Introduction to Computer Science and Programming in C

Session 24: December 2, 2008

Columbia University

Announcements

- Homework 4 is out, due last day of class: December 4 before class
- Final Review Thursday 12/4
- Final Exam: Tuesday, 12/16, 1:10 pm - 4:00 pm Mudd 233 (our normal room)

Review

- A look at C++ and Java

Artificial Intelligence

- General discussion of AI
- Problem Solving (by searching)
- Machine learning

- Some examples from *Artificial Intelligence: A Modern Approach* by Russell, Norvig

Intelligent Agents

- Act “rationally” (performance metric)
- Percepts, actions, goals, environment
- e.g., Medical diagnosis system
    - P: Symptoms, findings, patient’s answers
    - A: Questions, tests, treatments
    - G: Healthy patient, minimize costs
    - E: Patient, hospital

Types of Agents

- Reflex agents – simple reaction to percepts:
if *car-in-front-is-braking* then *initiate-braking*

- Internal state agents – store in memory information about the world

    - Updating state becomes one of the actions the agent must consider

- Goal-based – best action depends on goal

Problem Solving

- A formal definition of a problem includes:
  - state space, initial state, goal state
  - actions/operators, successor function
  - path, path cost, goal test
- Find a path to the goal state with minimal cost:
  - Search cost vs. path cost

Problem Example: 15 puzzle

- State: the location of 1-15 and a space on a 4x4 board.
- Operators: blank moves left, right, up, down
- Goal test: check if numbers are in order
- Path cost: each step costs 1

Search Strategies

- Search strategies can have these properties:
  - Completeness: will find solution if one exists
  - Time complexity: how many steps to soln
  - Space complexity: how much memory
  - Optimality: if multiple solns, will find best

Depth First

- Try first untested operator until no more operators possible
- If not at goal, backtrack and try next operator
- DFS fails if infinite search paths

Breadth First

- Try each operator at each level before trying next operator

Iterative Deepening

- Run depth first search to limited depths, iteratively increasing depth

Best-First Heuristic Search

- We can design a function that approximates how close each state is to solution based on prior knowledge
- e.g., in the 15 puzzle, we can use the number of moves if we directly move each block to its final destination
- Best-First search – explore state with least heuristic score h(x)

A* Search

- Explore state with the best current path cost plus heuristic cost: g(x)+h(x)
- Given certain mathematical properties of h(x), A* is optimal and often the best way to solve problems like 15-puzzle

Search Strategies

	Depth First	Breadth First	Iterative Deepening	"Best" First	A*

Time	O(b^m)	O(b^d)	O(b^d)	O(b^m)	O(b^d)
					O(p(d))

Space	O(bm)	O(b^d)	O(bd)	O(b^m)	O(b^m)
Optimal	No	Yes	Yes	No	Yes
Complete	No	No	Yes	No	Yes

A* demo

- http://www.cs.ualberta.ca/~aixplore/search/IDA/Applet/SearchApplet.html

Learning

- How do we store and update internal states to make rational decisions after seeing data?
- Machine learning: use Statistics and AI techniques together
- Example: Face recognition
- Supervised versus unsupervised
- Online versus batch learning

Learning

- Some goals of Machine Learning
    - Classification
    - Regression
    - Clustering