Agents and Environments

Agents include humans, robots, softbots, thermostats

The *agent function* maps from percept histories to actions

abstract mathematical description

The *agent program* runs on the physical architecture to produce the agent function.

concrete implementation
**Percepts**

- current location and contents
  - e.g., [A, dirty]

**Actions**

- Left, Right, Suck, NoOp
A Vacuum-Cleaner Agent

Agent Function

<table>
<thead>
<tr>
<th>Percept sequence</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>[A, Clean]</td>
<td>Right</td>
</tr>
<tr>
<td>[A, Dirty]</td>
<td>Suck</td>
</tr>
<tr>
<td>[B, Clean]</td>
<td>Left</td>
</tr>
<tr>
<td>[B, Dirty]</td>
<td>Suck</td>
</tr>
<tr>
<td>[A, Clean], [A, Clean]</td>
<td>Right</td>
</tr>
<tr>
<td>[A, Clean], [A, Dirty]</td>
<td>Suck</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Can it be implemented in a small program?

Agent Program

What is the right program?

```
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
```
Mike’s House as an Agent

Percepts

- motion detectors
- light level sensors
- door/window status
- sound level

Actions

- light intensity
- furnace on/off
- space heaters on/off
- water heater on/off
Rationality

Fixed *performance measure* evaluates sequence of actions performed by agent in environment.

- e.g., number of squares cleaned up by time $T$
- e.g., number of squares cleaned up per time step
- e.g., large penalty at time $T$ if any dirty squares remain

A rational agent chooses the action that maximizes the *expected value* of the performance measure given the percept sequence and *a priori* knowledge

i.e., could not build a better agent with the same built-in knowledge

does not imply that the agent is omniscient, clairvoyant, or successful

does require exploration, learning, autonomy
Specifying the Task Environment

To design a rational agent, we must specify the task environment.

PEAS

- performance measure
- environment
- actuators
- sensors

E.g., automated taxi

- performance measure: safety, speed, legality, comfort, profit
- environment: US streets & freeways, traffic, pedestrians, weather variation
- actuators: steering direction, accelerator, brake, horn, speaker/display
- sensors: cameras, sonar, speedometer, accelerometer, GPS, engine status
Internet Shopping Agent

User provides agent with a specification of product
  e.g., Treos 600 phone
  e.g., cell phone under $400 with PDA functions

Agent finds lowest prices on web.

Agent bids on Ebay to get a lower price.

PEAS

  performance?
  environment?
  actuators?
  sensors?
<table>
<thead>
<tr>
<th>aspects of environment</th>
<th>Solitaire</th>
<th>Backgammon</th>
<th>Internet Shopping</th>
<th>Taxi</th>
</tr>
</thead>
<tbody>
<tr>
<td>fully observable versus partially observable</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>next state of environment is deterministic versus stochastic</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>episodic or sequential task</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>static versus dynamic environment</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>discrete versus continuous (env., percepts, actions)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>single or multiple agents</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Environment Types

<table>
<thead>
<tr>
<th>aspects of environment</th>
<th>Solitaire</th>
<th>Backgammon</th>
<th>Internet Shopping</th>
<th>Taxi</th>
</tr>
</thead>
<tbody>
<tr>
<td>fully observable versus partially observable</td>
<td>partially</td>
<td>fully</td>
<td>partially</td>
<td>partially</td>
</tr>
<tr>
<td>next state of environment is deterministic versus stochastic</td>
<td>deterministic</td>
<td>stochastic</td>
<td>semi deterministic</td>
<td>stochastic</td>
</tr>
<tr>
<td>episodic or sequential task</td>
<td>sequential</td>
<td>sequential</td>
<td>sequential</td>
<td>sequential</td>
</tr>
<tr>
<td>static versus dynamic environment</td>
<td>static</td>
<td>static</td>
<td>semi static</td>
<td>dynamic</td>
</tr>
<tr>
<td>discrete versus continuous (env., percepts, actions)</td>
<td>discrete</td>
<td>discrete</td>
<td>discrete</td>
<td>continuous</td>
</tr>
<tr>
<td>single or multiple agents</td>
<td>single</td>
<td>multiple</td>
<td>multiple (auctions)</td>
<td>multiple</td>
</tr>
</tbody>
</table>
Agent Types

Four basic types (in order of increasing generality)

- simple reflex agents
- reflex agents with state
- goal-based agents
- utility-based agents

Any of these can be made into a learning agent.
Simple Reflex Agents

Agent

Environment

Sensors

What the world is like now

Condition–action rules

What action I should do now

Actuators
Reflex Agents With State

Agent

- State
  - How the world evolves
  - What my actions do
  - Condition-action rules

Environment

- Sensors
  - What the world is like now

- Actuators
  - What action I should do now
Goal-Based Agents

Agent

Environment

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

Actuators

Sensors
Utility-Based Agents

Agent

- State
- How the world evolves
- What my actions do
- Utility

Sensors

- 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

Environment

Actuators

- What action I should do now
Learning Agents

- **Critic**
- **Sensors**
- **Performance standard**
- **Learning element**
- **Problem generator**
- **Performance element**
- **Environment**
- **Agent**

Arrows indicate:
- Feedback
- Changes
- Knowledge

- Learning goals
- Actuators
### Wumpus World

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>START</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PIT</td>
</tr>
<tr>
<td>Stench</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gold</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PIT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Breeze</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Cave consisting of rooms connected by passageways
- Wumpus lurks in cave, eats anyone entering room
- Agent can shoot wumpus
- Some rooms contain bottomless pits
- One room contains pot of gold
Wumpus World

Performance measure

+1000 for picking up gold
-1000 for falling into a pit or being eaten by wumpus
-10 for shooting an arrow
-1 otherwise

Environment

4x4 grid of rooms
agent always starts at [1,1] facing right
gold and wumpus rooms chosen at random with all rooms other than [1,1] being equiprobable; gold and wumpus cannot be in the same room
each room has probability .2 of containing a pit (excluding [1,1] and room containing gold)
Wumpus World

Sensors

Is there a stench in the current room?
Is there a breeze in the current room?
Is there a glitter?
Did the agent perceive a bump (i.e., walked into a wall)?
Was a woeful scream heard (i.e., wumpus was killed)?

Actuators

move forward to adjacent room (but agent cannot walk off grid)
turn left 90 degrees
turn right 90 degrees
grab object in room (picks up gold if gold is in room)
fire an arrow in current direction (kills wumpus if in adjacent room)