COT 3100 — Recitation 3 — Logic makes the world go 'round

1. Below are the truth tables for the operators NAND, NOR, and XOR. Write definitions for these operators using only AND, OR, and NOT. Try to explain in words what each one means.

A	B	A NAND B
T	T	F
T	F	T
F	T	T
F	F	T

A	B	A NOR B
T	T	F
T	F	F
F	T	F
F	F	T

A	B	A XOR B
T	T	F
T	F	T
F	T	T
F	F	F

2. The NAND operator, described above, is a *sole sufficient operator*, which means that all the other operators can be rewritten using only NAND. This is an important result in chip design, for example, because it can make chips cheaper to build if they use all NAND gates. How would you translate "not P" using only combinations of the NAND operator?

3. Use the laws of logic to prove the following. Remember, for each line of your proof, you should include the rule and lines that support it.

$$\begin{array}{l} P \wedge \neg Q \\ R \vee S \\ S \rightarrow Q \\ \hline \therefore R \vee U \end{array}$$

Translate the following three stories into formal proofs. For each one, decide what the primitive statements should be, then describe the situation in logical symbols by writing a system of propositions using those primitives. *Next*, pick one story and use a truth table to determine its validity. *Finally*, pick a different story, and determine its validity by applying the laws of logic to its proposition system.

4. If I don't practice regularly, I won't make it onto *American Idol*. But I am practicing regularly, so I will make it.

5. Heads I win, tails you lose. We cannot both win and we will not both lose. Therefore I will certainly win.

6. Distributed database D has two client processes, P_1 and P_2. In order to complete transaction T in this database, a process must have an *exclusive lock* on each of the two data rows R_1 and R_2. If one process has an exclusive lock on anything, then the other process cannot use it. P_1 and P_2 both need to complete transaction T. Process P_1 starts by taking an exclusive lock on R_1. At the same time, process P_2 gets an exclusive lock on R_2. Now neither P_1 nor P_2 can complete the transaction.

Finished early? Why not translate AND and OR using only NAND...