

# Artificial Intelligence

## ENCS 434

### Introduction

**Aziz M. Qaroush**

# Syllabus

Course Information	
Course Title	Artificial Intelligence
Course Number	ENCS 434
Text Book	<ul style="list-style-type: none"><li>• Artificial Intelligence: A modern approach 3<sup>rd</sup> edition</li><li>• Artificial Intelligence: A Guide to Intelligent Systems 3<sup>rd</sup> edition</li></ul>
Instructor	Aziz Qaroush
Email	aqaroush@birzeit.edu
Office Hours	MW 11:00 – 2:00

Grading Scheme		
Assessment Type	Date	Weight
Midterm Exam	TBA	25%
Final Exam	TBA	35%
Assignments	TBA	10%
Projects	TBA	20
Quizzes	TBA	10%

# Syllabus

Topics	Time (Lectures )
• Introduction - Intelligent Agents	2
• Problem Solving by Search	4
• Heuristic (Informed) Search	4
• Constraint Satisfaction	3
• Games	3
• Knowledge-Based Agents - Propositional and First-Order Logic	3
• Inference in First-Order Logic, Logic Programming and Prolog	3
• Planning	4
• Fuzzy Logic	4
• Machine Learning - Basic Concepts, Decision Trees, Neural Networks	7
• Introduction to Natural Language Processing	2

# Course Learning Outcomes

- Upon completion of this course, you will have the ability to:
  - Understand the meaning of AI, its alternative approaches.
  - Know the techniques and technologies that currently exist and are "evolving" in the field of AI.
  - Know a variety of ways to represent and retrieve knowledge; Logic, semantic networks, frames, production rules.
  - Expand your knowledge about blind and heuristic search algorithms.
  - Know the fundamentals of AI programming languages; Prolog.
  - Know machine learning techniques and apply them in an AI programming language.
  - Understand the basic methods in planning and reasoning using both logic and uncertain inference.

# What is Artificial Intelligence ?

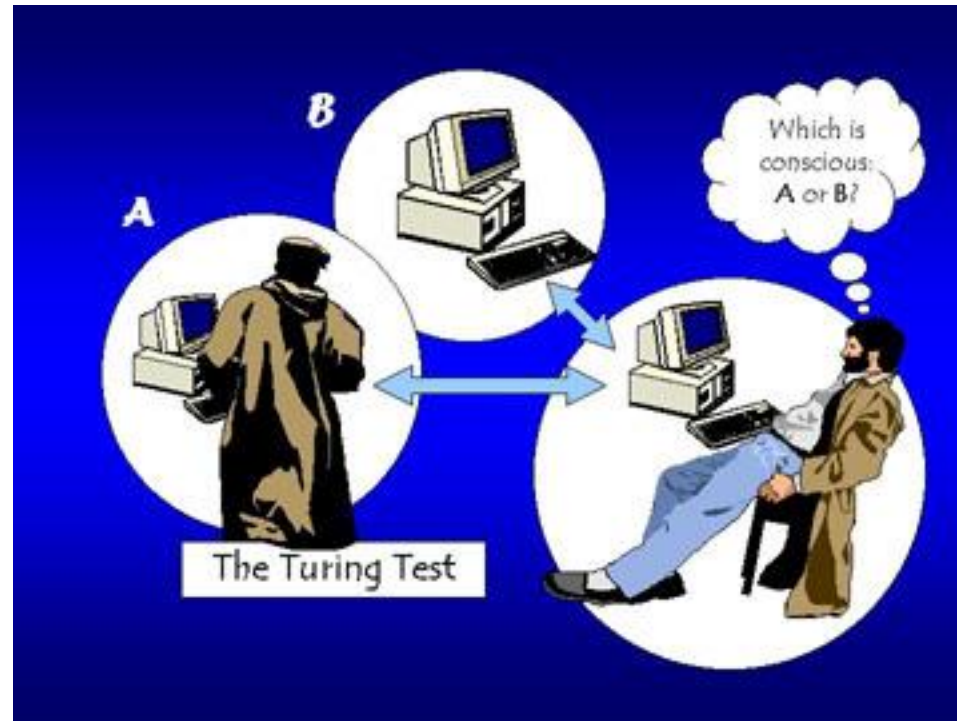
- Intelligence = Knowledge + ability to perceive, feel, understand, process, communicate, judge, and learn.
- **What is Artificial Intelligence?**
  - There is no official agreed upon definition of Artificial Intelligence.
  - In practice, it is an “umbrella term”
  - It is multidisciplinary subject
  - Technologies enter and exit the AI “umbrella” regularly.
- **General Goal:** replicate human intelligence

# What is Artificial Intelligence ?

- ❑ **Winston**: “AI is the study of ideas which enable computers to do things which make people seem intelligent.”
- ❑ **Steven Tanimoto**, “Computational techniques for performing tasks that apparently require intelligence when performed by humans.”
- ❑ **David Parnas**, “Artificial intelligence is to artificial flowers as natural intelligence is to natural flowers.”
- ❑ **Luger**: The branch of computer science that is concerned with automation of intelligent behavior.
- ❑ **Rich**: “AI is the study of how to make computers do things which, at the moment, people do better.”
- ❑ **Fahlman**: AI is the study of intelligence using the ideas and methods of computation.”
- ❑ **Found on the Web**: AI is the reproduction of the methods or results of human reasoning or intuition.
- ❑ **We can define it too**: AI is a field of computer science that simulates human performance to make a computer reasons in a manner similar to humans.

# What is Artificial Intelligence?

## Turing Test



**The computer passes the “test of intelligence” if a human, after posing some written questions, cannot tell whether the responses were from a person or not.**

# What is Artificial Intelligence?

- To give an answer, the computer would need to possess some capabilities:
  - Natural language processing: To communicate successfully.
  - Knowledge representation: To store what it knows or hears.
  - Automated reasoning: to answer questions and draw conclusions using stored information.
  - Machine learning: To adapt to new circumstances and to detect and extrapolate patterns.
  - Computer vision: To perceive objects.
  - Robotics to manipulate objects and move.



# Intelligent System Should do:

- ❑ How can we make computer based systems more intelligent?
- ❑ In practical terms, intelligent systems:
  - Should have the ability to *automatically perform tasks* that normally require a human expert.
  - Should have more *autonomy*; less requirement for human intervention or monitoring.
  - Should have *Flexibility in dealing with variability* in the environment in an appropriate manner.
  - Are *easier to use*: able to understand what the user wants from limited instructions.
  - *Can improve their performance* by learning from experience.

# Can we build hardware as complex as the brain?

## ❑ How complicated is our brain?

- A neuron, or nerve cell, is the basic information processing unit
- Estimated to be on the order of  $10^{12}$  neurons in a human brain
- Many more synapses ( $10^{14}$ ) connecting these neurons
- Cycle time:  $10^{-3}$  seconds (1 millisecond)

## ❑ How complex can we make computers?

- $10^6$  or more transistors per CPU
- Supercomputer: hundreds of CPUs,  $10^9$  bits of RAM
- Cycle times: order of  $10^{-8}$  seconds

## ❑ Conclusion

- **YES:** we can have computers with as many basic processing elements as our brain, but with
  - Far fewer interconnections (wires or synapses) than the brain
  - Much faster updates than the brain
- **But** building hardware is very different from making a computer behave like a brain!

# Must an Intelligent System be Foolproof?

- ❑ **A “foolproof” system is one that never makes an error:**

- Types of possible computer errors

- Hardware errors, e.g., memory errors
    - Software errors, e.g., coding bugs
    - “Human-like” errors

- Clearly, hardware and software errors are possible in practice

- What about “human-like” errors?

- ❑ **An intelligent system can make errors and still be intelligent**

- Humans are not right all of the time

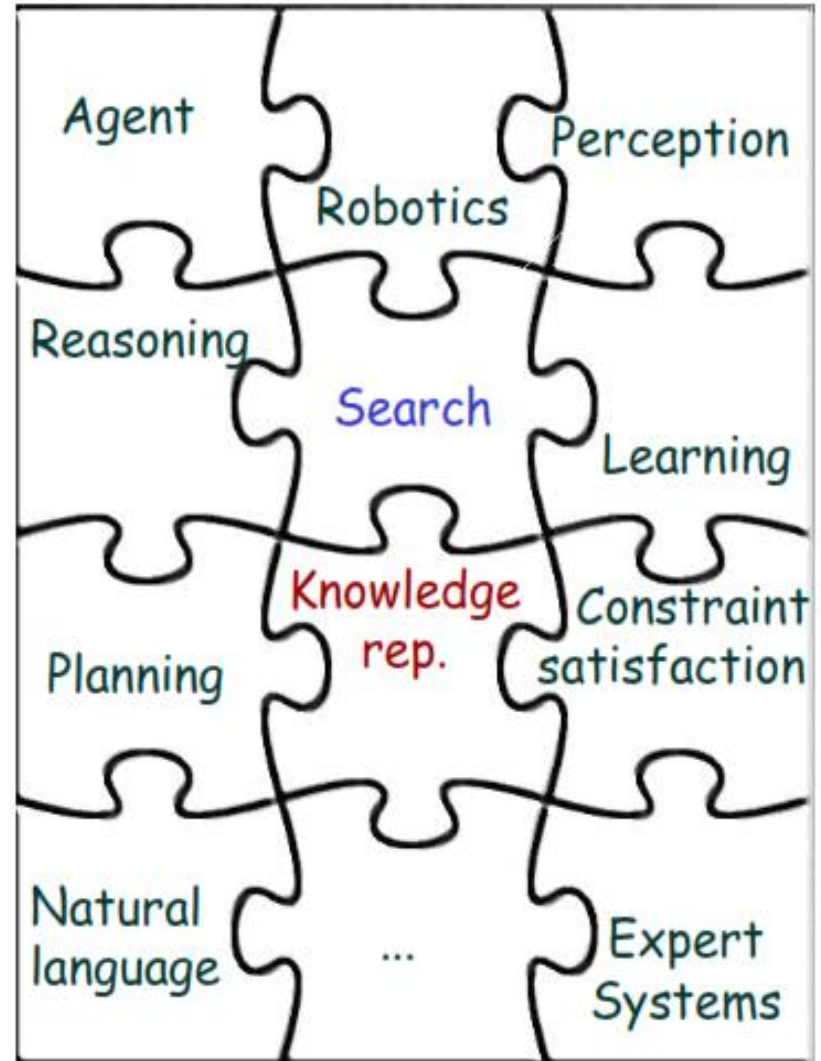
- We learn and adapt from making mistakes

- ❑ **Conclusion:**

- **NO:** intelligent systems will not (and need not) be foolproof

# Main Areas of AI

- Knowledge representation (including formal logic)
- Search, especially heuristic search (puzzles, games)
- Planning
- Reasoning under uncertainty, including probabilistic reasoning
- Learning
- Agent architectures
- Robotics and perception
- Natural language processing



# *Examples of AI Application systems:*

- Game Playing
- Autonomous Planning & Scheduling
- Natural Language Understanding
- Pattern Recognitions
- Robotics
- Automated theorem proving
- Web search Engines

---

# Agent

## Chapter 2



# Review: What is AI?

Views of AI fall into four categories:

Thinking humanly	Thinking rationally
Acting humanly	Acting rationally

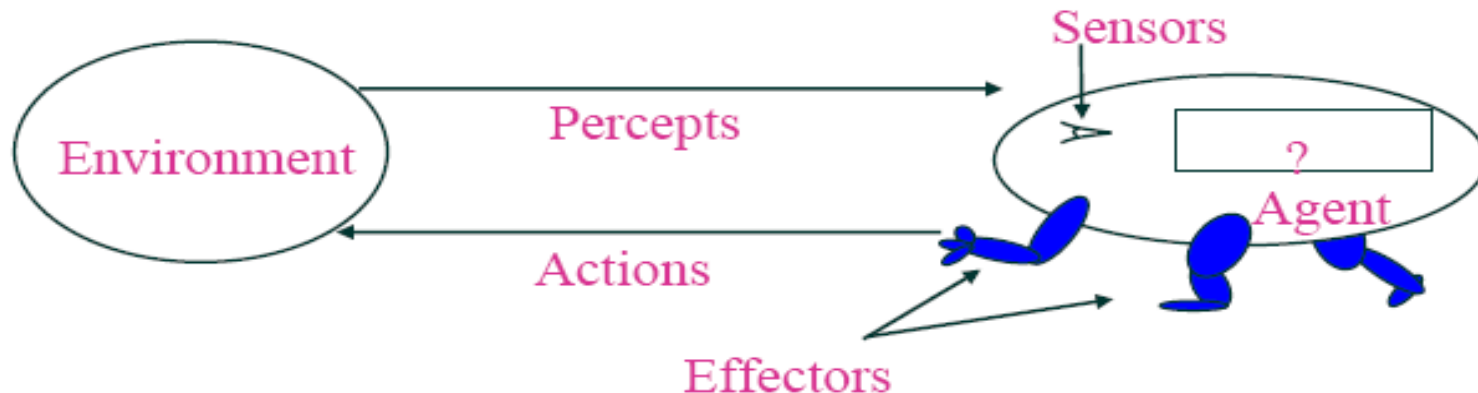
We will focus on "acting rationally"

## Acting rationally: rational agent

- ❑ **Rational** behavior: doing the right thing
- ❑ **The right thing**: which is expected to maximize goal achievement, given the available information.

# What is an Agent?

- in general, an entity that interacts with its environment
  - perception through sensors
  - actions through effectors or actuators



“Autonomous agents are computational systems that inhabit some complex dynamic environment, sense and act autonomously in this environment, and by doing so realise a set of goals or tasks for which they are designed”

- (*Pattie Maes, MIT Media Lab*)



# Examples of Agents

- **human agent**

- eyes, ears, skin, taste buds, etc. for sensors
- hands, fingers, legs, mouth, etc. for actuators
  - powered by muscles

- **robot**

- camera, infrared, bumper, etc. for sensors
- grippers, wheels, lights, speakers, etc. for actuators
  - often powered by motors

- **software agent**

- functions as sensors
  - information provided as input to functions in the form of encoded bit strings or symbols
- functions as actuators
  - results deliver the output

# Agents and Their Actions

- a *rational agent* does “the right thing”
  - the action that leads to the best outcome under the given circumstances
- an *agent function* maps percept sequences to actions
  - abstract mathematical description
- an *agent program* is a concrete implementation of the respective function
  - it runs on a specific agent architecture (“platform”)
- problems:
  - what is “the right thing”
  - how do you measure the “best outcome”

# Performance of Agents

- criteria for measuring the outcome and the expenses of the agent
  - often subjective, but should be objective
  - task dependent
  - time may be important
- **vacuum agent**
  - number of tiles cleaned during a certain period
    - based on the agent's report, or validated by an objective authority
    - doesn't consider expenses of the agent, side effects
      - energy, noise, loss of useful objects, damaged furniture, scratched floor
    - might lead to unwanted activities
      - agent re-cleans clean tiles, covers only part of the room, drops dirt on tiles to have more tiles to clean, etc.

# Rational Agent

- selects the action that is expected to maximize its performance
  - based on a performance measure
  - depends on the percept sequence, background knowledge, and feasible actions
- a rational agent is not omniscient
  - it doesn't know the actual outcome of its actions
  - it may not know certain aspects of its environment
- rationality takes into account the limitations of the agent
  - percept sequence, background knowledge, feasible actions
  - it deals with the expected outcome of actions

# Environments

- determine to a large degree the interaction between the “outside world” and the agent
  - the “outside world” is not necessarily the “real world” as we perceive it
    - it may be a real or virtual environment the agent lives in
- in many cases, environments are implemented within computers
  - they may or may not have a close correspondence to the “real world”

# Environment Properties

- fully observable vs. partially observable
  - sensors capture all relevant information from the environment
- deterministic vs. stochastic (non-deterministic)
  - changes in the environment are predictable
- episodic vs. sequential (non-episodic)
  - independent perceiving-acting episodes
- static vs. dynamic
  - no changes while the agent is “thinking”
- discrete vs. continuous
  - limited number of distinct percepts/actions
- single vs. multiple agents
  - interaction and collaboration among agents
  - competitive, cooperative

# Structure of Intelligent Agents

- **Agent** = Architecture + Program
- **architecture**
  - operating platform of the agent
    - computer system, specific hardware, possibly OS functions
- **program**
  - function that implements the mapping from percepts to actions
  - live in artificial environments where computers and networks provide the infrastructure

emphasis in this course is on the *program* aspect, not on the *architecture*

# PEAS Description

## **P**erformance Measures

used to evaluate how well an agent solves the task at hand

## **E**nvironment

surroundings beyond the control of the agent

## **A**ctuators

determine the actions the agent can perform

## **S**ensors

provide information about the current state of the environment



# PEAS: Taxi Driver Agent

□ **Example: Agent = 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



# Agent Programs

- the emphasis in this course is on programs that specify the agent's behavior through mappings from percepts to actions
  - less on environment and goals
- agents receive one percept at a time
  - they may or may not keep track of the percept sequence
- performance evaluation is often done by an outside authority, not the agent
  - more objective, less complicated
  - can be integrated with the environment program

# Skeleton Agent Program

- basic framework for an agent program

```
function SKELETON-AGENT(percept) returns action  
  static: memory
```

```
    memory := UPDATE-MEMORY(memory, percept)
```

```
    action := CHOOSE-BEST-ACTION(memory)
```

```
    memory := UPDATE-MEMORY(memory, action)
```

```
return action
```

# Look it up!

- simple way to specify a mapping from percepts to actions
  - tables may become very large
  - almost all work done by the designer
  - no autonomy, all actions are predetermined
    - with well-designed and sufficiently complex tables, the agent may appear autonomous to an observer, however
  - learning might take a very long time
    - so long that it is impractical
    - there are better learning methods

# Table Agent Program

- agent program based on table lookup

```
function TABLE-DRIVEN-AGENT(percept) returns action  
  static: percepts // initially empty sequence*  
           table   // indexed by percept sequences  
                // initially fully specified  
  
  append percept to the end of percepts  
  action      := LOOKUP(percepts, table)  
  
return action
```

\* Note: the storage of percepts requires writeable memory

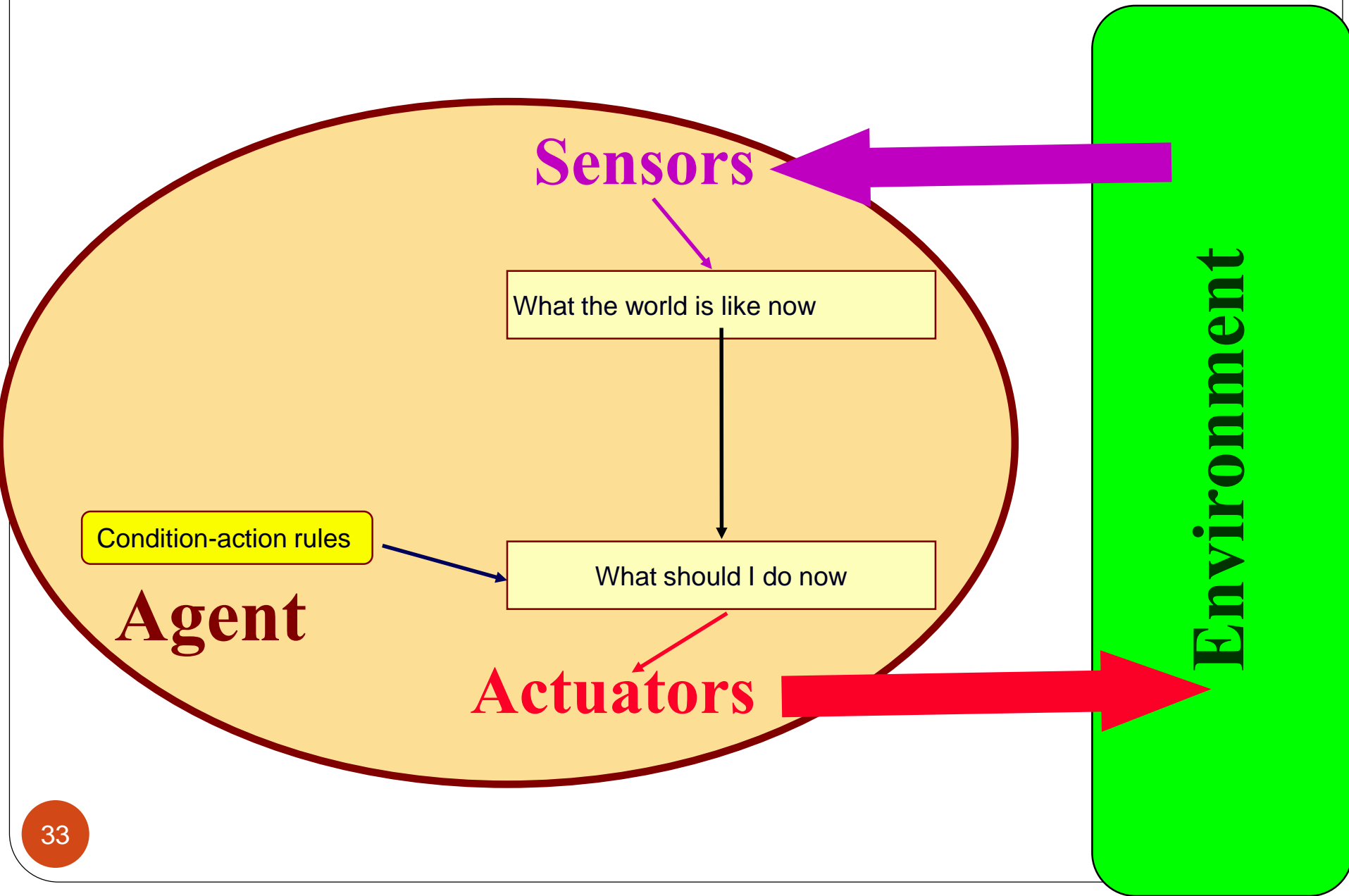
# Agent Program Types

- different ways of achieving the mapping from percepts to actions
- different levels of complexity
  - simple reflex agents
  - model-based agents
    - keep track of the world
  - goal-based agents
    - work towards a goal
  - utility-based agents
  - learning agents

# Simple Reflex Agent

- instead of specifying individual mappings in an explicit table, common input-output associations are recorded
  - requires processing of percepts to achieve some abstraction
  - frequent method of specification is through condition-action rules
    - *if percept then action*
  - efficient implementation, but limited power
    - environment must be fully observable
- Problems
  - Table is still too big to generate and to store (e.g. taxi)
  - Takes long time to build the table
  - No knowledge of non-perceptual parts of the current state
  - Not adaptive to changes in the environment; requires entire table to be updated if changes occur

# Reflex Agent Diagram





# Reflex Agent Program

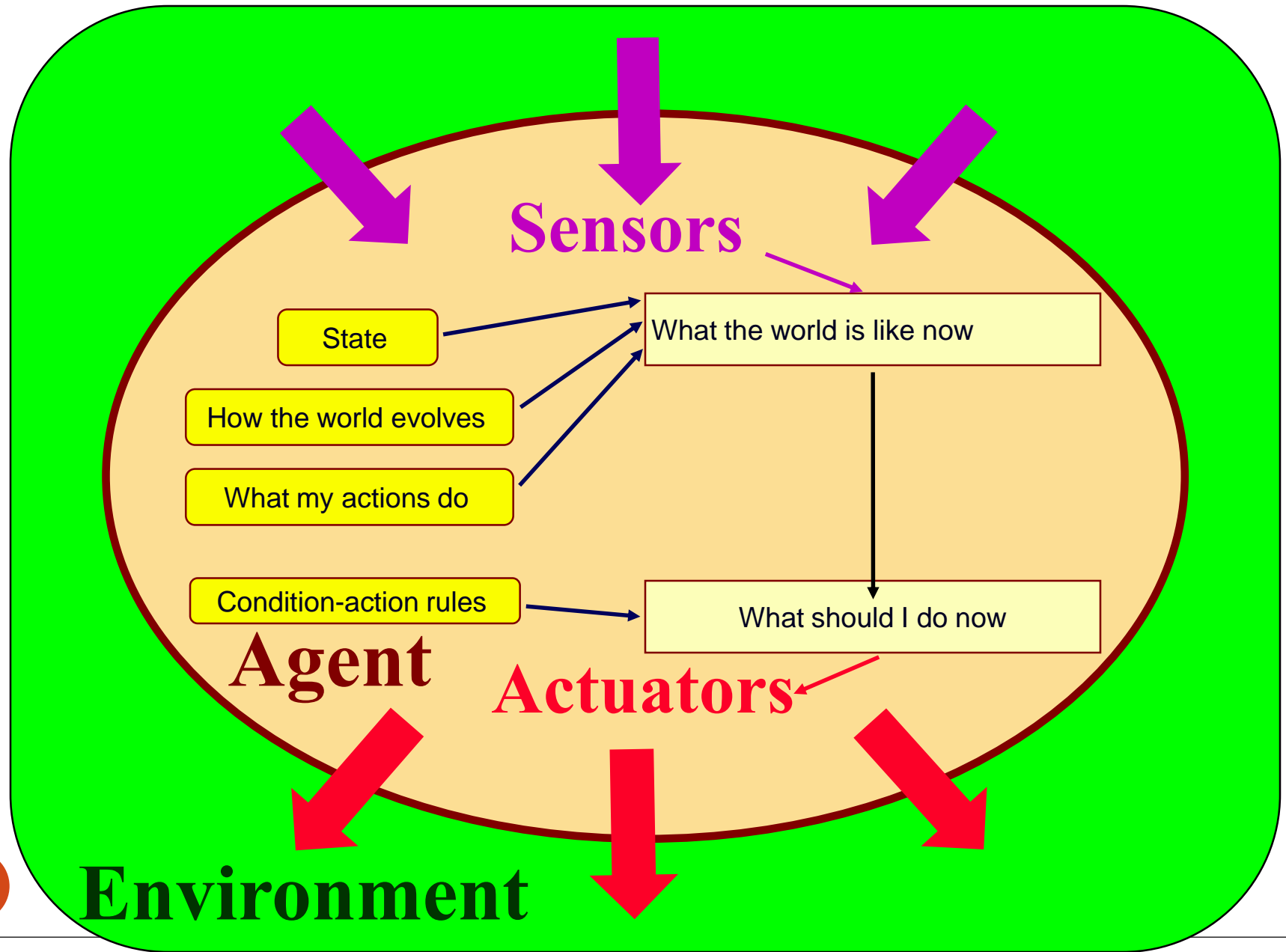
- application of simple rules to situations

```
function SIMPLE-REFLEX-AGENT(percept) returns  
  action  
  static: rules//set of condition-action rules  
  
  condition      := INTERPRET-INPUT(percept)  
  rule           := RULE-MATCH(condition, rules)  
  action         := RULE-ACTION(rule)  
  
return action
```

# Model-Based Reflex Agent

- an internal state maintains important information from previous percepts
  - sensors only provide a partial picture of the environment
  - helps with some partially observable environments
- the internal states reflects the agent's knowledge about the world
  - this knowledge is called a *model*
  - may contain information about changes in the world
    - caused by actions of the action
    - independent of the agent's behavior

# Model-Based Reflex Agent Diagram



# Model-Based Reflex Agent Program

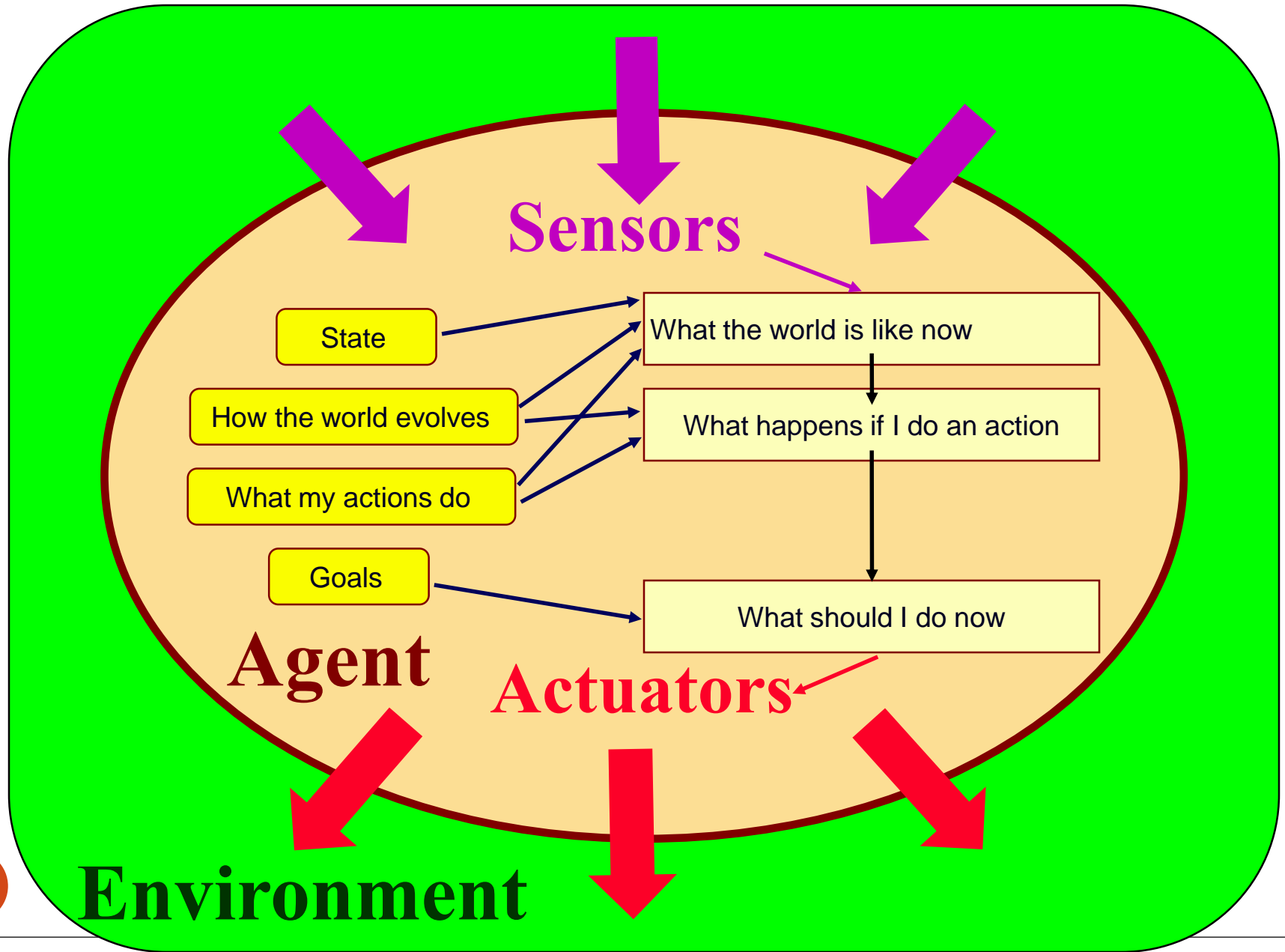
- application of simple rules to situations

```
function REFLEX-AGENT-WITH-STATE(percept) returns action  
  static: rules      //set of condition-action rules  
           state      //description of the current world state  
           action     //most recent action, initially none  
  
  state      := UPDATE-STATE(state, action, percept)  
  rule       := RULE-MATCH(state, rules)  
  action     := RULE-ACTION[rule]  
return action
```

# Goal-Based Agent

- the agent tries to reach a desirable state, the *goal*
  - may be provided from the outside (user, designer, environment), or inherent to the agent itself
- results of possible actions are considered with respect to the goal
  - easy when the results can be related to the goal after each action
  - in general, it can be difficult to attribute goal satisfaction results to individual actions
  - may require consideration of the future
    - what-if scenarios
    - search, reasoning or planning
- very flexible, but not very efficient

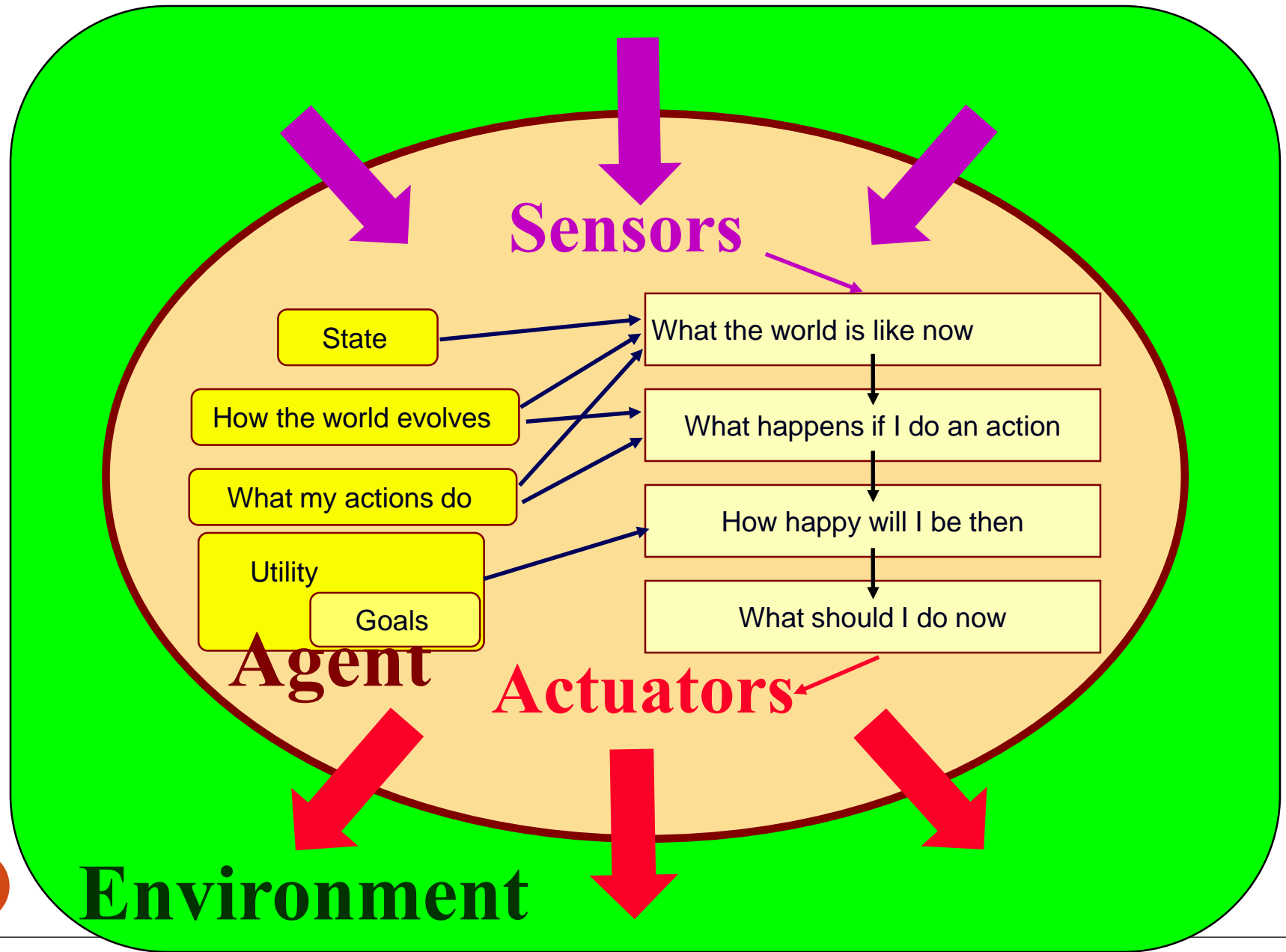
# Goal-Based Agent Diagram



# Utility-Based Agent

- more sophisticated distinction between different world states
  - a utility function maps states onto a real number
    - may be interpreted as “degree of happiness”
  - permits rational actions for more complex tasks
    - resolution of conflicts between goals (tradeoff)
    - multiple goals (likelihood of success, importance)
    - a utility function is necessary for rational behavior, but sometimes it is not made explicit

# Utility-Based Agent Diagram

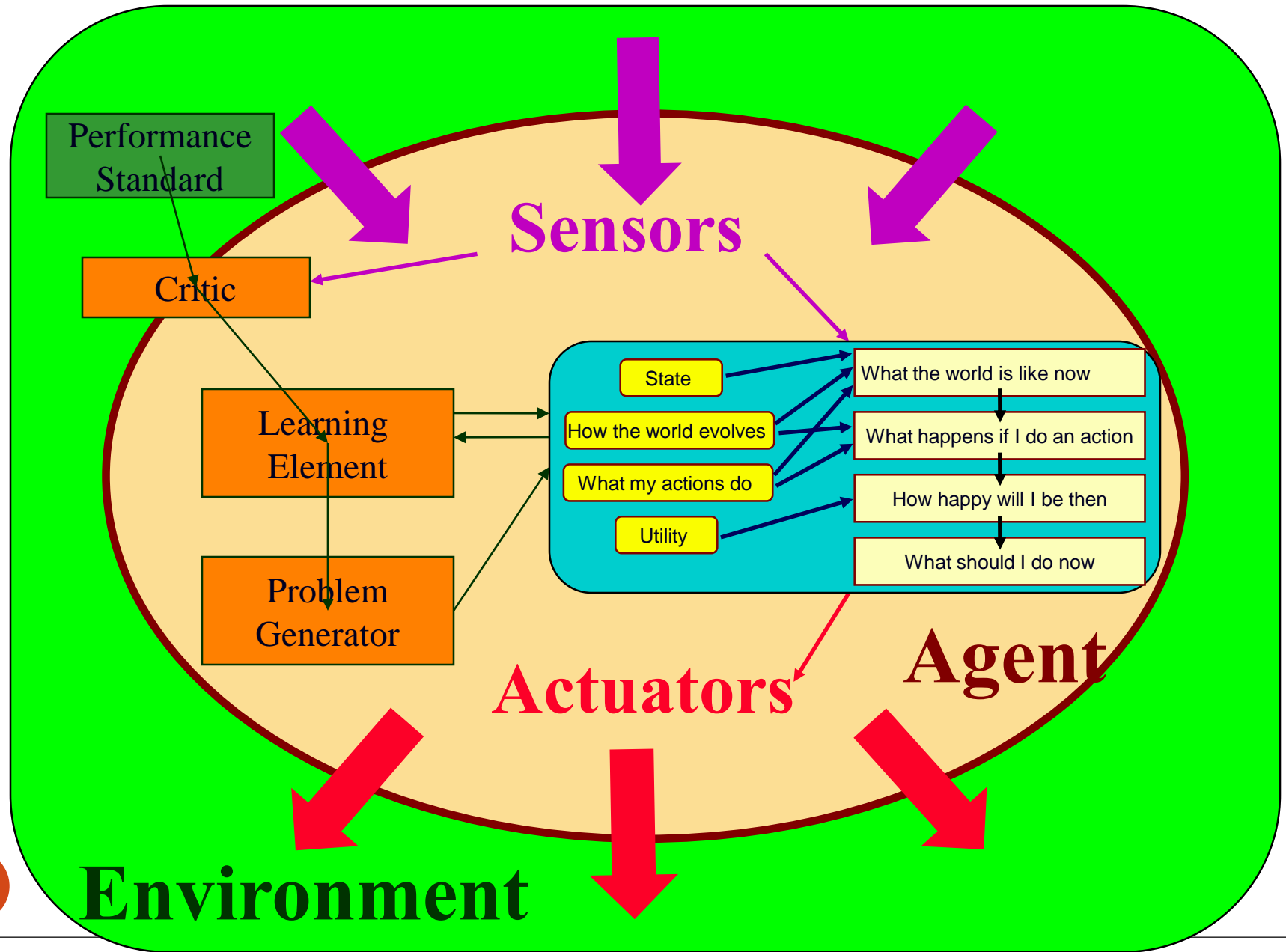




# Learning Agent

- performance element
  - selects actions based on percepts, internal state, background knowledge
  - can be one of the previously described agents
- learning element
  - identifies improvements
- critic
  - provides feedback about the performance of the agent
  - can be external; sometimes part of the environment
- problem generator
  - suggests actions
  - required for novel solutions (creativity)

# Learning Agent Diagram



# Chapter Summary

- What is an AI
- agents perceive and act in an environment
- ideal agents maximize their performance measure
  - autonomous agents act independently
- basic agent types
  - simple reflex
  - reflex with state
  - goal-based
  - utility-based
  - Learning
- some environments may make life harder for agents
  - inaccessible, non-deterministic, non-episodic, dynamic, continuous