AI基础

Lecture 1：Introduction and Intelligent Agents

Bin Yang

School of Data Science and Engineering

byang@dase.ecnu.edu.cn

Outline

- Course organization
    - Lectures + mini-projects
    - Staff: Lecturers, teaching assistants
- Introduction to AI
    - What is AI?
    - AI history
- Agents
    - The nature of environments
    - The structure of agents

Course structures

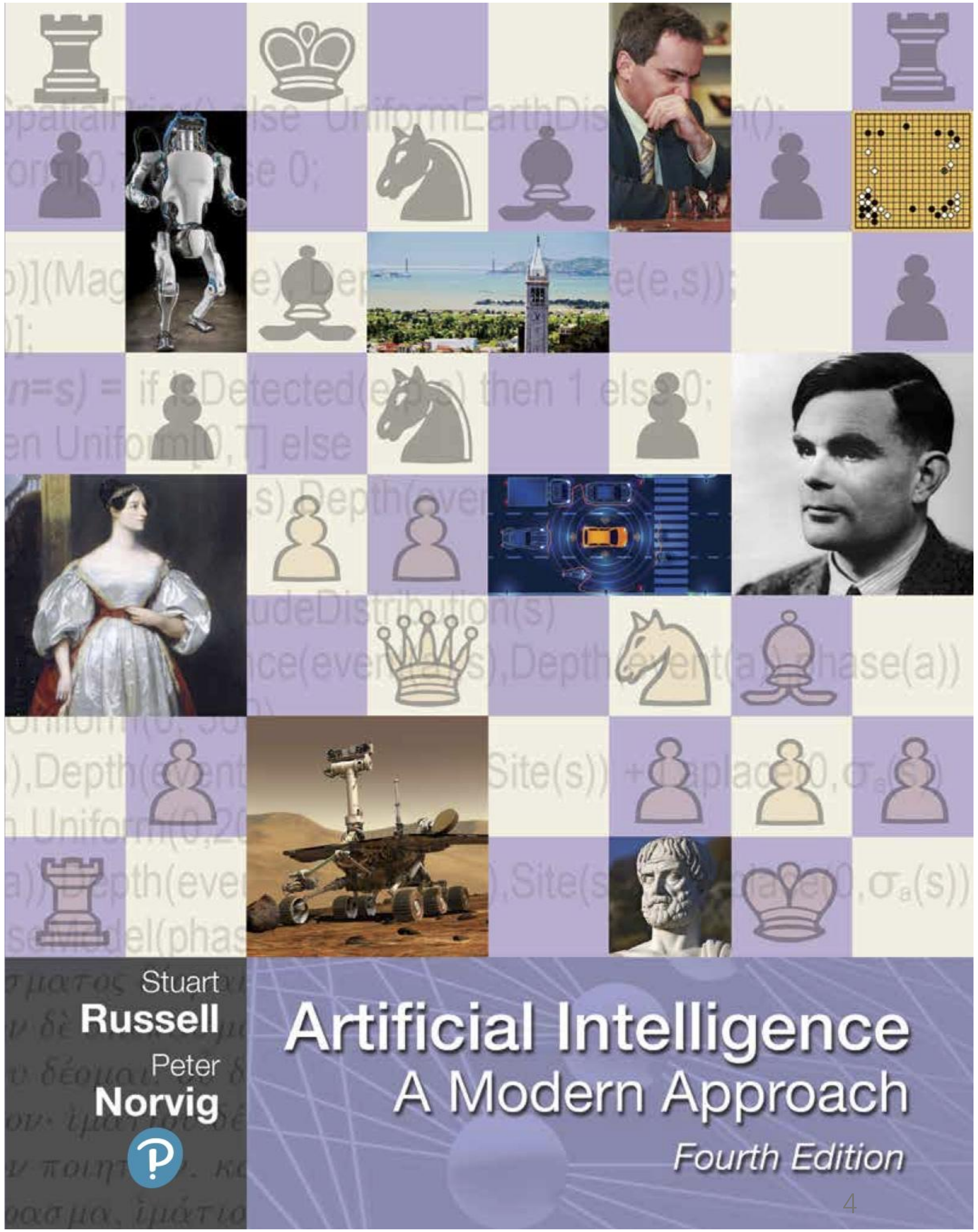
- Lectures: Wednesdays, 9:50 to 12:15, 文附楼302
  - Given by lecturers

- Mini-projects: Fridays, 9:50 to 11:25， 文附楼302
  - Supervised by Tas

- 5 * 45 mins per week
- 17 weeks in total

Textbook

- Russell & Norvig, Artificial Intelligence: A Modern Approach.



Staff

- Lecturers:
  - 杨彬
    - byang@dase.ecnu.edu.cn
  - 树扬
    - yshu@dase.ecnu.edu.cn

- TAs:
  - 田锦东, 1st year PhD student jdtian@stu.ecnu.edu.cn
  - 欧阳彪, 3rd year PhD student, bouyang@stu.ecnu.edu.cn

- Decision Intelligence Lab, School of Data Science and Engineering
  - https://decisionintelligence.github.io/index

Outline

- Course organization
    - Lectures + mini-projects
    - Staff: Lecturers, teaching assistants

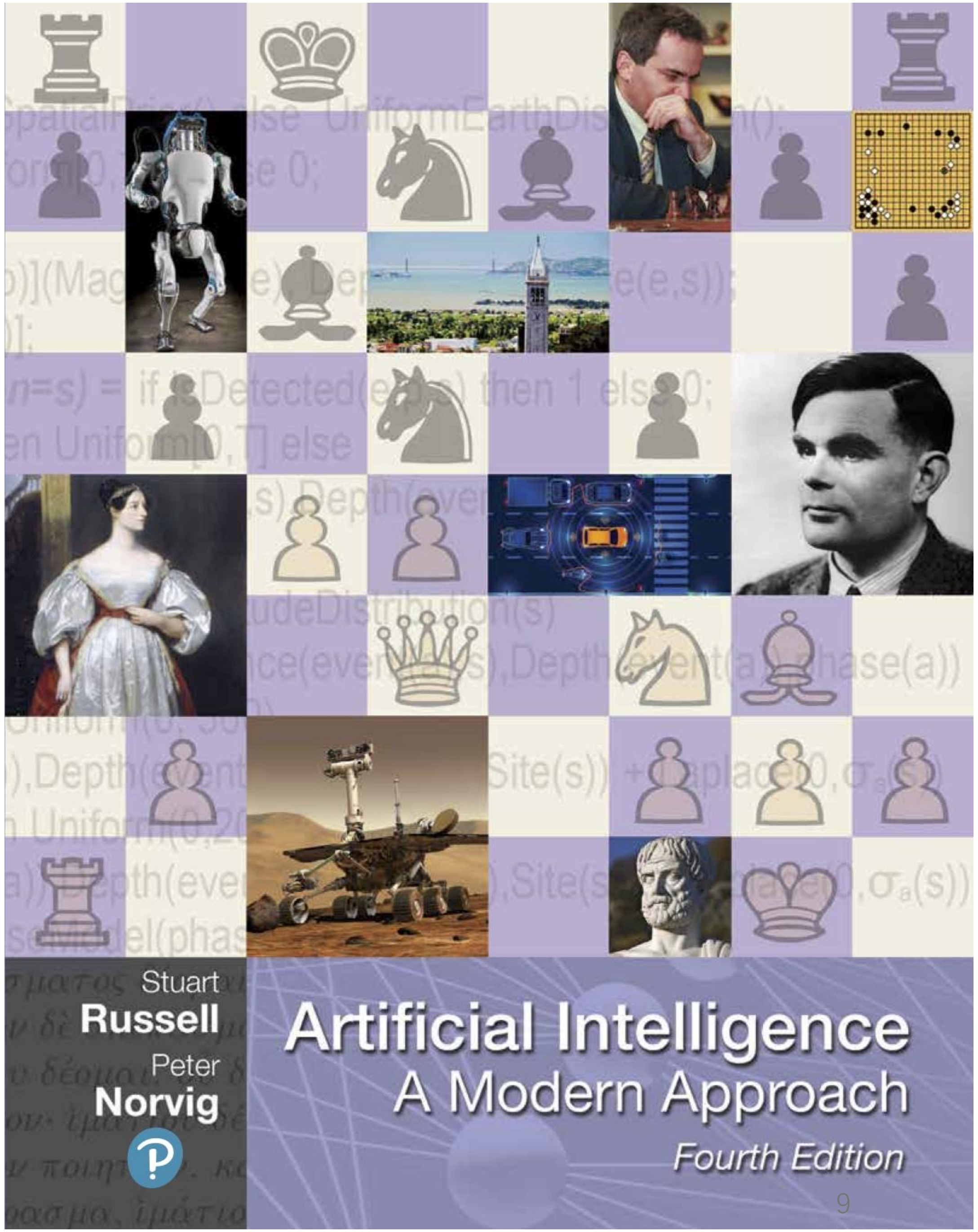
- Introduction to AI
    - What is AI?
    - AI history
- Agents
    - The nature of environments
    - The structure of agents

Intended learning outcomes (ILOs)

- What is AI?
  - Understand the four approaches
    - Human vs rationality
    - Behavior vs thought
  - Beneficial machines, value alignment
- AI history
- Agents
  - Key concepts: agent, environment, agent function, agent program, rational agent
  - Performance measure and rational agent
  - Task environments, and their categorization
  - Agent programs, and their categorization

What is AI?

- 1997, Chess, Deepblue vs. Kasparov
- 2016, Go, AlphaGo vs. Lee Sedol
- Atlas humanoid robot, Boston Dynamics
- Ada Lovelace, Alan Turing
- Self-driving cars
- Mars Exploration Rover robot
- Aristotle
- UN Comprehensive Nuclear-Test-Ban Treaty Organization for detecting nuclear explosions from seismic signals



Intelligence and Artificial Intelligence

- Intelligence
  - How we **think** and **act**.
  - How our brain can perceive, understand, predict, and manipulate a world far larger and more complicated than itself.

- Artificial intelligence
  - Not just understanding but also building intelligent entities-machines that can compute how to act effectively and safely in a wide variety of novel situations.
  - General subfields: learning, reasoning, perception
  - Specific subfields: playing games, proving mathematical theorems, writing poetry, driving a car, or diagnosing diseases

Categorization

	Human	Rational
Behavior	Act like people Acting Humanly The Turing test approach	Act rationally The rational agent approach
Thought	Think like people Thinking humanly The cognitive modeling approach	Think rationally The “laws of thought” approach

Human: Fidelity to Human Performance
Rational: doing the “right thing”
Thought: Internal thought processes and reasoning
Behavior: External characterization

Turing test

- “Can a machine think?”
- A computer passes the test if a human interrogator, after posing some written questions, cannot tell whether the written responses come from a person or a computer.
- To pass the test, the computer would need the following capabilities:
  - Natural language processing
  - Knowledge representation
  - Automated reasoning
  - Machine learning
- Physical simulation of a person is unnecessary to demonstrate intelligence

Total Turing test

- Interaction with objects and people in the real world. Need additional capabilities
  - Computer vision and speech recognition
  - Robotics

- The six disciplines compose most of AI.
- But AI researchers do not really focus on passing the Turing test, believing that studying the underlying principles of intelligence is more important.

- “Machines that fly so exactly like pigeons that they can fool even other pigeons.”

Think humanly

- A program thinks like a human.
- We need to know how humans think.
    - Introspection
    - Phycological experiments
    - Brain imaging

- Cognitive science
    - Interdisciplinary field: computer models from AI + experimental techniques from psychology
    - To construct precise and testable theories of the human mind

Think rationally: The “laws of thought” approach

- Aristotle, Syllogisms
  - an instance of a form of reasoning in which a conclusion is drawn (whether validly or not) from two given or assumed propositions (premises), each of which shares a term with the conclusion, and shares a common or middle term not present in the conclusion
  - Socrates is a man
  - All men are mortal
  - Socrates is moral
- Logic
  - Develop precise notation for statements about objects in the world and the relations among them.
  - Solve problem described in logical notation
  - Build on such programs to create intelligent systems
- Logic requires knowledge of the world to be certain. Probability theory fills the gap.

Acting rationally, rational agent

- A rational agent is one that acts to achieve the best outcome, or the best expected outcome (when there is uncertainty)

- “Laws of thought” approach (think rationally) makes correct inferences, which is sometimes part of being a rational agent.
  - But not always, e.g., reflex action
- All the skills needed for passing the Turing test (act humanly) allow an agent to act rationally.
- Knowledge representation and reasoning (think humanly) enable agents to reach good decisions.

Acting rationally, rational agent

- Benefits compared to the other three approaches
  - More general than “laws of thought,” i.e., think rationally
  - More amenable to scientific development
- Rational agent approach to AI has prevailed throughout most of the field’s history.
- AI has focused on the study and construction of agents that **do the right thing**.
  - **What is the right thing is defined by the objective that we provide to the agent.**
- **Standard model**
  - Not only in AI, but also in control theory, operations research, statistics, economics
- Limited rationality
  - Acting appropriately when there is not enough time to do all the computation one might like

Is the standard model always right model?

- When the standard model is applicable?
  - Task such as chess, where the task comes with an objective built in.
- When the standard model is not that applicable anymore?
  - When it is difficult to specify the objective completely and correctly.
    - For example, in designing a self-driving car, one might think that the objective is to reach the destination safely. But driving along any road incurs a risk of injury due to other errant drivers, equipment failure, and so on; thus, a strict goal of safety requires staying in the garage. There is a tradeoff between making progress towards the destination and incurring a risk of injury. How should this tradeoff be made? Furthermore, to what extent can we allow the car to take actions that would annoy other drivers? How much should the car moderate its acceleration, steering, and braking to avoid shaking up the passenger? These kinds of questions are difficult to answer a priori.
- Value alignment problem
  - The values or objectives put into the machine must be aligned with those of the human.
- A system with an incorrect objective will have negative consequences
  - The more intelligent the system, the more negative the consequences

Beneficial machines

- We don’t want intelligent machines to pursue their objectives, but our objective.
- When we ourselves cannot transfer those objectives perfectly to the machine, then we need a new formulation
  - The machine is uncertain with what are the objectives
  - When a machine knows that it doesn’t know the complete objective, it has an incentive to act cautiously, to ask permission, to learn more about our preferences through observation, and to defer to human control.
- We want agents that are **provably beneficial** to humans

Risks and benefits of AI

- Benefits
  - Our entire civilization is the product of our human intelligence
  - When accessing to substantially greater machine intelligence, the ceiling on our ambitions is raised substantially
  - “First solve AI, then use AI to solve everything else.”
- Risks, misuse of AI
  - Lethal autonomous weapons.
  - Surveillance and persuasion
  - Biased decision making
  - Impact on employment
  - Safety critical applications
  - Cybersecurity

Risks and benefits of AI

- Human-level AI
  - Machine should be able to learn to do anything a human can do
- Artificial general intelligence (AGI)
- Artificial superintelligence (ASI)
  - Intelligence that far surpasses human ability

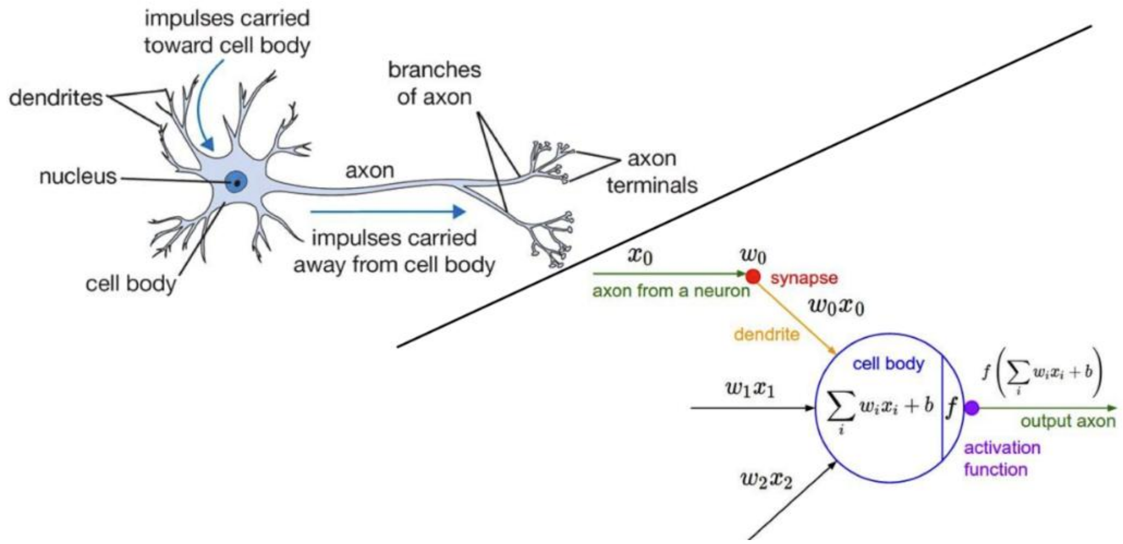
- We design the AI systems, so if they do end up ”taking control,” as Turing suggests, it would be the result of a design failure.

- King Midas problem

The foundations of AI

- Philosophy
  - Can formal rules be used to draw valid conclusions?
  - How does the mind arise from a physical brains?
  - Where does knowledge come from？
  - How does knowledge lead to action?
- Mathematics
  - What are the formal rules to draw valid conclusions?
  - What can be computed?
  - How do we reason with uncertainty?
- Economics
  - How should we make decisions in accordance with our preferences?
  - How should we do this when others may not go along?
  - How should we do this when the payoff may be far in the future?

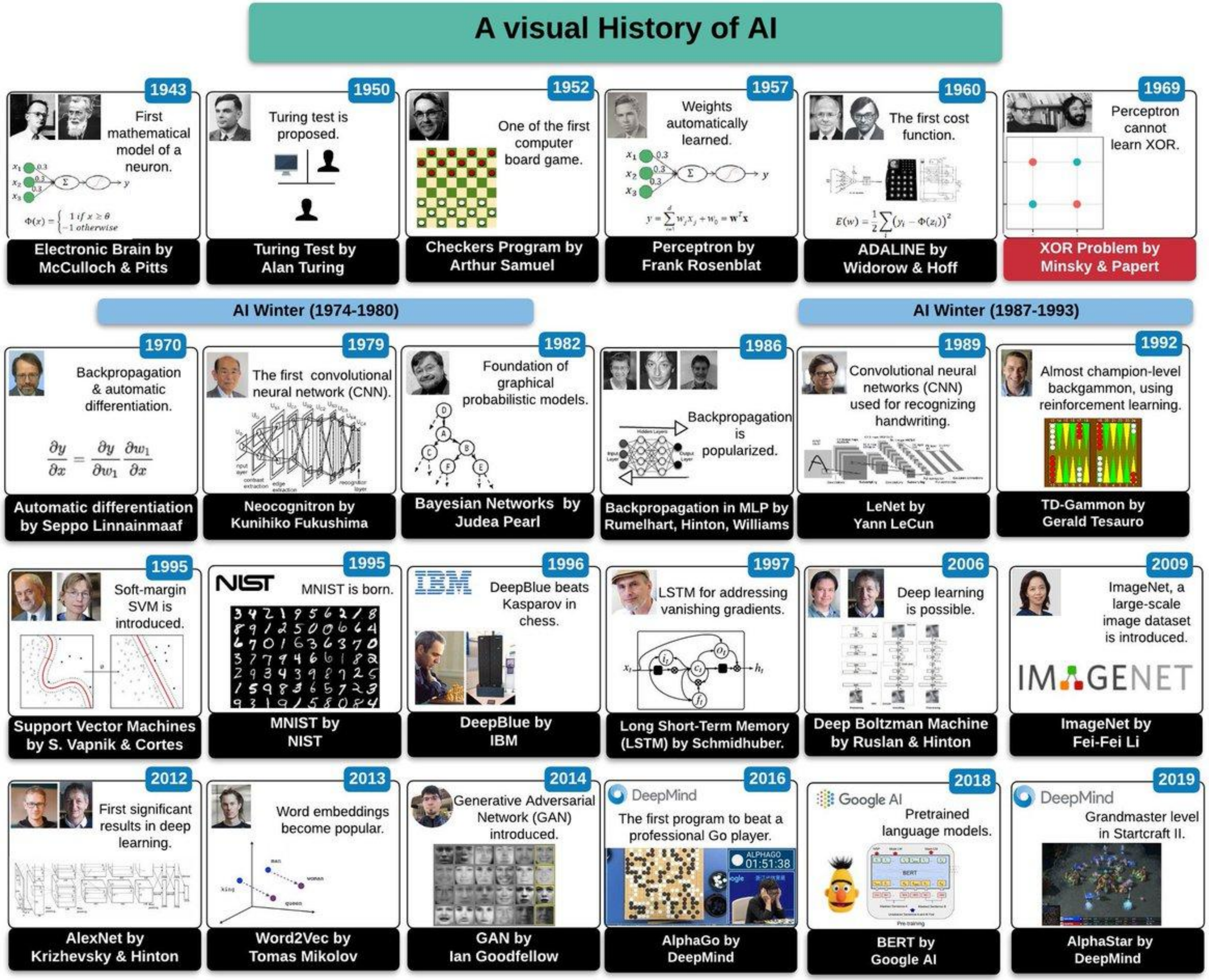
The foundations of AI



- Neuroscience
  - How do brains process information?
- Psychology
  - How do humans and animal think and act?
- Computer engineering
  - How can we build an efficient computer?
- Control theory and cybernetics
  - How can artifacts operate under their own control?
- Linguistics
  - How does language relate to thought?

AI history

- 1943—1956: The inception of AI
    - 1943: McCulloch & Pitts: Boolean circuit model of brain
    - 1950: Turing's “Computing Machinery and Intelligence”
- 1952—69: Early enthusiasm, great expectations. “Look, Ma, no hands!”
    - 1950s: Early AI programs, including Samuel's checkers program, Newell & Simon's Logic Theorist, Gelernter's Geometry Engine
    - 1956: **Dartmouth meeting: “Artificial Intelligence” adopted**
    - 1965: Robinson's complete algorithm for logical reasoning
- 1969—86: Expert systems
    - 1969—79: Early development of knowledge-based systems
    - 1980—88: Expert systems industry booms
    - 1988—93: Expert systems industry busts: “AI Winter”
- 1990—2012: Statistical approaches + subfield expertise
    - Resurgence of probability, focus on uncertainty
    - General increase in technical depth
    - Agents and learning systems… “AI Spring”?
- 2012—: Excitement: Look, Ma, no hands!
    - Big data, big compute, neural networks
    - Some re-unification of subfields
    - AI used in many industries



A visual History of AI

1943 First mathematical model of a neuron.
$\Phi(x) = \begin{cases} 1 & if\ x \geq \theta \\ -1 & otherwise \end{cases}$
Electronic Brain by McCulloch & Pitts

1950 Turing test is proposed.
Turing Test by Alan Turing

1952 One of the first computer board game.
Checkers Program by Arthur Samuel

1957 Weights automatically learned.
$y = \sum_{i=1}^{d} w_i x_i + w_0 = \mathbf{w}^T\mathbf{x}$
Perceptron by Frank Rosenblatt

1960 The first cost function.
$E(w) = \frac{1}{2}\sum_i (y_i - \Phi(z_i))^2$
ADALINE by Widrow & Hoff

1969 Perceptron cannot learn XOR.
XOR Problem by Minsky & Papert

AI Winter (1974-1980)

AI Winter (1987-1993)

1970 Backpropagation & automatic differentiation.
$\frac{\partial y}{\partial x} = \frac{\partial y}{\partial w_1}\frac{\partial w_1}{\partial x}$
Automatic differentiation by Seppo Linnainmaaf

1979 The first convolutional neural network (CNN).
Neocognitron by Kunihiko Fukushima

1982 Foundation of graphical probabilistic models.
Bayesian Networks by Judea Pearl

1986 Backpropagation is popularized.
Backpropagation in MLP by Rumelhart, Hinton, Williams

1989 Convolutional neural networks (CNN) used for recognizing handwriting.
LeNet by Yann LeCun

1992 Almost champion-level backgammon, using reinforcement learning.
TD-Gammon by Gerald Tesauro

1995 Soft-margin SVM is introduced.
Support Vector Machines by S. Vapnik & Cortes

1995 NIST MNIST is born.
MNIST by NIST

1996 IBM DeepBlue beats Kasparov in chess.
DeepBlue by IBM

1997 LSTM for addressing vanishing gradients.
Long Short-Term Memory (LSTM) by Schmidhuber.

2006 Deep learning is possible.
Deep Boltzman Machine by Ruslan & Hinton

2009 ImageNet, a large-scale image dataset is introduced.
IMAGENET
ImageNet by Fei-Fei Li

2012 First significant results in deep learning.
AlexNet by Krizhevsky & Hinton

2013 Word embeddings become popular.
Word2Vec by Tomas Mikolov

2014 Generative Adversarial Network (GAN) introduced.
GAN by Ian Goodfellow

2016 DeepMind The first program to beat a professional Go player.
AlphaGo by DeepMind

2018 Google AI Pretrained language models.
BERT by Google AI

2019 DeepMind Grandmaster level in Startcraft II.
AlphaStar by DeepMind

Parisa Rashidi, May 2020. CC BY 4.0

The state of the art

- Stanford 100 Year study on AI report
- Substantial increases in the future uses of AI applications
  - Self-driving cars, healthcare diagnostics and targeted treatment, physical assistance for elder care
- Society is now at a crucial juncture in determining how to deploy AI-based technologies in ways that promote rather than hinder democratic values such as freedom, equality, and transparency.
- Publications, sentiment, students, diversity, conferences, industry, internationalization, vision, speed, language, human benchmark.

SOTA Applications

- Robotic vehicles
- Legged locomotion
- Autonomous planning and scheduling
- Machine translation
- Speech recognition
- Recommendations
- Game playing
- Image understanding
- Medicine
- Climate science

What can we do now?

- √ Win against any human at chess?
- √ Win against the best humans at Go?
- √ Play a decent game of table tennis?
- X Unload any dishwasher in any home?
- √ Drive safely along the highway?
- ? Drive safely along streets of Shanghai?
- √ Buy a week's worth of groceries on the web?
- X Buy a week's worth of groceries at grocery store?
- ? Discover and prove a new mathematical theorem?
- X Perform a surgical operation?
- √ Translate spoken Chinese into spoken English in real time?
- √ Win an art competition?
- √ Write an intentionally funny story?
- X Construct a building?

Outline

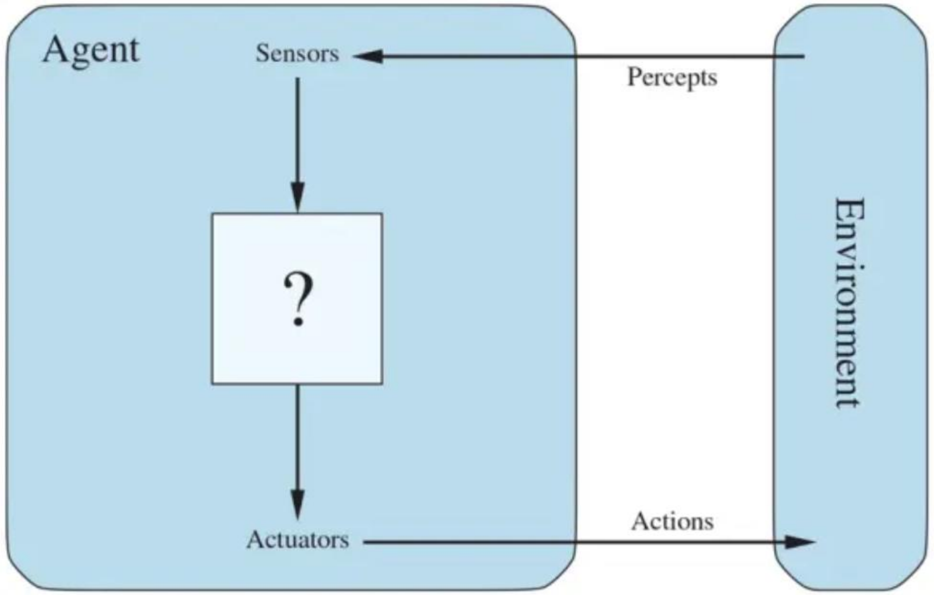
- Course organization
    - Lectures + mini-projects
    - Staff: Lecturers, teaching assistants

- Introduction to AI
    - What is AI?
    - AI history
- Agents
    - The nature of environments
    - The structure of agents

What are agents?

- An agent is anything that
  - is able to perceive its **environment** through **sensors**, and
  - is able to act upon that **environment** through **actuators**
- A human agent
  - Eyes and ears as sensors
  - Hands and legs as actuators
- A robotic agent
  - Cameras and infrared as sensors
  - Various motors as actuators

Figure 2.1

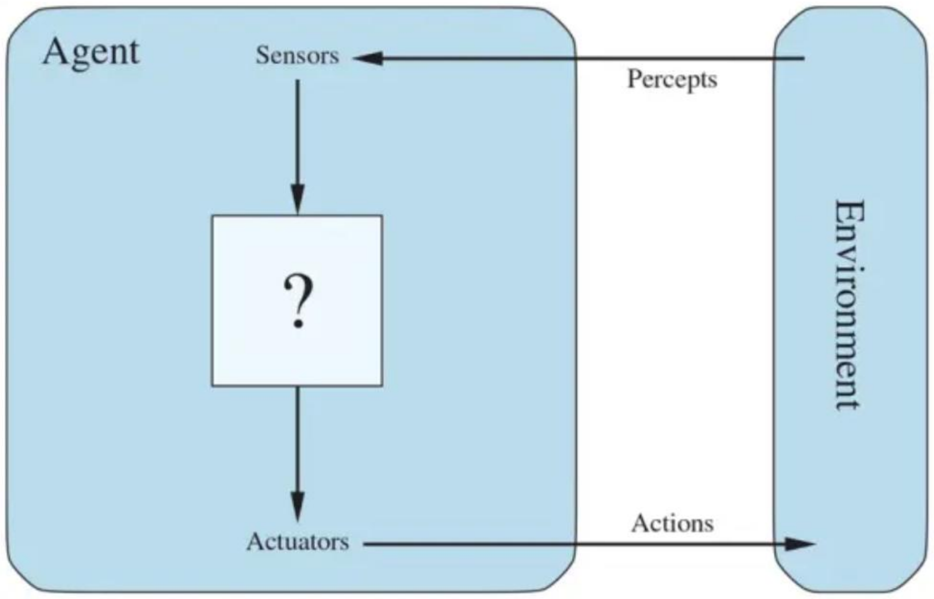


Agents interact with environments through sensors and actuators.

Agent function vs. Agent program

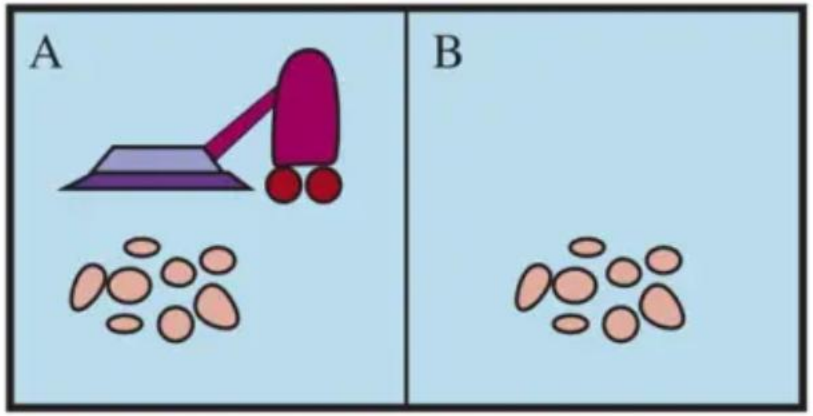
- **Percept** refers to the content an agent’s sensors are perceiving
- **Percept sequence** is the complete history of the agent has ever perceived so far.
- An agent’s choice of action
  - Its built-in knowledge
  - Entire percept sequence observed to date
- Agent function maps
  - From any given percept sequence
  - To an action
- Agent function vs. agent program
  - The former: abstract mathematical description
  - The latter: concrete implementation, running on some physical system.

Figure 2.1



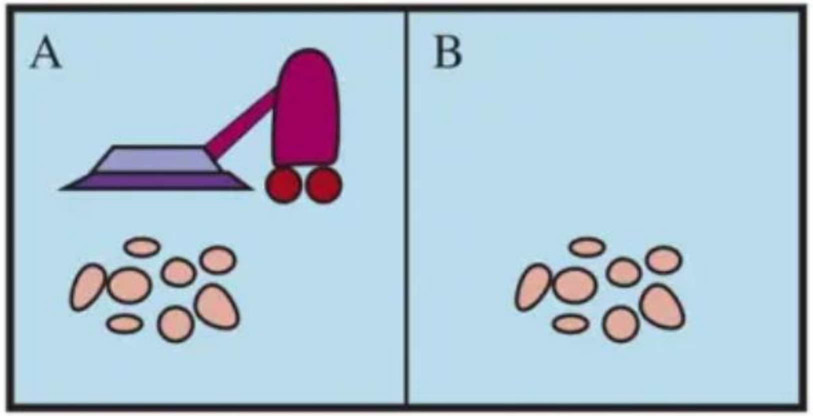
Agents interact with environments through sensors and actuators.

A vacuum-cleaner example

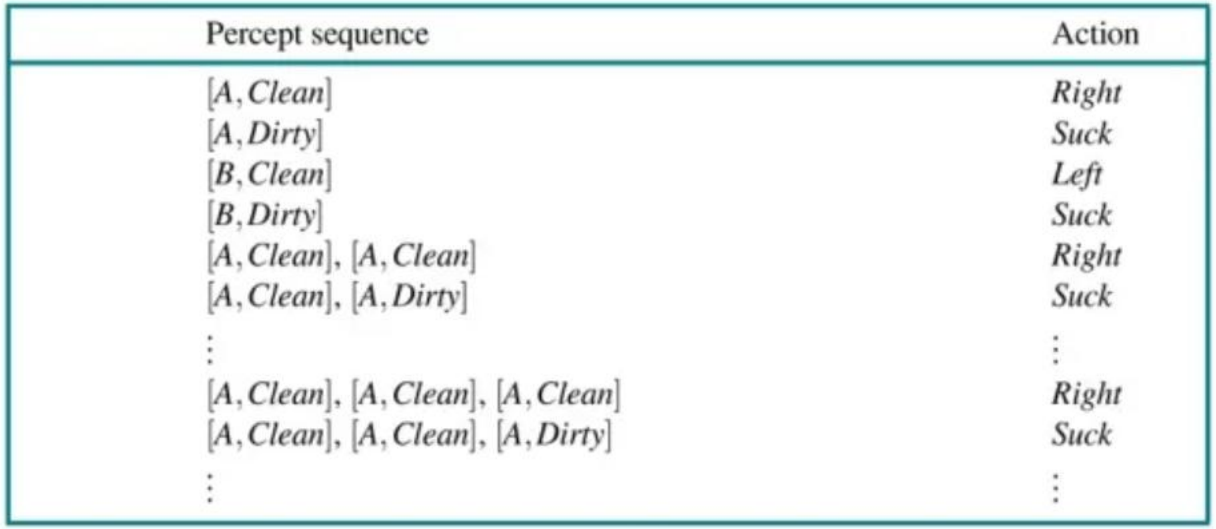


- A robotic vacuum-cleaning agent
- The world consists of two squares A and B
- The agent can perceive
  - Which square it is in
  - Whether there is dirt in the square
- Actions
  - Move to right
  - Move to left
  - Suck up the dirt
  - Do nothing

A vacuum-cleaner example



- Agent function
  - If the current square is dirty, then suck.
  - Otherwise, move to the other square.



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*
⋮	⋮

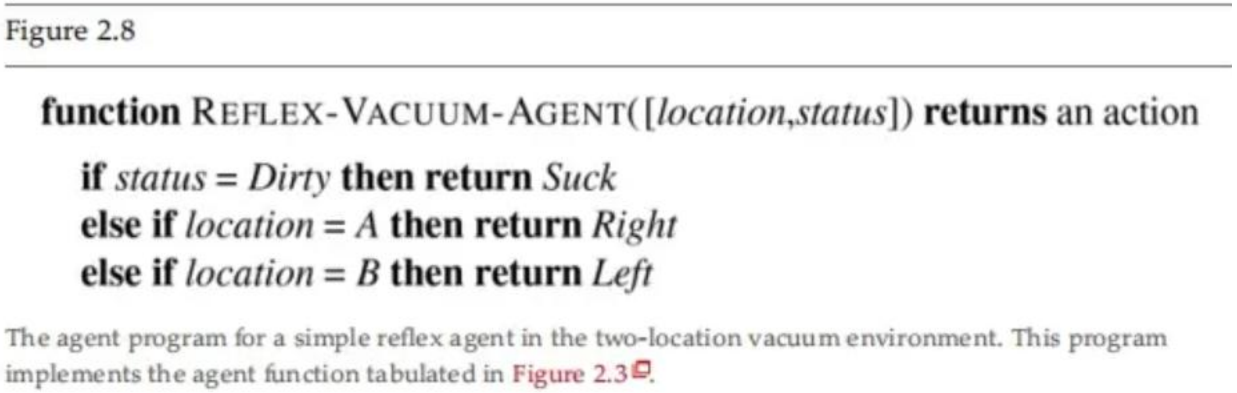


Figure 2.8

```
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
```

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

Agent program

What are good agents? Rational agents.

- A rational agent is one that does the right thing.
- What is rational depends on
  - The performance measure that defines the criterion of success
  - The agent’s prior knowledge of the environment
  - The actions that the agent can perform
  - The agent’s percept sequence to date
- Definition of a 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.
  - Maximize its expected performance measure/utility function
- Mini quiz?
  - Is the vacuum agent defined in the pervious slide a rational agent?
- Omniscience vs rationality
  - Being rationality is not the same as perfection.

The nature of environments

- Specifying the **task environment**, **PEAS**
  - **P**erformance, **E**nvironment, **A**ctuators, **S**ensors

P	E	A	S
Safe, fast, legal, comfortable, minimize impact on other road users	Roads, other traffic, police, pedestrians, weather	Steering, accelerator, brake, signal, horn	Cameras, radar, speedometer, GPS, engine sensors, accelerometer

Properties of task environments

- Fully observable vs. partially observable
  - Whether an agent's sensors give it access to the complete state of the environment at each time point
  - Noisy and inaccurate sensors
- Single-agent vs. multiagent
  - Competitive vs. cooperative
- Deterministic vs. nondeterministic
  - Deterministic environment: the next state of the environment is completely determinized by the current state and the action executed by the agent(s)
  - Nondeterministic vs stochastic
    - there's a chance of rain vs. there's a 25% chance of rain (explicitly deals with probabilities)

Properties of task environments

- Episodic vs. sequential
  - In each episode, the agent receives a percept and then performs a single action. The next episode does not depend on the actions taken in pervious episodes.
  - Current decision could affect all future decisions.
- Static vs. dynamic
  - Whether the environment can change while an agent is deliberating
- Discrete vs. continuous
  - State of the environment
  - Chess vs. taxi-driving
- Known vs unknown
  - Known environments: the outcomes (probabilities) for all actions are given.

- Hardest case
  - Partially observable, multiagent, nondeterministic, sequential, dynamic, continuous, and unknown.
  - Self-driving

Examples of task envrinment

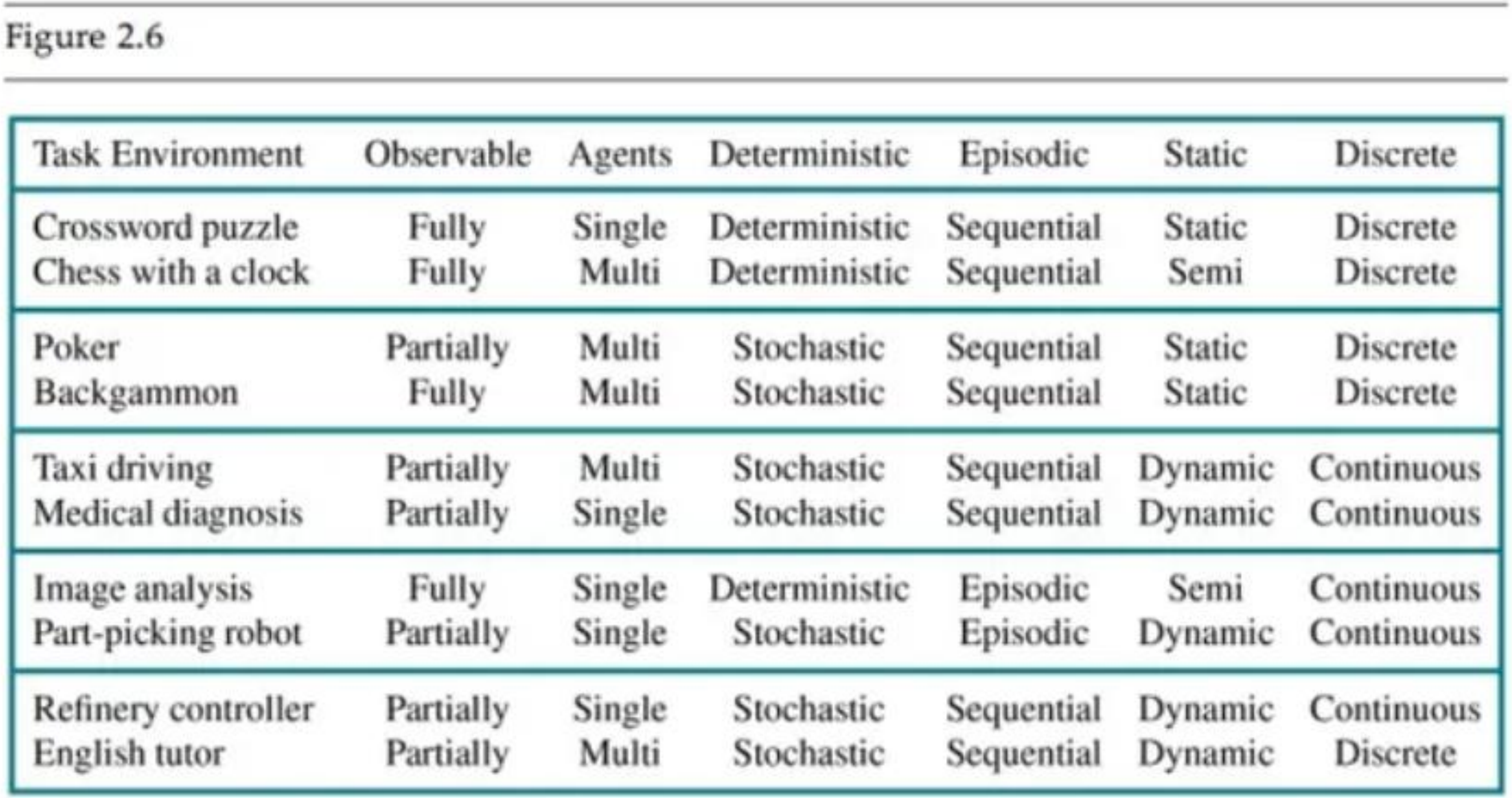


Figure 2.6

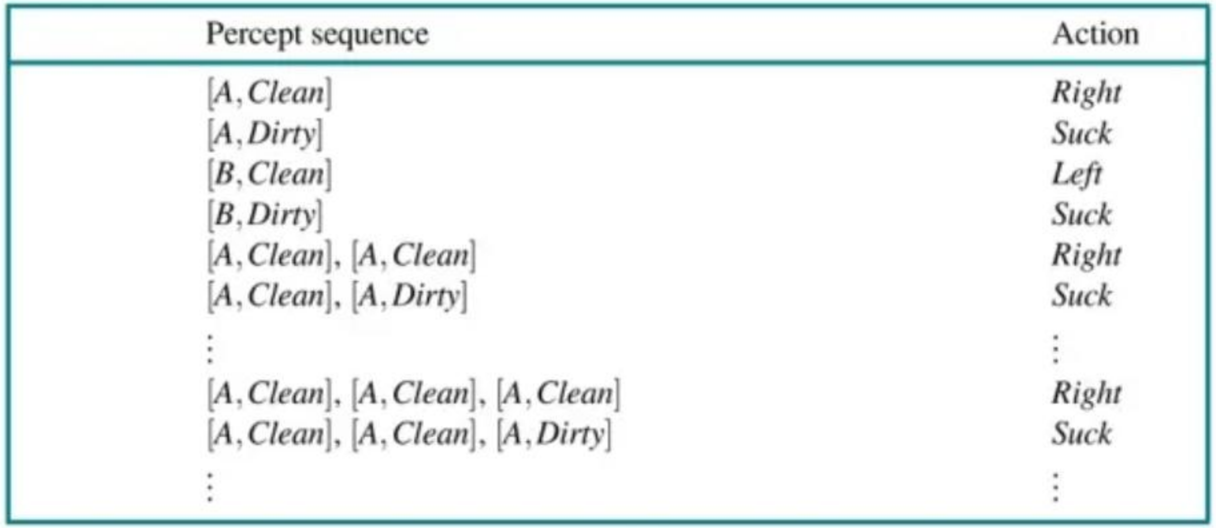
Task Environment	Observable	Agents	Deterministic	Episodic	Static	Discrete
Crossword puzzle	Fully	Single	Deterministic	Sequential	Static	Discrete
Chess with a clock	Fully	Multi	Deterministic	Sequential	Semi	Discrete
Poker	Partially	Multi	Stochastic	Sequential	Static	Discrete
Backgammon	Fully	Multi	Stochastic	Sequential	Static	Discrete
Taxi driving	Partially	Multi	Stochastic	Sequential	Dynamic	Continuous
Medical diagnosis	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
Image analysis	Fully	Single	Deterministic	Episodic	Semi	Continuous
Part-picking robot	Partially	Single	Stochastic	Episodic	Dynamic	Continuous
Refinery controller	Partially	Single	Stochastic	Sequential	Dynamic	Continuous
English tutor	Partially	Multi	Stochastic	Sequential	Dynamic	Discrete

Four basic kinds of agent types

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

Simple reflex agent

- Select actions based on the **current** percept, ignoring the rest of the percept history.
- Agent function vs. agent program
  - 4^T vs 4, as history is ignored.



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*
⋮	⋮

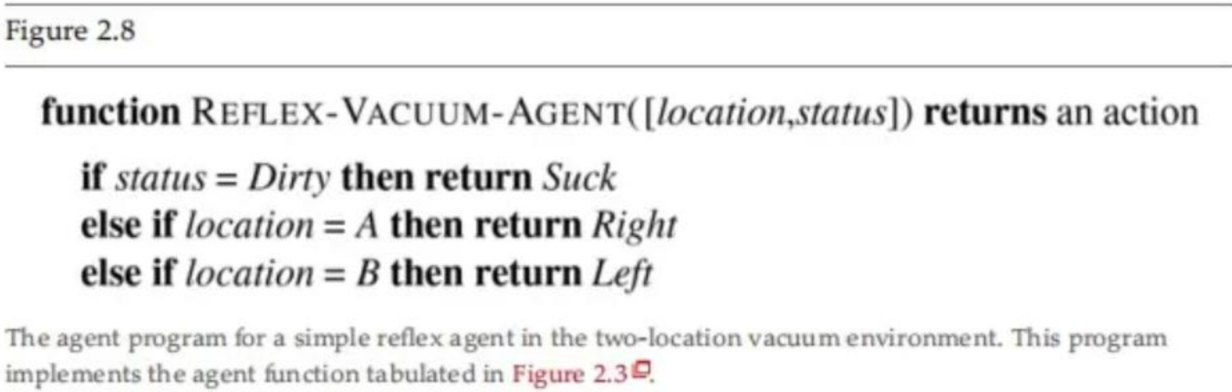


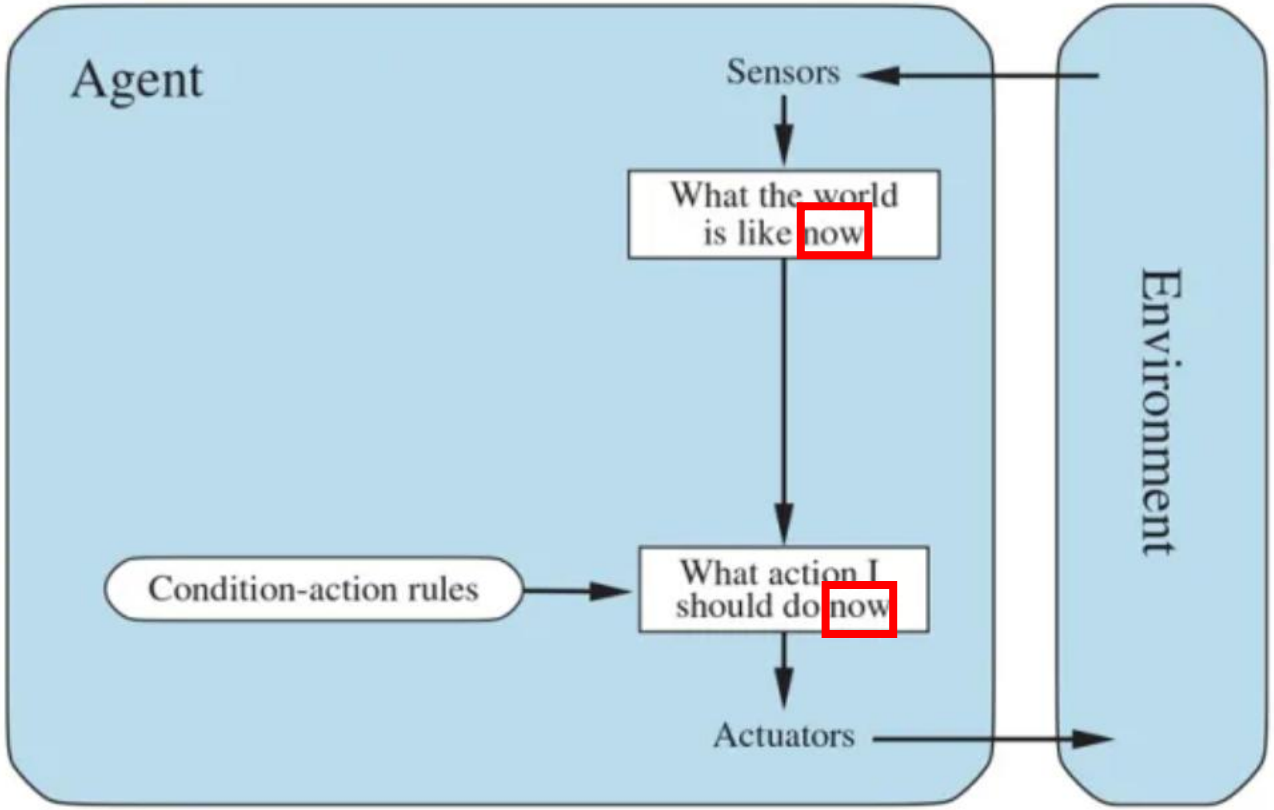
Figure 2.8

```
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
```

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

Simple reflex agent

- Condition-action rules
  - If car-in-front-is-braking
  - Then initiate-braking



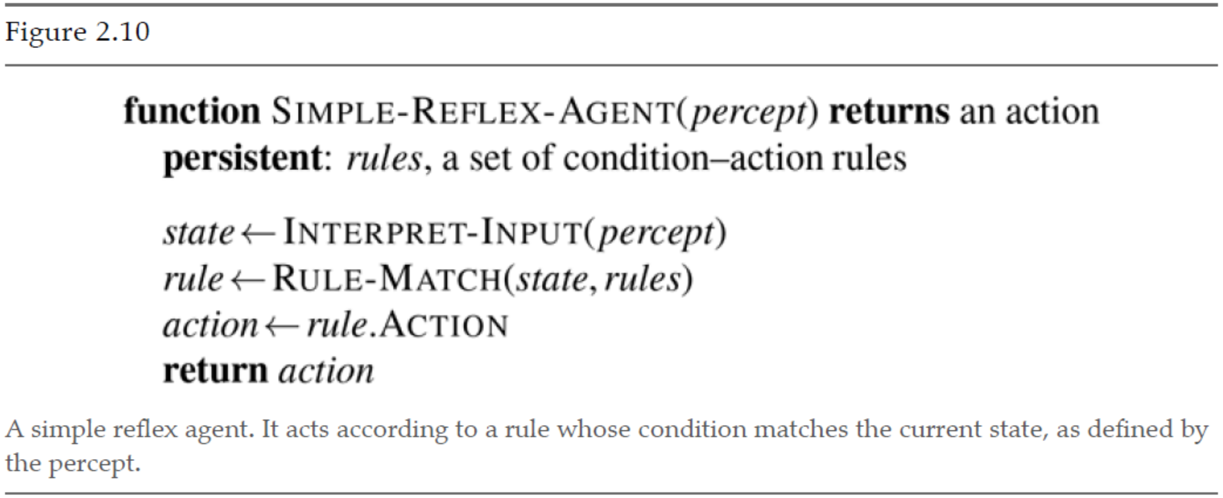


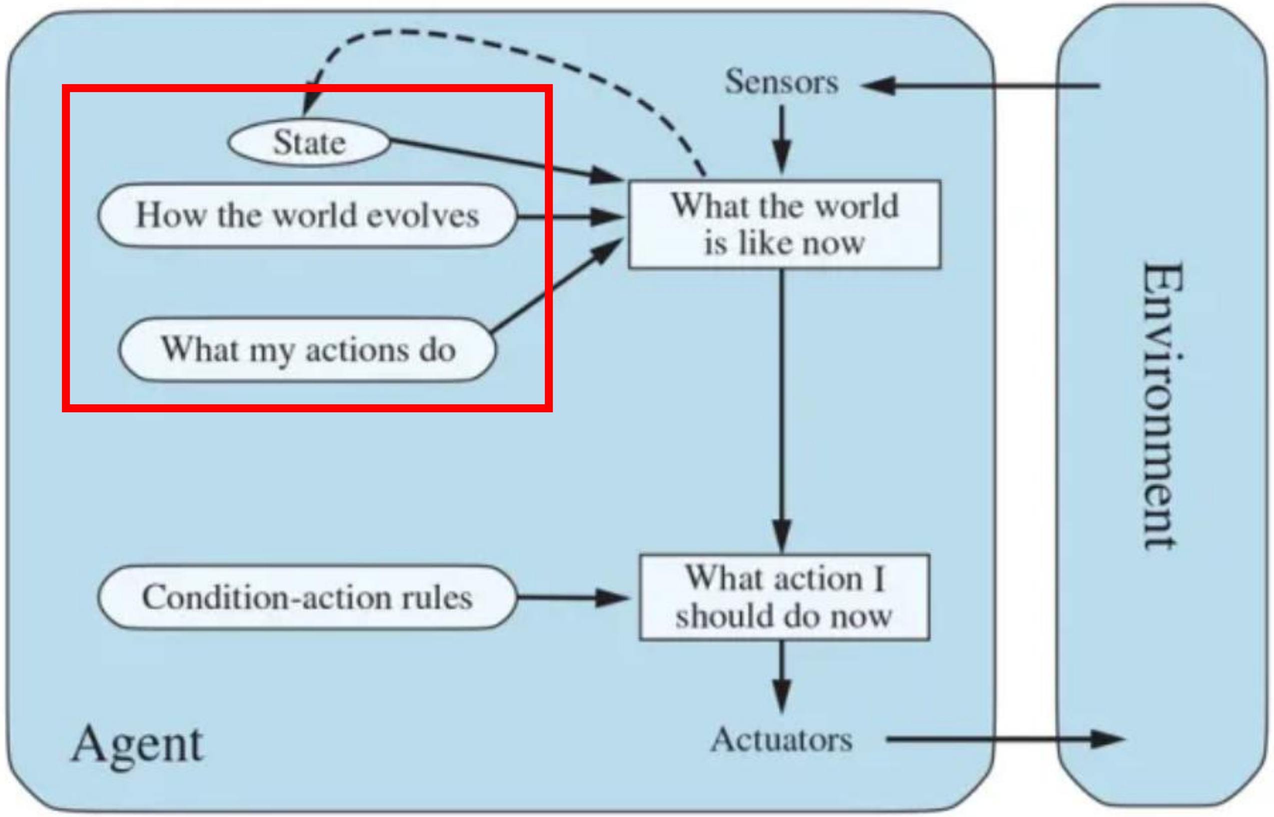
Figure 2.10

```
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
```

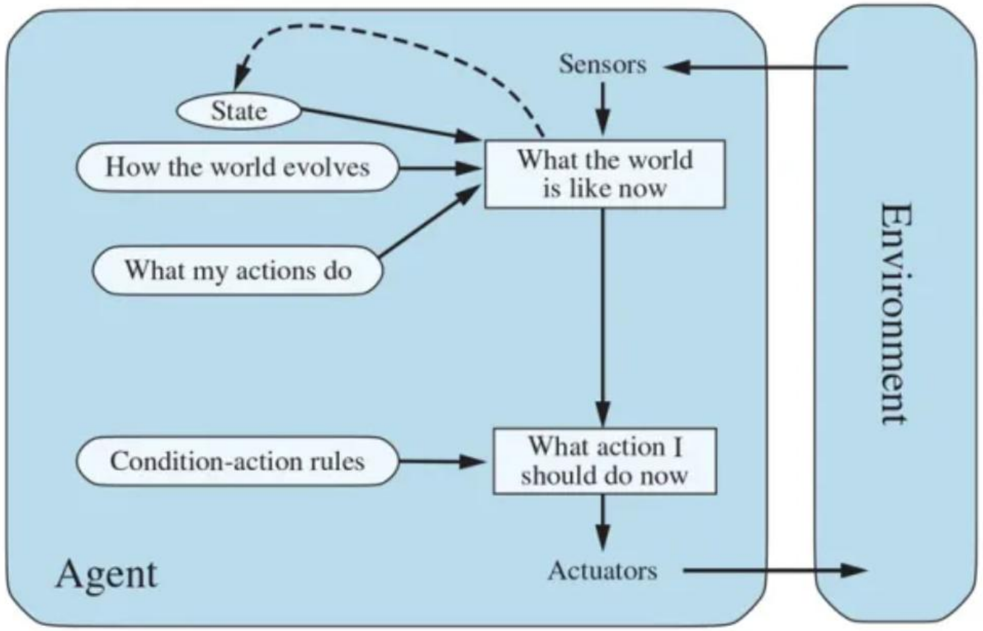
A simple reflex agent. It acts according to a rule whose condition matches the current state, as defined by the percept.

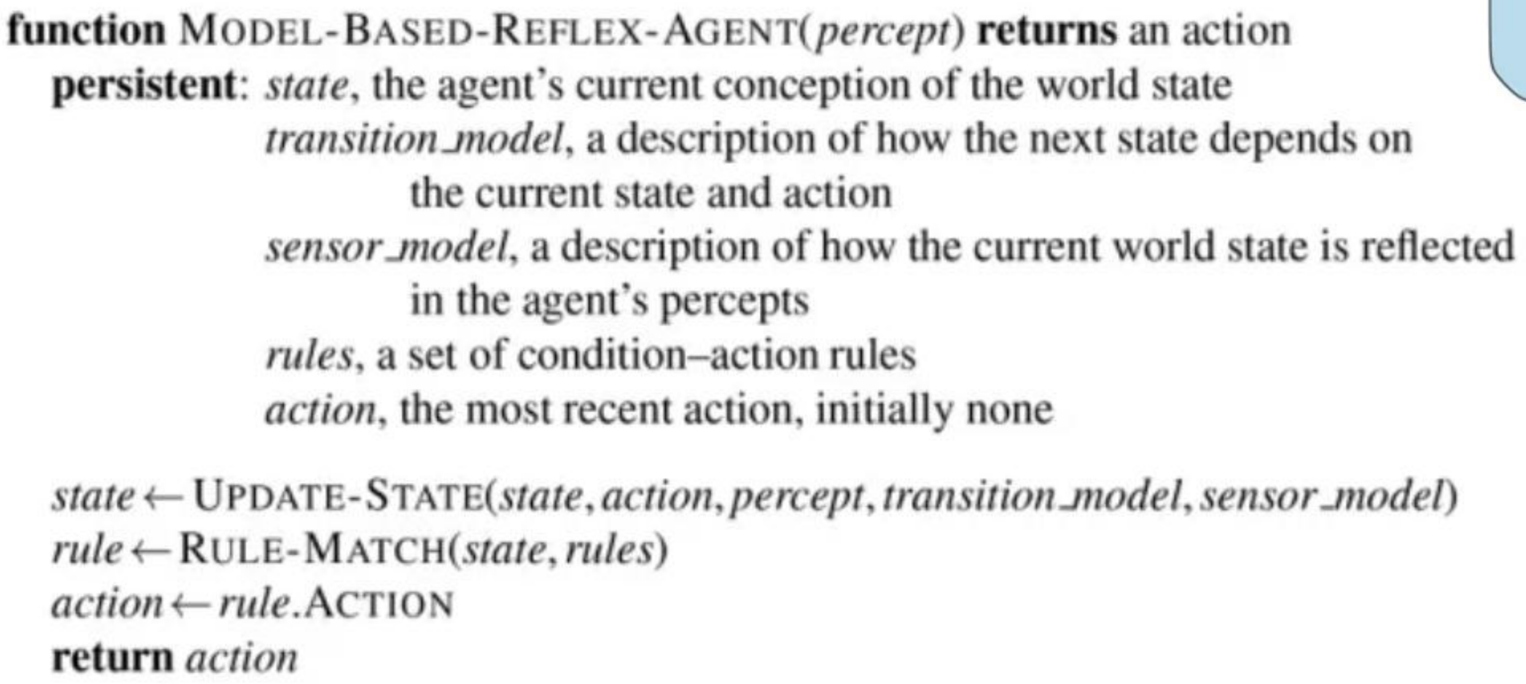
Model-based reflex agent

- The agent keeps track of the part of the world it can't see now.
  - Maintain some sort of internal state that depends on the percept history
  - Reflects some of the unobserved aspects of the current state.
- An effective way to handle partial observability



Model-based reflex agent

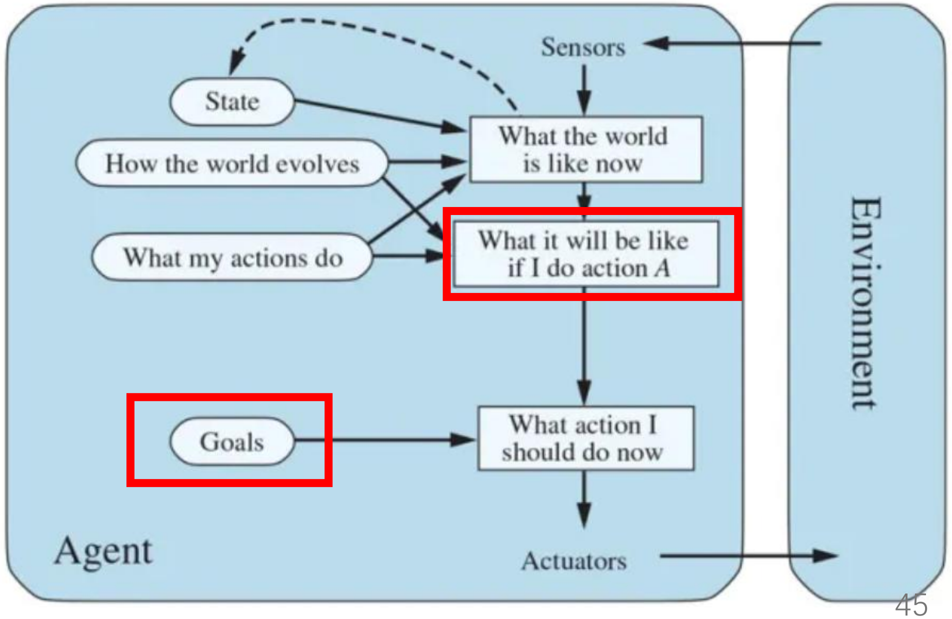




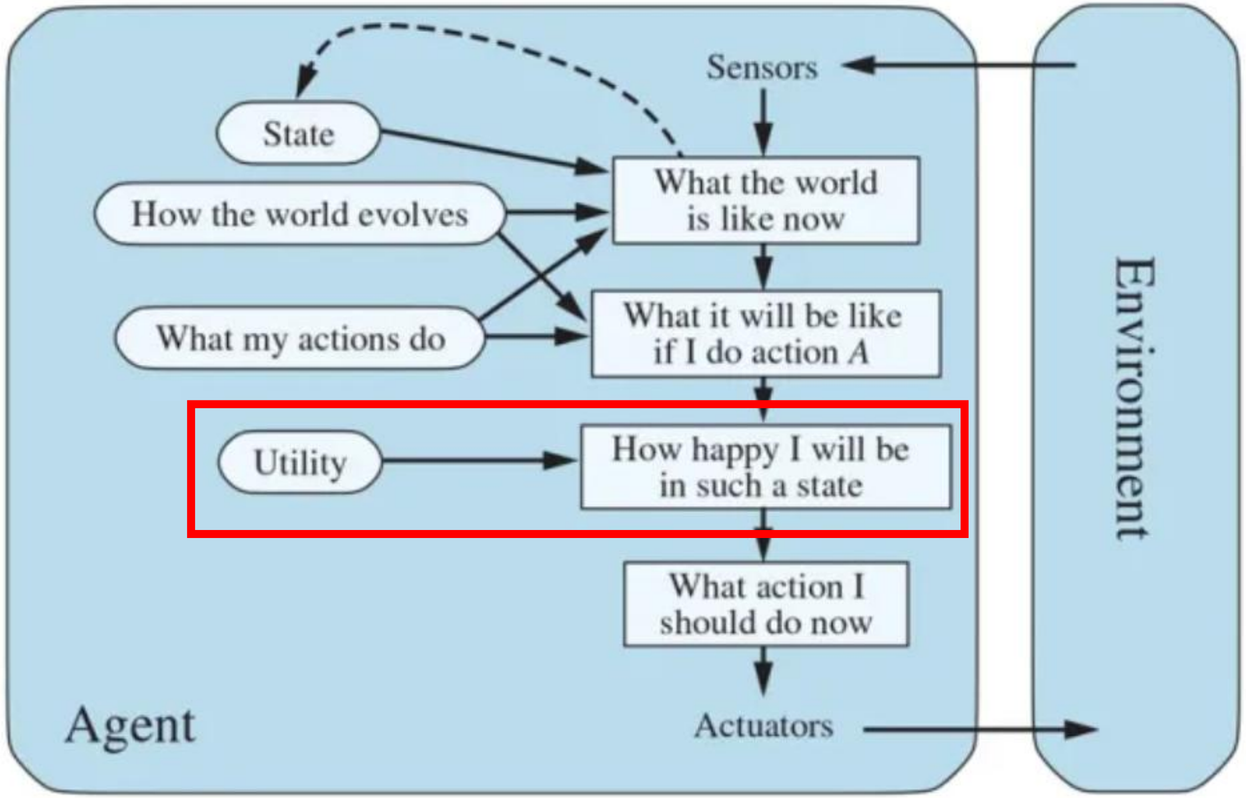
```
function MODEL-BASED-REFLEX-AGENT(percept) returns an action
  persistent: state, the agent's current conception of the world state
              transition_model, a description of how the next state depends on
                    the current state and action
              sensor_model, a description of how the current world state is reflected
                    in the agent's percepts
              rules, a set of condition–action rules
              action, the most recent action, initially none

  state ← UPDATE-STATE(state, action, percept, transition_model, sensor_model)
  rule ← RULE-MATCH(state, rules)
  action ← rule.ACTION
  return action
```

Goal-based agents

- Knowing current state of the environment is not always enough to decide what to do.
- At a road junction, the taxi can go left, right or straight. The correct decision is based on where the taxi is trying to go, the destination.
- What action to do next
  - The current state + goal
- Goal satisfaction results
  - Immediately from a single action
  - Consider long sequences of actions
- Less efficient, but more flexible

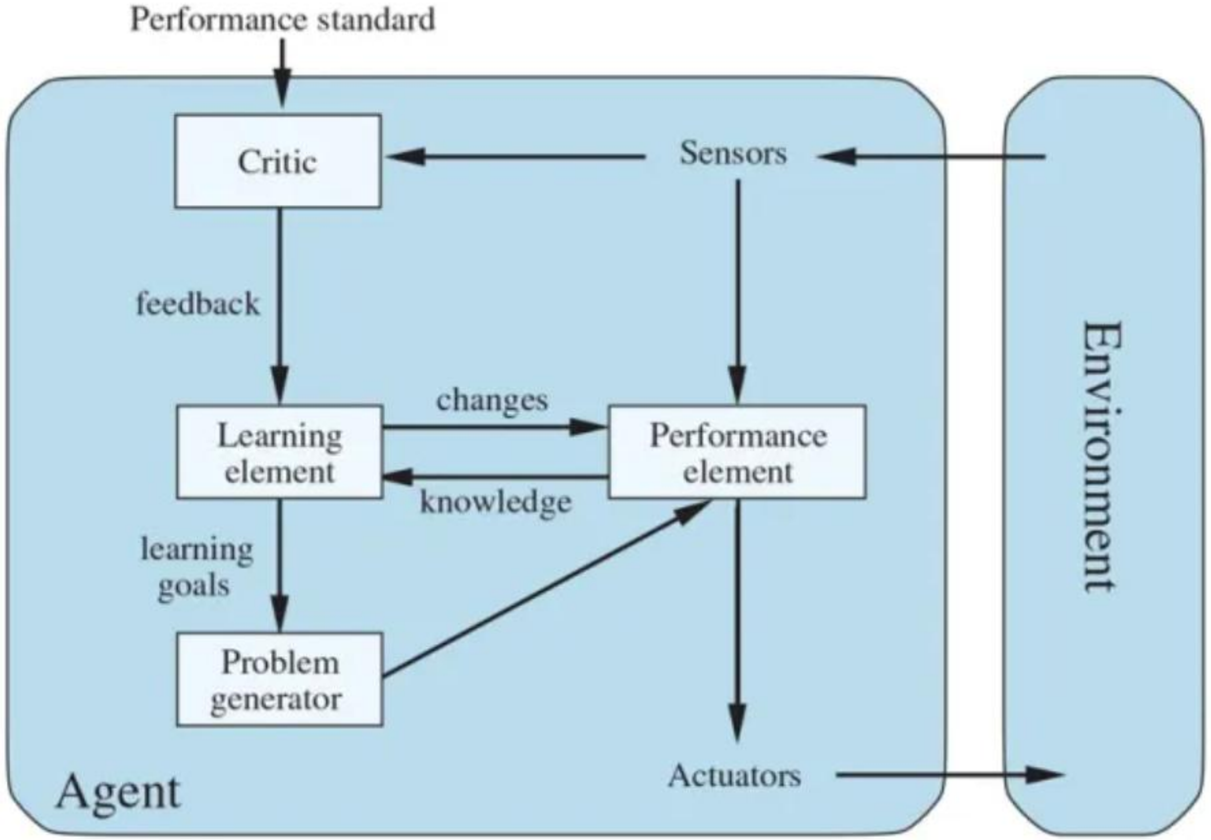

Sensors
State
What the world is like now
How the world evolves
What my actions do
What it will be like if I do action *A*
Goals
What action I should do now
Agent
Actuators
Environment

Utility-based agents


Sensors
State
How the world evolves
What my actions do
What the world is like now
What it will be like if I do action A
Utility
How happy I will be in such a state
What action I should do now
Agent
Actuators
Environment

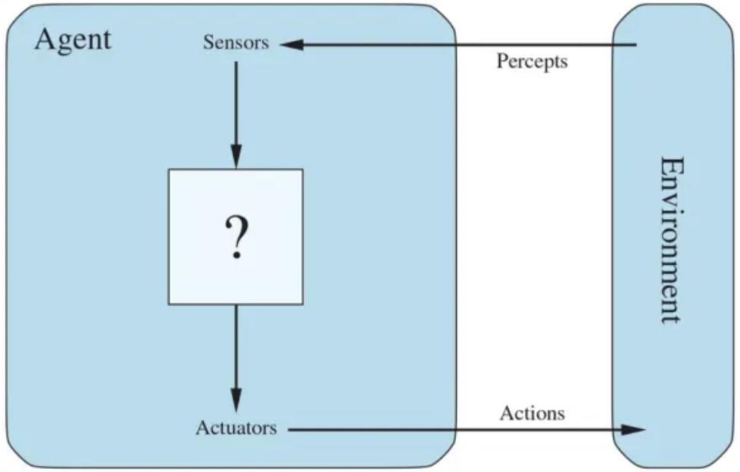
- Goals alone may not be enough
  - Many action sequences will get the taxi to its destination (all achieving the goal)
  - Some are quicker, safer, more reliable, cheaper, greener.
- Goals provide happy vs unhappy.
- Utility functions provide how happy.
  - Provide tradeoff between conflicting goals (speed vs. safety)
- Expected utility, decision making under uncertainty
  - When facing nondeterminism and partial observability

Learning agents

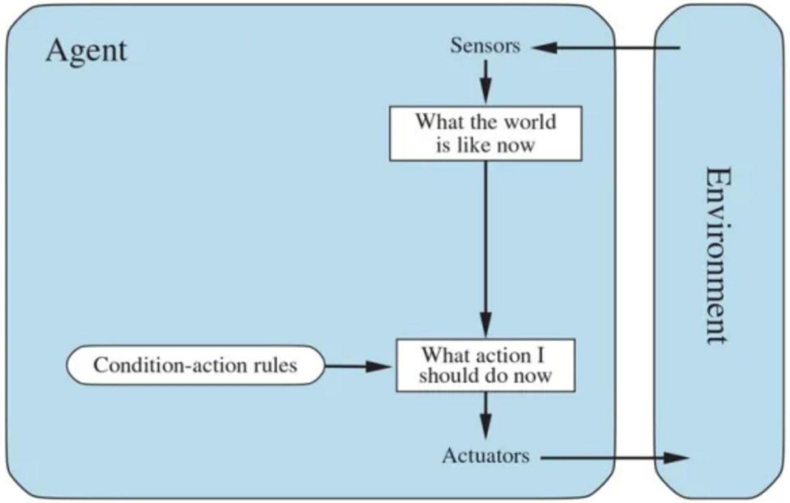


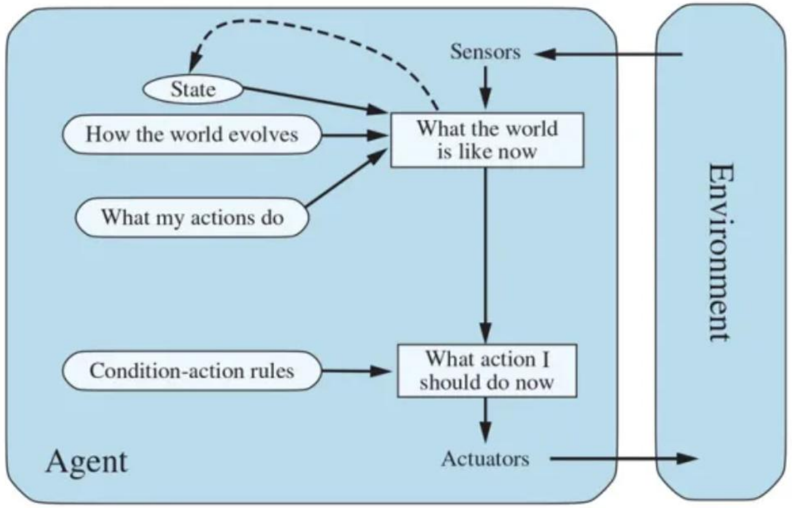
- Orthogonal to the four types
- Learning allows the agent to operate in initially unknown environments and become more competent.
- Performance element: previous agent, selection actions.
- Learning element: making improvements
- Critic: tells the learning element how the agent is doing, good or bad.
- Problem generator: suggesting actions that will lead to new and informative experiences.

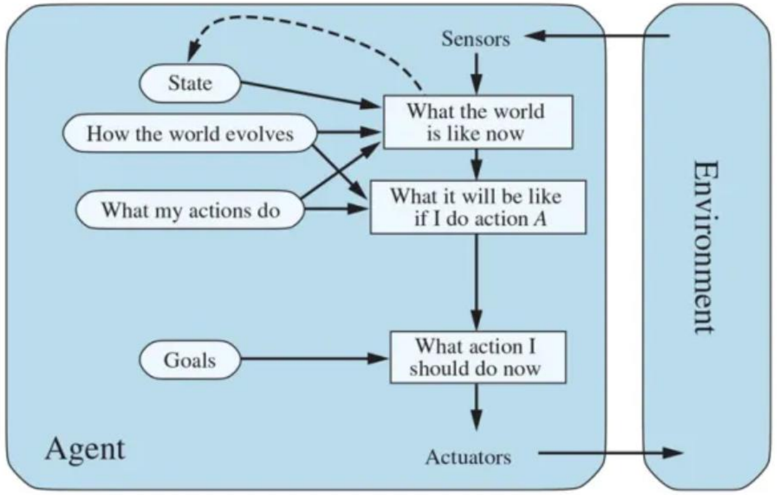
Figure 2.1

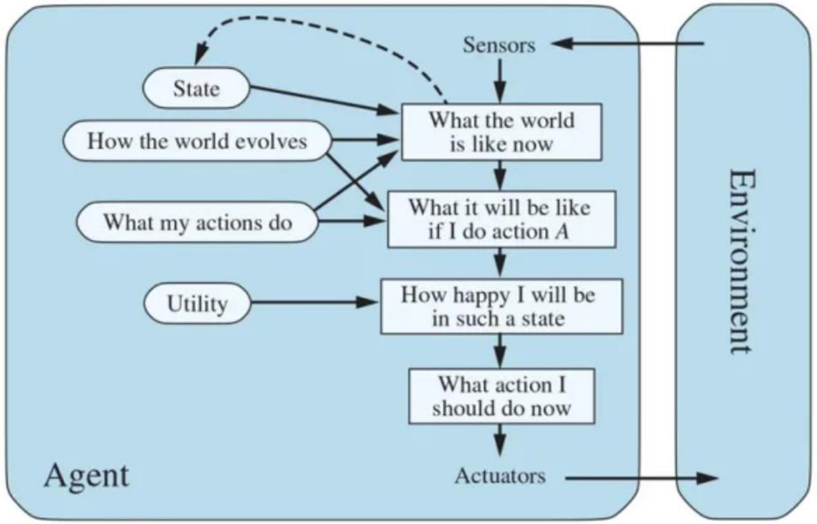


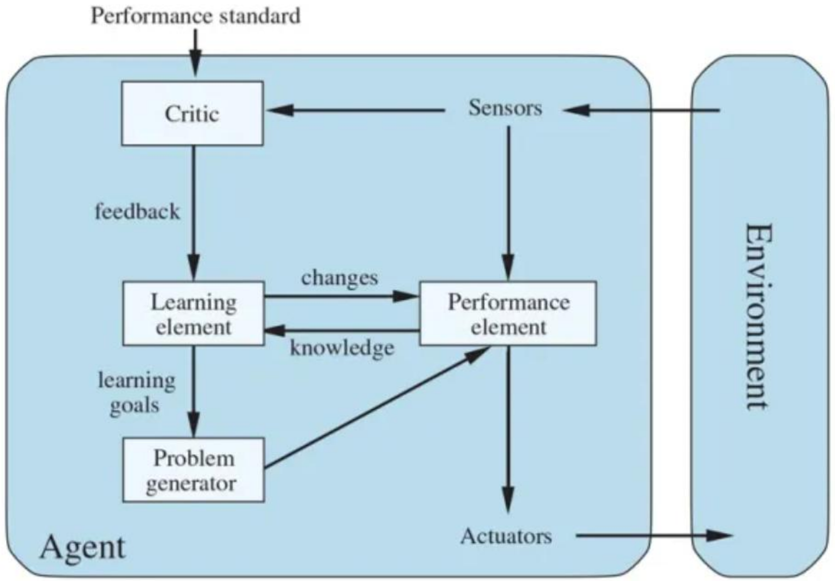
Agents interact with environments through sensors and actuators.






Sensors
State
How the world evolves
What the world is like now
What my actions do
What it will be like if I do action A
Goals
What action I should do now
Agent
Actuators
Environment


Sensors
State
How the world evolves
What the world is like now
What my actions do
What it will be like if I do action A
Utility
How happy I will be in such a state
What action I should do now
Agent
Actuators
Environment



Intended learning outcomes (ILOs)

- What is AI?
  - Understand the four approaches
    - Human vs. rationality
    - Behavior vs thought
  - Beneficial machines
    - Value alignment, AI risks
- AI history
- Agents
  - Key concepts: agent, environment, agent function, agent program, rational agent
  - Performance measure and rational agent
    - Maximize expected performance measure
  - Task environments, and their categorization
    - PEAS, Fully/Partially observable, single-/multi-agent, (non)deterministic, episodic vs. sequential, static vs. dynamic, discrete vs. continuous, and (un)known.
  - Agent programs, and their categorization
    - Simple reflex agents, model-based reflex agents, Goal-based agents, Utility-based agents
    - Learning agents