

Lecture

Foundations of Artificial Intelligence

Part 3 – Agents

Dr. Mohsen Mesgar

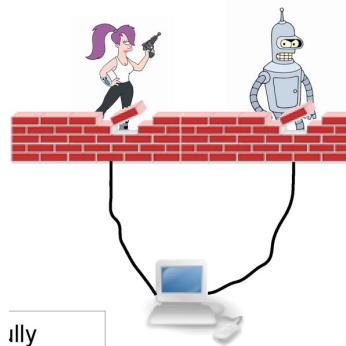
Universität Duisburg-Essen

Recall...

- **What is (artificial) intelligence?** The ability to acquire and apply knowledge and skills to achieve complex goals.

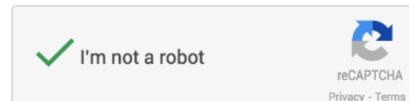
Recall...

- **What is (artificial) intelligence?** The ability to acquire and apply knowledge and skills to achieve complex goals.
 - Turing Test



- Captchas

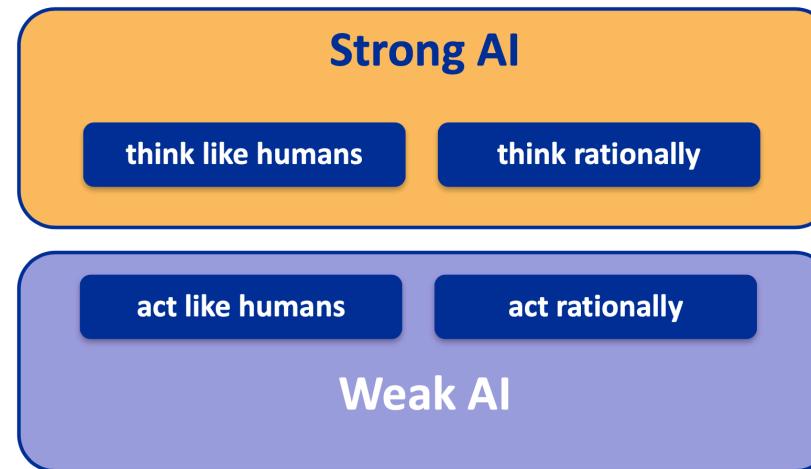
CAPTCHA:
Completely Automated Public Turing test to tell Computers and Humans Apart



Recall...

- **What is (artificial) intelligence?** The ability to acquire and apply knowledge and skills to achieve complex goals.
 - Turing Test
 - Captchas

- Strong AI vs weak AI



- Ethical concerns

Any other open questions?



Goal: In this lecture you learn ...

- PEAS framework
- Performance measure
- Environment
 - How to **formulate** it?
- Agent
 - **Architecture** and **program**
 - **Agent types**
 - **Simple reflex** agents
 - **Model-based** agents
 - **Goal-based** agents
 - **Utility-based** agents
 - **Learning-based** agents

(Artificial) Intelligence

- AI is the ability to achieve a **goal** or conduct a task
- A clear definition of **goal/task** helps find and formulate interesting problems

What is the goal of sudoku?

- To fill a 9×9 grid with digits so that each column, each row, and each of the nine 3×3 subgrids that compose the grid contain all of the digits from 1 to 9

5	3			7				
6			1	9	5			
	9	8				6		
8				6			3	
4			8	3			1	
7				2			6	
	6				2	8		
			4	1	9		5	
			8			7	9	

What is the goal of chess?

- **To checkmate the opponent's king,** whereby the king is under immediate attack (in "check") and there is no way for it to escape.



What is the goal of backgammon?

- **To win money by capturing the pot,**
which contains bets made by various
players during the hand.



What is the goal of poker?

- To move the fifteen pieces around the board and be first to bear off, i.e. remove them from the board.



What is the goal of taxi driving?

- Driving safely from source to destination with enough hospitality
- ...



What is the goal of image analysis?

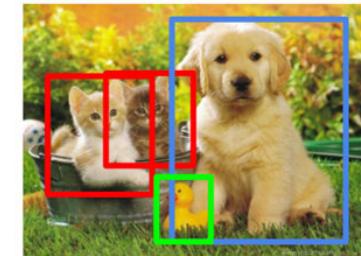
- Interpreting the content of an image
 - Detecting objects
 - Classifying an image

Classification



CAT

Object Detection



CAT, DOG, DUCK

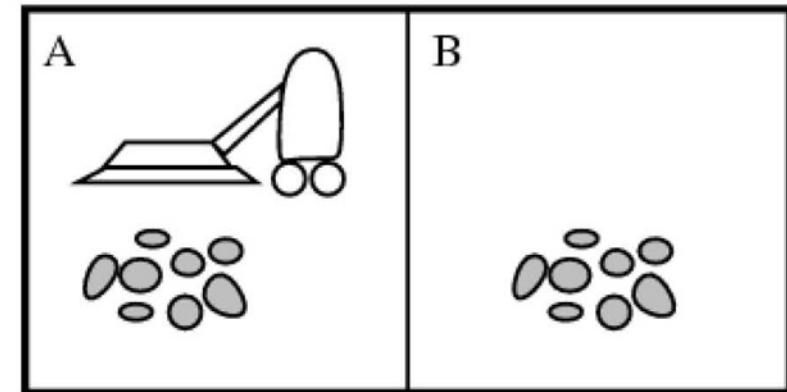
What is the goal of part picking?

- A simple goal of part-picking is to choose parts that come off a production line or from a previous operation



What is the goal of the vacuum-cleaner world?

Goal: To clean a house consisting of two rooms where each room may or may not have dust



PEAS Framework

PEAS framework

- This framework specifies the setting for achieving a goal or conducting a task using AI

PEAS framework

- This framework specifies the setting for achieving a goal or conducting a task using AI
 - PEAS:
 - Performance measure,
 - Environment,
 - Actuators,
 - Sensors
-
- ```
graph LR; PM[Performance measure] --- Env[Environment]; PM --- Agent[Agent]; PM --- AS[Actuators, Sensors]; subgraph AS []; direction TB; AS1[Actuators] --- AS2[Sensors]; end; AS --- Agent;
```

# Performance Measure

A function that evaluates how successfully a defined goal is achieved?

Vacuum-World

Reward for the amount of dust cleaned

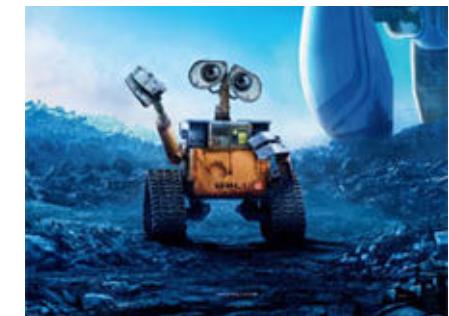
- one point per square cleaned up in time T
- can be maximized by dumping dust on the floor again...

Reward for clean floors

- one point per clean square per time step
- might move around too much

Penalty for consumed energy

- minus one per move?

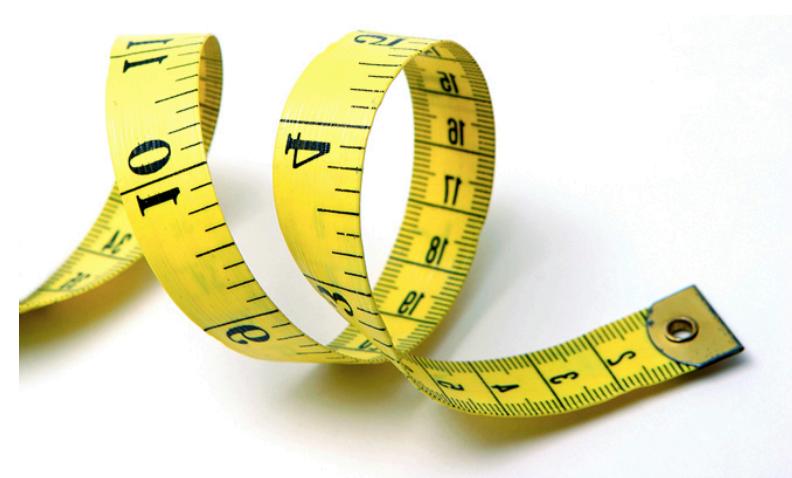


# Performance Measure

General rule for designing performance measures:

- **Based on desired goal**
- **Not on desired AI behavior**

→ What do you want?



# Environment

# Formulation: Environment variables

- **Observability:** fully vs partial
- **Certainty:** deterministic vs strategic vs stochastic
- **Temporal succession:** episodic vs sequential
- **Continuity:** static vs dynamic
- **Scale:** discrete vs continuous
- **Population:** single-agent vs multi-agent

# Observability

## Fully observable

- the complete state of the environment can be observed (relevant parts)
- no need to keep track of internal states

## Partially observable

- parts of the environment cannot be observed



# Observability

- Fully vs partial

| Task   | Observability |
|--------|---------------|
| Sudoku | Fully         |

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| 5 | 3 |   | 7 |   |   |   |
| 6 |   |   | 1 | 9 | 5 |   |
|   | 9 | 8 |   |   |   | 6 |
| 8 |   |   | 6 |   |   | 3 |
| 4 |   | 8 | 3 |   |   | 1 |
| 7 |   | 2 |   |   |   | 6 |
|   | 6 |   |   | 2 | 8 |   |
|   |   | 4 | 1 | 9 |   | 5 |
|   |   | 8 |   | 7 | 9 |   |

# Observability

- Fully vs partial

| Task             | Observability |
|------------------|---------------|
| Sudoku           | Fully         |
| Chess with clock | Fully         |



# Observability

- Fully vs partial

| Task             | Observability |
|------------------|---------------|
| Sudoku           | Fully         |
| Chess with clock | Fully         |
| Poker            | Partial       |



# Observability

- Fully vs partial

| Task             | Observability |
|------------------|---------------|
| Sudoku           | Fully         |
| Chess with clock | Fully         |
| Poker            | Partial       |
| Backgammon       | Fully         |



# Observability

- Fully vs partial

| Task             | Observability |
|------------------|---------------|
| Sudoku           | Fully         |
| Chess with clock | Fully         |
| Poker            | Partial       |
| Backgammon       | Fully         |
| Taxi Driving     | Partial       |



# Observability

- Fully vs partial

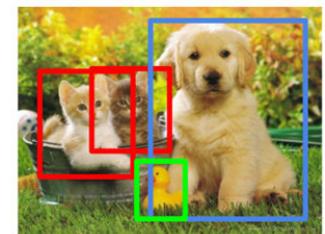
| Task             | Observability |
|------------------|---------------|
| Sudoku           | Fully         |
| Chess with clock | Fully         |
| Poker            | Partial       |
| Backgammon       | Fully         |
| Taxi Driving     | Partial       |
| Image Analysis   | Fully         |

Classification



CAT

Object Detection



CAT, DOG, DUCK

# Observability

- Fully vs partial

| Task             | Observability |
|------------------|---------------|
| Sudoku           | Fully         |
| Chess with clock | Fully         |
| Poker            | Partial       |
| Backgammon       | Fully         |
| Taxi Driving     | Partial       |
| Image Analysis   | Fully         |
| Part-picking     | Partial       |



## Deterministic

- next environment state is determined only by current state and executed action

## Strategic

- only the opponents' actions cannot be foreseen

## Stochastic

- next environment state is uncertain (e.g., throwing dices).



# Certainty

- Deterministic vs strategic vs stochastic

| Task   | Certainty     |
|--------|---------------|
| Sudoku | Deterministic |

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| 5 | 3 |   | 7 |   |   |   |
| 6 |   |   | 1 | 9 | 5 |   |
|   | 9 | 8 |   |   |   | 6 |
| 8 |   |   | 6 |   |   | 3 |
| 4 |   | 8 | 3 |   |   | 1 |
| 7 |   | 2 |   |   |   | 6 |
|   | 6 |   |   | 2 | 8 |   |
|   |   | 4 | 1 | 9 |   | 5 |
|   |   | 8 |   | 7 | 9 |   |

# Certainty

- Deterministic vs strategic vs stochastic

| Task             | Certainty     |
|------------------|---------------|
| Sudoku           | Deterministic |
| Chess with clock | Strategic     |



# Certainty

- Deterministic vs strategic vs stochastic

| Task             | Certainty     |
|------------------|---------------|
| Sudoku           | Deterministic |
| Chess with clock | Strategic     |
| Poker            | Stochastic    |



# Certainty

- Deterministic vs strategic vs stochastic

| Task             | Certainty     |
|------------------|---------------|
| Sudoku           | Deterministic |
| Chess with clock | Strategic     |
| Poker            | Stochastic    |
| Backgammon       | Stochastic    |



# Certainty

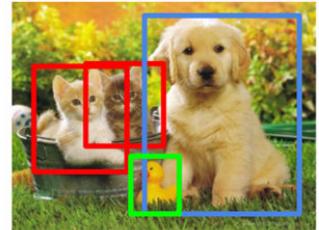
- Deterministic vs strategic vs stochastic

| Task             | Certainty     |
|------------------|---------------|
| Sudoku           | Deterministic |
| Chess with clock | Strategic     |
| Poker            | Stochastic    |
| Backgammon       | Stochastic    |
| Taxi Driving     | Stochastic    |



# Certainty

- Deterministic vs strategic vs stochastic

| Task             | Certainty     | Classification                                                                             | Object Detection                                                                                      |
|------------------|---------------|--------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|
| Sudoku           | Deterministic |                                                                                            |                                                                                                       |
| Chess with clock | Strategic     |                                                                                            |                                                                                                       |
| Poker            | Stochastic    |                                                                                            |                                                                                                       |
| Backgammon       | Stochastic    |                                                                                            |                                                                                                       |
| Taxi Driving     | Stochastic    |                                                                                            |                                                                                                       |
| Image Analysis   | Deterministic | <br>CAT | <br>CAT, DOG, DUCK |

# Certainty

- Deterministic vs strategic vs stochastic

| Task             | Certainty     |
|------------------|---------------|
| Sudoku           | Deterministic |
| Chess with clock | Strategic     |
| Poker            | Stochastic    |
| Backgammon       | Stochastic    |
| Taxi Driving     | Stochastic    |
| Image Analysis   | Deterministic |
| Part-picking     | Stochastic    |



# Temporal succession

## Episodic

- If the choice of an action does not depend on previous actions

## Sequential

- the current action affect future actions

# Temporal succession

- Sequential vs Episodic

| Task   | Succession |
|--------|------------|
| Sudoku | Sequential |

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| 5 | 3 |   | 7 |   |   |   |
| 6 |   |   | 1 | 9 | 5 |   |
|   | 9 | 8 |   |   |   | 6 |
| 8 |   |   | 6 |   |   | 3 |
| 4 |   | 8 | 3 |   |   | 1 |
| 7 |   | 2 |   |   |   | 6 |
|   | 6 |   |   | 2 | 8 |   |
|   |   | 4 | 1 | 9 |   | 5 |
|   |   | 8 |   | 7 | 9 |   |

# Temporal succession

- Sequential vs Episodic

| Task             | Succession |
|------------------|------------|
| Sudoku           | Sequential |
| Chess with clock | Sequential |



# Temporal succession

- Sequential vs Episodic

| Task             | Succession |
|------------------|------------|
| Sudoku           | Sequential |
| Chess with clock | Sequential |
| Poker            | Sequential |



# Temporal succession

- Sequential vs Episodic

| Task             | Succession |
|------------------|------------|
| Sudoku           | Sequential |
| Chess with clock | Sequential |
| Poker            | Sequential |
| Backgammon       | Sequential |



# Temporal succession

- Sequential vs Episodic

| Task             | Succession |
|------------------|------------|
| Sudoku           | Sequential |
| Chess with clock | Sequential |
| Poker            | Sequential |
| Backgammon       | Sequential |
| Taxi Driving     | Sequential |



# Temporal succession

- Sequential vs Episodic

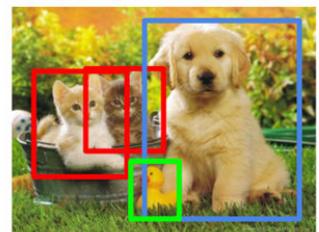
| Task             | Succession |
|------------------|------------|
| Sudoku           | Sequential |
| Chess with clock | Sequential |
| Poker            | Sequential |
| Backgammon       | Sequential |
| Taxi Driving     | Sequential |
| Image Analysis   | Episodic   |

Classification



CAT

Object Detection



CAT, DOG, DUCK

# Temporal succession

- Sequential vs Episodic

| Task             | Succession |
|------------------|------------|
| Sudoku           | Sequential |
| Chess with clock | Sequential |
| Poker            | Sequential |
| Backgammon       | Sequential |
| Taxi Driving     | Sequential |
| Image Analysis   | Episodic   |
| Part-picking     | Episodic   |



## Static

- the environment does not change

## Semi-dynamic

- the environment does not change but the performance score may

## Dynamic

- the environment may change

# Continuity

- Static vs semi-dynamic vs dynamic

| Task   | Continuity |
|--------|------------|
| Sudoku | Static     |

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| 5 | 3 |   | 7 |   |   |   |
| 6 |   |   | 1 | 9 | 5 |   |
|   | 9 | 8 |   |   |   | 6 |
| 8 |   |   | 6 |   |   | 3 |
| 4 |   | 8 | 3 |   |   | 1 |
| 7 |   | 2 |   |   |   | 6 |
|   | 6 |   |   | 2 | 8 |   |
|   |   | 4 | 1 | 9 |   | 5 |
|   |   | 8 |   | 7 | 9 |   |

# Continuity

- Static vs semi-dynamic vs dynamic

| Task             | Continuity |
|------------------|------------|
| Sudoku           | Static     |
| Chess with clock | Semi       |



# Continuity

- Static vs semi-dynamic vs dynamic

| Task             | Continuity |
|------------------|------------|
| Sudoku           | Static     |
| Chess with clock | Semi       |
| Poker            | Static     |



# Continuity

- Static vs semi-dynamic vs dynamic

| Task             | Continuity |
|------------------|------------|
| Sudoku           | Static     |
| Chess with clock | Semi       |
| Poker            | Static     |
| Backgammon       | Static     |



# Continuity

- Static vs semi-dynamic vs dynamic

| Task             | Continuity |
|------------------|------------|
| Sudoku           | Static     |
| Chess with clock | Semi       |
| Poker            | Static     |
| Backgammon       | Static     |
| Taxi Driving     | Dynamic    |



# Continuity

- Static vs semi-dynamic vs dynamic

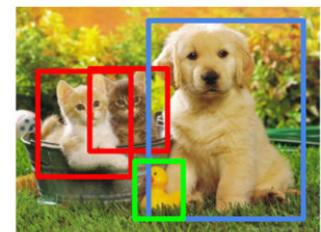
| Task             | Continuity |
|------------------|------------|
| Sudoku           | Static     |
| Chess with clock | Semi       |
| Poker            | Static     |
| Backgammon       | Static     |
| Taxi Driving     | Dynamic    |
| Image Analysis   | Semi       |

Classification



CAT

Object Detection



CAT, DOG, DUCK

# Continuity

- Static vs semi-dynamic vs dynamic

| Task             | Continuity |
|------------------|------------|
| Sudoku           | Static     |
| Chess with clock | Semi       |
| Poker            | Static     |
| Backgammon       | Static     |
| Taxi Driving     | Dynamic    |
| Image Analysis   | Semi       |
| Part-picking     | Dynamic    |

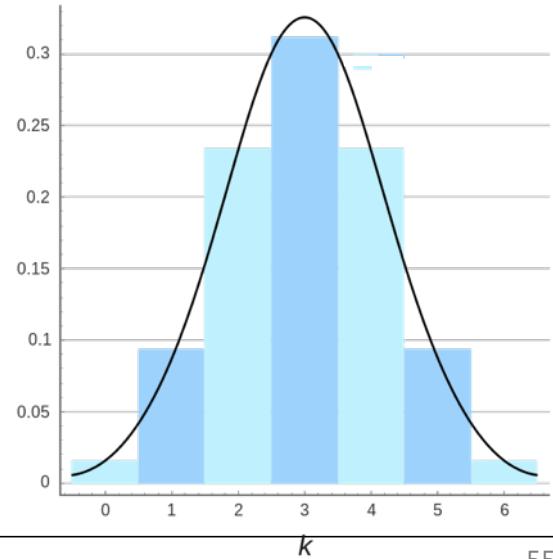


## Discrete

- A finite number of environment states / percepts

## Continuous

- A continuous range of values
  - This distinction applies separately to
    - actions
    - states
    - percepts
- Can be mixed in individual tasks



# Scale

- Continuous vs discrete

| Task   | Scale    |
|--------|----------|
| Sudoku | Discrete |

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| 5 | 3 |   | 7 |   |   |   |
| 6 |   |   | 1 | 9 | 5 |   |
|   | 9 | 8 |   |   |   | 6 |
| 8 |   |   | 6 |   |   | 3 |
| 4 |   | 8 | 3 |   |   | 1 |
| 7 |   | 2 |   |   |   | 6 |
|   | 6 |   |   | 2 | 8 |   |
|   |   | 4 | 1 | 9 |   | 5 |
|   |   | 8 |   | 7 | 9 |   |

# Scale

- Continuous vs discrete

| Task             | Scale    |
|------------------|----------|
| Sudoku           | Discrete |
| Chess with clock | Discrete |



# Scale

- Continuous vs discrete

| Task             | Scale    |
|------------------|----------|
| Sudoku           | Discrete |
| Chess with clock | Discrete |
| Poker            | Discrete |



# Scale

- Continuous vs discrete

| Task             | Scale    |
|------------------|----------|
| Sudoku           | Discrete |
| Chess with clock | Discrete |
| Poker            | Discrete |
| Backgammon       | Discrete |



# Scale

- Continuous vs discrete

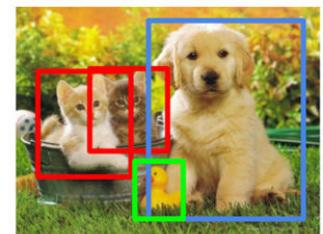
| Task             | Scale      |
|------------------|------------|
| Sudoku           | Discrete   |
| Chess with clock | Discrete   |
| Poker            | Discrete   |
| Backgammon       | Discrete   |
| Taxi Driving     | Continuous |



# Scale

- Continuous vs discrete

| Task             | Scale      | Classification | Object Detection |
|------------------|------------|----------------|------------------|
| Sudoku           | Discrete   |                |                  |
| Chess with clock | Discrete   |                |                  |
| Poker            | Discrete   |                |                  |
| Backgammon       | Discrete   |                |                  |
| Taxi Driving     | Continuous | CAT            | CAT, DOG, DUCK   |
| Image Analysis   | Continuous |                |                  |



CAT

# Scale

- Continuous vs discrete

| Task             | Scale      |
|------------------|------------|
| Sudoku           | Discrete   |
| Chess with clock | Discrete   |
| Poker            | Discrete   |
| Backgammon       | Discrete   |
| Taxi Driving     | Continuous |
| Image Analysis   | Continuous |
| Part-picking     | Continuous |



## Single-agent

- No other agents
- Other agents may be part of the environment though

## Multi-agent

- Does the environment contain other agents whose performance measures depend on my actions?
- Other agents may be *co-operative* or *competitive*

# Population

- Single-agent vs multi-agent

| Task   | Population |
|--------|------------|
| Sudoku | Single     |

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| 5 | 3 |   | 7 |   |   |   |
| 6 |   |   | 1 | 9 | 5 |   |
|   | 9 | 8 |   |   |   | 6 |
| 8 |   |   | 6 |   |   | 3 |
| 4 |   | 8 | 3 |   |   | 1 |
| 7 |   | 2 |   |   |   | 6 |
|   | 6 |   |   | 2 | 8 |   |
|   |   | 4 | 1 | 9 |   | 5 |
|   |   | 8 |   | 7 | 9 |   |

# Population

- Single-agent vs multi-agent

| Task             | Population |
|------------------|------------|
| Sudoku           | Single     |
| Chess with clock | Multi      |



# Population

- Single-agent vs multi-agent

| Task             | Population |
|------------------|------------|
| Sudoku           | Single     |
| Chess with clock | Multi      |
| Poker            | Multi      |



# Population

- Single-agent vs multi-agent

| Task             | Population |
|------------------|------------|
| Sudoku           | Single     |
| Chess with clock | Multi      |
| Poker            | Multi      |
| Backgammon       | Multi      |



# Population

- Single-agent vs multi-agent

| Task             | Population |
|------------------|------------|
| Sudoku           | Single     |
| Chess with clock | Multi      |
| Poker            | Multi      |
| Backgammon       | Multi      |
| Taxi Driving     | Multi      |



# Population

- Single-agent vs multi-agent

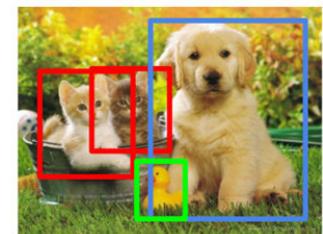
| Task             | Population |
|------------------|------------|
| Sudoku           | Single     |
| Chess with clock | Multi      |
| Poker            | Multi      |
| Backgammon       | Multi      |
| Taxi Driving     | Multi      |
| Image Analysis   | Single     |

Classification



CAT

Object Detection



CAT, DOG, DUCK

# Population

- Single-agent vs multi-agent

| Task             | Population |
|------------------|------------|
| Sudoku           | Single     |
| Chess with clock | Multi      |
| Poker            | Multi      |
| Backgammon       | Multi      |
| Taxi Driving     | Multi      |
| Image Analysis   | Single     |
| Part-picking     | Multi      |



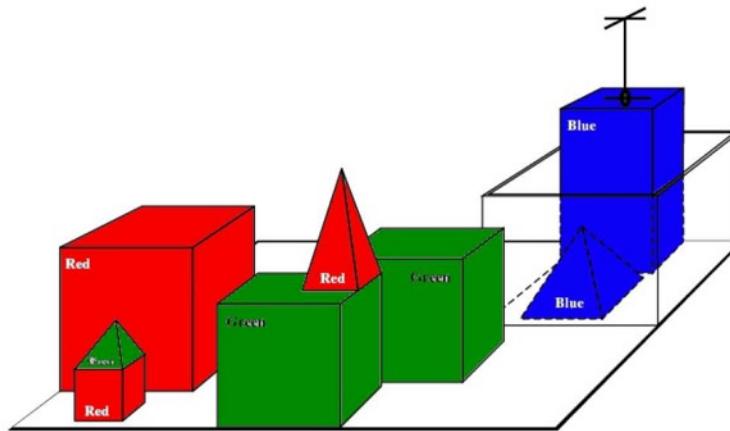
# Environment descriptions

| Task           | Observability | Certainty     | Succession | Continuity | Scale      | Population |
|----------------|---------------|---------------|------------|------------|------------|------------|
| Sudoku         | Fully         | Deterministic | Sequential | Static     | Discrete   | Single     |
| Chess w. clock | Fully         | Strategic     | Sequential | Semi       | Discrete   | Multi      |
| Poker          | Partially     | Stochastic    | Sequential | Static     | Discrete   | Multi      |
| Backgammon     | Fully         | Stochastic    | Sequential | Static     | Discrete   | Multi      |
| Taxi Driving   | Partially     | Stochastic    | Sequential | Dynamic    | Continuous | Multi      |
| Image Analysis | Fully         | Deterministic | Episodic   | Semi       | Continuous | Single     |
| Part-picking   | Partially     | Stochastic    | Episodic   | Dynamic    | Continuous | Multi      |

# Challenging environments

Rather simple:

- fully observable
- deterministic
- episodic
- static
- discrete
- single-agent



Pretty hard:

- ◊ partially observable
- ◊ stochastic
- ◊ sequential
- ◊ dynamic
- ◊ continuous
- ◊ multi-agent



# Agent

- **What is an agent?**

An entity that **perceives (observes)** an environment through its **sensors** and **acts** upon that environment through **actuators**

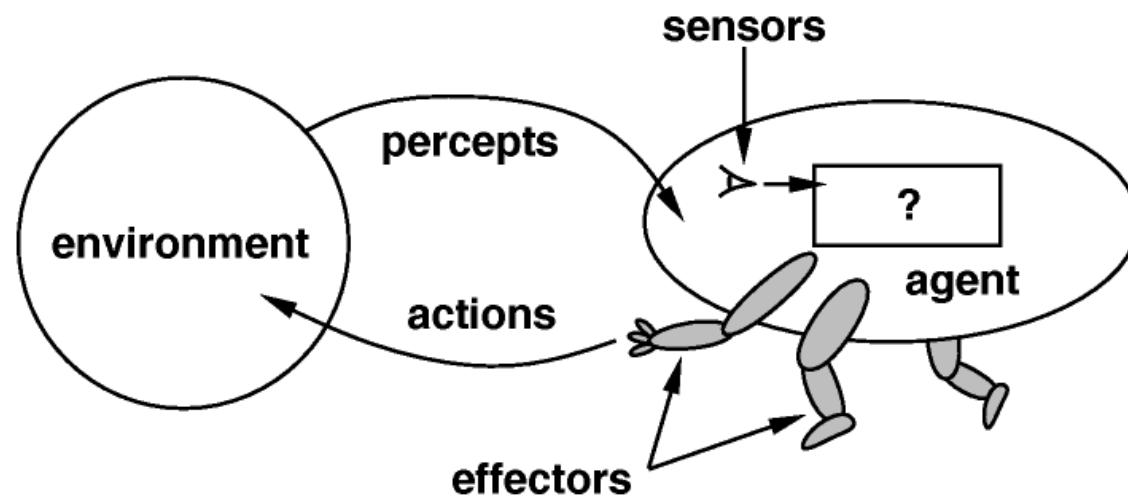
- **Human agents**

- Sensors: eyes, ears, and other organs for sensors
- Actuators: hands, legs, mouth, and other body parts for actuators

- **Artificial agents**

- Sensors: Camera and infrared range finders, microphone, keyboard, ...
- Actuators: Various motors for movement, displays, sound generator, ...

# Agents and environment



# Example: The vacuum-cleaner world

A robot-vacuum-cleaner that operates in a very simple world.

## Environment:

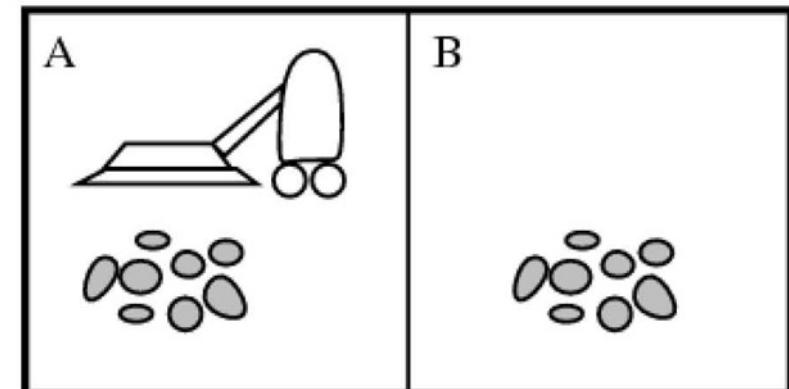
- Virtual house with room A and room B

## Percepts:

- The robot can sense pairs [**<location>**, **<status>**]
  - Location: room A or room B
  - Status: The room is Clean or Dirty

## Actions:

- Left, Right, Suck, NoOp



# 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.

# Rational Agent

## Rational ≠ omniscient

- Percepts may not supply all relevant information
- E.g., in poker, don't know cards of others.

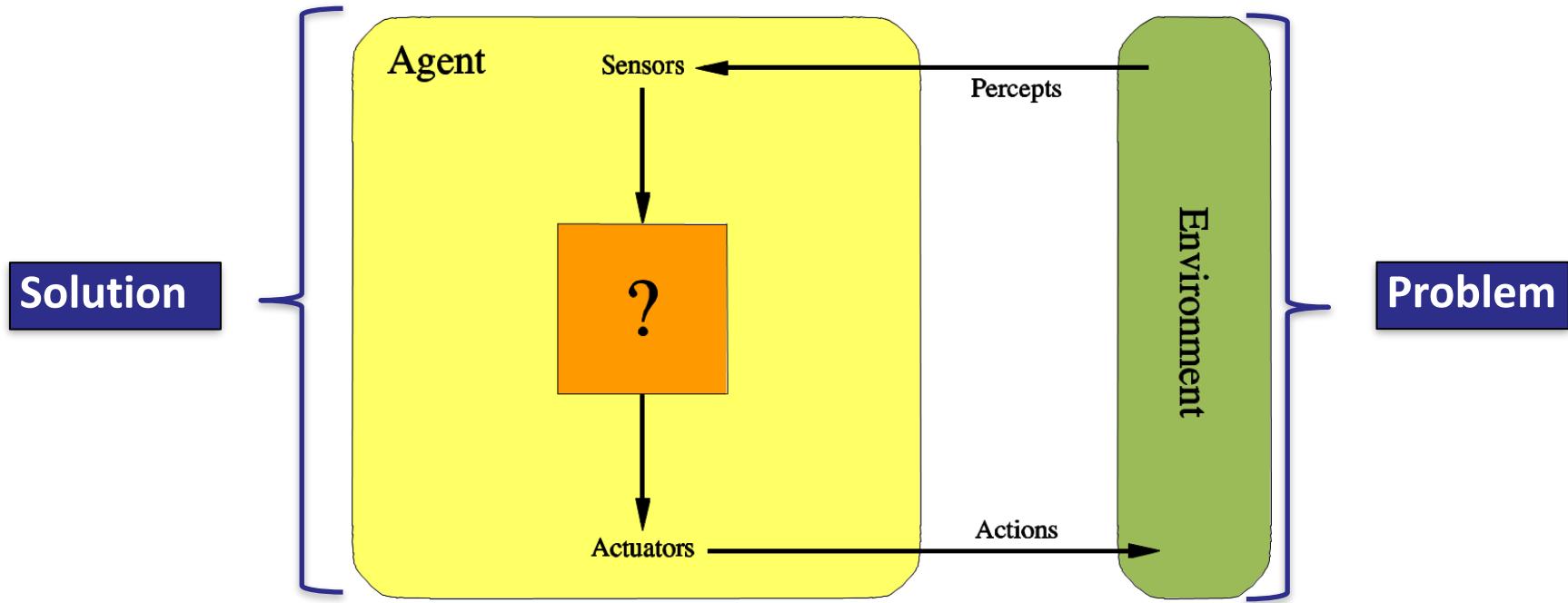
## Rational ≠ successful

- Rationality maximizes expected (not actual) performance
- Successful but not rational:
  - playing in the lottery/casino
- Rational but not successful
  - Agent is destroyed by a meteorite while behaving fully rational

# Autonomy in Agents

- The **autonomy** of an agent is the extent to which its actions are determined by its own experience, rather than knowledge of designer.
- **No autonomy** – ignores environment/data
- **Complete autonomy** – must act randomly/no program
- Example: baby learning to crawl
- **Ideal:** agents to have some autonomy
  - Possibly become more autonomous with experience

# Agent function



An **agent function** maps percept histories to actions

- internally represented by the **agent program**

# Agent architecture and program

- The **agent program** runs on the **physical architecture** to produce ***agent function***
- Agent = **architecture + program**

# A simple vacuum-cleaner agent

**Agent's function** is a look-up function

| 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 program for the simple vacuum agent



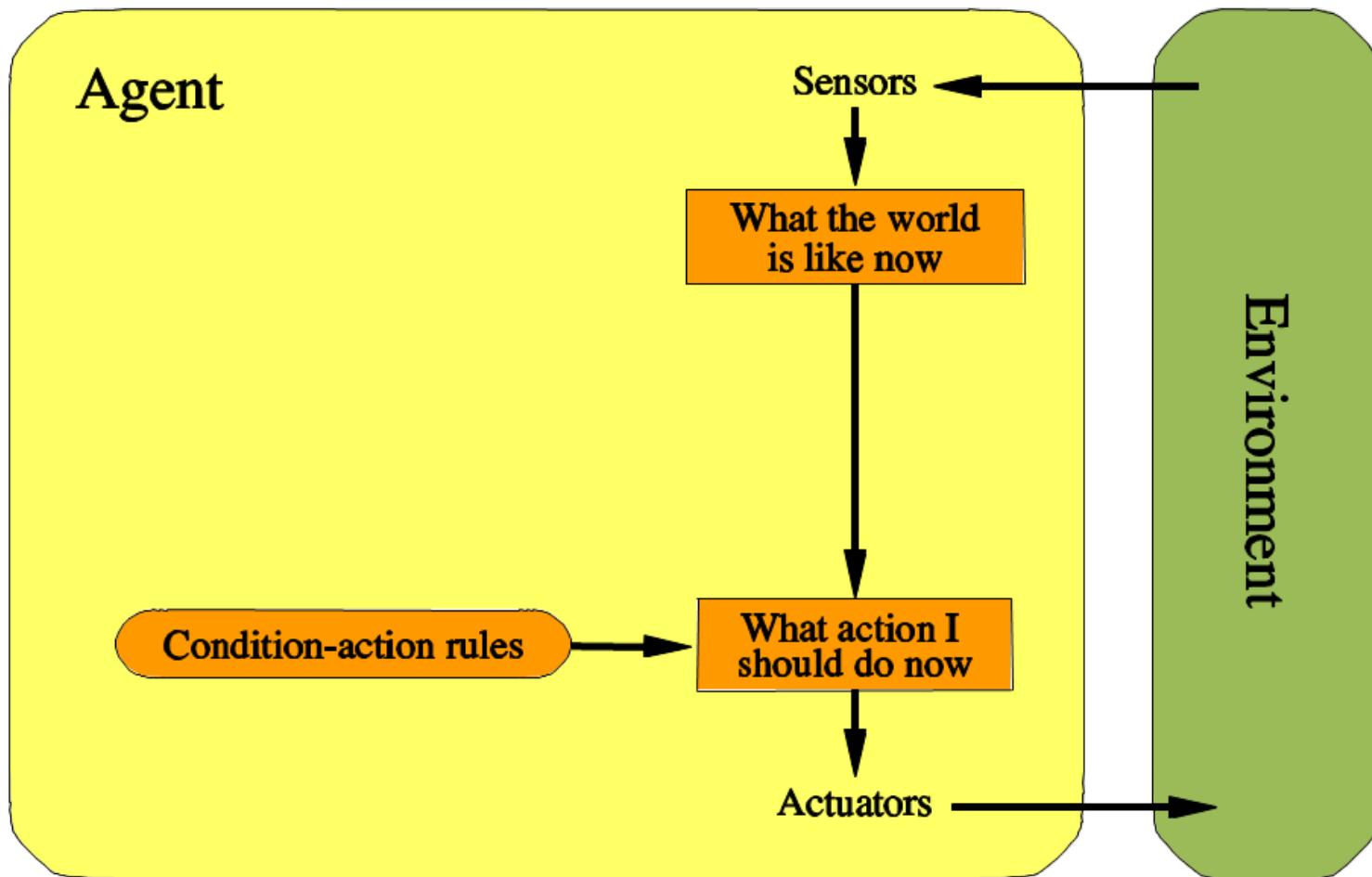
| Percept    | Action       |
|------------|--------------|
| [A, clean] | <i>Right</i> |
| [A, dirty] | <i>Suck</i>  |
| [B, clean] | <i>Left</i>  |
| [B, dirty] | <i>Suck</i>  |

```
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
```

# Agent types

- Simple reflex agent
- Model-based agent
- Goal-based agent
- Utility-based agent
- Learning agent

# Simple reflex agent



# Simple reflex agent

Selected action is exclusively based on current percept

- Ignores the percept history

Implemented through condition-action rules

- Large reduction in possible percept-action situations

Will make a very bad chess player

- Does not look at the board, only at the opponent's last move

# Simple reflex agent

Rules are just used as a concept here

- actual implementation could, e.g. be logical circuitry

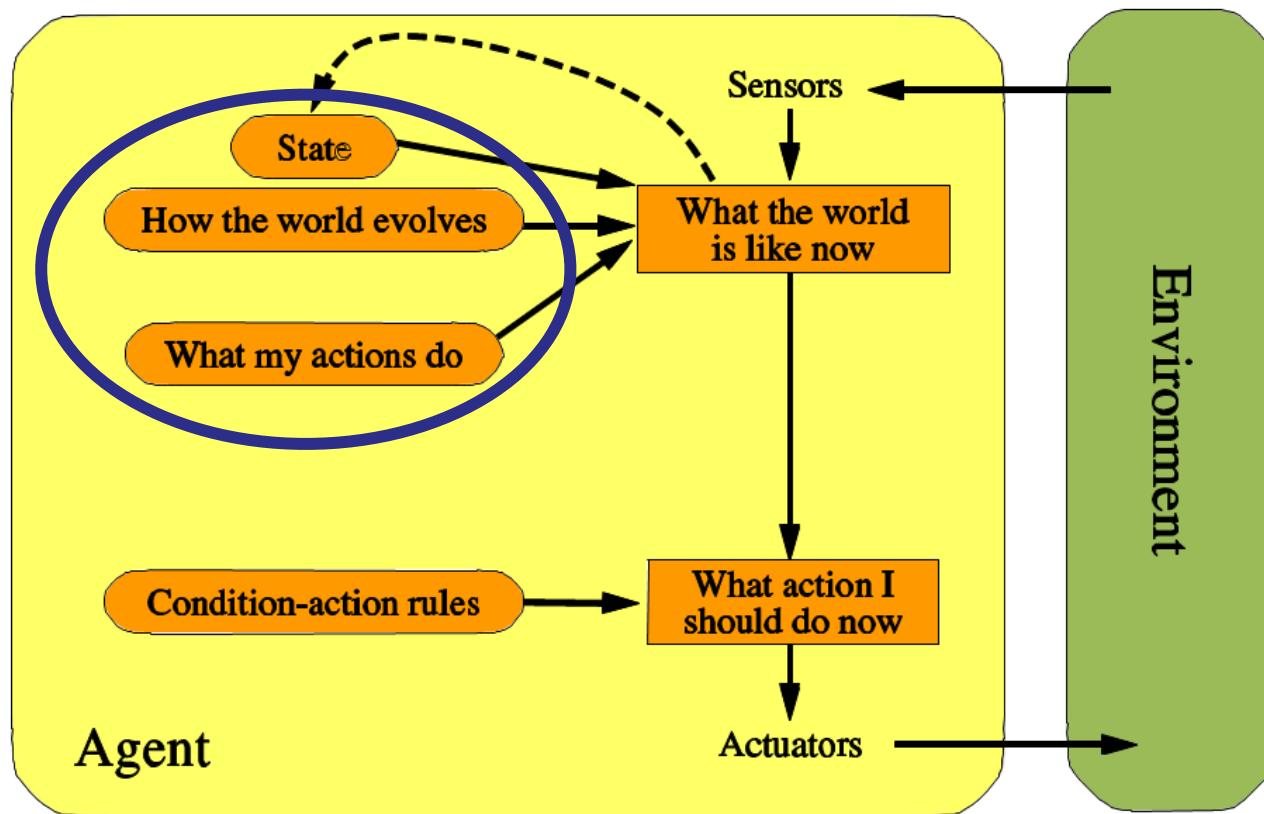
Will only work if the environment is **fully observable**

- everything important needs to be determinable from the sensory input
- otherwise infinite loops may occur
  - e.g. in the vacuum world the agent may not know whether to move right or left to get to the next room
  - possible solution: randomization

# Model-based agent

Keep track of the state of the world

- better way to fight partial observability



# Model-based agent

Input is mapped onto an internal state description (a world model)

- chess agent: keep a model of the current board situation

Internal state is also used for interpreting subsequent percepts

- the world model may include effects of own actions!

```
function REFLEX-AGENT-WITH-STATE(percept) returns an action
```

```
 static: state, a description of the current world state
```

```
 static: rules, a set of condition-action rules
```

```
 static: action, the most recent action, initially none
```

```
 state ← UPDATE-STATE (state, action, percept)
```

```
 rule ← RULE-MATCH (state, rule)
```

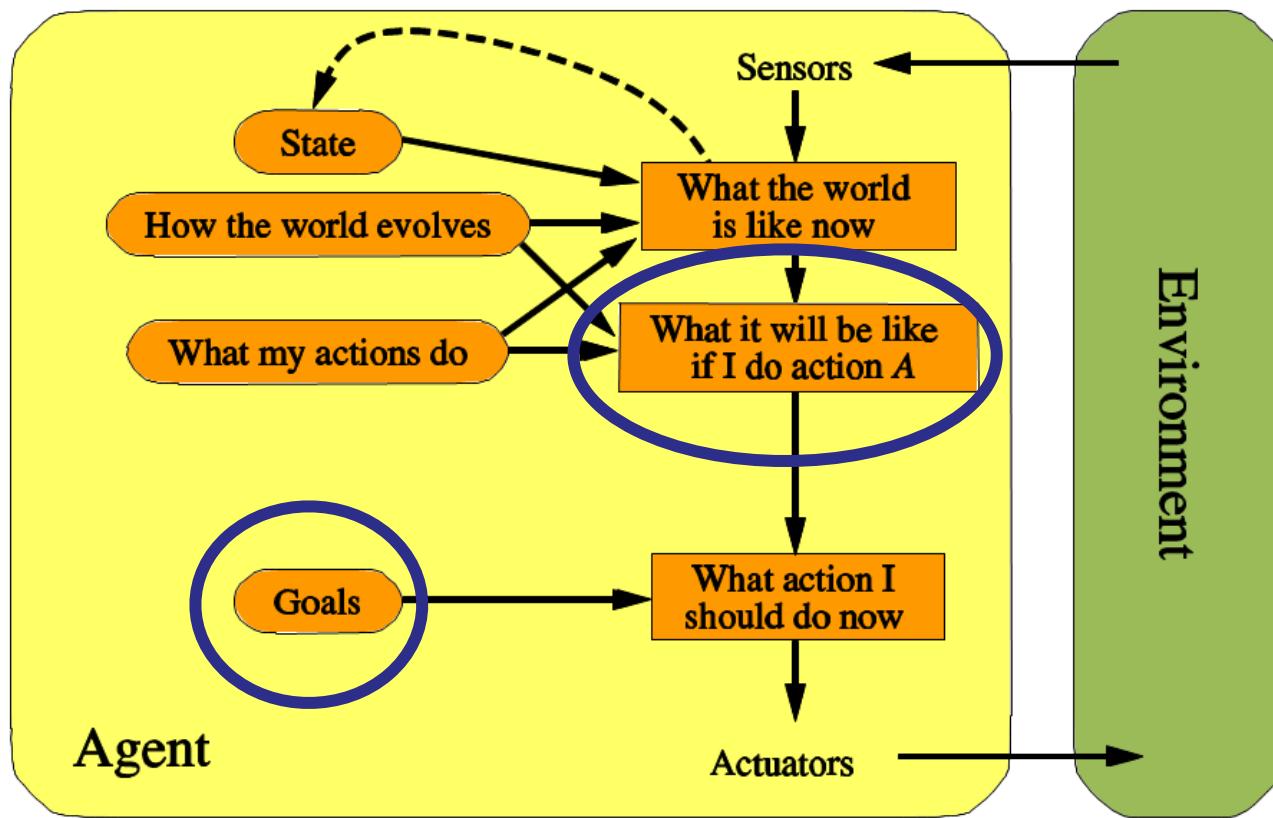
```
 action ← RULE-ACTION[rule]
```

```
 return action
```

# Goal-based Agent

The agent knows what states are desirable

- choose an action that leads to a desirable state

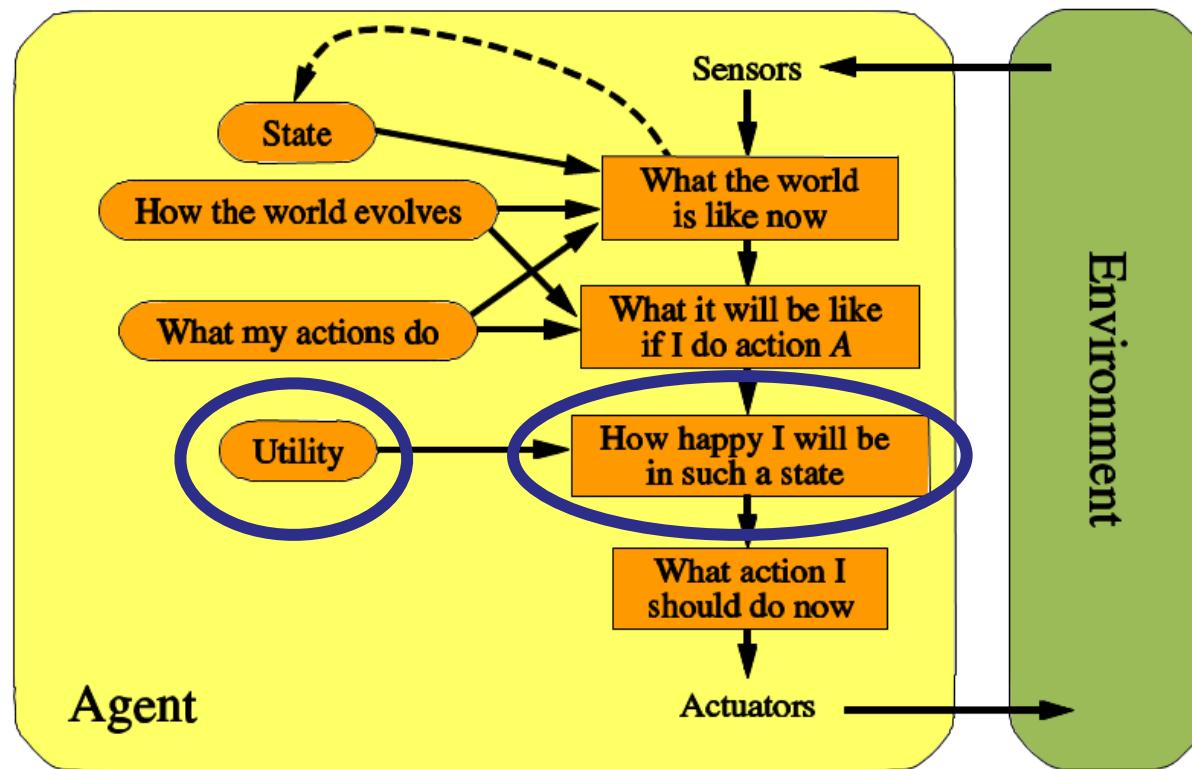


# Goal-based agent

- Choose an action that leads to the goal. **Problem:**  
A goal can often only be reached after a sequence of actions  
→ search and planning algorithms required
- Main difference from previous agents: oriented towards the future.
  - ◊ *What will happen if I do such-and-such?*
  - ◊ *Will this make me reach my goal?*
- Goals provide just a binary happy/unhappy distinction. Certain goals can be reached in different ways
  - *In the taxi driving task, the destination can be reached in different ways ..*
  - Some ways are quicker, safer, more reliable, cheaper, ...

# Utility-based agent

- Utility functions provide a continuous scale



# Utility-based agent

Goals provide just a binary happy/unhappy distinction

- utility functions provide a continuous scale
- → have a higher utility

Utility function

- maps a state (or a sequence of states) onto a real number

Improves on goals:

- selection between conflicting goals (e.g. speed and safety)
- selection between goals based on trade-off between likelihood of success and importance of goal

# Learning-based agents

All previous agent programs assume some knowledge for selecting actions.

- Where does this knowledge come from?

Learning mechanisms can be used for acquiring knowledge.

- Teach the agent instead of instructing it
- Every part of the previous agents can be improved with learning

Advantage: robustness in initially unknown environments

# Reinforcement learning

## Learning Scenario

- a learning agent
- $S$ : a set of possible states
- $A$ : a set of possible actions
- a state transition function  $\delta: S \times A \rightarrow S$
- a reward function  $r: S \times A \rightarrow \mathbb{R}$
- Environment:  
the agent repeatedly chooses an action according to some policy  $\pi: S \rightarrow A$
- this will put the agent into a new state according to  $\delta$
- in some states the agent receives feedback from the environment  
→ reinforcement

- The ability to acquire and apply knowledge and skills to achieve complex goals.

# Summary

## PEAS framework

- Performance, environment, actions, sensors

## Environment variables

- Observability, certainty, temporal succession, continuity, scale, population

## Performance measures

- Careful what you ask for

## Agent Types

- Simple reflex agent
- Model-based agent
- Goal-based agent
- Utility-based agent
- Learning agent

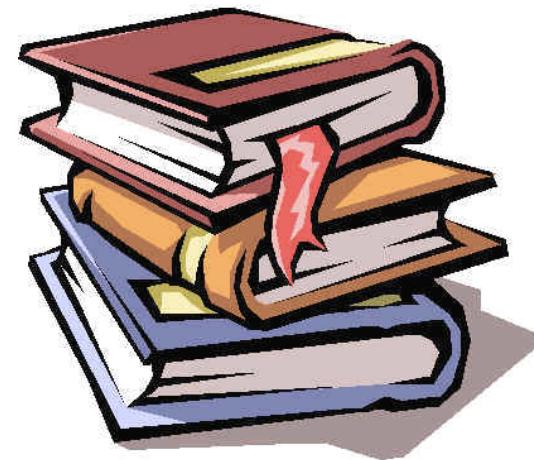
# Readings

Mandatory

- Russell & Norvig, Section 2 *Intelligent Agents*, 2.1 – 2.4

Optional

- Russell & Norvig, Rest of Section 2



# Next lecture: Search Problems

