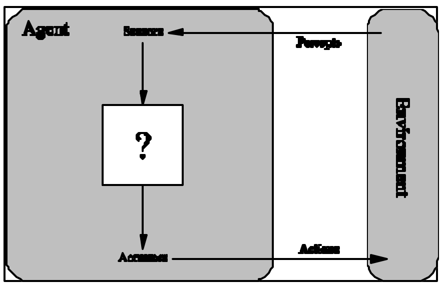
Agents and Environments



Example: Vacuum Cleaner Agent

- agent: robot vacuum cleaner
- environment: floors of your apartment
- sensors:
  - dirt sensor: detects when floor in front of robot is dirty
  - bump sensor: detects when it has bumped into something
  - power sensor: measures amount of power in battery
  - bag sensor: amount of space remaining in dirt bag
- effectors:
  - motorized wheels
  - suction motor
  - plug into wall? empty dirt bag?
- percepts: "Floor is dirty"
- actions: "Forward, 0.5 ft/sec"

Rational Agent

- Performance Measure: Criteria for determining the quality of an agent's behavior
  - Example: dirt collected in 8 hour shift
- Avoiding Omniscience
  - An omniscient agent is one that can predict the future perfectly. We don't want this!
- Agent: Mapping from percept sequences to actions

Defn: Ideal Rational Agent

- For each percept sequence, choose the action that maximizes the expected value of the performance measure given only builtin knowledge and the percept sequence

Policies

- Policy: A mapping from percept sequences to actions
- Agent programming: designing and implementing good policies
- Policies can be designed and implemented in many ways:
  - Tables
  - Rules
  - Search algorithms
  - Learning algorithms

Implementing Agents Using Tables

```
function TABLE-DRIVEN-AGENT(percept) returns action
   static: percepts, a sequence, initially empty
           table, a table, indexed by percept sequences, initially fully specified

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

- Problems:
  - Space: For chess this would require 35^{100} entries
  - Design difficulty: The designer would have to anticipate how the agent should respond to every possible percept sequence

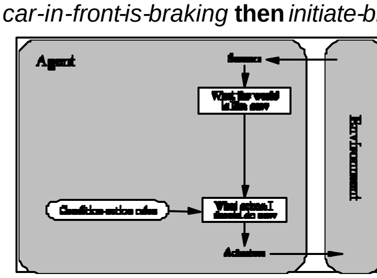
Avoiding Tables

- Compact Representations of the Table. Many cells in the table will be identical.
  - Irrelevant Percepts: Example: If the car in front of you slows down, you should apply the breaks. The color and model of the car, the music on the radio, the weather, and so on, are all irrelevant.
  - Markov Environments: Example: In chess, only the current board position matters, so all previous percepts dictate the same move.

    Environments where this is always true are called *Markov Environments*

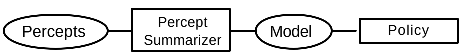
Example of Compact Representation: Implementing Agents using Rules

If *car-in-front-is-braking* **then** *initiate-braking*

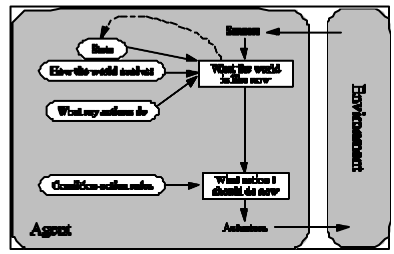


Avoiding Tables (2)

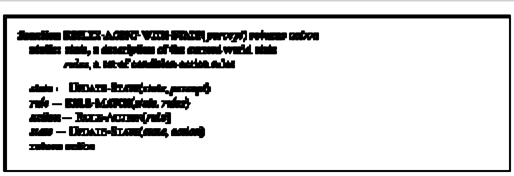
- Summarizing the Percept Sequence
  - By analyzing the sequence, we can compute a *model* of the current state of the world. With this state, the agent can act as if the world is a Markov environment



Summarizing Percepts as Environment Model

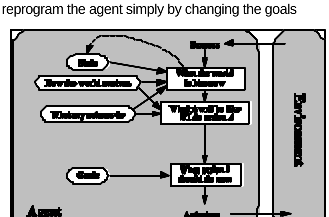


Pseudo-Code



Goal-Based Agents

- Generate possible sequences of actions
- Predict resulting states
- Assess goals in each resulting state
- Choose an action that will achieve the goal
- We can reprogram the agent simply by changing the goals

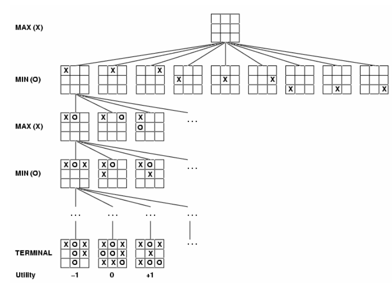


Goal-Based Agents compute the desired action on demand

- In many cases, the agent can *compute* the desired action rather than looking it up. This trades extra CPU time to reduce memory.
  - Example: Deep Blue

Oregon State University – CS430 Intro to AI

Example of Computing Table Dynamically



Oregon State University – CS430 Intro to AI

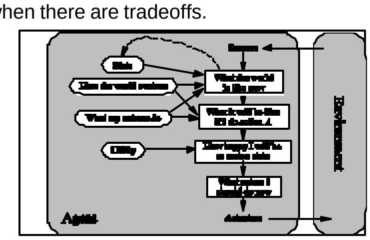
Problems with Computing Table Dynamically

- Search space may be exponentially large
  - Computing the best action may be computationally intractable
- World may change while we are searching
  - In a dynamic environment, we must act promptly
- Knowledge of the world may be incomplete or wrong
  - We may not be able to accurately predict the future

Oregon State University – CS430 Intro to AI

Utility-Based Agents

- In some applications, we need to make *quantitative* comparisons of states based on *utilities.* Important when there are tradeoffs.



PEAS Descriptions

- P: Performance Measure
- E: Environment
- A: Actuators
- S: Sensors

Examples of agent types

[redacted]

Different Kinds of Environments

- Fully-observable vs. Partially-observable
  - Fully-observable = Markov
- Deterministic vs. Stochastic
  - Strategic: deterministic except for the actions of other agents
- Episodic vs. Sequential
- Static vs. Dynamic
- Discrete vs. Continuous
- Single agent vs. Multiagent

Examples of Environments

Env	Observable	Deterministic	Episodic	Static	Discrete	Agents?
Crossword puzzle	Fully	Deterministic	Sequential	Static	Discrete	Single
Chess w/clock	Fully	Strategic	Sequential	Semi	Discrete	Multi
Poker	Partially	Strategic	Sequential	Static	Discrete	Multi
Backgammon	Fully	Stochastic	Sequential	Static	Discrete	Multi
Taxi driving	Partially	Stochastic	Sequential	Dynamic	Continuous	Multi
Medical Dx	Partially	Stochastic	Sequential	Dynamic	Continuous	Single
Image analy	Fully	Deterministic	Episodic	Semi	Continuous	Single
Part-picking	Partially	Stochastic	Episodic	Dynamic	Continuous	Single
Refinery contr	Partially	Stochastic	Sequential	Dynamic	Continuous	Single
English tutor	Partially	Stochastic	Sequential	Dynamic	Discrete	Multi

Advantages of Simpler Environments

- Observable: policy can be based on only most recent percept
- Deterministic: predicting effects of actions is easier
- Episodic: Do not need to look ahead beyond end of episode
- Static: Can afford lots of time to make decisions
- Discrete: Reasoning is simpler

Learning Agents