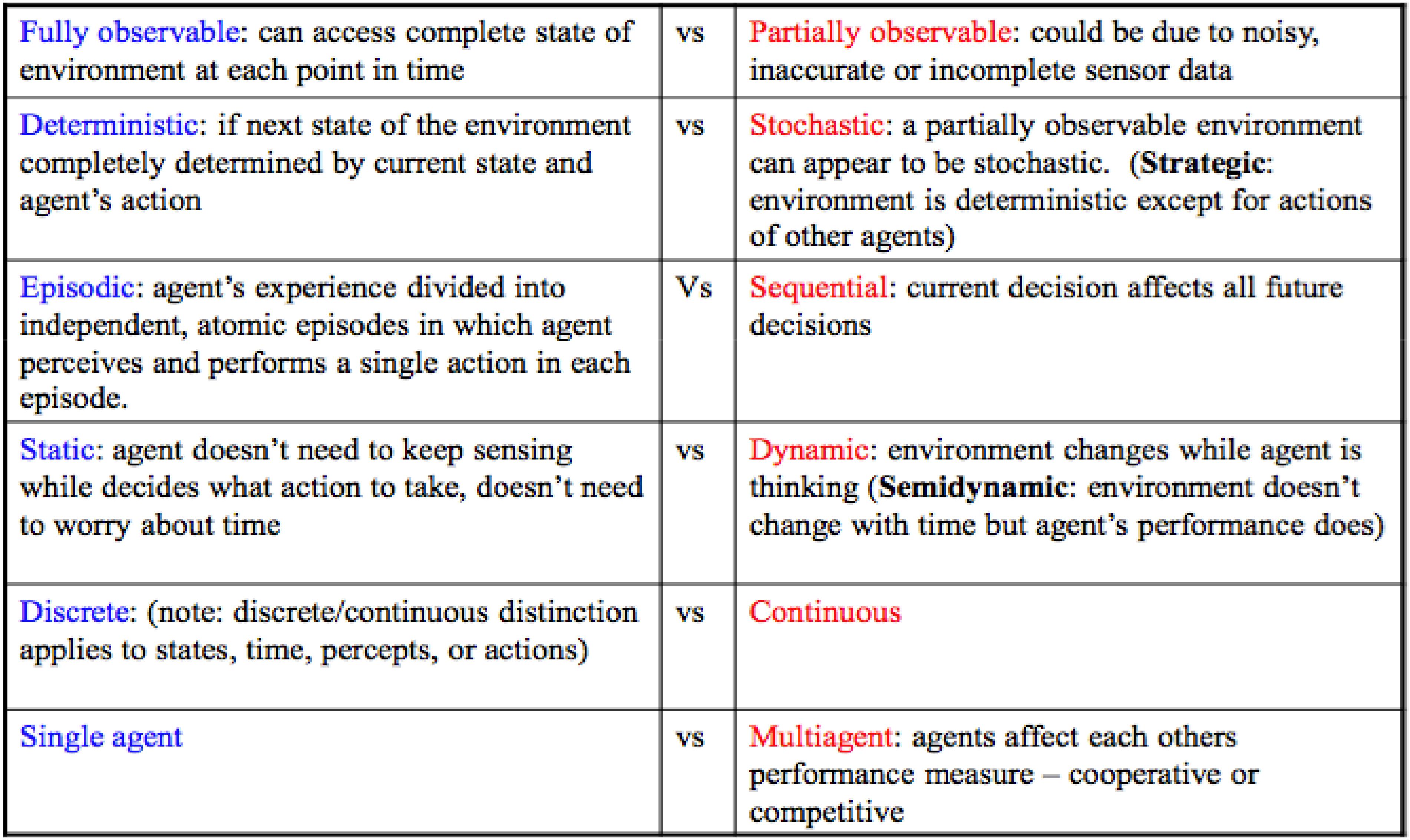
Discrete	States, percepts and actions are countable, e.g. *chess, poker*
Continuous	States, percepts or actions are real-valued, e.g. *driving*



Single vs. Multiple Agents

Single	Single agent, e.g. *crossword, face detection*
Multiple	More than one agent, e.g. *poker, driving*

Summary of Environment types



Fully observable: can access complete state of environment at each point in time	vs	Partially observable: could be due to noisy, inaccurate or incomplete sensor data
Deterministic: if next state of the environment completely determined by current state and agent's action	vs	Stochastic: a partially observable environment can appear to be stochastic. (**Strategic**: environment is deterministic except for actions of other agents)
Episodic: agent's experience divided into independent, atomic episodes in which agent perceives and performs a single action in each episode.	Vs	Sequential: current decision affects all future decisions
Static: agent doesn't need to keep sensing while decides what action to take, doesn't need to worry about time	vs	Dynamic: environment changes while agent is thinking (**Semidynamic**: environment doesn't change with time but agent's performance does)
Discrete: (note: discrete/continuous distinction applies to states, time, percepts, or actions)	vs	Continuous
Single agent	vs	Multiagent: agents affect each others performance measure – cooperative or competitive

Environment Types

- **The simplest environment is:**

  - Fully observable, deterministic, episodic, static, discrete, and single-agent

- **Most real situation/hardest:**

  - Partially observable, stochastic, sequential, dynamic, continuous, and multi-agent

Please check Table 2.6

Agent Program

- The job of AI is to design the agent program that implements the agent function mapping percepts to actions. This program runs physically on architecture.

Agent = Architecture + Program

- All agent programs have the same skeleton: They take the current percept as input from the sensors and return an action to the actuators.

Agent Program

Five basic agent types in order of increasing generality:

- **Table-Driven Agents**
  - Use a percept sequence/ action table in memory to find the next action. They are implemented by a (large) lookup table.
- **Simple Reflex agents**
  - Are based on condition-action rules and implemented with an appropriate production (rule-based) system. They are stateless devices, which do not have memory of past world states.
- **Model-based reflex**
  - Have internal state, which is used to keep track of past states of the world.
- **Agents with Explicit Goals**
  - Are agents, which in addition to state information have a kind of goal information that describes desirable situations. Agents of this kind take future events into consideration.
- **Utility-Based Agents**
  - Base their decision on classic axiomatic utility-theory in order to act rationally.

Learning Agents

- All previous agent programs describe methods for selecting actions, but: ***How the agent programs come into being?*** **(Learning)**
- Turing in 1950 proposes "to build" learning machines and then "to teach" them according to TWO concepts:
  - ✓ Teach them instead of instructing them,
  - ✓ Learning allows the agent "to operate" in unknown environments and "to become" more competent than its initial knowledge alone might allow **(robustness)**

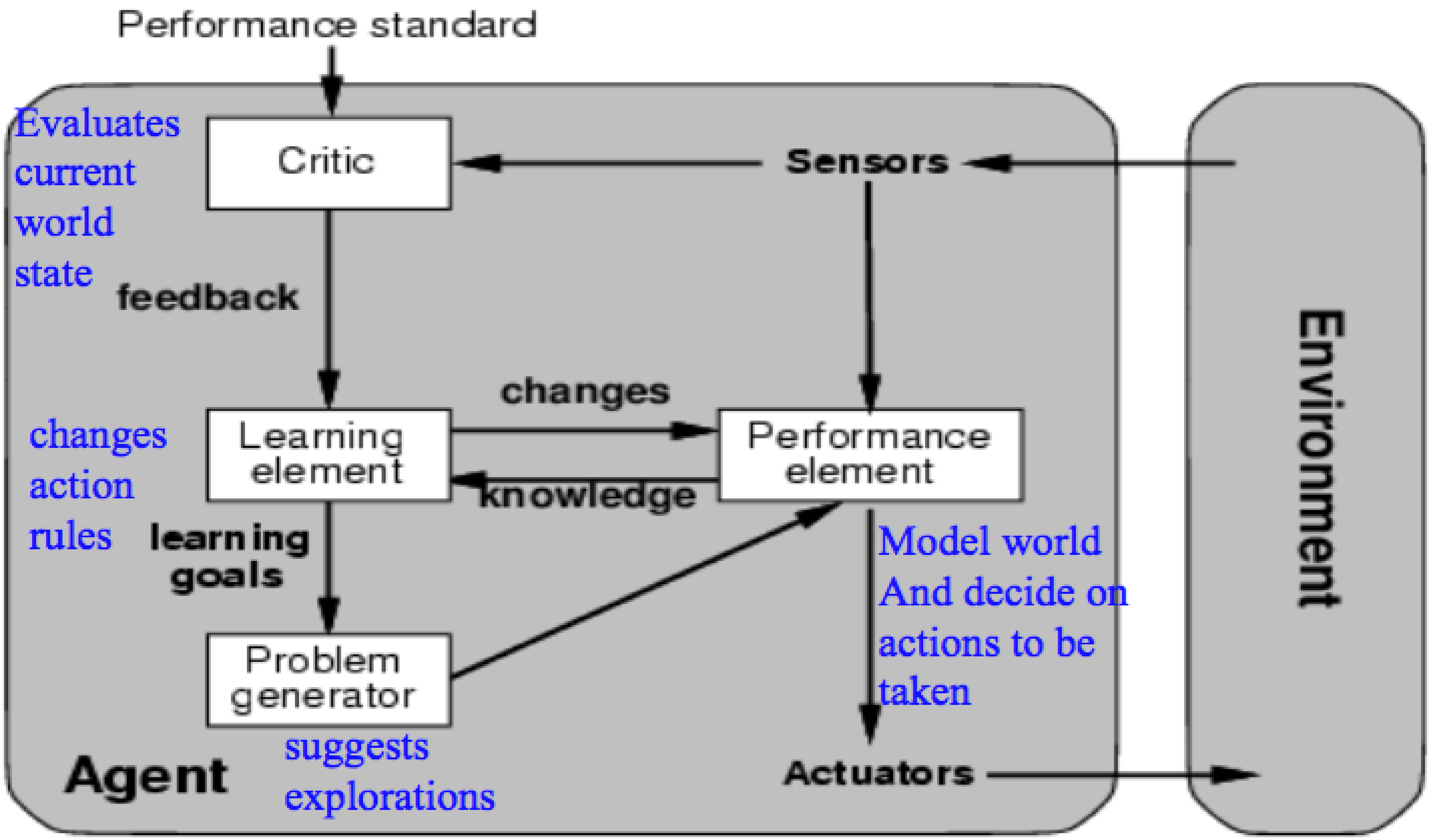
- Learning has FOUR elements:
  - Learning element
  - Performance element
  - Critic
  - Problem generator

Learning Agents

- ***Performance element***: selecting actions based on percepts
- ***Learning element:*** that receives information from the critic and makes appropriate improvements to the performance element.
- ***Critic***: tell the learning element how well the agent is doing with respect to a fixed performance standard
  - It is a necessary element (the percept itself has no meaning for agent's success), like in a chess game (Checkmate!)
- ***Problem generator***: suggest actions that will lead to new and informative experiences
  - Note that not all learning agents will need a problem generator – a teacher

Learning Agents

- How does an agent improve over time?
- By monitoring it's performance and suggesting better modeling, new action rules,



Chapter Summary

- What Is Agent?: Anything that can be *viewed as* perceiving its environment through sensors and acting upon that environment through its effectors to maximize progress towards its goals.
- Rational Agent
- Rational Action: The action that maximizes the expected value of the performance measure given the percept sequence to date.
- Environment Types: **Environments** are accessible, non-deterministic, non-episodic, dynamic, and continuous.
- Examples: Agents and Task Environments
- PEAS (Performance measure, Environment, Actuators, Sensors)
- Agent Types
  - Table Driven Agents use a percept-action table in memory to find the next action.
  - Reflex agents respond immediately to percepts.
  - Goal-based agents act in order to achieve their goals.
  - Utility-based agents maximize their own utility function.
  - Learning agents improve their performance over time.

Chapter Summary

What you should know

- What it means to be rational
- Be able to do a PEAS description of a task environment
- Be able to determine the properties of a task environment
- Know which agent program is appropriate for your task

In-class Exercise (1)

Develop a PEAS description of the task environment for a face-recognition agent

- ❑ Performance Measure
- ❑ Environment
- ❑ Actuators,
- ❑ Sensors

In-class Exercise (2)

Select a suitable agent design for the face-recognition agent

- ❑ Simple reflex agents,
- ❑ Model-based reflex agents,
- ❑ Goal-based agents,
- ❑ Utility-based agents

EXERCISES

Please try to solve the following exercises:

2.1

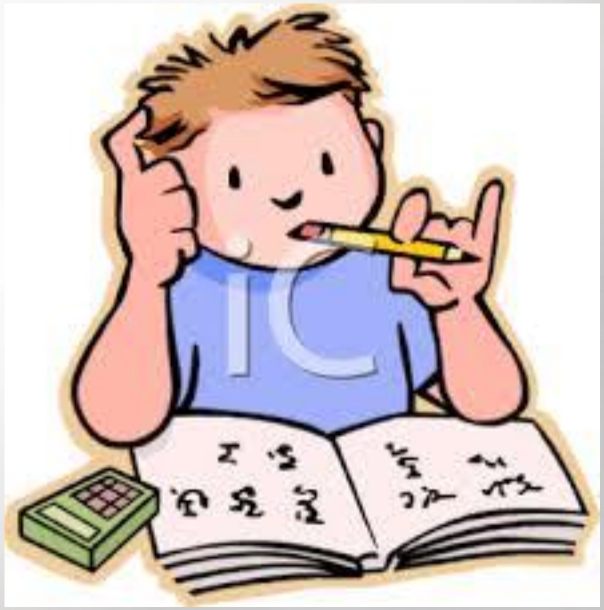
2.2

2.3

2.5

2.6

2.7 and beyond



Any questions?