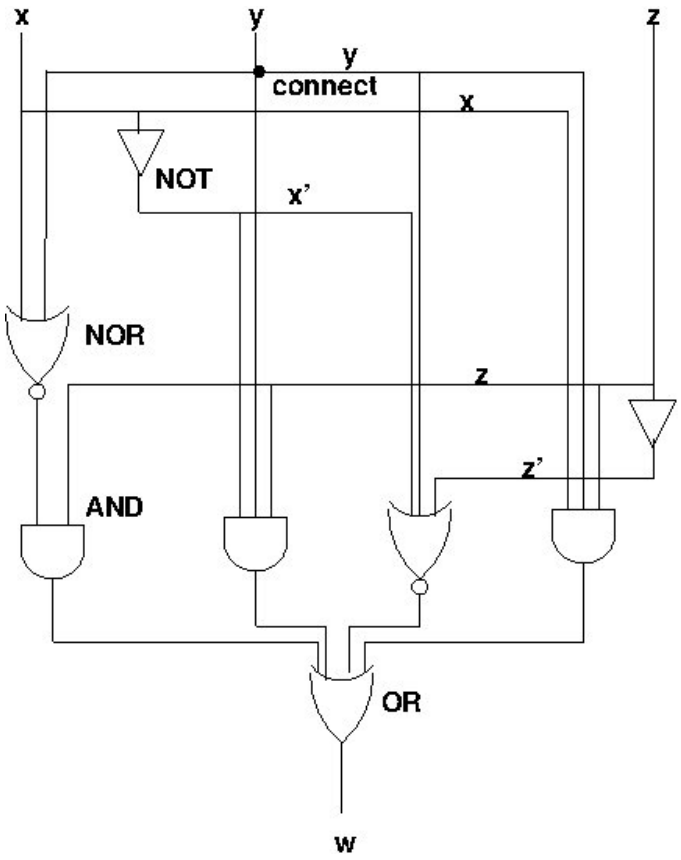
173 Logic and Prolog 2013
With Answers and FFQ

Please write your name on the bluebook. You may have two sides of handwritten notes. There are 75 points (minutes) total. Stay cool and please write neatly.

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



A. (5 min) Write down a propositional logic expression for the function computed by the circuit. Your expression should:
1. Only use $\wedge, \vee, \neg, (,)$. Or, use "sum of product" notation, e.g. $x + \bar{y}z$ for $x \vee (\neg y \wedge z)$, and
2. Be as literally related to the circuit as possible. So for instance the lowest OR gate generates an expression with four sub-expressions, connected by three $\vee$s.

B. (5 min) Anyhow you like (truth table, inspection of circuit or PC expression...), figure out the Boolean function implemented by the circuit (w as function of x, y, z). Express your function as an eight-row, four-column (i.e. [x y z w]) table.

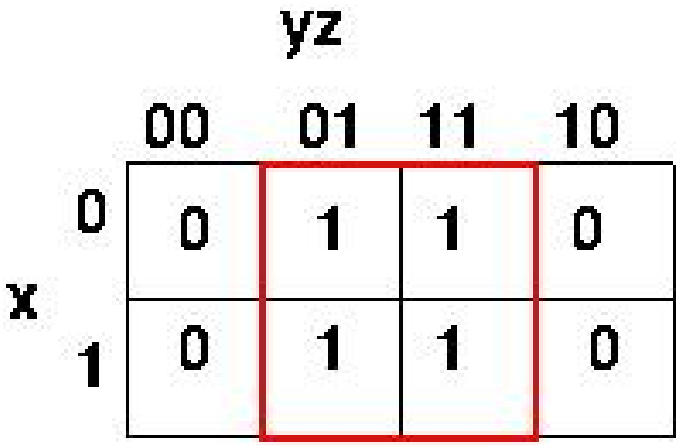
C. (5 min) Make a Karnaugh map for your Boolean function, and draw or describe the resulting minimized circuit.

Ans. A. $(\neg(x \vee y) \wedge z) \vee (\neg x \wedge y \wedge z) \vee \neg(\neg x \vee y \vee \neg z) \vee (x \wedge y \wedge z)$ Bars over multiple letters can yield a formula equivalent to the the circuit, but I can't easily do that in LaTeX; the "stacked bar" notation simplifies to what turn out to be the four minterms of the function: $\bar{x}\bar{y}z + \bar{x}yz + x\bar{y}z + xyz$.

B.

x	y	z	w
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

C.

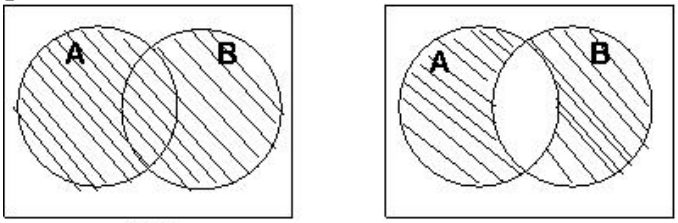


Minimal circuit is the zero-gate wire from z to w.

2. Models (10 min)

A. (5 min). Let's use model theory to prove $(A \wedge B)$ is not the same as $(A \Rightarrow B)$. Sure, they're different *by definition*, but what fun is that? Instead, prove it with Venn diagrams.

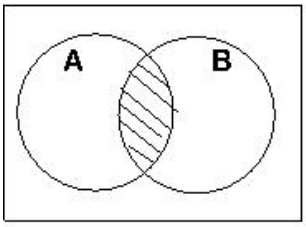
Use two circles labeled A and B for each expression and shade the area that represents the models of the expressions: if they are different, you're done. For example, here are the diagrams for $(A \vee B)$ and (A XOR B), XOR being 'exclusive OR': clearly they're different, so OR and XOR are not equivalent.



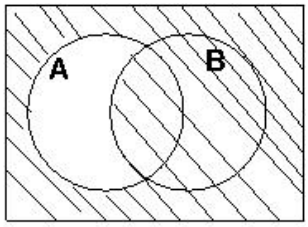
B. (5 min). Not using Venn diagrams, but arguing from the definition of entailment and the four-row, three-column boolean function table of $(P \vee Q)$ as a function of P, Q, show that $P \models (P \vee Q)$.

Ans.

A:



AND



IMPLIES

B:

row	P	Q	(P V Q)
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	1

We want to show the models of P are a subset of the models of $P \vee Q$, or that variable assignments (rows) that make P true are a subset of those that make $P \vee Q$ true, or that the set of rows $\{3, 4\}$ is a subset of rows $\{2, 3, 4\}$, which is true. We're done.

3. English to FOPC (15 min)

Choose the FOPC expressions that best expresses the English sentence below **Any number, from all to none**, of the answers may seem OK to you. List those that do. Optionally (partial credit?), say why you don't like the others. Basic point score is:
$\lceil 10 \cdot \max(0, \frac{r-w}{n}) \rceil$, with r right answers, w wrong answers, and n possible right answers.

The domain is people. Here, $I(w)$ is a predicate meaning "I know w", $Y(w)$ means "you know w", $C(u, v)$ means "u is more conservative than v". $[\cdots]$ is the same as $(\cdots)$.

FFQ Comment: Some of these are really subtle! The answers I like are pretty much agreed to by many students, but in any case they show that the semantics of everyday english is sometimes not that easy to nail down or is often just ambiguous or does not deal with all the cases.

A. (10 min.): *Everybody I know is more conservative than anybody you know.*

1. $\forall x[I(x) \Rightarrow (\forall y(Y(y) \Rightarrow C(x, y)))]$
2. $\forall x \forall y(I(x) \wedge Y(y) \wedge C(x, y))$
3. $\neg(\exists x \exists y(I(x) \wedge Y(y) \wedge C(y, x)))$
4. $\forall y \exists x([I(x) \wedge Y(y)] \Rightarrow C(x, y))$
5. $\forall x \forall y[(I(x) \wedge Y(y)) \Rightarrow C(x, y)]$

Ans: 1 yes, 2 no (I don't know everyone), 3 yes, 4 no I may not know anyone, 5 seems ok. There is thruout a kind of semantic ambiguity in orig. English: suppose we both know the same person? Some careful axiomatizations and maybe more detailed rules needed to get out of this...

B. (5 min.) Add a function $p(w)$ that returns a parent of w, and formalize **both** obvious readings of the ambiguous statement *I know someone whose child is more conservative than his grandparent*.

Ans:
$\exists x(I(p(x)) \wedge C[x, p(p(x))])$
$\exists x(I(p(x)) \wedge C[x, p(p(p(x)))])$

FFQ ALERT! Many people wanted to use two variables. Maybe $I(p(x))$ didn't occur to them. That led to something like $I(x) \wedge (x = p(y))$, which isn't terrible but we don't actually have = in our suite of predicates, and = is a bit funny anyway in FOPC-land, AND we don't need it. Also $p(x)$ was sometimes confused with (or changed to) another predicate we don't have, $P(x)$. which presumably would be true if x is a parent.

4. CNF and Resolution (10 min)

A. (5 min.) Put this FOPC expression into Conjunctive Normal Form.
$\forall x \forall y \exists z(P(z) \Rightarrow \neg[Q(f(z)) \vee R(x, y)])$.

Ans:
$(\neg P(s(x, y)) \vee \neg Q(f(s(x, y))))$
$(\neg P(s(x, y)) \vee \neg R(x, y))$
$s(x, y)$ a Skolem function.

FFQ ALERT! A very depressing number of people don't know what CNF is, much less how to convert either PL or FOPC sentences into CNF. What can I say? Basic skill, on all previous tests. Maybe I softpedaled it in class? Needed for next question, too.

FFQ: about 2/3 of people got to some equivalent of $\neg P \vee (Q \wedge R)$, with or without the ()s, and stopped. That's not CNF, is it? Distribute!

FFQ: Even worse was the Skolemization issue. Most people didn't seem to get the idea at all. If they sort of did, there was still the issue that our z is universally quantified-over twice. So for every value of x and y, or for every (x,y) pair, the function needs to produce a z that can depend on the values of both x and y, so the skolem function looks like s(x,y).

B. (5 min.) Put these PC expressions into CNF and use resolution to prove S by deriving the null clause.

Axioms:
$P \Rightarrow (Q \wedge (R \vee S))$
$Q \Rightarrow \neg(T \vee R)$
P
To Prove:
S

Ans.
See **FFQ Alert!** on CNF above!

Axiom clauses:
1. $\neg P \vee Q$
2. $\neg P \vee R \vee S$

3. $\neg Q \vee \neg T$
4. $\neg Q \vee \neg R$
5. P

negation of conclusion

6. $\neg S$

Proof:

7. $1, 5 \rightarrow Q$
8. $2, 5 \rightarrow (R \vee S)$
9. $4, 7 \rightarrow \neg R$
10. $8, 9 \rightarrow S$
11. $10, 6 \rightarrow \square$

FFQ Alert! Resolution is only guaranteed to work on CNF clauses. If you don't put the axioms and conclusion into CNF, you do crazy things like say $\neg S$ and $S \wedge T$ resolve to T...wrong! As we said several times, to do resolution inference, put everything into CNF. If you can't put PC sentences into CNF, you're stuck.

5. Short Answers (15 min).

A. True or False: If a proof system is sound, it is complete, but not the reverse.

B. True or False: P true and Q true satisfies $P \Rightarrow Q$...

C. True or False: ...thus $P \Rightarrow Q$ is valid.

D. Here, like Prolog, constants and functions are lower case and variables are upper case. "substitute c for W" is denoted $W \setminus c$.
True or false: A unifier of
$f(S, X, g(a, Y), Y)$ and
$f(W, Z, g(W, U), b)$ is
$\{Y \setminus b, \ U \setminus b, \ W \setminus a, \ S \setminus a, \ X \setminus c, \ Z \setminus c)\}$.

E. If the last answer was False, say so and go to F. If it was True, then are the above substitutions a most general unifier? If not, what IS a most general unifier for the clauses?

F. Fill in the blank: Truth tables are not a scalable way to prove or falsify an arbitrary propositional logic sentence because they _____.

G. True or False: Resolution theorem proving will prove any set of FOPC clauses consistent or inconsistent in some finite time.

H. True or False: One can construct a NAND function from AND, OR, and NOT functions and *vice versa*.

Ans:
FFQ Alert! A: learn definitions of sound and complete!
G: Inference with Resolution is only semi-decidable: Only guaranteed to complete, and then only in finite but possibly very long time, if clauses are INconsistent.

A Half true, so False. B true ; C false (not satisfied for all domain models); D True; E putting individual c in for Z, X not most general. $Z \setminus X$ would make it a MGU; F. have exponential cost; G false, it's only semi-decidable. H true: we're supposed to recall that NAND is a universal function like NOR, and that AND, OR, NOT are also a universal set, so with either we can generate any Boolean function.

6. Psychoanalyzing Prolog (10 min.)

Here are a fact and rules defining `dust`, a set operation (not its real name). Sets are represented as lists with no repeated members.

```
dust([],X,X).
dust([X|R],Y, Z)  :- member(X,Y), !, dust(R,Y,Z).
dust([X|R],Y, [X|Z])  :- dust(R,Y,Z).
```

A: (5 min) What is this set operation's usual name?

B. (5 min): Consider `dustnocut`,

```
dustnocut([],X,X).
dustnocut([X|R],Y, Z)  :- member(X,Y), dustnocut(R,Y,Z).
dustnocut([X|R],Y, [X|Z])  :- dustnocut(R,Y,Z).
```

Which is the same as `dust` except for the *Cut* goal in 2nd rule.

What does prolog respond to the following queries?

```
?- dustnocut([a], [b],L).
?- dustnocut([a,b], [a],L).
```

Answers:

A.1 Set Union

FFQ Alert! The Q. is "what SET operation?". Set ops are member, union, intersection, ... not list ops like append, nor some op like search. This whole section is easier if you recognize set union. Describing back what the code does is not really what I was after, and limits your thought leverage. Only a little better is describing a list operation. In fact I gave partial credit since union looks a bit like `append` – in fact enough alike that append gives some correct-for-wrong-reason answers to part B!

B.

```
?- dustnocut([a],[b],L).
L = [a, b].

?- dustnocut([a,b],[a],L).
L = [b, a] ;
L = [a, b, a].
```

It's continuing to union even though `a` is a member of 2nd list, since it's retrying the now-cutless rule.