

First Midterm and Some Answers CSC 242

Write your **NAME** legibly on your bluebook(s). There are 75 minutes worth of problems. Work all problems. You may use two double-sided pages of notes. Please hand your notes in with your bluebook. **Please start each new problem on a separate page.**

1. In the Barber's Chair (10 min)

"Nose hair?"

"Nah, just trim the eyebrows."

"Ya know, I shave everybody in this burg what don't shave themselves, and they's the only ones I shave. That's four bits."

"With the haircut?"

"Sorry, that'll be a buck."

"Sign says six bits. Anyone ever say you're kinda self-contradictory?"

Express the barber's claim about shaving in first order predicate calculus, put it into clause form, and use resolution to establish that it's inconsistent. You don't need any other axioms. Use the domain of people living along with the barber in his home town, a single predicate $S(x, y)$ meaning " x shaves y ", and call the barber B .

Comment: I always wanted to start an exam with nasal hair.

Ans: people had some trouble with this but hard to say why. It doesn't get much easier!

$$\forall x[\neg S(x, x) \Leftrightarrow S(B, x)]$$

write as two \Rightarrow s, which you then remove by replacing $A \Rightarrow B$ with $\neg A \vee B$, and you wind up with two CNF clauses,

$$[S(x, x) \vee S(B, x)], [\neg S(B, x) \vee \neg S(x, x)].$$

To match you do a grand unification: substituting B for x propagates everywhere, and you get

$$[S(B, B) \vee S(B, B)], [\neg S(B, B) \vee \neg S(B, B)],$$

and the only tricky part is you really should factor to remove the repetition: $[A \vee A]$ goes to $[A]$. As it stands it looks like the two negative literals will cancel their counterparts. This idea of one literal canceling several others could actually be built into a sort of 'super-resolution' inference rule, it just traditionally is not. The process of factoring traditionally does the job, and is arguably simpler to implement. I accepted the super-resolution approach, and in the end you can get the null clause either way, so the original assertion is inconsistent. Notice that the penultimate situation is asserting that the barber both shaves himself and doesn't, which is why the classical question "Does the barber shave himself?" puts its finger on the problem (which is Russell's paradox, of course).

2. Mere Formality (10 min)

We have a domain of geographical regions and map colors, a function $MapColor(x)$ and predicates $x = y$, $In(x, y)$, $Borders(x, y)$, and $Country(x)$, all with obvious meanings. We have constants for some regions. Given the English sentences below, say whether each associated

logical expression (1) correctly expresses the English sentence, (2) is syntactically invalid and thus meaningless, (3) is syntactically valid but does not express the meaning correctly.

A. No region in South America borders any region in Europe.

- (i) $\neg[\exists c, d \text{ In}(c, \text{SouthAm}) \wedge \text{In}(d, \text{Europe}) \wedge \text{Borders}(c, d)]$.
- (ii) $\forall c, d [\text{In}(c, \text{SouthAm}) \wedge \text{In}(d, \text{Europe}) \Rightarrow \neg \text{Borders}(c, d)]$
- (iii) $\neg \forall c \text{ In}(c, \text{SouthAm}) \Rightarrow \exists d \text{ In}(d, \text{Europe}) \wedge \neg \text{Borders}(c, d)$.
- (iv) $\forall c \text{ In}(c, \text{SouthAm}) \Rightarrow \forall d \text{ In}(d, \text{Europe}) \Rightarrow \neg \text{Borders}(c, d)$.

B. No two adjacent countries have the same map color.

- (i) $\forall x, y \neg \text{Country}(x) \vee \neg \text{Country}(y) \vee \neg \text{Borders}(x, y) \vee \neg (\text{MapColor}(x) = \text{MapColor}(y))$.
- (ii) $\forall x, y \text{ Country}(x) \wedge \text{Country}(y) \wedge \text{Borders}(x, y) \wedge \neg (x = y) \Rightarrow \neg (\text{MapColor}(x) = \text{MapColor}(y))$.
- (iii) $\forall x, y \text{ Country}(x) \wedge \text{Country}(y) \wedge \text{Borders}(x, y) \wedge \neg (\text{MapColor}(x) = \text{MapColor}(y))$.
- (iv) $\forall x, y (\text{Country}(x) \wedge \text{Country}(y) \wedge \text{Borders}(x, y)) \Rightarrow \text{MapColor}(x \neq y)$.

Ans:

A. 1, 1, 3, 1 or 2

B. 1, 1, 3, 2

People OK with this. A(iv) is nasty since it's pretty obscure what the precedence of $\Rightarrow \dots \Rightarrow$ is, so some disambiguating parentheses would have been a good idea. Grade is 10 - number of wrong answers.

3. Alphabet Soup (15 min)

A. (5 min) From Max's point of view, $\alpha - \beta$ pruning is based on two simple observations. Put these underlying intuitions or justifications into one clear, specific English sentence.

Ans. maybe I should have emphasized the last 5 words of the Q. Lots of ways to describe this but clarity and brevity aren't that easy.

B. (5 min) Can you apply $\alpha - \beta$ pruning to a **three-person** zero-sum game, where the players take turns? If so how, if not why not?

Ans. You can. People didn't have much trouble here. One way to think of it is a regular minimax situation with the opposition getting two moves in a row.

C. (5 min) Consider $\alpha - \beta$ pruning in a two-person **non-zero-sum game** in which each player has a different utility function, and both utility functions are known to both players. Can pruning happen if there are no constraints on the two terminal utilities? Can pruning happen if the player's utility functions on any state differ by at most a constant (making the game almost cooperative)? Say why or why not for both questions.

Ans. In general not (2nd and 3rd ed: p. 167). An unexamined leaf node may have the best outcome for both players. So pruning can only hurt Max (or Min) and the players effectively wind up collaborating by being selfish. This same example could be true of course in the constrained utilities of the 2nd part of the question, so the constraint doesn't help pruning.

4. Satisfied Now? (10 min)

A. In backtracking search for CSP solutions, we know that the MRV heuristic chooses the next variable to assign that has the maximum number of possible legal values ... no, wait, M is actually for "minimum" I think. Anyway, having chosen a variable there's another heuristic that says its possible values should be explored in an order such that the values

chosen first are those that constrain their neighboring variables values least. Or should that be “most?”

Dang, can you straighten me out here? Which variable and which of its values do these heuristics prefer and why?

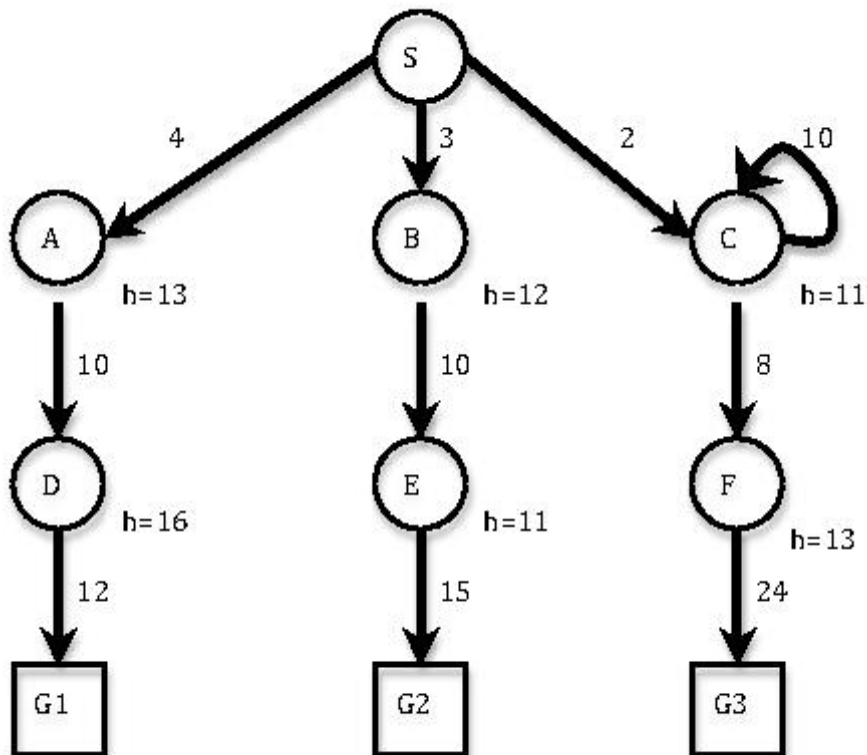
Ans. MinRV variable but choose the value that is least constraining.

B. I just found out that most published Sudoku puzzles have a unique solution. Explain (both whether and why) this is good or bad news if I’m planning to use a local search method (like hill-climbing with the min-conflicts heuristic) for my new Sudoku-solver?

Ans. If this were a ‘convex’ problem with a nice smooth ascent everywhere to the unique solution and no local minima, it’d be superb news. Somehow with Sudoku you get the feeling that can’t be true. One feels almost certainly that this is bad news – prob. lots of local minima, only one global max. Seems to be borne out by structure of actual automatic solvers, by the way.

5. Search This! (15 min)

Apply four different tree-searching algorithms to the ‘tree’ below. It’s not a tree but we’re pretending we don’t know that: you have an OPEN list but no CLOSED list. The figure shows the names of the nodes: G1, G2, and G3 are search-ending goal states, and S is the start state. The arcs are labeled with the cost of following them, and the heuristic h value (estimated cost to goal) is given for the relevant interior nodes. For each type of search, please list **in order** the nodes that are explored (taken off the OPEN list, checked for goal, successors put on OPEN list). Hints: Each search starts by putting S on the OPEN list so each answer (A-D) starts with S. Also, some searches may explore a state more than once.



- A. Greedy (best first) Search.
- B. Iterated Depth First Search.
- C. Uniform Cost Search.
- D. A* Search.
- E. Just checking. We pay A* search to find the optimal (cheapest) path to the goal. Did it work? If so, how do you know and if not why not?

Ans. These examples are hard to do right, and I didn't quite: I missed that tie in f values between C and F in the A* search, sorry.

- A. Greedy uses the h values only: SCCCC....
- B. SSABCSADBECCFSAD G1
- C. SCBAFCEDC G1 as path costs go 0 2 3 4 10 12 13 14 22 26
- D. SCBAFE G2 or SCBACFE G2 as f goes 13 15 17 23-23 tie between C and F, 24, 28.
- E. G1 is optimal goal, not G2 – people did notice D's f overestimates distance to goal, so heuristic not admissible. Now it's not tactful to be impolite about the h values. Sure they are pretty weak and don't help much, seem disconnected from reality, but you could maybe invoke the “max of current f and parent's f” trick if you wanted improvement (I don't know if it would help here, actually), but in any event, stupid, weak, unintuitive h's will never cause you to miss the optimal solution unless they are actually inadmissible, and here only one is.

Some people had heuristic search going SCCCC..., but we pay the g in $f=g+h$ to increase as we go along, so you'll eventually find something better to do than go around in a loop.

Some people sneaked BFS in there, often instead of uniform cost search.

6. Random Gibberish (15 min)

A probabilistic context-free grammar (PCFG) is a context-free grammar augmented with a probability value on each rule, so that the PCFG specifies a probability distribution over all allowable strings in the language. Specifically, the rules for each nonterminal are annotated with probabilities that sum to 1; these define how likely it is that each rule is applied to expand that nonterminal; and all such rule choices are made independently. For example, the following is a PCFG for simple verb phrases with rule probabilities on the left.

- 0.1 : $VP \rightarrow Verb$
- 0.2 : $VP \rightarrow Copula Adjective$
- 0.5 : $VP \rightarrow Verb\ the\ Noun$
- 0.2 : $VP \rightarrow VP\ Adverb$
- 0.5 : $Verb \rightarrow \mathbf{is}$
- 0.5 : $Verb \rightarrow \mathbf{shoots}$
- 0.8 : $Copula \rightarrow \mathbf{is}$
- 0.2 : $Copula \rightarrow \mathbf{seems}$
- 0.5 : $Adjective \rightarrow \mathbf{unwell}$
- 0.5 : $Adjective \rightarrow \mathbf{well}$
- 0.5 : $Adverb \rightarrow \mathbf{well}$
- 0.5 : $Adverb \rightarrow \mathbf{badly}$
- 0.6 : $Noun \rightarrow \mathbf{duck}$
- 0.4 : $Noun \rightarrow \mathbf{well}$

A. (4 min) Which of the following have a nonzero probability of being generated as complete VPs?

- (i) **shoots the duck well well well**
- (ii) **seems the well well**
- (iii) **shoots the unwell well badly**

Ans. i is OK: three adverbial VPs, a Verb the Noun and Verb. ii can't parse: seems is only a copula, so can't be followed by the. iii can't parse: unwell is only an adjective and only a noun could have followed the the.

B. (5 min) What is the probability of generating **is well well**?

Ans. Takes a little clarity of thought, **ESPECIALLY** because (see C. below) the sentence is syntactically ambiguous and has two parses. All choices in the parse need to be ANDed, but some choices lead to more than one nonterminal at once, so the answer isn't directly related to paths in the tree.

I get $(.2 \cdot .2 \cdot .8 \cdot .5 \cdot .5)$ for the five rule applications in one parse and $(.2 \cdot .2 \cdot .1 \cdot .5 \cdot .5 \cdot .5)$ for the other. or .008 and .0005, but I like those totals worse than the factors for obvious reasons.

C. (3 min) What types of ambiguity are exhibited by the phrase in part B. above?

- (i) lexical
- (ii) syntactic
- (iii) referential
- (iv) none

Ans. only syntactic (more than one parse) that I can see. Given the "part of speech", the 'same-looking' words are all determined uniquely. Lexical ambiguity is like "I went to the woods and shot a few bucks" versus "I went to Vegas and shot a few bucks." The lexical ambiguity comes not from having two shots and two bucks, but that both the shots are verbs, and both the bucks are nouns. No reference is going on: referential ambiguity is like "Tom told Fred he was cold. He got up and shut the window."

D. (3 min) Given a PCFG with no productions of the form $X \rightarrow \epsilon$ (with ϵ meaning the null string), is it possible to calculate the probability that it generates a string of exactly 10 words? If so how and if not why not?

Ans. just enumerate the trees for 10-long sentences and add up the probabilities as above.