Logic Midterm, CSC173, 28 Oct. 2008 (with FFQ)

Please write your name on the bluebook. This is a closed-book exam. There are 75 possible points (one per minute). Stay cool and write neatly. Answers are on the Web site.

Start of FFQ Feature

The whole purpose of the Karnaugh map is to make smaller circuits by eliminating variables that are unused (perhaps in sub-circuits). Our example is a function that is always true, so All the variables can be eliminated: the map says a 0-gate solution is possible, and that's what you made it and paid it for, and then you ignore it and give me 3-variable-input circuits that produce T for all inputs! Yikes! This particular idea (KMs) is all about engineering: it's easy to make a correct circuit, the idea is to make a correct small circuit!

The most common mistake was probably in 4A: unify
A. ?- f(h(X),Y) = f(g(Z),Z).

This will NOT unify since the function h(.) can never unify with a different function g(.).

3A. There are at least two ways to go wrong in clausifying your conclusion before negating. There is a potential basic syntactic problem with negating a clause: you can turn it into a non-clause. If clausification comes last, you always have clauses, by definition. If you clausify first, you can get forced backwards by producing non- clauses (non CNF). Eg: Start with $S = A \wedge B$, a fine litte prop. calc. sentence that's already in conjunctive normal form. So clausifying S doesn't change it: $C(S) = S$ in our notation. But then negating S leaves a result that's not in CNF: $\neg(A \wedge B)$. So you'd have to RE-clausify, which would get you the right answer but proves that doing these operations in the wrong order won't work formally.

There's a more subtle way to get in trouble: nothing is formally wrong but the result is too weak to be useful so the completeness of the resolution inference system is ruined: The problem is with existential quantifiers ... Skolemization gets you. e.g.
$\exists(x)(F(x))$
Wrong: clausify first (skolemize), then negate: So $F(Bob)$ (Bob is Skolem constant) negates to $\neg F(Bob)$ which is true but a very weak specialized clause that won't unify with, say, $F(Fred)$ if $Fred$ is another constant.

Correct: negate, then clausify:
Negation is $\forall(x)(\neg(F(x))$, which clausifies to $\neg F(x)$ which is more general and will unify with more clauses, like $F(Fred)$.

In 3B I actually should have said instead of "a person" (possibly himself) instead of "another person" so we didn't have to formalize that a person couldn't love himself. Which only one person did, so this wasn't a big hangup.

Worse was that a lot of youse took my "not both" syntax to mean "both not". This wound you up with an $(L \vee K)$ clause instead of an $(L \wedge K)$ clause (or CLAUSE B and CLAUSE C) below. This means you couldn't prove anything. I gave full credit for a correct attempt, but I must say: reading skills today, tsk tsk tsk... According to what I meant, then, My version of 3B is: Given:

$$\forall(x)\exists(y)(L(x,y) \Rightarrow A(y))$$

Clausification; This Skolemizes via Skolem function $l(x)$ to:

$$L(x, l(x)) \Rightarrow A(l(x))$$

and finally clausifies to
CLAUSE A:

$$\neg(L(x, l(x)) \vee A(l(x))$$

Also given:

$$\neg\exists(v)[\neg(L(T, x) \wedge K(T, v, Z))]$$

Clausification: negating the exists and DeMorganing...

$$\forall(v)\neg[\neg L(T, v) \vee \neg K(T, v, Z)]$$

pushing thru the outer negation by DeMorgan again leaves two clauses,
CLAUSE B:

$$L(T, v)$$

and CLAUSE C:

$$K(T, v, Z).$$

We want to know who is amiable, so we assert there isn't anyone amiable and prove our resulting set of clauses inconsistent.

Negate the assertion; The negation of the assertion that there is anyone amiable – $\exists(w)(A(w))$ – is the universally-quantified
CLAUSE D:

$$\neg A(w).$$

So now we prove stuff : resolving CLAUSE A with CLAUSE B yields thru the magic of unification

$$A(l(T))$$

which resolves with CLAUSE D to give the null clause with the substitution of $l(T)$ for v. Thus $l(T)$, the unnamed person whom Tim loves, is amiable. Actually CLAUSE B says Tim loves everyone, so that would actually imply everyone is amiable, eh? But it doesn't seem we can prove that thru resolution (unless I'm missing something!).

End of FFQ Feature

1 Propositional Calculus, Karnaugh Maps, Circuits (15 min)

A. (5 min) Make a truth table for this PC formula:

$\neg(P \vee Q) \Rightarrow (P \Rightarrow R)$

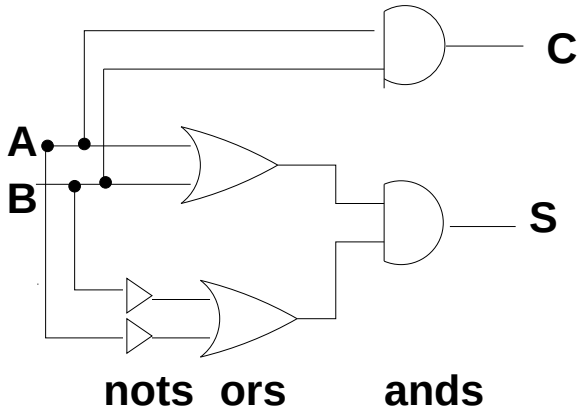
(Be very careful here ... this has to be right or nothing below works!)

B. (5 min) Make a Karnaugh map for this "function" (that is, for the column under that middle connecting $\Rightarrow$.)

C. (5 min) Convert the Karnaugh map into a simple circuit.

2 PC Inference (15 min)

I claim the following circuit (gate functions labelled below) takes binary inputs A and B and computes C the carry bit and S the sum bit in the binary sum of A and B. Prove it (hint: using truth tables).



3 FOPC Inference (25 min)

A. (5 min): We're warned to negate the conclusion *before* putting it into clause form, not *after*. That must mean that negation ($\neg$) does not commute with the operation $C(\cdot)$ of conversion into clause form. That is, for some FOPC sentence S, it must be true that $\neg(C(S)) \neq C(\neg S)$. Give such a sentence S and demonstrate the results of doing negation and clause-conversion in the "correct" and "wrong" order.

B. (20 min): Put the following assertions into FOPC formulae, convert to clause form, and use resolution to answer the question "Who (if anyone) is amiable?". Make sure you show what converts to what, what resolves to what, etc. so the process is clear, not just the answer.

Use the predicate notation $L(x, y), A(x), K(x, y, z)$ for "x Loves y, x is Amiable, and x, y, z all Know each other". Use constants T for Tim and Z for Zelda, variables u, v, w, x, y, z as necessary.

In the following statements, the labels P and Q should not be part of your FOPC axiomatization, (but you may want to use variables to formalize their meaning). They are being used to try to keep the English clear (thus making my prose sound like a math class: sigh.)

A. *For every person P there is another person Q such that if P loves Q then Q is amiable.*
B. *There is no person P such that:*
it is false that both
a) Tim loves P
b) Tim, P, and Zelda all know each other.

4 Unification (10 min)

What is the most general unifier of the following clauses? That is, what does Prolog print given the following input lines?

```
A. ?- f(h(X),Y) = f(g(Z),Z).

B. ?- f(X,Y) = f(Z,Z).

C. ?-  p(X, g(a, f(Y)), Y) = p(h(Z,b),Z,g(w,b)).
```

5 Pro and Con: (10 min)

As feature editor at GoogleGeekBlog.com, you're left high and dry when your star programming-language critic suddenly departs to join Queue Luu Breeze's posse.

So now you've got ten minutes to write both sides of a "Prolog: Pro and Con" sidebar (maybe 300 words): Throbbing with gratitude for that superb CSC173 course and its kindly old professor (what *was* his name?), you race for the deadline:

A. Pro: Prolog Better Than Sex!

B. Con: Prolog Worse Than Sucks!