Midterm

CSC 242

March 2006

*Write your **NAME** legibly on the bluebook. Work all problems. You may use two double-sided pages of notes. Please hand your notes in with your bluebook. The best strategy is not to spend more than the indicated time on any question (minutes = points).*

1. Propositional Calculus: 15 Min.

A (5 Min.): *"Anything follows from a contradiction."*

Formalize this statement in Propositional Calculus and prove or disprove it with a truth table.

Ans: Only hard part was the formalization of the concept. A contradiction is something and its opposite, so we want to prove $(A \wedge \sim A) \Rightarrow B$. Easy with truth table.

B (10 Min.): Given these PC statements ($\sim$ means "not"): $(A \Leftrightarrow B), (C \Rightarrow (B \vee D))$, $(\sim (D \wedge E)), (\sim (C \Rightarrow D)), (E \vee \sim F)$.

Prove A by resolution (put into clause form and derive the null clause by contradiction using only the resolution rule).

Ans: Easy.

2. Heuristic Search: 10 Min.

This moving-block puzzle has three black and three white tiles and a blank space. Object is to get all the white tiles to left of all black tiles: position of blank is not important.

B	**B**	**B**		**W**	**W**	**W**

A tile may move into adjacent empty location at cost of 1. Or it can hop over one or two other tiles into the empty position at a cost of the number of tiles hopped over.

A (5 Min.): Analyze the state space: are there loops, about how big is it, what is the branching factor, etc?

Ans. According to the rules you can move "backwards", though it may be you needn't to solve it, in which case it would be a smart idea to re-write the rules. However, as they are, the rules allow loops, so you're in a graph search. There are six opening moves, but there can be fewer later (if the hole gets to an end, say). There are a finite number of distinct states. I figure you can put the blank anywhere (7 choices), then you can distribute the 3 whites anywhere in the remaining six spots (C(6,3) or "six choose three" positions). (The whites are indistinguishable, so they can be arranged (6*5*4) / 3! ways. We have so far 7*5*4 choices. The remaining indistinguishable

blacks are forced into the remaining 3 spots and were're done with 140 different states in the graph. Another way to think about it is ALL arrangements (permutations of the 7 labels) divided by the number of indistinguishable arrangements of the white and black squares, or $7!/(3! + 3!)$.

B (5 Min.): Propose a heuristic for an A* search on this problem. Is it admissible? Is it consistent (monotonic)?

Ans. Most common were "number of whites to the right of blacks" (teleport move simplification) and "distance of whites to their final destinations (a simplification of no jumps and no opponents that dominates teleport). Both are monotonic and thus admissible.

3. FOPC: 10 Min.

A (5 Min.): Translate this English sentence into FOPC.

"Everybody regrets something his father did."

Ans. Lots of choices, but this would work:

$$\forall x \exists y (Person(x) \wedge Act(y) \wedge Did(Father(x), y) \wedge Regrets(x, y))$$

B (5 Min.): Now translate your FOPC for the sentence into clause form.

Ans. The above would yield the And of these clauses, with Skolem function A that produces the act done by x's Father.

$$Person(x), Act(A(x)), Did(Father(x), A(x)), Regrets(x, A(x))$$

4: Minimax 10 Min.

Consider a three-person zero-sum game of perfect information.

A (5 Min.): Can the minimax algorithm be modified to apply to this situation? If so how, and if not why not?

B (5 Min.): Will $\alpha - \beta$ pruning work in this situation? Why or why not?

Ans. Good way to think of this is Max vs. Min1 and Min2. As turns go round the table, Max can think of the two Mins as one team who is out to get him – the worst case (his Min outcome) is that they collude against Max. So he can just treat them like a powerful opponent who makes two moves at a time, if you like. So it's really just minimax, and also the motivation and mechanism for $\alpha - \beta$ pruning are intact and work fine.

5. Unification: 10 Min.

You are given the clauses:

$$P(y, f(y), C, g(u, v, A), h(B))$$

$$P(x, z, w, g(h(w), t, t), x)$$

with variables t, u, v, w, x, y, z; constants A, B, and C; functions f, g, h and predicate P.

A (5 Min.): If it exists, find the most general unifying substitution and show the result of unifying the clauses.

B (5 Min.): How would your answer to **A** change if the clauses were

$$P(y, f(y), C, g(u, v, A), h(B))$$

$$P(y, z, w, g(h(w), u, u), v)$$

Ans. A: $P(h(B), f(h(B)), C, g(h(C)), A, A, h(B)$.

B: There was a misprint in that I wanted the last v to be a y. Rats. Oh well, as it is, there actually IS a difference and the answer is

$P(y, f(y), C, g(h(C)), A, A, h(B)$.

6. Un-Natural Language: 20 Min.

Regular expressions *regexp*s are basic to editors (vi, Emacs) and pattern-matching programs and utilities like Perl, `grep, awk, sed`, etc. Let's consider a small typical subset of a regexp language.

A regexp (sometimes called a pattern) is a string of characters from the set: the lower-case letters *a,b,c,d*, the symbols (,),|, *, and ?, and the operator D. We think of the regexp as matching a string. Strings of letters are patterns that match themselves. More complex patterns are made as follows: Matching parentheses group subpatterns into a pattern. The ? and the * operate on the previous pattern and mean "repeat zero or one time" and "repeat zero or more times", respectively. | means "(exclusive) or": $(x \mid y \mid z)$ matches just one of x or y or z. The operator D means "match any non-null string of (numerical, base 10) digits". So $(D(ab)?cd*)? \mid fubar$ matches the null string, as well as *fubar, 367c, 3567abc, 999999cdddd,* (but not *c, 55dddd, or 112fubar*)

We know a "regular grammar," which generates or parses regular expressions, has only one sort of rule: each rule has a single non-terminal on the LHS and a terminal symbol, optionally followed by a non-terminal, on the RHS.

Write a regular grammar that will generate (or recognize) all the strings that match our example regexp above, *viz.* $(D(ab)?cd*)? \mid fubar$.

Ans.

I dunno... maybe something like – this is almost certainly suboptimal. Actually it helps to draw the FSM and write the grammar from that.

```
S -> Terminal
S -> fubar  (this is shorthand for six productions leading to Terminal)
S -> 0T      (do the D operator.. need this style for ‘‘fubar’’ too)
S -> 1T
...
S -> 9T

T -> 0T
T -> 1T
...
T -> 9T    (pesky having to do this twice but I don't see a way out)

T -> aU    (decide to do ab)
T -> cW    (or do c immediately)
U -> bV    (finish ab)
V -> cW    (now do c)
W -> dW    (emit d*)
W -> Terminal
```