Agents
What’s an agent?

- Russell and Norvig:
  “An agent is anything that can be viewed as perceiving its environment through sensors and acting on that environment through actuators.” (p. 32)

- Examples:
  - The automatic driver
  - Internet shopper
  - Backgammon player
  - Chemical plant controller
  - Spam detector
The agent and the environment

An agent:
- Works in a particular environment
- Has goals
- Perceives the environment
- Performs actions to achieve its goals.
Example: the automated driver

- Performance measures?

- Environment:
  - Roads, other traffic, pedestrians, weather

- Actuators:
  - Steering, accelerator, brake, turn signal, horn

- Sensors:
  - Cameras, LIDAR, RADAR, GPS engine and motion sensors, microphone
Example: the automated driver

- Possible performance measures:
  - Safe, fast, legal, comfortable trip.

- Environment:
  - Roads, other traffic, pedestrians, weather

- Actuators:
  - Steering, accelerator, brake, turn signal, horn

- Sensors:
  - Cameras, LIDAR, RADAR, GPS, engine and motion sensors, microphone
**Environments**

**Fully vs. partially observable**: can the sensors detect all aspects that are relevant to the choice of action.

<table>
<thead>
<tr>
<th></th>
<th>Crossword puzzle</th>
<th>Backgammon</th>
<th>Internet shopping</th>
<th>Automated driver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observable</td>
<td>fully</td>
<td>fully</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>Deterministic</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Episodic</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Static</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Discrete</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Agents</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Environments

**Deterministic vs. stochastic:** is the next environment state completely determined by the current state?

<table>
<thead>
<tr>
<th>Observable</th>
<th>Crossword puzzle</th>
<th>Backgammon</th>
<th>Internet shopping</th>
<th>Automated driver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deterministic</td>
<td>deterministic</td>
<td>stochastic</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>Episodic</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Static</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Discrete</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Agents</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Environments

### Episodic vs. sequential: can the agent’s experience be divided into steps where the agent’s action depends only on the current episode?

<table>
<thead>
<tr>
<th>Observable</th>
<th>Crossword puzzle</th>
<th>Backgammon</th>
<th>Internet shopping</th>
<th>Automated driver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observables</td>
<td>fully</td>
<td>fully</td>
<td>?</td>
<td>partially</td>
</tr>
<tr>
<td>Deterministic</td>
<td>deterministic</td>
<td>stochastic</td>
<td>?</td>
<td>stochastic</td>
</tr>
<tr>
<td>Episodic</td>
<td>sequential</td>
<td>sequential</td>
<td>sequential</td>
<td>sequential</td>
</tr>
<tr>
<td>Static</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Discrete</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Agents</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Environments**

**Static vs. dynamic:** can the environment change while the agent is choosing an action?

<table>
<thead>
<tr>
<th>Agents</th>
<th>Crossword puzzle</th>
<th>Backgammon</th>
<th>Internet shopping</th>
<th>Automated driver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observable</td>
<td>fully</td>
<td>fully</td>
<td>?</td>
<td>partially</td>
</tr>
<tr>
<td>Deterministic</td>
<td>deterministic</td>
<td>stochastic</td>
<td>?</td>
<td>stochastic</td>
</tr>
<tr>
<td>Episodic</td>
<td>sequential</td>
<td>sequential</td>
<td>sequential</td>
<td>sequential</td>
</tr>
<tr>
<td>Static</td>
<td>static</td>
<td>static</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>Static vs. dynamic</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**Agents**

- Crossword puzzle
- Backgammon
- Internet shopping
- Automated driver

**Characteristics:**
- Observable
- Deterministic
- Episodic
- Static
- Discrete
**Environments**

**Discrete vs. continuous:** This distinction can be applied to the state of the environment, the way time is handled and to the percepts/actions of the agent.

<table>
<thead>
<tr>
<th></th>
<th>Crossword puzzle</th>
<th>Backgammon</th>
<th>Internet shopping</th>
<th>Automated driver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observable</td>
<td>fully</td>
<td>fully</td>
<td>?</td>
<td>partially</td>
</tr>
<tr>
<td>Deterministic</td>
<td>deterministic</td>
<td>stochastic</td>
<td>?</td>
<td>stochastic</td>
</tr>
<tr>
<td>Episodic</td>
<td>sequential</td>
<td>sequential</td>
<td>sequential</td>
<td>sequential</td>
</tr>
<tr>
<td>Static</td>
<td>static</td>
<td>static</td>
<td>dynamic</td>
<td>dynamic</td>
</tr>
<tr>
<td>Discrete</td>
<td>discrete</td>
<td>discrete</td>
<td>discrete</td>
<td>continuous</td>
</tr>
<tr>
<td>Agents</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Environments

**Single vs. multi-agent:** Does the environment contain other agents who are also maximizing some performance measure that depends on the current agent’s actions?

<table>
<thead>
<tr>
<th></th>
<th>Crossword puzzle</th>
<th>Backgammon</th>
<th>Internet shopping</th>
<th>Automated driver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observable</td>
<td>fully</td>
<td>fully</td>
<td>?</td>
<td>partially</td>
</tr>
<tr>
<td>Deterministic</td>
<td>deterministic</td>
<td>stochastic</td>
<td>?</td>
<td>stochastic</td>
</tr>
<tr>
<td>Episodic</td>
<td>sequential</td>
<td>sequential</td>
<td>sequential</td>
<td>sequential</td>
</tr>
<tr>
<td>Static</td>
<td>static</td>
<td>static</td>
<td>dynamic</td>
<td>dynamic</td>
</tr>
<tr>
<td>Discrete</td>
<td>discrete</td>
<td>discrete</td>
<td>discrete</td>
<td>continuous</td>
</tr>
<tr>
<td>Agents</td>
<td>single</td>
<td>multi</td>
<td>multi</td>
<td>multi</td>
</tr>
</tbody>
</table>
The vacuum world

- Environment: squares A and B
- Percepts: [location and content] e.g. [A, Dirty]
- Actions: left, right, suck, no-op
A simple agent function

```python
def REFLEX-VACUUM-AGENT (location, status) :
    if status == Dirty:
        return Suck
    elif location == A:
        return Right
    elif location == B:
        return Left
```

Is this the best agent for the job?
Rational agents

- An agent has a performance measure according to what is wanted in the environment (goal).
- Performance measures for the vacuum world?

- A rational agent chooses an action which maximizes the expected value of the performance measure given the percept sequence and its built-in knowledge.
Agent structure

- Agent: architecture + program.
- The **agent program**: maps percepts to actions.
- The agent program receives as input the current percept and returns an action for the agent’s actuators.
Simple reflex agent

- Selects action only on the basis of the current percept
- Large reduction in possible percept/action combinations

function REFLEX-VACUUM-AGENT ([location, status])
  if status == Dirty then return Suck
  else if location == A then return Right
  else if location == B then return Left
Simple reflex agent

- Selects action only on the basis of the current percept.
- Large reduction in possible percept/action combinations.

Will our automated-driver agent work as a simple reflex agent?
Model-based reflex agent

- Maintain an internal state.
- Update the state using information on “how the world works” (the model of the world).
- Automated-driver agent: needs to keep track of cars in his blind spot.
Goal-based agents

- Our automated-driver agent needs to get somewhere: it has a *goal*. Chooses actions to achieve goal.

- **Search** and **planning** are subfields of AI devoted to finding a sequence of actions that achieve the agent’s goals.
Utility-based agents

- Utility function maps a (sequence of) state(s) onto a real number.
- Certain goals can be reached in different ways.
  - Some are better, have a higher utility.
- Improves on goals:
  - Selecting between conflicting goals.
  - Select appropriately between several goals that have varying probability of success.
Learning agents

- All previous agent-programs describe methods for selecting *actions*.
  - Yet it does not explain the origin of these programs.
  - Learning mechanisms can be used to perform this task.
  - Teach them instead of instructing them.
  - Advantage is the robustness of the program toward initially unknown environments.
Learning agents

- **Performance element**: selects actions based on percepts.
  - Corresponds to the previous agent programs

- **Learning element**: introduce improvements in performance element.
  - Critic provides feedback on agents performance based on fixed performance standard.