

Final

CSC 242

5 May 2000

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, total of 140). Open book, open notes.

Note to Instructor: use of book may be really bad. It seems that if people didn't find it in the book they gave up, and furthermore the book is wrong in many places. See below for e.g.s.

1. FOPC 15 Mins.

Suppose we are attempting to resolve the following well-formed logical formulae (wffs).

- (1) $\forall(a)loves(father(a), a)$
- (2) $\forall(x)\forall(y)(loves(y, x) \Rightarrow loves(x, y))$.

a.(1 min) Put the wffs into clause form; call the results (1') and (2').

Ans: Performance on this question was *Truly Terrible!* Problem may be that the book doesn't use the word "clause"! If true ignorance is as bad as this question shows, I've got to change things in a serious way!

$loves(father(a), a), \neg loves(y, x) \vee loves(x, y)$

b.(3 min) What results can you get by unifying (1') and (2')?

Ans: Either $(a/x)(father(a)/y)$ or $(x/a)(father(a)/y)$

c.(3 min) What can be generated as a result of resolving the unified clauses?

Ans: Either $loves(a, father(a))$ or $loves(x, father(x))$ depending on which unification was produced in step b. These two clauses are equivalent, they just use different variable names.

d.(8 min) What does this example show about the order in which the substitutions determined by the unification procedure must be performed? (Hint: try one in the other order).

Ans: Have to apply substitutions in same order they are generated. Else the variable substitutions won't be consistent. Here they must be applied right to left. To see why, notice that if you apply $(a/x)(father(a)/y)$ in the other order, you get $loves(x, father(a))$.

2. Minimax and $\alpha - \beta$ 15 Min.

a.(1 min) Why does search in game-playing programs proceed forward from a current position rather than backwards from a goal?

Ans: There may be lots of "goal" (winning) states, only one current state.

b.(2 min) Is minimax a depth-first or breadth-first search? (Maybe a-b search is a better question.. the text waffles).

Ans: depth first, but the book has some misleading stuff about “generate the game tree down to the lowest level”, which is contradictory and largely wrong, no?

c.(2 min) In alpha-beta pruning, the order in which you consider successor positions makes a difference. What difference, and why does it make a difference?

Ans: You can search deeper, because if you find good positions early they generate more pruning in the alpha-beta algorithm.

d.(10 min) Your game has a branching factor of b and your simple minimax program searches to a depth of d . It takes time E to evaluate a position. It takes time T to generate a move, but by spending some more time, $(T + \Delta T)$, per move you figure you should be able to generate moves in an optimal order and use $\alpha - \beta$ pruning. What is the maximum ΔT you can afford and still search to a depth $\geq d$?

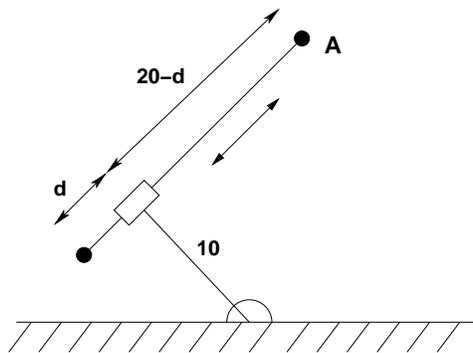
Ans:

$$\Delta T \leq \frac{(b^d - \sqrt{(b)^d})(T + E)}{\sqrt{(b)^d}}$$

Depending on the assumptions people made, they more or less came up with something like this. Most of credit comes from noticing what a-b pruning can do to your branching factor.

3. Kinematics: 15 Mins.

The two-fisted robot diagrammed below has one revolute joint at the base and one prismatic (sliding) joint at right angles to the first link. There are two active ends, shown by black disks. The lower link is 10 long. The upper links are the two parts of a continuous straight sliding bar of length 20: as one link gets longer, the other gets shorter. Assume the joints are small (in fact, points). The robot can't pass through the floor.



a. (1 min.) How many degrees of freedom does the diagrammed robot have?

Ans: Two

b. (2 min.) If there are areas of the workspace that can be reached in more than two ways, where are they (approximately) ?

Ans: None

c. (2 min) If there are areas of the workspace that can be reached in more than one way, where are they (approximately)?

Ans: There's an area in the top center of the working area that can be reached two ways...maybe vaguely triangular shaped with the tip pointing down, symmetrical around the centerline...

d. (10 min) Write the two forward kinematic equations for the (x, y) position of the active end A given the angle θ and the displacement d .

Ans:

$$x = (20 - d) \cos(\theta - 90) + 10 \cos(\theta)$$

$$y = (20 - d) \sin(\theta - 90) + 10 \sin(\theta)$$

And if you're clever you can rewrite $\sin(\theta - 90)$ as $-\cos(\theta)$, $\cos(\theta - 90)$ as $\sin(\theta)$.

4. Alarums and Excursions: 15 min.

Reassured by the evening paper that the chance of a burglary in your neighborhood is only .1 per night, you read a chapter of Borestein's *Bayesian Epistemics* and serenely drop off to sleep. You are awakened by your burglar alarm at 3:37 AM. Your razor-sharp brain immediately recalls from the manufacturer's specifications that the burglar alarm will sound with .9 probability if there is a burglar, but with probability .2 it may sound when no burglar is present. In your habitual neat, complete, precise, and self-explanatory style, you make calculations on the flyleaf of your book while you grope for your Colt Python. You rapidly calculate the probability that there actually IS a burglar in the house given the alarm....

Ans:

$$p(b | a) = \frac{p(a | b)p(b)}{p(a | b)p(b) + p(a | \neg b)p(1 - p(b))}$$

$$p(b | a) = (.9 \cdot .1) / ((.9 \cdot .1) + (.2 \cdot .9)) = 1/3$$

5. Utility Functions: 15 Min.

You recall that the utilities of action sequences are easier to analyze if the utility function is *separable* (Russell and Norvig p. 502). That is,

$$U_h([s_0, s_1, \dots, s_n]) = f(s_0, U_h([s_1, \dots, s_n])).$$

For example, to show that

$$U_h([s_0, s_1, \dots, s_n]) = \sum_{i=0}^n s_i^2$$

is separable, write $U_h([s_0, s_1, \dots, s_n]) = s_0^2 + \sum_{i=1}^n s_i^2$

Say whether the following utilities are separable. If not, say why not, and if so, give the $f(\cdot)$ that demonstrates separability.

a. (3 Min) $U_h([s_0, s_1, \dots, s_n]) = \prod_{i=0}^n s_i$

Ans: Yes, $U_h([s_0, s_1, \dots, s_n]) = s_0 * \prod_{i=1}^n s_i$

b. (3 Min) $U_h([s_0, s_1, \dots, s_n]) = [\sum_{i=0}^n s_i]^2$

Ans: Yes, you notice that (call the sum of $s_1, \dots, s_n = S$) $(s_0 + S)^2 = s_0^2 + 2s_0S + S^2$, where you know S^2 from before and you can compute the new squared sum from that, using the square root function to get S . This was the only tricky one.

c. (3 Min) $U_h([s_0, s_1, \dots, s_n]) = \text{average}(s_0, s_1, \dots, s_n)$

Ans:

Nope, don't know N to compute running average.

d.(3 Min) $U_h([s_0, s_1, \dots, s_n]) = \text{median}(s_0, s_1, \dots, s_n)$

Ans: Nope, even worse. You have to know all past individual s's and sort s0 into them.

e. (3 Min) $U_h([s_0, s_1, \dots, s_n]) = \max(s_0, s_1, \dots, s_n)$

Ans: Yes, $U_h([s_0, s_1, \dots, s_n]) = \max(s_0, \max(s_1, \dots, s_n))$

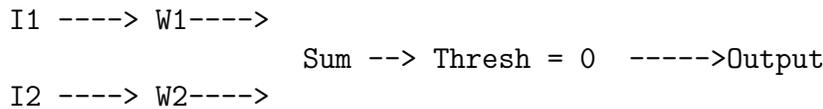
6. Perceptrons: 15 Min.

You want to train a two-input, one-output perceptron to recognize whenever input I1 is positive. The inputs and desired output levels are:

In	Output
I2:	
-1	-1 1
1	-1 1

In I1:	-1 1

Your perceptron looks like this



So if $I_1W_1 + I_2W_2 \geq 0$, output is 1 and if $I_1W_1 + I_2W_2 < 0$, output is -1.

The initial weights of the perceptron are $W_1 = .2$ and $W_2 = .8$

a.(2 min) You present the input (-1, 1): what answer does your perceptron generate and is it correct? (Show enough work to convince me you know what happens).

Ans: $.2*(-1) + .8(1) = .6$, which is > 0 , so output is 1. This is wrong, since for I1 = -1 you wanted -1 as output.

b.(5 min) Train the perceptron with this input of (-1, 1); apply the perceptron learning rule with a learning rate of $\alpha = .1$. What are the new weights?

Ans: $w1 < -.2 + .1(-1)(-1 - (1)) = oldw1 + a * I1 * Err(\text{True} - \text{Output})$ $w2 < -.8 + .1(-1)(-1 - (1)) = oldw2 + a * I2 * Err(\text{True} - \text{Output})$

So new w1 is .4, new w2 is .6

c. (8 min) If you keep training with the input (-1, 1) will your perceptron converge to the correct behavior? How can you convince yourself that these weights work for all inputs?

Ans: The book tells us that it will converge and you get some points for quoting that back to me. I wanted to see a few more iterations and several people gave me the following answer, which I think is right. (.4, .6) doesn't work but one more iteration brings the weights to (.6, .4) which DO work and learning stops. In fact These weights work for all inputs, as you can see by drawing the separating "plane" (line really) $i_2 = (-3/2)i_1$, which does the right thing: (1,-1) and (1,1) are on one side, (-1,1) and (-1,-1) are on the other.

7. Grammar: 15 Mins.

a. (10 min) Write a context free grammar for the set of sentences of the form:

1. The man died.
2. The man the knife cut died.
3. The slug the cat the boy hated bit died.

All sentences start with “The” and end with “died.”

Use nonterminal symbols for Noun, (Transitive) Verb, and Adjectival Phrase (AP). You can use single terminal symbols “the” for a noun determiner, “died” for an intransitive verb. You can assume a large list of terminals like hammer, boy, dog, coyote, tree, spider, slug, bandicoot,... for Noun, and like hit, hated, encouraged, bit, ignored, worshipped,... for Verb. (Hint: there are about 5 rules including the two described in the last sentence.)

Ans:

Answers all over the lot on this. Unsurprising since there are infinite ways to do it. Pretty much my judgement call on whether the crucial imbedding step was noticed. Several answers iterate separately (learn to spell separate, dammit!) on the nouns and verbs, which definitely isn’t English: “The man the dog the bug the cat ate died.”

S → the Noun died

S → the Noun AP died

AP → the Noun Verb

AP → the Noun AP Verb

Verb → ate, hit, etc...

Noun → man, bird, car, pen, ...

b. (5 min.) Sentences 1 and 2 are certainly grammatical, and your grammar produces them in a straightforward and simple way using standard concepts and simple context-free rules. But at some point these embedded modifiers may seem ungrammatical. How can the grammar be extended to fix this problem?

Ans: You could extend the grammar to be context sensitive, basically to count the depth of embeddings and now allow more than some fixed number. Or you could stay context free and introduce explicit nonterminals for “AP level 1”, “AP level 2” up to the number of levels you allow.

8. Ambiguity: 10 Mins.

Your speech recognizer returns this sentence during a dialog: “He thaw the tall man with the telescope just now”. Identify as many types of ambiguity as you can in this sentence, and say how a natural language system could resolve them.

Ans:

I gave full credit pretty much for 3 types of ambiguity and the general observation that lexicon and context are about all you have to go on.

Syntactic (what “with the telescope” modifies), semantic (different parsings mean different meanings). The “thaw” is perhaps a sort of ambiguity in that the word was ambiguous enough in sound to the voice recognizer that it got it wrong, so there is a sort of phonetic ambiguity going on. There’s also referential ambiguity (Who is “he?” and also who is “the man” if “he” (the subject) used a telescope to see). Also there is some pragmatic ambiguity (what does “just now” mean exactly? relative to what?). There is some vagueness in the word “tall”, which could also be a pragmatic ambiguity.

Pretty much the answer to resolving all this ambiguity is “context”, but there is a lot of technical stuff you could say about lexical information restricting certain senses in the context of other words, and there is lots of literature on word-sense disambiguation and referential ambiguity... Check out the book if I don’t get back here with more details.

9. Robot and Vision Quickies: 15 Mins.

a. (2 Min) How is the Fourier Transform useful in image filtering?

Ans: You could say a lot: high and low-pass filtering, radial filtering, thinking about image in terms of spatial energy.

b. (3 Min) You want to detect bright vertical stripes 4 pixels wide in an image. What is a filter that responds strongly locally to such image structure and not to a uniformly bright area? (hint: it can be a one-dimensional ($1 \times n$) template.)

Ans: Good: -1 -1 1 1 1 1 -1 -1 or -2 1 1 1 1 -2

Sample worse answers were: -2 4 2 and 2 2 2 2.

c. (2 Min) In two dimensions, draw two clusters A and B, and a feature vector \mathbf{x} that obviously belongs to A, for which the cluster centroid classification rule misclassifies \mathbf{x} as belonging to B.

Ans: Several ways to do this.

d. (2 Min) Why do robot arms often have six degrees of freedom?

Ans: So they can reach some working volume dextrously, with any orientation of the manipulator. NOT because the puma or the human hand does.

e. (2 Min) What is the aperture problem in optic flow computation?

Ans: It would have helped to be in class and to take notes. I think one person got this. Locally, using brightness discontinuities (like edge operators) you can only sense flow in the direction of the brightness gradient (perpendicular to the edge). This is “normal flow”. If you can track a point, you get the true flow.

f. (2 Min) In a shape from shading algorithm, what is regularization and what problem is it solving?

Ans: See above comment. The Regularization we saw adds constraint to shape-from-X solutions in the form of a smoothness criterion.

g. (2 Min) What is a Lambertian surface?

Ans: Clearly the book had this one, nicely indexed. Without that I would predict crappy performance as in parts e. and f. Maybe I’m just a cynical old crab... A flat (matte) surface; a lambertian planar surface looks equally bright from any direction. The amount of light it emits is proportional to the cosine of the angle between the emission angle and the surface normal (1 in the normal direction, 0 along the tangent direction). This means that the perceived radiance is proportional to the cosine of the angle between the light source and the surface normal.

10. Grab Bag: 10 Mins

a. (2 Min.) What is *shaping*, and how does it help in reinforcement learning?

Ans: Shaping gives rewards when the learner is somehow “headed in the right direction” toward the goal, even if the goal is not reached yet. It speeds convergence.

b. (2 Min.) What are some functions of the lexicon in a NLU system?

Ans: gives word meanings, tells what roles word can fill in sentences or in phrase or grammatical constructions, gives irregular conjugations or tense forms,...

c. (2 Min.) Knowledge of exact state utilities is not always needed to determine optimal policy. Why is this, and what learning method(s) would you use to sidestep computing accurate utilities?

Ans: The policy may deal with discrete variables (E,W, N,S) while utilities are continuous, so you might get an optimal policy with only approximate utilities. Policy iteration searches for good policies and only figures out utilities approximately as a side effect.

d. (2 Min.) What application domain does the TRIPS system use?

Ans: Moving people around from place to place, say for military or civil emergencies.

e. (2 Min.) What is an example of a nonholonomic system?

Ans: A wheeled vehicle. Again, the book has this definition but has it wrong. If I'd noticed that I could have asked a really good question, like "why is the book's definition wrong?". Oh well...