

Comprehensive Final Exam Answers with FFQ(TM) Feature.

CSC 242

May 2004

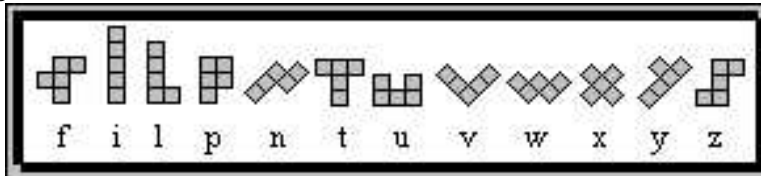
Write your **NAME** legibly on the bluebook. Work all problems. Best strategy is not to spend more than the indicated time on any question (minutes = points). There are 120 points. Open book, open notes.

The FFQ side comments appear in answers in italics like this.

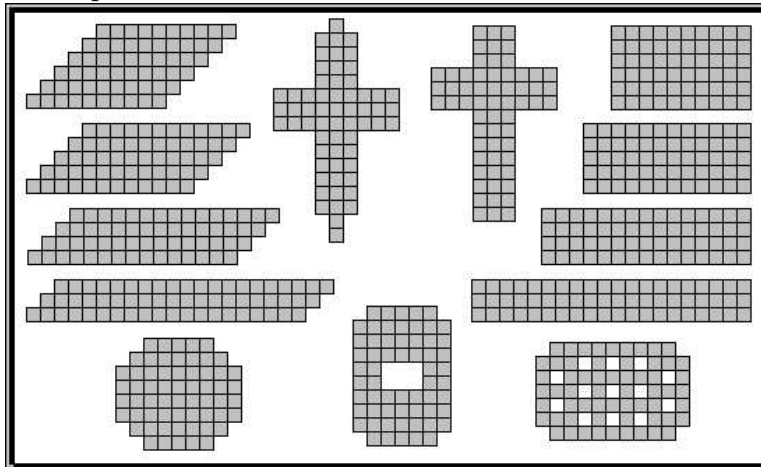
1. Search (10 mins)

Here are two well-known puzzles. Describe in detail how you would apply state space search to each. If you see any difficulties with that approach to either puzzle, describe them.

A. Pentominoes

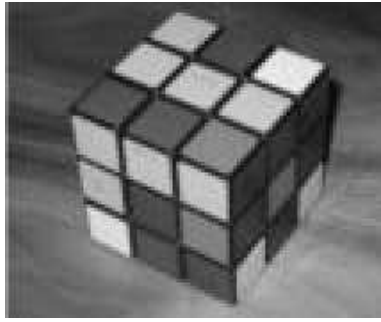


The idea is to assemble these twelve shapes (all the “4-connected” shapes of five squares) like a jigsaw puzzle into other shapes like those shown below.



B. Rubik's Cube

See Fig. below. Each layer in each of the three dimensions can be rotated independently of the others. The idea is to perform rotation operations until each cube face has a uniform color.



Answers:

A. Pentominoes are a natural for state-space search, and in fact I have a screensaver that does it. Goal state is an array with legal squares where you want tiles and illegal ones where you don't. Initial state, it's empty. Operations are simply placing shapes of tiles in the array. Shapes can be represented as smaller arrays. Placing them includes rotating the shape representation and again symmetry arguments can save search. If can place shape legally (all inside goal shape, no overlap with previously placed shapes) do it. If done, declare victory. If not, iterate. If can't place tile, backtrack. Usual thing. Some heuristics might involve which tiles you put down first, where to start filling in, etc.

B. Rubik Representation is a moderately interesting issue: I'll be looking for good ideas here. after a little thought I don't see anything much better than representing each primitive operations as a permutation of the 54 faces. However, there some symmetries that could be used to keep from exploring the same operation twice.

FFQ: Only a VERY FEW people noticed you do NOT want to tackle RC with SSSearch! Nobody who's ever tried would dream of such a crazy idea. So maybe it's a generational thing and noone does RC any more or maybe it's a failure of imagination for not seeing that ops screw up your state in a serious way, but lots of points were lost by blindly following some recipe, despite my asking if this was a good problem for SSS. With such simple operations, Rubik's cube would be a nightmare for state space search since every simple operation screws up the effect of other operations. One way to make progress is to develop macro-operations that actually undo the damage they do on the way to making some simple change you want. Given them as primitive operations a search approach should be possible, but I don't know the details of how complete such an operator set is, and I think there are still plenty of things to go wrong... anyway this is a hard one.

2. More Search (15 min)

Motivated by the idea of iterative deepening search, the text mentions (p. 79) Iterative Lengthening search, pointing to exercise 3.11 to explore difficulties arising from non-uniform step costs. Again Exercise 3.11 turns up on page 101 in the context of iterative-deepening A*, where non-uniform step costs cause trouble. So...here's —

(Exercise 3.11) Iterative lengthening search is an iterative analog of uniform cost search. The idea is to use increasing limits on path cost. If a node is generated whose path cost exceeds the current limit, it is immediately discarded. For each new iteration, the limit is set to the lowest path cost of any node discarded in the previous iteration.

A (5 min). Show that this algorithm is optimal for general path costs.

B (5 min). Consider a uniform tree with branching factor b , solution depth d , and unit step costs. How many iterations will iterative lengthening require.

C (5 min). Now consider step costs drawn from a continuous range $[0, 1]$ with a minimum

positive cost ϵ . How many iterations are required in the worst case?

Answer:

A: Algorithm expands nodes in order of increasing path cost: therefore the first goal it encounters will be the goal with cheapest cost.

B: It will be the same as iterative deepening, i.e. d iterations, in which $O(b^d)$ nodes are generated.

C: d/ϵ . The message is that if you want to guarantee optimality, you might have to spend a lot of effort crawling along making really short steps.

FFQ: people didn't do well on part C.

3. Short Answers (15 mins)

Define, discuss, or describe these terms in a sentence or two:

1. Case Grammar
2. Alpha-Beta Filter
3. Conceptual Dependency
4. Modulation Transfer Function
5. Power Spectrum
6. Low-Pass Filter
7. Hough Transform
8. Cubist Vision

Ans:

FFQ: this was designed to smoke out the class-cutters and it certainly did that, which is why 3/4 of people got really crappy grades here. It, however, also smoked out the NON-note takers, which is why the remaining grades weren't much better. Let's all come to class and let things wash in one ear and out the other... hmmm.

1. Case Grammar: Identify roles in generic situations (e.g. actor, instrument, recipient) and incorporate them into grammar. Result less dependent on syntax, more on semantics, helps lexical and structural disambiguation.
2. Alpha-Beta Filter: Kalman Filter restricted to constant-noise and simple kinematic (e.g. constant velocity, constant acceleration) situations. Used for tracking. *FFQ: NOT alpha-beta pruning!! Trick question, you decide...*
3. Conceptual Dependency: Roger Schank's interlingua, or canonical system for representing sentences in terms of the and basic action-types. Goal is to have one unambiguous representation of a sentence from which paraphrases and translations can be produced. *FFQ: Yeah, sounds like something about independence or how concepts influence each other but it ain't.*
4. Modulation Transfer Function: Fourier transform of impulse response: gives attenuation and amplification of frequencies by the linear system being described.

5. Power Spectrum: Squared modulus of Fourier transform: How much energy is in signal at each frequency. *NON-FFQ: Congrats! Lots of people cottoned onto this concept (can I say that?)*
6. Low-Pass Filter: throws away high-frequency components of a signal. *FFQ: Low-PASS filter, not Low-BLOCK filter!!*
7. Hough Transform: mode-based estimation strategy based on voting: explanation of data with most votes wins. Deals with outliers better than mean.
8. Cubist Vision: Randal Nelson's four-tier approach to vision using curve segments, contextual patches, and multiple views.
- 9.

4. Resolution (30 Mins)

A. (25 Mins) Here's a theory of admiration:

1. *Everyone admires a hero.*
2. *A failure admires everyone.*
3. *Anyone who is not a hero is a failure.*

With these assumptions, we want to prove the claim that

4. *There is a pair of individuals (not necessarily distinct) who admire one another.*

Put the sentences into FOPC, likewise with the negated conclusion, put everything into clauses, and prove the collection unsatisfiable by resolution. Showing a nice tree of what cancels is a friendly way to give the proof. (Hint: Factoring!)

FFQ: MY fault. These bloody FOPC questions are SO delicate! No wonder I hate making them up. So I stole this one from another final and of course SNAFU. That first premise is a canonically ambiguous one and it's not easy to rephrase: "people admire heros" might be better. Anyway what I wanted is below. It's the obvious reading, and it's what I meant to get across when I answered the in-test question. Several people invented an unnecessary "Person(x)" predicate that muddied the waters...

Answer:

FOPC:

1. $\forall x \forall y (H(x) \rightarrow A(y, x))$
2. $\forall x \forall y (F(x) \rightarrow A(x, y))$
3. $\forall x (\neg H(x) \rightarrow F(x))$
4. $\exists x \exists y (A(x, y) \wedge A(y, x))$
5. = $\neg 4. \forall x \forall y (\neg A(x, y) \vee \neg A(x, y))$

Clauses

1. $\neg H(w) \vee A(z, w)$
2. $\neg F(u) \vee A(u, v)$

3. $H(x) \vee F(x)$
 5. $\neg A(s, t) \vee \neg A(t, s)$

Resolution Proof:

- 3 and 1 give 6: $F(w) \vee A(z, w)$ (or equivalent)
 2 and 6 give 7: $A(z, w) \vee A(w, v)$, which by Factoring is say 8 $A(w, v)$.
 5 by factoring gives say 9 $\neg A(t, s)$.
 8 and 9 give null

B. (5 Mins) What is the most general unifier of these two wffs? w, x, y, z are (universally-quantified of course) variables, a is a constant, P is a predicate symbol, f, g are function symbols.

$P(x, x, f(y))$ and $P(g(w, z), g(a, z), z)$

Answer:

Substitutions $z/f(y), x/g(w, f(y)), w/a$ turns both into $P(g(a, f(y)), g(a, f(y)), f(y))$

5. Neural Net Theory (20 mins)

(Exercise 20.17) Suppose a neural net has linear activation functions: each unit's output is simply $g(x) = cx + d$. (Different linear functions per unit do not change the answer, however.)

A. Assume one hidden layer. For a given assignment to the weights \mathbf{W} , write down the equations for the value of the units in the output layer as a function of \mathbf{W} and the input layer values \mathbf{I} , without any explicit mention of the activation (output) of the units in the hidden layer. Show that there is a network with no hidden units that computes the same function.

Given this results, what are your intuitions or insights about n -layer linear networks?

Answer: The hidden layer outputs are:

$$H_j = g\left(\sum_k W_{k,j} I_k\right) = c \sum_k W_{k,j} I_k + d.$$

Thus the final outputs are:

$$O_j = g\left(\sum_j W_{j,i} H_j\right) = c \left(\sum_j W_{j,i} \left(c \sum_k W_{k,j} I_k + d\right)\right) + d.$$

So this IS just linear in the inputs:

$$O_j = c^2 \sum_k I_k \sum_j W_{k,j} W_{j,i} + d \left(1 + c \sum_j W_{j,i}\right),$$

and we can compute the same function as the two-layer network using just a one-layer perceptron with weights

$W_{k,i} = \sum_j W_{k,j} W_{j,i}$ and activation function $g(x) = c^2 x + d(1 + \sum_j W_{j,i})$.

All this notation can probably be collapsed into about two lines of matrix algebra involving multiplication of input vectors by weight matrices and similarly computing the outputs by multiplication and vector addition....

This reduction can clearly reduce any n layer net to an $n - 1$ layer net so by induction any linear activation function net is equivalent to some (linear-function representing) simple perceptron.

6. Probabilistic Reasoning (20)

A. (3 mins) Behind door number one: fifty \$5 bills, one bicycle worth \$1000, and 49 worthless movie posters for the airline-edited, Spanish-dubbed version of *Gigli*. Behind door number two: a \$10 bill. You get to choose a door and one randomly-picked item from behind it will be given to you. Is there a rational choice for doors and if so what and why (mathematically speaking).

Answer: Of course there's a rational approach – you just need to maximize your expected utility. If you were purely rational and had a linear utility for money, you'd expect \$12.50 in “value” from door 1 but only \$10 in value from door 2. But door 2 is a sure thing and most of the value from 1 comes from a bike, which has its own attractions or problems, and 99% of the time you get less than from door 1. So utility theory can be deployed to express your risk-taking or other priorities and come up with the best choice for you. What I would do would be to value the poster and the \$5 bills with a (negative) benefit that amounted to how hard I would kick myself for not taking the sure-thing \$10. It doesn't take much risk-aversion to decide to go for door 2.

FFQ: Idea is you get ONE shot at this deal, not an infinite set of trials. There was a range of risk preferences divided between “go for the bike” and “go for the sure thing”. I gave 1 point for the obvious justification for Door 1, 1 more point for noticing there are such things as nonlinear and risk-averse or risk-seeking utilities, but NObody got 3 points for telling me how to implement them.

B. (17 mins) You're shocked! Shocked! Your professor just uttered a Politically Incorrect Statement!! Call this event S . Of course this means he is probably Insensitive (event I) or Misinformed (event M). But just how probable is probable?

Luckily, thanks to a recent self-study of UR faculty, you know certain probabilities: $P(I), P(M), P(S | I), P(S | M), P(S | \neg I), P(S | \neg M)$ (Prior probabilities of professors being Insensitive, being Misinformed, and the conditional probabilities of them uttering PI statements given that they are misinformed, insensitive, well-informed, sensitive)

You want to know $P(I \vee M | S)$ – the probability of event (I or M). Write down this probability in terms of priors and conditionals you know. (If you need to know more probabilities, say so and assume you know them). The faculty study states that insensitivity and misinformation covary quite significantly, BUT for the purposes of this exercise you may assume (the “explaining away” simplification) that I and M are conditionally independent given S .

Answer:

FFQ: Conditional independence does NOT mean you can ignore the third term in this next equation. TOTAL independence doesn't mean it either! If it did, then you can prove that the probability of getting at least one head in two independent fair coin tosses is 1! (Exercise for the reader.)

$$P(I \vee M | S) = P(I | S) + P(M | S) - P(I, M | S),$$

where

$$P(I | S) = \frac{P(S | I)P(I)}{P(S | I)P(I) + P(S | \neg I)(1 - P(I))}$$

and

$$P(M | S) = \frac{P(S | M)P(M)}{P(S | M)P(M) + P(S | \neg M)(1 - P(M))}$$

By Bayes' Rule, and

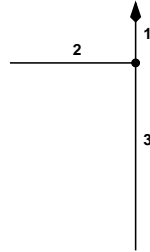
$$P(I, M | S) = P(I | S)P(M | S)$$

by the conditional independence assumption.

FFQ: People did OK on this but usually only OK. You do NOT need to know $P(S)$, you compute it, and $P(S)$ is NOT = 1 because “it happened”. $P(S)$ is the Prior probability of S .

7. Configuration Space: 10 Min.

Consider a 2-D holonomic robot, like a rigid shape cut out of a magnetized hockey puck that can move freely in 2-D but whose nose always faces north. In fact consider the specific shape below. It’s a very skinny robot with a very skinny left “arm”. Its dimensions, direction it faces (arrow pointing north), and its origin (dot) are shown. Hence (hint) its configuration is uniquely determined by the (x, y) location of its origin).

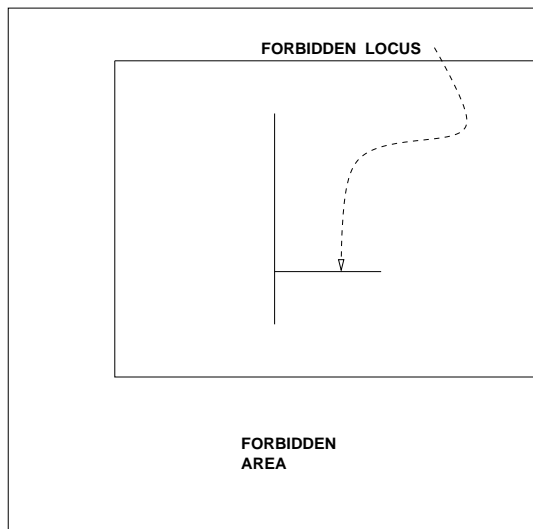


A. (5 min) This robot moves continuously in a square 10×10 unit arena oriented with its walls facing NSEW and a single point obstacle (like a very thin flagpole) located in the middle at $(x, y) = (5, 5)$. Draw its configuration space.

B. (5 min) De-magnetize the robot: it can now rotate (hint: its configurations are (x, y, θ) triples). What does the resulting configuration space look like? (Hint: words can be as good as, and easier than, a picture!).

Answer:

A: This is called the “Minkowski sum:” Get the configuration space equivalent to the problem of a point robot in a different arena by replacing every point in the obstacles by the mirror image of the robot shape.



B. Add a third dimension for θ , the robot heading, that is just a twisted version of the figure. basically pull it out of the page while twisting.