Artificial Intelligence

2. Intelligent Agents

Shashi Prabh

School of Engineering and Applied Science
Ahmedabad University

Contents

Goal:

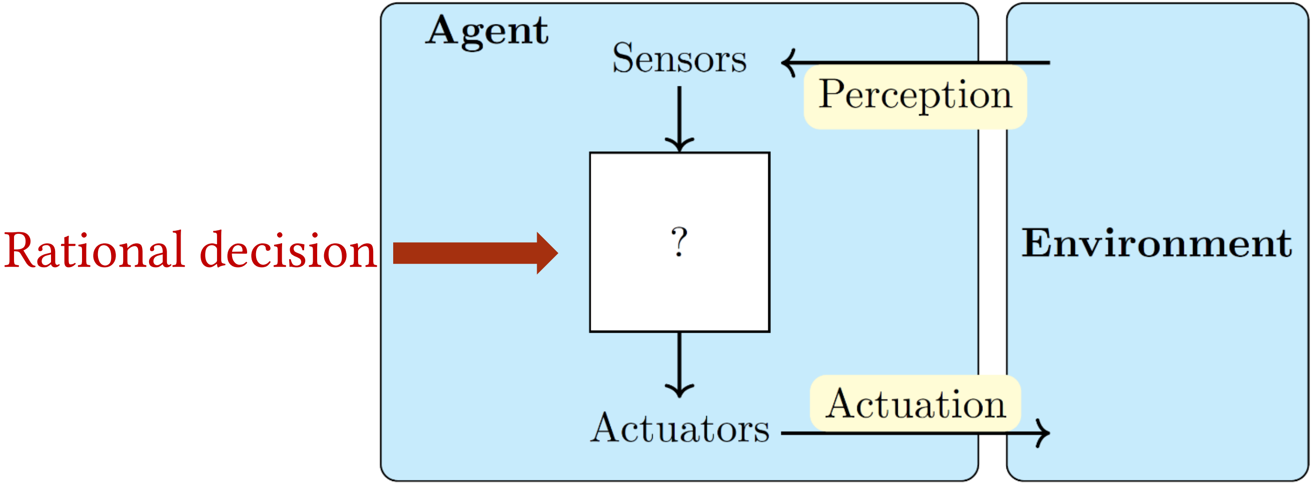
- Use the concept of rationality to develop a set of design principles for building AI systems a.k.a. intelligent agents.
  - This sets the stage for exploring specific AI techniques, such as, search, logic and learning, in future lectures.

Topics

- Agents
- Rationality
- Environment characterization
- Agent types

Agents

- Agent is an entity that perceives its environment through sensors and acts upon that environment through actuators
  - A tool for analysing systems
  - Examples?



Agents interact with environments through sensors and actuators.

Agent examples

Agent	Sensors	Actuators
Humans	Eyes, ears, skin	Hands, legs, vocal chord
Robots	Camera, sound sensor, IR range finder	Motors, pumps, drills, displays
Software	File content, human input, packets received over network	Write file, display/print information, Send packets over network

Agents

- **Environment:** part of the universe whose state is relevant for designing the agent
- **Percept:** content gathered from the environment by the agent's sensors
- **Percept sequence:** complete history of an agent's percepts
  - Determines the choice of actions
- **Agent function** (f): mapping of a percept sequence to an action
  - Determines an agent's behavior
- **Agent program:** Implementation of f

Quiz

What are the two basic components of an agent's interaction with the environment?

A. Sensors and actuators

B. Percepts and actions

C. Rationality and actions

D. Inputs and outputs

Quiz

What are the two basic components of an agent's interaction with the environment?

A. Sensors and actuators

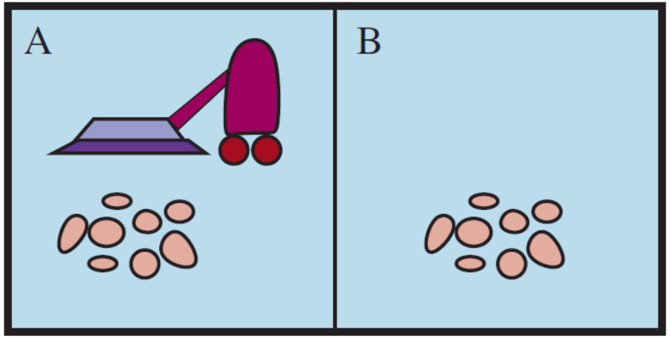
B. Percepts and actions

C. Rationality and actions

D. Inputs and outputs

Vacuum-Cleaning Agent

- Environment/World consists of 2 squares, A and B
- Percepts available
  - Location (A or B)
  - Is the room dirty?
- Actions available
  - Move to left or right
  - Suck up the dirt
  - Do nothing
- Agent function: If the square is dirty, then suck. Otherwise, move to the other square
  - Is it a good agent design?



Vacuum-Cleaning Agent 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, $Clean$], [A, $Clean$], [A, $Clean$]	$Right$
[A, $Clean$], [A, $Clean$], [A, $Dirty$]	$Suck$
⋮	⋮

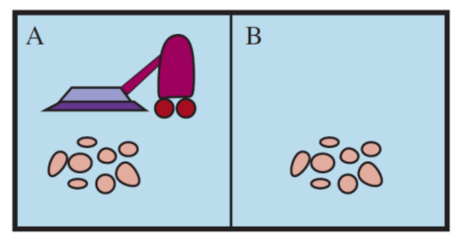
- Specifying the actions differently leads to various vacuum-cleaning agents
    - What is the right way?

Vacuum-Cleaning Agent Program

- The vacuum-cleaning agent function can be implemented as

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*



- Reflex Agent: Use the latest percept to determine the action
  - Implemented using condition-action rules
    - Simple and fast: 4^t entries in the table to 4 entries
      - Thermostats, running away from a snake, moving hand away from a hot surface

Rationality: Performance Measure

- What makes an agent good / bad / stupid?
  - A rational agent is supposed to do the "right thing"
    - What does "right" mean? How to define it?
- **Consequentialism as performance measure:** behaviour is evaluated by its consequences
  - Dangerous: lying gets more profit. Hence it is a good behaviour...
  - As the result of actions, environment transitions through a sequence of states
  - A consequentialist performance measure evaluates an agent based on the desirability of the sequence of environment states

Rationality: Performance Measure

- Performance measure evaluates the desirability of a sequence of states
  - Needs to be specified by the designer/user
    - Machines don't have innate preferences or desires
  - It is usually hard to formulate it

Rationality: Performance Measure Example

- Option 1: The amount of dirt cleaned-up in a day
  - Bad idea. Why?
- Option 2: Is the floor clean?
  - + 2 for clean floor (every minute), -3 for dirty floor, -1 for noise
- Weiner: "the purpose to put into the machine is the purpose we really desire."
- General guideline: Performance measure should reflect what one wants to achieve not how the agent should behave (consequentialism…)

Rationality

- Rationality at any given time depends on
  - Performance measure (the criterion of success)
  - Agent's prior knowledge of the environment
  - Available actions
  - Percept sequence
- Rational agent takes actions that are expected to maximize performance measure
- **Rationality ≠ Omniscience**
  - Do not expect the agent to do what turns after the fact to be the best action

Quiz

Which of the following is a rational agent?

A. Always acts based on past experience

B. Takes the shortest path to the goal

C. Always chooses the best known action (even if suboptimal)

D. Acts to maximize expected performance, given its knowledge

Quiz

Which of the following is a rational agent?

A. Always acts based on past experience

B. Takes the shortest path to the goal

C. Always chooses the best known action (even if suboptimal)

D. Acts to maximize expected performance, given its knowledge

Nature of Environments

- Task environments are the problems of which rational agents are the solutions
- Task environment specification : Performance, Environment, Actuators, Sensors (PEAS)
- PEAS description of self-driving car environment?

Nature of Environments

- PEAS description of self-driving car task environment

Agent Type	Performance Measure	Environment	Actuators	Sensors
Taxi driver	Safe, fast, legal, comfortable trip, maximize profits, minimize impact on other road users	Roads, other traffic, police, pedestrians, customers, weather	Steering, accelerator, brake, signal, horn, display, speech	Cameras, radar, speedometer, GPS, engine sensors, accelerometer, microphones, touchscreen

Examples

- PEAS examples

Agent Type	Performance Measure	Environment	Actuators	Sensors
Medical diagnosis system	Healthy patient, reduced costs	Patient, hospital, staff	Display of questions, tests, diagnoses, treatments	Touchscreen/voice entry of symptoms and findings
Satellite image analysis system	Correct categorization of objects, terrain	Orbiting satellite, downlink, weather	Display of scene categorization	High-resolution digital camera
Part-picking robot	Percentage of parts in correct bins	Conveyor belt with parts; bins	Jointed arm and hand	Camera, tactile and joint angle sensors
Refinery controller	Purity, yield, safety	Refinery, raw materials, operators	Valves, pumps, heaters, stirrers, displays	Temperature, pressure, flow, chemical sensors
Interactive English tutor	Student's score on test	Set of students, testing agency	Display of exercises, feedback, speech	Keyboard entry, voice

Properties of Task Environments

Task environments vary along several significant dimensions:

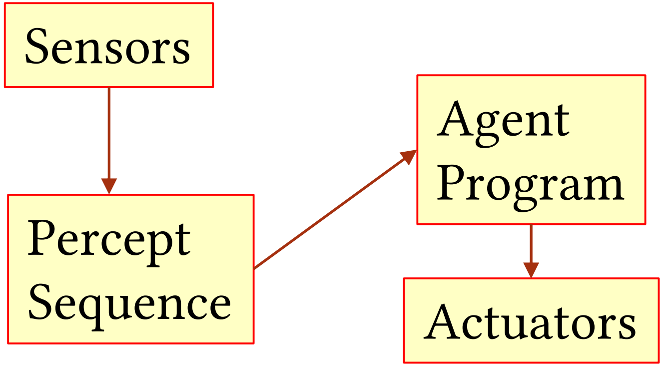
1. Fully or partially observable
2. Single-agent or multi-agent
3. Deterministic or nondeterministic
4. Episodic or sequential
5. Static or dynamic
6. Discrete or continuous
7. Known or unknown

Properties of Task Environments

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

Structure of Agents

- Agent program : implementation of agent function
  - Takes percept sequence as input
  - Outputs action
- Agent architecture : composition of physical device(s) running the agent program
  - Sensors
  - Actuators
  - Computing device
- **Agent = Program + Architecture**



Agent Program : Table-Driven Agent

- Keeps track of percept sequence
- Maintains a mapping between percept sequence and action in a table
  - Looks-up in the table to find the action
- Not practical - Table can get very large
  - 10^{150} entries for a chess program, 10^{80} atoms in the universe

function TABLE-DRIVEN-AGENT(*percept*) **returns** an action
  persistent: *percepts*, a sequence, initially empty
    table, a table of actions, indexed by percept sequences, initially fully specified

  append *percept* to the end of *percepts*
  action ← LOOKUP(*percepts*, *table*)
  return *action*

Table-Driven Agent

```python
def TableDrivenVacuumAgent():
    """Tabular approach towards vacuum world as mentioned in [Figure 2.3]
    >>> agent = TableDrivenVacuumAgent()
    >>> environment = TrivialVacuumEnvironment()
    >>> environment.add_thing(agent)
    >>> environment.run()
    >>> environment.status == {(1,0):'Clean' , (0,0) : 'Clean'}
    True
    """
    table = {((loc_A, 'Clean'),): 'Right',
             ((loc_A, 'Dirty'),): 'Suck',
             ((loc_B, 'Clean'),): 'Left',
             ((loc_B, 'Dirty'),): 'Suck',
             ((loc_A, 'Dirty'), (loc_A, 'Clean')): 'Right',
             ((loc_A, 'Clean'), (loc_B, 'Dirty')): 'Suck',
             ((loc_B, 'Clean'), (loc_A, 'Dirty')): 'Suck',
             ((loc_B, 'Dirty'), (loc_B, 'Clean')): 'Left',
             ((loc_A, 'Dirty'), (loc_A, 'Clean'), (loc_B, 'Dirty')): 'Suck',
             ((loc_B, 'Dirty'), (loc_B, 'Clean'), (loc_A, 'Dirty')): 'Suck'}
    return Agent(TableDrivenAgentProgram(table))
```

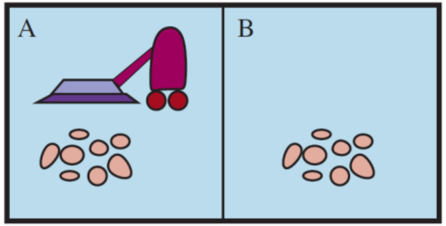
How to write programs that produce rational behaviour from a reasonably short program than from an enormous table?

Agent Program : Reflex Agent

- Use the latest percept to determine the action
  - Simple and fast: Implemented using condition-action rules
  - Running away from a snake, moving hand away from a hot surface

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*

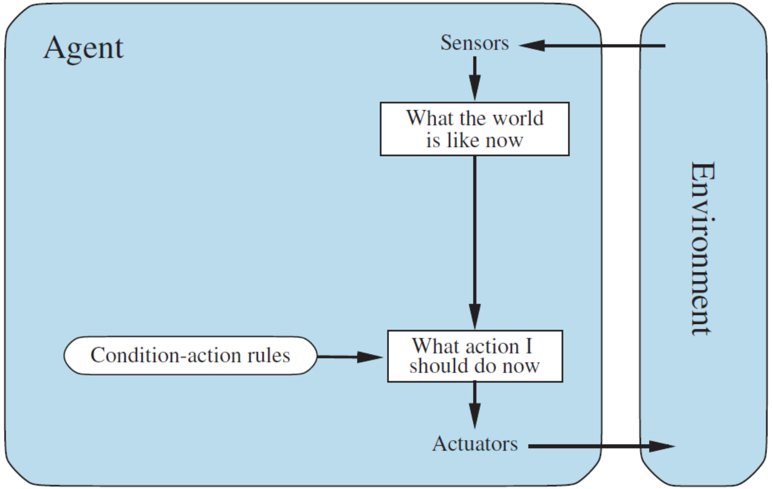


- Vacuum-world agent: **4**^{t} entries to **4** entries

Agent Program : Reflex Agent

- Reflex agents are appropriate when the latest percept can correctly determine the rational action
  - Problematic in environments that are not fully observable
    - No location sensor in the vacuum-world can cause infinite loop
      - Randomization can help

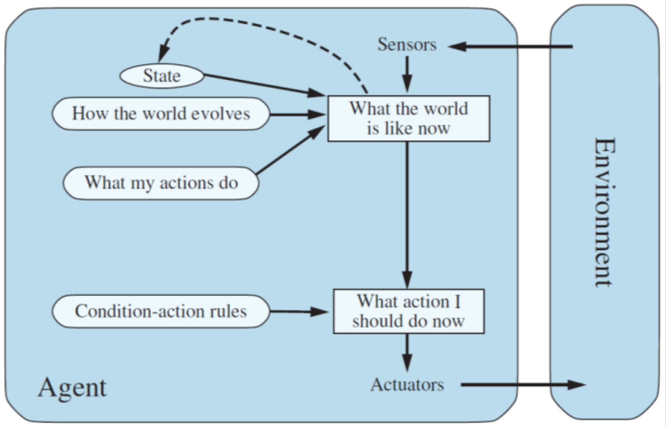
  - How to deal with partial observability??



Model-Based Reflex Agent

- The agent maintains internal state to handle partially observable environments using a model of how the environment works.
    - To gain an approximate information about the unobserved aspects
- It chooses an action as a reflex agent would do

Model-Based Agent

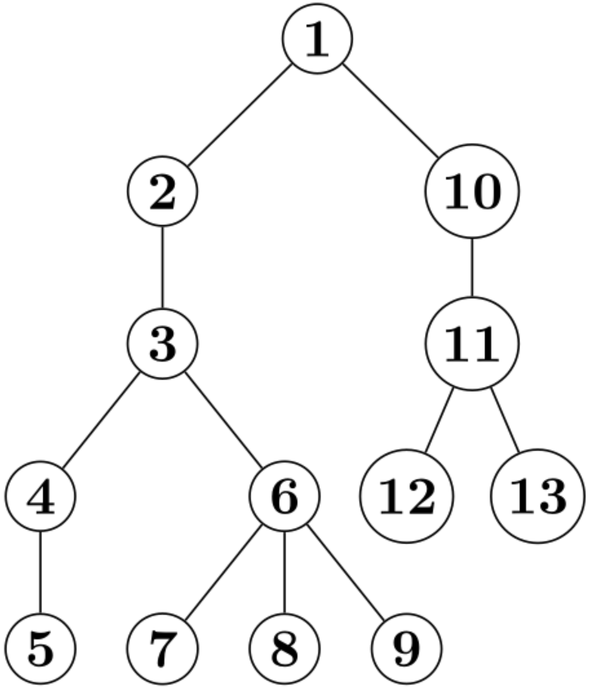


A model-based reflex agent.

- A model-based agent needs
  - A model of the evolution of the world and the result of agent's actions: **Transition Model**
    - Relationship between speed and distance covered
  - A model of how the state of the world is reflected in its percepts: **Sensor Model**
    - Darkness and on headlights means the sun has set

A Digression…

- DFS, BFS



Navigation example

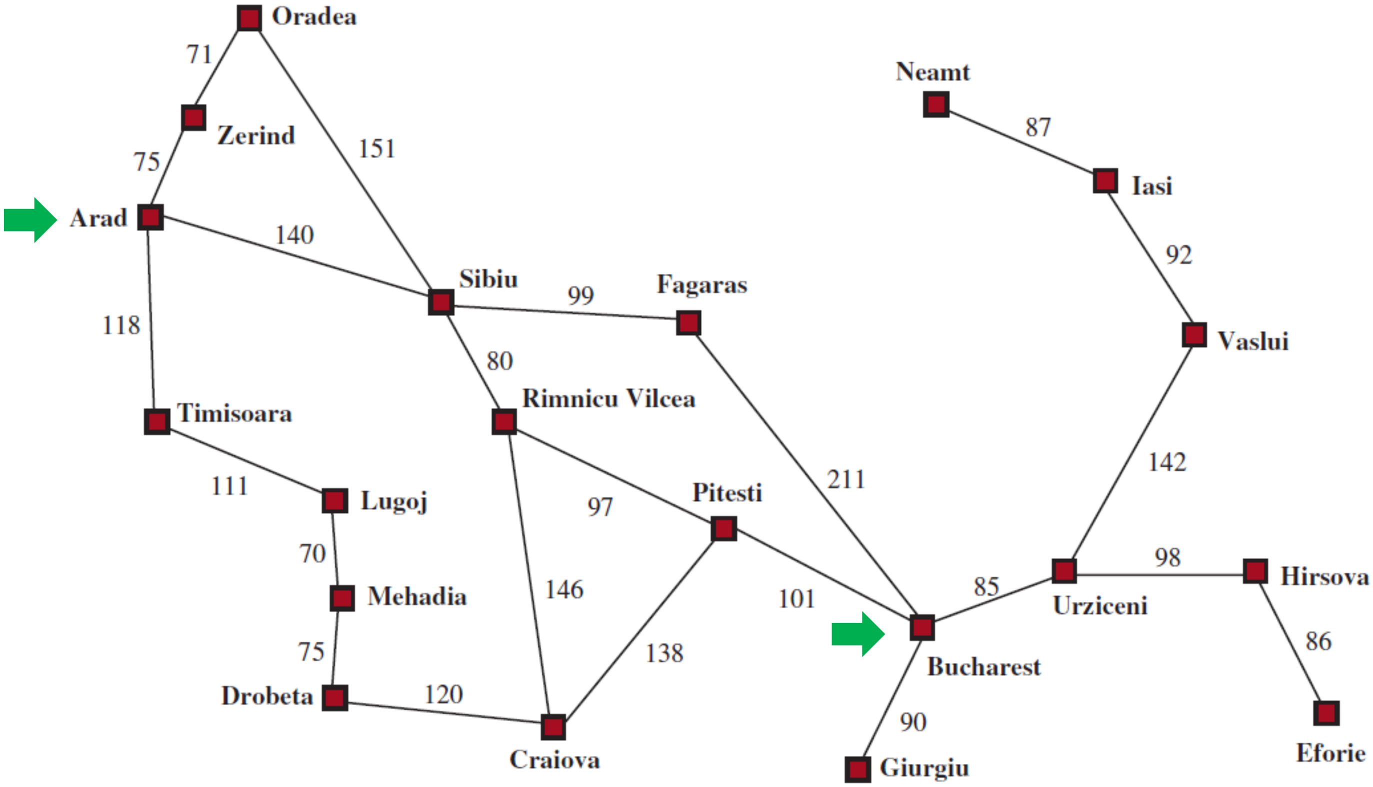


Figure 3.1 A simplified road map of part of Romania, with road distances in miles.

Navigation example

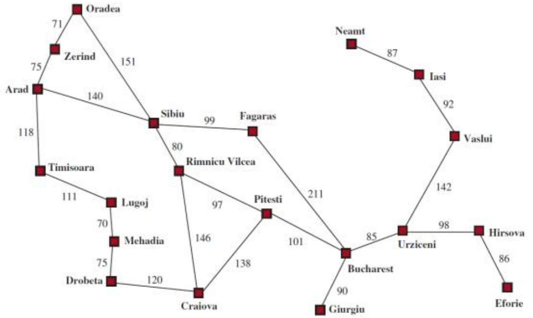


Figure 3.1 A simplified road map of part of Romania, with road distances in miles.

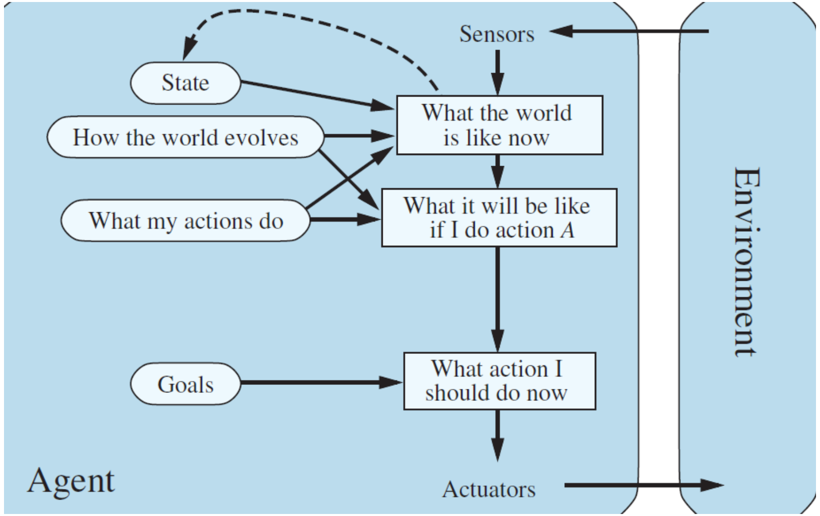
- Find a sequence of actions that form a path to the destination (goal state)
  - Called problem-solving agent
  - The computational process it undertakes is search
- Steps
  - Goal Formulation
  - Problem formulation: a description of the states and actions to reach the goal
  - Search: simulates sequences of actions in its model
  - Execution

Goal-Based Agent

- When many seemingly equally rational actions but with potentially very different outcomes are available, knowledge of only the environment is not enough
    - What to do at an intersection?
- Makes decisions based on explicit goals, considering future consequences
    - Use cases: Path planning in navigation

Goal-Based Agent

- A goal is needed to select actions
  - No direct mapping from percept to actions
- Goal-based action selection:
  - Trivial if episodic
  - Search
  - Planning
- Goal-based agents are flexible
  - Can change goals to do different tasks



Utility-Based Agents

- Goals can be achieved in different ways
  - There can be many ways to go from A to B
  - Goals provide binary classification
    - achieved (good) or not achieved (bad)
  - Want to maximize the performance measure
- An utility-based agent uses a utility function, which is a mapping from a sequence of states to a real number representing its desirability.
  - It selects actions that lead to maximum utility.

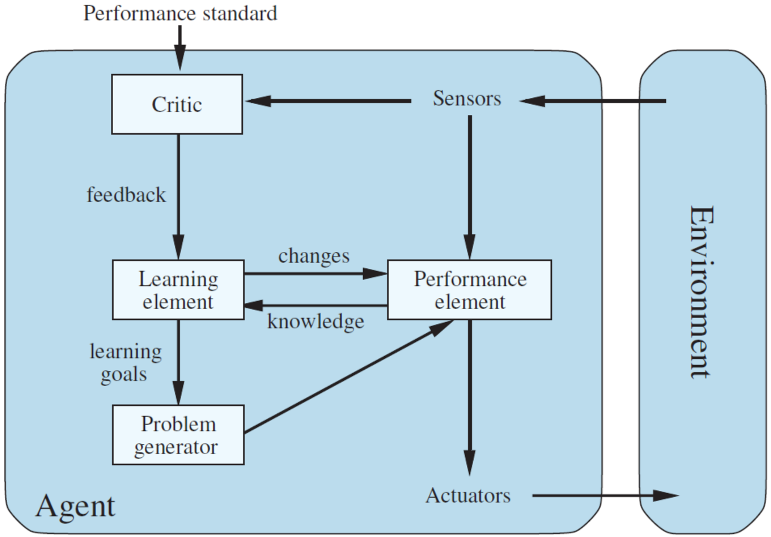
Learning Agents

- Learning agents are needed in unknown environments
- Improves its performance over time by learning from experience instead of a fixed set of rules
- Instead of writing (many, complex) agent programs, write a learning machine and teach it (Turing, 1950).

Learning Agents

Components of a learning agent:

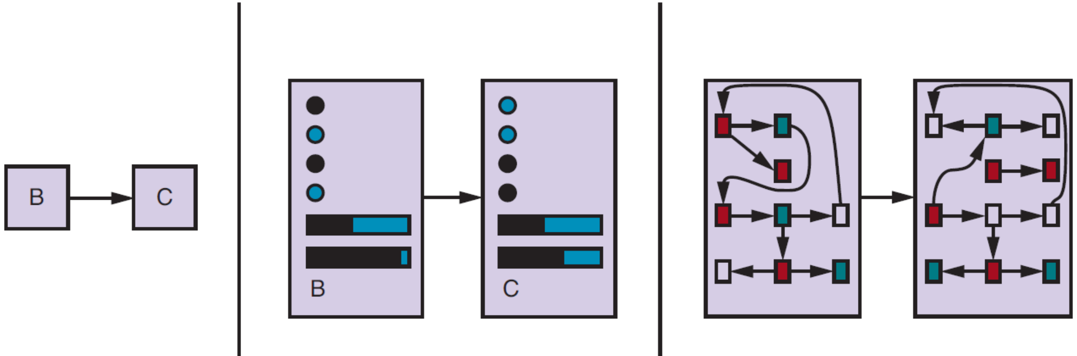
- Performance element selects actions
  - Was the entire agent before
- Critic: how well is the agent doing?
- Learning element
  - Uses feedback from critic on performance to make modifications in the performance element
- Problem generator suggests new actions for exploration
  - Can be suboptimal in the short term



State representations

- **Atomic**: A state has no internal representation
  - Indivisible black box
- **Factored**: State is represented by a vector of attribute values
- **Structured**: State includes objects and interacts with other objects

- From simplest to most complex
- From least to most expressive
- From least to most compact



(a) Atomic (b) Factored (c) Structured

Quiz

1. Which type of agent chooses actions by looking ahead to future states and selecting those that achieve a goal?
   A. Reflex agent
   B. Model-based agent
   C. Goal-based agent
   D. Learning agent

2. How is a model-based agent different from a reflex agent?
   A. It uses a model of the environment to track its state
   B. It acts randomly when it sees something unfamiliar
   C. It uses a reward function to choose between goals
   D. It has no access to percepts

Quiz

1. Which type of agent chooses actions by looking ahead to future states and selecting those that achieve a goal?
   A. Reflex agent
   B. Model-based agent
   C. Goal-based agent
   D. Learning agent

2. How is a model-based agent different from a reflex agent?
   A. It uses a model of the environment to track its state
   B. It acts randomly when it sees something unfamiliar
   C. It uses a reward function to choose between goals
   D. It has no access to percepts

Quiz

3. Which of the following would most likely be a reflex agent? A utility-based agent?
   A. A thermostat that turns on the heater if temperature < 20°C
   B. A chess-playing AI that evaluates board positions using win probability
   C. A web crawler that explores pages without a goal
   D. A GPS system that always follows the shortest path regardless of traffic

4. Which of these best describes a learning agent?
   A. Has perfect knowledge of the environment
   B. Repeats the same action for a given percept
   C. Improves its performance through feedback and experience
   D. Always chooses randomly to explore better options

Quiz

3. Which of the following would most likely be a reflex agent? A utility-based agent?
   A. A thermostat that turns on the heater if temperature < 20°C
   B. A chess-playing AI that evaluates board positions using win probability
   C. A web crawler that explores pages without a goal
   D. A GPS system that always follows the shortest path regardless of traffic
4. Which of these best describes a learning agent?
   A. Has perfect knowledge of the environment
   B. Repeats the same action for a given percept
   C. Improves its performance through feedback and experience
   D. Always chooses randomly to explore better options

Summary

- Agents perceive their environment through sensors and act upon it using actuators.
- A rational agent is one that selects actions to maximize its performance measure based on perception history and prior knowledge.
- The PEAS (Performance, Environment, Actuators, Sensors) framework describes the task environment of an agent.
- The agent's environment is classified based on properties like observability, determinism, single/multi agent, dynamism etc.
- Simple reflex agents, model-based reflex agents, goal-based agents, utility-based agents and learning agents are examples of agent types

Activities

Vacuum-World Agent Exercise

1. Run and understand the following implementation of a table-driven vacuum-world agent
2. Change the code such that the rooms get dirty again after 1 time step (if you wish, you can choose a random interval between 1 and 2 steps)
3. Make necessay changes to the code such that the agent runs for at-least 4 time staeps
4. Change the implementation to a reflex agent such that the vacuum keeps cleaning for ever

[17]:
```
from agents import Agent,Thing
import random
import numbers
```

1. Reading assignment: Chapter 2
2. Assignments: PS 1, agent.ipynb, exercise_1_agent.ipynb
3. Project: Proposal due on August 25, before class, hard copy

Next lecture: Search-Based Agents, Chapter 3