

Logic Midterm, CSC173, Oct. 27, 2009

Please write your name on the bluebook. This is a closed-book exam. There are 75 regular and 5 extra-credit points. Stay cool and write neatly.

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

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

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

(Hint: get this right!)

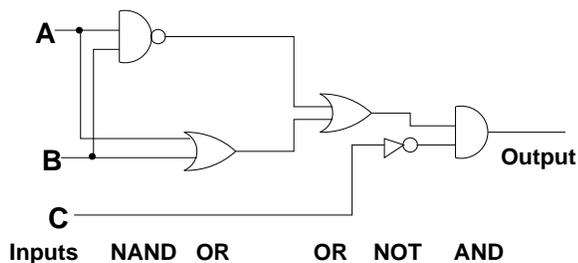
B. (5 min) Make a Karnaugh map for the resulting Boolean function (the column under the second \Rightarrow .)

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

[FNFQ: almost everyone did perfect, and those that didn't were almost perfect].

2 My Excellent Circuit (15 min)

Whew, I worked all night designing this circuit. Before applying for a provisional patent, I thought I'd run it by you. Can you please use a truth table to figure out what Boolean function is being computed? In the unlikely event you think of any practical improvements, tell me and we can share the profits.



[FNFQ: almost everyone did perfect.]

3 FOPC Inference (20 min)

A. (20 min): Put the following assertions into FOPC formulae, convert to clause form, and use resolution to answer the question (state an appropriate conclusion, negate it, resolve...). 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 $S(x, t)$, $U(x, t)$, $L(s, t)$ for “ x is sad at time t ”, “ x has ulcers at time t ”, and “time s is later than time t ”, respectively.

Let's assume a “typed” calculus, with people $(x, y, z\dots)$ and times (s, t, u, \dots) : so no need to assert that “ t is a time” for instance.

Given:

A. *Someone is sad sometime.*

B. *If anyone's ever sad, he'll get ulcers later.*

Question:

C. *Who gets ulcers and when?*

CB's Ans:

FOPC:

A. $\exists x \exists t S(x, t)$

B. $\forall x \forall t \exists s [S(x, t) \Rightarrow (U(x, s) \wedge L(s, t))]$

Now because of the way the English reads, many people wanted to do this (note the dot):

B. $\exists x \exists t. (S(x, t)) \Rightarrow \exists s (U(x, s) \wedge L(s, t)).$

This works too, since arrow removal means negating the double \exists on the LHS of the \Rightarrow gives us a negated double \forall , so we wind up the same as in the first formalization. In fact, I think you can freely choose \forall or \exists for the outer two variables and always get to the same place. People got mixed up with scope (which IS confusing in this latter formalization... hence the dot in my formalization above). You still wind up needing both a Skolem *function* and some Skolem *constants* (for statement A.) Last, I should have been more careful with S for sad, s for a time, and then of course some people added an S for the skolem function. Very confusing.

C. $\exists x \exists t U(x, t)$ Claim someone gets ulcers sometime

Here many people over specified this claim with the AND of some other predicates like the time being after some other time, etc. That misses the point that you want the question answering system to tell you the time.

Clauses:

A. $S(\text{Bob}, \text{Monday})$ Bob and Monday skolem constants

B1. $\neg S(x, t) \vee U(x, T(x, t))$, T a skolem function

B2. $\neg S(x, t) \vee L(T(x, t), t)$

negate C: $\neg U(x, t)$.

Resolutions:

A. and B1 give D: $U(\text{Bob}, T(\text{Bob}, \text{Monday}))$

A and B2 give E: $L(T(\text{Bob}, \text{Monday}), \text{Monday})$

negated C and D give \square

So we conclude that Bob got ulcers and the time he did was T, a particular time later than Monday.

4 Unification (10 min)

What is the most general unifier of the following clauses, written with prolog conventions (upper-case variables, lower-case predicates and functions).

That is, what does Prolog print given the following input lines (order of output substitutions is irrelevant).

A. $f(g(j(Z,b,c))) = f(h(j(a,b,c)))$.

B. $g(X,h(a,Y)) = g(Y, h(a, k(l,m)))$.

C. $f(X,g(Y,W,U),h(X,Z)) = f(Z,g(d,V,Y),h(b,W))$.

Ans:

A. false.

B. $X = k(l, m)$, $Y = k(l, m)$.

C. $X = b$, $Y = d$, $W = b$, $U = d$, $Z = b$, $V = b$.

5 Symbology meets Prolog (15 min)

You're applying to Robert Langdon's Harvard lab as a researcher. He's on the phone (something about Rosicrucians and Kazhakstani plutonium), but hands you an ancient (1970's, by the typeface) manuscript and waves you off, flashing his right hand open thrice. You've clearly got 15 minutes to decipher this code and tell him what the incantation prints out — otherwise you might have to take that Rochester job.

```
favefun(L,Ans) :- funhelp(L,0,1,Ans).
funhelp([],_,X,X).
```

```
funhelp([H|T],S,P,N) :- S1 is S + 1, P1 is P * S1,
    write(H), write(' '),write(T), write(' '), write(S1), write(' '), write(P1), nl,
    funhelp(T,S1,P1,N).
```

```
?- favefun([brown, prose, sucks],BigTime).
```

Ans:

[FNFQ: Almost everyone got this perfect: I'm impressed (or suspicious).]

```
?- favefun([brown, prose, sucks],BigTime).
```

```
brown [prose, sucks] 1 1
```

```
prose [sucks] 2 2
```

```
sucks [] 3 6
```

```
BigTime = 6.
```

6 Cut and Run ! for Extra Credit (5 min worth)

In Prolog, what is the *cut* (!) operation? Describe one of its good points and one of its bads points?