Activity 2.1 Propositional Logic
CS 536, Wed Jan 11, 2012

A. Why?
• Reviewing/overviewing logic is necessary because we'll be using it in the course.

B. Objectives
At the end of this activity you should:
• Be able to follow proofs of propositional formulas.
• Be able to generate proofs of simple propositional formulas, given a set of rules.

C. Questions
1. Fill in the missing rule names in the proof below of $\neg(p \leftrightarrow q) \Leftrightarrow (\neg q \wedge p) \vee (p \wedge \neg q)$, using the rules from the lecture notes.

$\neg(p \leftrightarrow q)$	
$\Leftrightarrow \neg((p \rightarrow q) \wedge (q \rightarrow p))$	by defn $\leftrightarrow$
$\Leftrightarrow \neg(p \rightarrow q) \vee \neg(q \rightarrow p)$	________
$\Leftrightarrow (p \wedge \neg q) \vee (q \wedge \neg p)$	________
$\Leftrightarrow (q \wedge \neg p) \vee (p \wedge \neg q)$	________

2. Write a formal proof that shows that $(p \rightarrow p \vee q) \Leftrightarrow T$ (sometimes called the "$\vee$ introduction" rule) is a tautology.

$p \rightarrow p \vee q$	
$\Leftrightarrow$ ________	by ________

etc

D. Solutions
1. DeMorgan's law; Negation of $\rightarrow$ twice; commutativity of $\vee$.

2. $(p \rightarrow p \vee q)$

$\Leftrightarrow \neg p \vee (p \vee q)$	defn $\rightarrow$
$\Leftrightarrow (\neg p \vee p) \vee q$	commutativity of $\vee$
$\Leftrightarrow T \vee q$	excluded middle
$\Leftrightarrow T$	domination

Activity 2.2 Predicate Logic

A. Why?

- We'll be using predicates to write specifications for programs.

B. Outcomes

At the end of this activity you should

- Be able to read and write predicates.
- Be able to logically negate predicates.
- Be able to translate informal descriptions of properties on integers and arrays into formal predicates and predicate functions.

C. Questions

1. What do we get if we add the redundant parentheses back to $(\forall x \,.\, \forall y \,.\, \exists z \,.\, x \neq y \rightarrow x \leq z \wedge z \leq y \vee x > z \wedge z \geq y)$? (Don't bother putting parentheses around individual variables.
2. What is the minimal parenthesization for $(\forall x \,.\, (\exists y \,.\, x > y) \wedge (\exists y \,.\, x < y))$?
3. In general, if $\forall x.\forall y.\, p(x, y)$ is true, is $\forall y.\forall x \,.\, p(x, y)$ true? What about $\exists x \,.\, \exists y \,.\, p(x, y)$ and $\exists y \,.\, \exists x \,.\, p(x, y)$?
4. Find a predicate equivalent to $\neg(\forall x \,.\, \exists y \,.\, p(x,y))$ that doesn't have $\neg$ in front of any quantifier. (It's okay for the $\neg$ to appear inside the body of a quantified predicate.) Hint: Use DeMorgan's laws to take and move the logical negation inward.
5. Repeat the question above on $\neg(\exists y \,.\, \forall x \,.\, p(x,y))$.
6. Write the definition of a predicate function *Repeats*(b, m) that is true exactly when $b[0], b[1], ..., b[m-1]$ and $b[m], b[m+1], ..., b[2\,m-1]$ are equal (pointwise). E.g., if b contains 1, 3, 5, 1, 3, 5 then *Repeats*$(b, 3)$ is true.

D. Solutions

1. $(\forall x \,.\, (\forall y \,.\, (\exists z \,.\, ((x \neq y) \rightarrow (((x \leq z) \wedge (z \leq y)) \vee ((x > z) \wedge (z \geq y)))))))$
2. $\forall x \,.\, (\exists y \,.\, x > y) \wedge \exists y \,.\, x < y$
3. 3 Yes, and yes
4. $\neg(\forall x \,.\, \exists y \,.\, p(x,y))$
   iff $\exists\, x \,.\, \neg\exists\, y \,.\, p(x, y)$ DeMorgan's Law
   iff $\exists\, x \,.\, \forall\, y \,.\, \neg p(x, y)$ DeMorgan's Law

To continue, we'd need to know the structure of $p(x, y)$. E.g., if $p(x, y) \equiv x < y \rightarrow y < z \wedge f(x) = 2$, then

$\neg p(x, y)$
iff $\neg(x < y \rightarrow y < z \wedge f(x) = 2)$ Defn of p
iff $x < y \wedge \neg(y < z \wedge f(x) = 2)$ Negation of $\rightarrow$
iff $x < y \wedge (\neg(y < z) \vee \neg(f(x) = 2))$ DeMorgan's Law
iff $x < y \wedge (y \geq z \vee f(x) \neq 2)$ Negation of comparison, 3 times

5. $\neg(\exists y . \forall x . p(x,y))$
iff $\forall y . \neg(\forall x . p(x,y))$ DeMorgan's Law
iff $\forall y . \exists x . \neg p(x,y)$ DeMorgan's Law

6. $Repeats(b, m) \equiv 2m \leq \text{sizeof}(b) \wedge \forall j . 0 \leq j < m \rightarrow b[j] = b[m+j]$

