CS4618: Artificial Intelligence I

Agents

Derek Bridge
School of Computer Science and Information Technology
University College Cork

Initialization

In [1]:

```
%reload_ext autoreload
%autoreload 2
%matplotlib inline
```

In [2]:

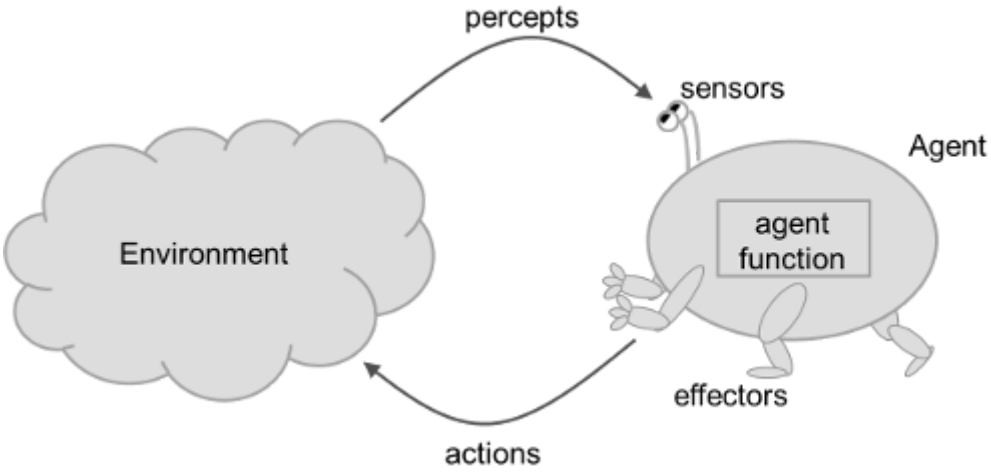
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Intelligence

- My far-from-perfect definition:
  A system's degree of intelligence is defined in terms of its capacity to act autonomously and rationally when faced with disorder, uncertainty, imprecision and intractability.
- Key points:
  - Intelligence is not a binary concept; it's a matter of **degree**
  - **Autonomous**, e.g. not under remote control; e.g. skills acquired by learning rather than instinct/pre-programming
  - **Rational**: acting so as to achieve your **goals**, given your **beliefs**
  - Certain situations are more challenging than others: **disorder**, **uncertainty**, **imprecision**, **intractability** (see also discussion of Environments, below)

Agents

- An **agent** is anything that can be viewed as **perceiving** its environment through **sensors** and **acting** upon that environment through **effectors**



Robots

- Robots are **embodied** agents, situated in **physical** environments

Software agents

- Software agents (sometimes called softbots) are situated in **virtual** environments

Sense, Plan, Act

- Sense
  - Use sensors to find things out about the environment
- Plan
  - Decide on the next action(s)
- Act
  - Use effector(s) to carry out the chosen action(s)

Action function

- The task of the **Plan** phase is to implement an **action function** that maps
  - from **percept sequences**
  - to the **actions** the agents can perform
- In intelligent agents, this function exhibits high degrees of
  - **autonomy** and
  - **rationality**

Environments

- Fully observable vs. partially observable
- Deterministic vs. stochastic
- Single-step vs. sequential
- Static vs. dynamic
- Discrete vs. continuous
- Single-agent vs. multi-agent

- Exercise: classify the environments of a chess program, a spam filter, a robot vacuum cleaner, and an autonomous car

Agents

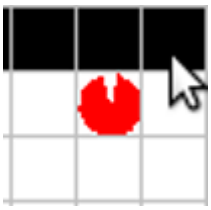
- Reactive agents
- Deliberative agents

Table-Driven Agents

- At each point in time:
  - $\boldsymbol{s} = SENSE()$;
  - $a = LOOKUP(\boldsymbol{s}, table)$;
  - $EXECUTE(a)$;

Class exercise: Table-Driven Wall-Following Agent

1. Suppose the agent has 8 touch sensors, each returning 0 or 1



Sensors return 11000001

How many table entries will there be?

Class exercise

2. In fact, only three sensors are needed:

How many table entries will there be?

Class exercise

3. The actions are:
   - MOVE:
     - this moves the agent one cell forward
   - TURN(d, n) where d = LEFT or d = RIGHT and n = 0, 1, 2, etc.:
     - this turns the agent to the left or right, n lots of 45°

Fill in the table so that the agent walks the walls anticlockwise

Percept	Action
000	
001	
010	
011	
100	
101	
110	
111	

Discussion

- Is this agent autonomous?
- When is the table-driven approach a *possible* approach?
- When is it a *practicable* approach?