Problem Solving

Hwk 5 due today.
Hwk 6 available; due 4/16

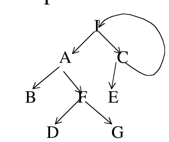
I. What is Problem Solving?
II. Problem spaces etc
III. Types of Problems
IV. Strategies for Solving
V. Insight problems
VI. Expertise

I. What is Problem Solving

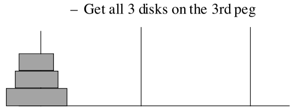
- What makes a problem a problem?
  - Goal State - outcome
  - Initial State - different from goal state
  - Operators - actions, processes to get from Initial to Goal
- A problem state is a description of the current status of all the elements of a problem.
- Examples?
  - Doing HW 5
  - Getting up in the morning
  - Being happy
  - Doing a crossword puzzle

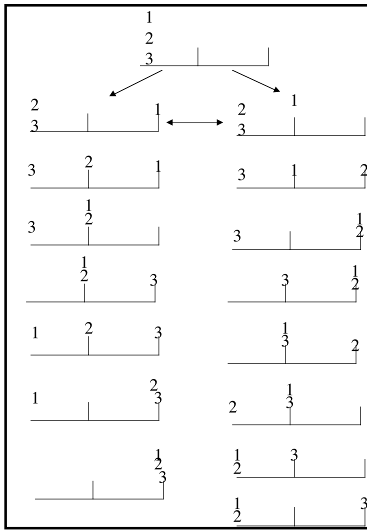
II. Problem Space

- Solving a problem then involves moving from initial state to goal state.
- At each step, an operator is applied.
- Visual depiction of states and transitions leads to *problem space,* the set of all possible problem states and their operator connections



- Functions of Operators
  - Describing the Problem Space
    - Operators generate the space of possible states from initial to goal state.
    - Problem:
      - For the following TOH problem, define the space of possible states.
      - Get all 3 disks on the 3rd peg





III. Types of Knowledge used in Problem Solving

- Declarative vs Procedural
- Explicit vs Implicit Knowledge

Problem solving can involve various types of knowledge depending on the type of problem being solved and the expertise of the problem solver

How psychologists Classify Problems

- Knowledge Rich vs Lean
  - Rich - requires specific, domain knowledge.
  - Lean - no specific domain knowledge required
- Insight vs routine
  - Insight - Cognitive Restructuring, sudden solution
  - Routine - familiar, known pathway or types of operators to solution

When you aren't familiar with the domain

- Solver doesn't know a relevant *algorithm,* a defined set of steps guaranteed to find a solution
- Many problems (esp knowledge lean) may be solved by *heuristics*, generic strategies for solving
  - Working forward: onward & upward
  - Working backward: from the goal
  - Means-ends analysis (see slide)
  - Generate-and-test: trial & error
  - Hill-climbing (see slide)

V. Strategies for solving: **Means-Ends Analysis**

What is the biggest difference between current & goal state?

Can we immediately eliminate that difference with an operator?

If yes & operator can be applied, :)

If yes but obstacle to applying oper, then new subgoal is to get situation where operator can be applied

Example: h s student -> $$$

Current state: _____________

Goal State: ______________

1 Biggest difference: ______

2 Operator to eliminate? _____

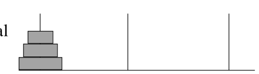
3 Obstacle preventing operator? ___

4 New subgoal: _____________

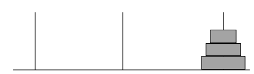
Repeat at step 1, with new subgoal

Tower of Hanoi Problem

Initial

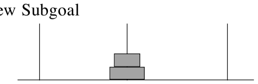


Goal



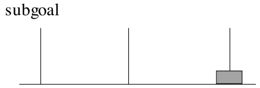
Biggest difference? ____________
Operator to eliminate difference? _________
Obstacle preventing this operator? ________

New Subgoal



Biggest difference? _____________
Operator to eliminate difference? ____________
Obstacle preventing this operator? ________

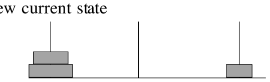
New subgoal



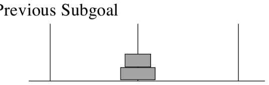
Biggest difference? ____________
Operator to eliminate difference? _________
Obstacle preventing this operator? ________

No… then just apply operator; subgoal achieved!

New current state

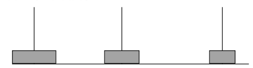


Previous Subgoal

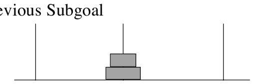


Biggest difference? _____________
Operator to eliminate difference? ____________
Obstacle preventing this operator? ________
No… then just apply operator...

New current state

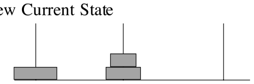


Previous Subgoal



Biggest difference? _____________
Operator to eliminate difference? ____________
Obstacle preventing this operator? ________
No… then just apply operator; subgoal achieved!

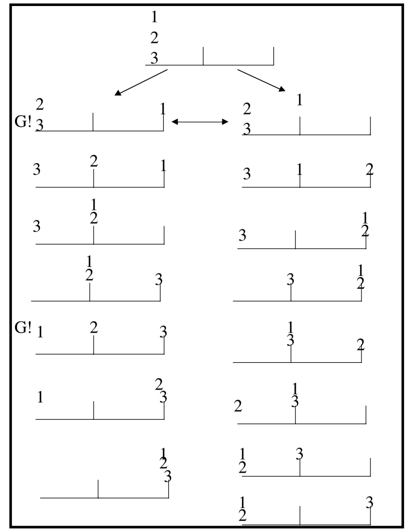
New Current State



Do you remember the (sub)goal before this?
Biggest difference? _____________
Operator to eliminate difference? ____________
Obstacle preventing this operator? ________
No… then just apply operator...

Why care?

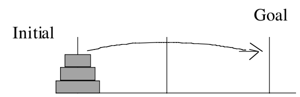
- Means-ends analysis seems to fit problem solvers' data
  - It is a strategy people naturally use!
- Means-ends analysis enables computer programs to solve complex problems (AI)
- Means-ends analysis is a strategy we can all consider to help us solve difficult (well-defined) problems.
  - Subgoaling helps in complex problems!



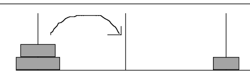
V. Strategies for solving: **Hill Climbing**

- reduce distance from initial state to goal state
- If operator doesn't reduce distance, discard and find new operator.
- Each step is local (i.e., can't subgoal a priori to figure out which step in the long run is going to be closer to the goal)

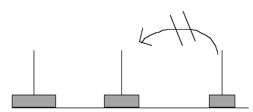
Example: Get all 3 disks on Peg 3



Move gets closer to goal of all disks on 3rd peg



Closer to goal since Peg 2 is closer to goal peg



Hill-climbing fails because requires moving away from goal state (reversing previous move)

Insight Problems

- Insight Problems
  - Require re-thinking or re-representation of problem
  - How we move through problem space depends on how we represent the problem (i.e., what is in each state, what the operators are)
  - Natural representation may produce complex p space (long solution path), but with new representation, new problem space produces short path)
  - Think back to mutilated checkerboard
  - Another example: *kids* solving balance beam task

Insight & Incubation

- Sudden solution discovery
  - For insight problems subjects do not report being close to solution 15 sec prior to solving (Metcalfe & Weibe, 1987)
- Verbalization hurts performance
- Time away often leads to solution upon return. (Incubation Effects)

Consistent w/ need to re-represent…

Is insight problem solving special?

More on incubation

Silveira's incubation study (necklace)

- 3 groups worked for 30 minutes (no break, 30 min break, 4 hour break)
- Results - more success with longer break

- Why does it help?
  - Upon return, it's easier to get a fresh view (representation) of the problem
  - Memory for operators generated may decay, letting new ones have a chance

What else makes problems hard?

- Complexity of problem space:
  - More branches, more possible errors
- Problems with equally complex p spaces can still differ in difficulty
  - Problem isomorphs have same pspace
  - When current state & operators harder to remember, isomorph harder to solve
- Mental Sets
  - Get into a "rut" of solving problems one way, it's hard to see other ways
  - Functional fixedness - see an object as having only 1 (typical) function
  - Think back to 2-string problem

When you don't know the answer...

- Fully explore the problem first
  - Spend time exploring different pathways to solutions, encoding all the aspects of the problem.
- Allow sufficient time for incubation to occur
  - Step away from the problem
  - Relax, let your mind wander

- Will these strategies help more for ill/well-defined problems? Insight/routine problems?

Expertise

- If you want to solve problems better (or teach) … ask :
- What makes an expert an expert?

1. Knowledge
   - Amount, Org'n & Use
     - Chase & Simon Chess Experts
     - Larkin et al - Physics
     - Chi - Dinosaurs
2. Schemas
   - Large & interconnected
   - Organization based on deep conceptual similarity
   - Evidence experts' schemas different
     - Classification of problems
     - How they determine and describe a solution method

3. Time allotment
   - Experts spend more time on determining a representation
   - Experts spend less time on implementing the solution strategy
4. Operator Selection
   - Experts work forward
   - Novices work backward
5. Automaticity
   - Consolidation of sequences of steps that require little conscious control.
     - Benefits
       - Frees up WM for other things, e.g., monitoring
6. Talent or just good practice?