Overview

CPSC 447 - Artificial Intelligence I

What is Artificial Intelligence?

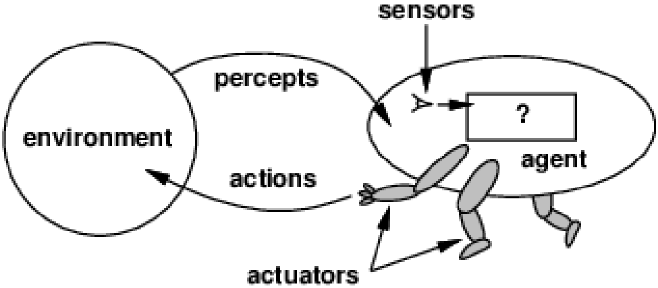
- Computational models of human behavior?
  (Programs the behave externally like humans)
- Computational models of human thought?
  (Programs that operate internally the way humans do)
- Computational systems that behave intelligently?
  (What does it mean to behave intelligently?)
- Computational systems that behave rationally.
- AI applications

Agents

- An *agent* is software that gathers information about an environment and takes actions based on that information.
- Examples:
    - a robot
    - a factory
    - a traffic control system
    - ...

Agents and the Environment

- Formalizing the problem of building an agent.



World Model

- $\mathcal{A}$ - the action space
- $\mathcal{P}$ - the percept space
- $\mathcal{E}$ - the environment: $\mathcal{A}* \rightarrow \mathcal{P}$
- $\mathcal{S}$ - the internal state (may not be visible to agent)
- $\mathcal{S} \rightarrow \mathcal{P}$ - the perception function
- $\mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ - the world dynamics

Agent Design

- $\mathcal{U}$ - utility function: $\mathcal{S} \to \mathbb{R}$ (or $\mathcal{S}^* \to \mathbb{R}$)
- The agent design problem: find $\mathcal{P}^* \to \mathcal{A}$
- The agent function maps percept histories to actions.
- The agent function should maximize the utility of the resulting sequence of states.

Rationality

- A *rational agent* chooses whichever action maximizes the *expected value* of the performance measure given the percept sequence to date.
- In other words, a rational agent takes actions it believes will achieve its goals
- Rational $\neq$ omniscient
  - percepts may not supply all relevant information
- Rational $\neq$ clairvoyant
  - action outcomes may not be as expected
- Hence, rational $\neq$ successful
- Rational $\Rightarrow$ exploration, learning, autonomy

Limited Rationality

- Problem: the agent may not be able to compute the best action (subject to its beliefs and goals).
- So, we want to use *limited rationality*: “acting in the best way you can subject to the computational constraints that you have”
- The (limited rational) agent design problem: find $\mathcal{P}^* \to \mathcal{A}$
  - mapping of sequences of percepts to actions
  - maximizes the utility of the resulting sequence of states
  - subject to our computational constraints

Thinking

- Is finding $\mathcal{P}^* \to \mathcal{A}$ AI? Aren't the agents supposed to think?
- Why is it useful to think? If you were endowed with an optimal table of reactions ($\mathcal{P}^* \to \mathcal{A}$) why do you need to think?
- Problem: the table is too big; there are too many world states and too many sequences of percepts.
  - In some domains, the required reaction table can be specified compactly.
  - In other domains, we can take advantage of the fact that most things that could happen do not actually happen in practice.

Learning

- Learning is useful when the agent does not know much about the initial environment or the environment can change.
- An agent can use sequences of percepts to estimate missing details in the world dynamics.
- Learning is similar to perception; they both find out about the world based on experience.
  - Perception – short time scale
  - Learning – long time scale

Specifying the Task Environment

- A *task environment* is the “problem” that the rational agent is a “solution” for.
- Specifying a task environment (PEAS)
  - Performance measure
  - Environment
  - Actuators
  - Sensors

Example Task Environment Specification

Taxi driver agent

- Performance measure? safe, fast, legal, comfortable trip, maximize profits
- Environment? roads, traffic, pedestrians, customers
- Actuators? steering, accelerator, brake, signal, horn
- Sensors? cameras, sonar, GPS, odometer, ...

Properties of Task Environments

- Fully Observable vs. Partially Observable
  - Can you observe the state of the world directly?
- Deterministic vs. Stochastic
  - Does an action map one state to a single other state?
- Episodic vs. Sequential
  - Do current decisions affect future decisions?
- Static vs. Dynamic
  - Can the world change while you are thinking?
- Discrete vs. Continuous
  - Are percepts and actions discrete or continuous?
- Single agent vs. Multiagent
  - Does an agent need to consider the actions of other agents?

Example Task Environments

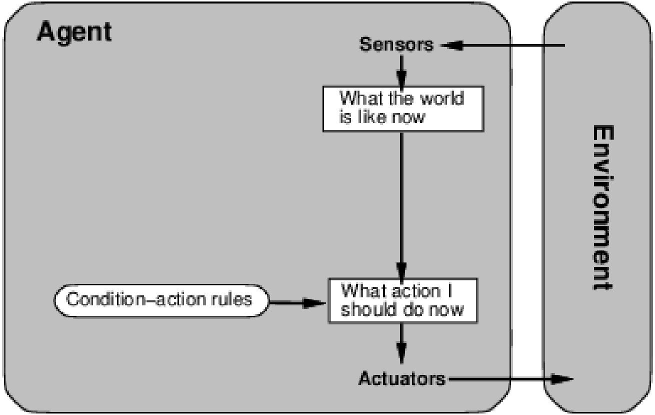
	Solitaire	Backgammon	Internet shopping	Taxi
Observable?	Yes	Yes	No	No
Deterministic?	Yes	No	Partly	No
Episodic?	No	No	No	No
Static?	Yes	Semi	Semi	No
Discrete?	Yes	Yes	Yes	No
Single-agent?	Yes	No	Yes	No

The environment type largely determines the agent design.

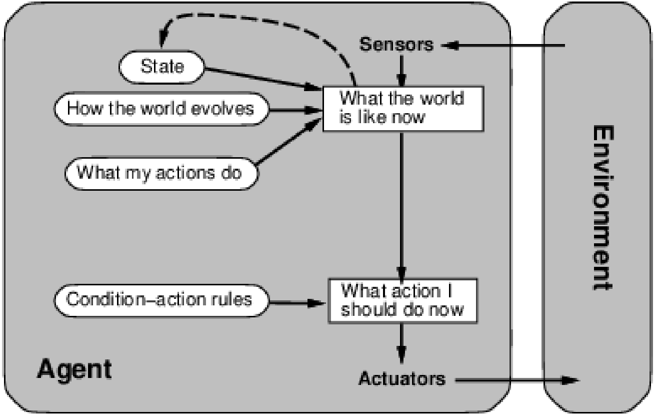
Agent Types

- Four basic agent types in increasing generality.
    - simple reflex agents
    - reflex agents with state
    - goal-based agents
    - utility-based agents
- All the basic agent types can be turned into learning agents.

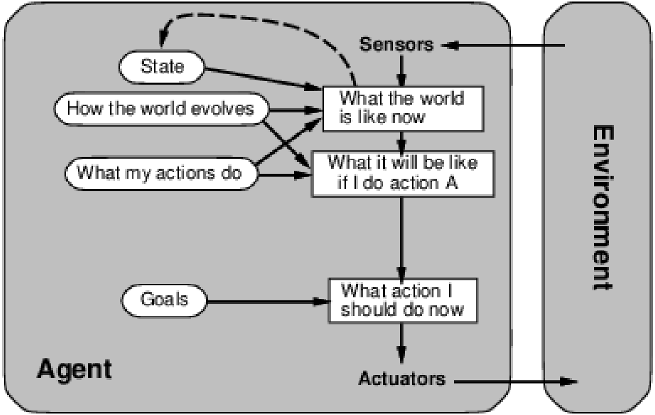
Simple Reflex Agents



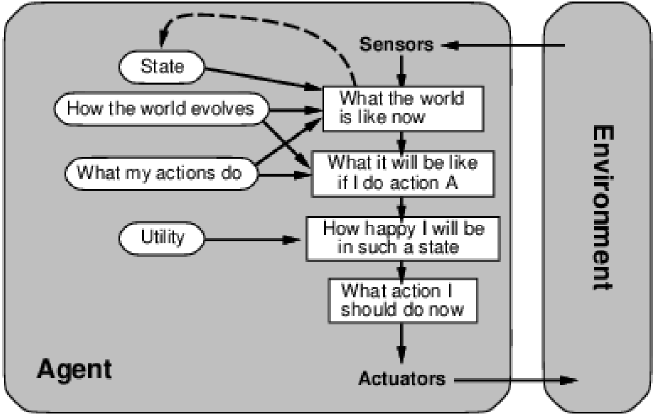
Reflex Agents with State



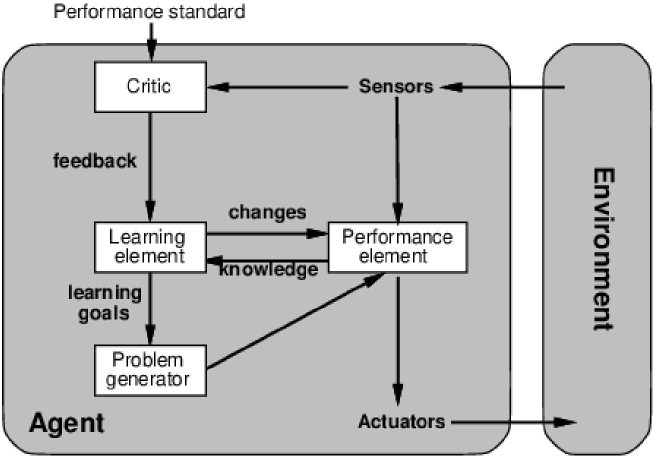
Goal-Based Agents


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
Goals
What action I should do now
Agent
Actuators
Environment

Utility-Based Agents


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

Learning Agents


Performance standard
Critic
Sensors
feedback
changes
Learning element
knowledge
Performance element
learning goals
Problem generator
Agent
Actuators
Environment