CS70, Fall 2003 **Discussion #1** **Amir Kamil**
UC Berkeley **8/29/03**

Topics: Administrivia, Course Overview, Bad Proofs, Knights and Knaves, Quantifiers

1 Administrivia

1. About me:
   - Amir Kamil
   - 4th year EECS major
   - Email: `kamil@eecs.berkeley.edu`
   - Webpage: `http://www.cs.berkeley.edu/~kamil`
   - Be warned that I have a twin brother who is a CS major (and a CS186 TA), so if you get an unexpected reaction when you ask one of us for help, it's probably him.
2. Course:
   - You can and should work together on homeworks with up to two other people, but your writeup must be entirely your own.
   - Homeworks must be turned in at the homework box in 283 Soda before class on Thursdays.
3. Section:
   - Notes will be available on my webpage, probably as PDFs. It may take a week for them to appear.

2 Course Overview

CS70 is similar to Math 55, but is focused on topics more applicable to CS. To be honest, the course is much more difficult than Math 55, though grading will take into account the added difficulty. Here is a list of topics we plan on covering:

1. Propositional Logic and Proofs
2. Induction
3. Protocol Design and Analysis
4. Number Theory
5. Combinatorics
6. Probability Theory
7. Computation Theory

3 Bad Proofs

Consider the following proof:

Claim: $2 = 1$.
Proof:

1. Pick non-negative integers a and b such that $a = b$.
2. $a = b$
3. $a^2 = ab$
4. $a^2 - b^2 = ab - b^2$
5. $(a + b)(a - b) = (a - b)b$
6. $a + b = b$
7. $2b = b$
8. $2 = 1$

Is this proof correct? Obviously not, since the conclusion is absurd. So where did the error occur? A careful decomposition of the reasoning behind the proof reveals that between steps 5 and 6, the author of the proof used the justification $x = y \implies x/z = y/z$. However, this is only true if $z \neq 0$, which is not the case here. This shows why each step in a proof must be justified rigorously.

4 Knights and Knaves

Recall the knights and knaves problems from lecture. Knights always tell the truth, and knaves always lie. Suppose Alice says "At least one of us is a knight," and Bob says "At least one of us is a knight." Can we determince if Alice is a knight or a knave?

The first thing we have to do is define propositions. So let's define

$$P = \text{"Alice is a knight"},$$
$$Q = \text{"Bob is a knight"}.$$

Now we must state our givens in terms of the propositions. The statement "at least one of us is a knight" can be expressed as $P \vee Q$. So the givens are

$$P \implies P \vee Q, \text{ i.e. Alice is a knight implies at least one of them is a knight,}$$
$$Q \implies P \vee Q, \text{ i.e. Bob is a knight implies at least one of them is a knight,}$$
$$\neg P \implies \neg(P \vee Q), \text{ i.e. Alice is a knave implies neither of them is a knight,}$$
$$\neg Q \implies \neg(P \vee Q), \text{ i.e. Bob is a knave implies neither of them is a knight.}$$

Let's attempt a proof by enumeration. The truth table for the above propositions is in table 1. Looking up the rows in which all givens are satisfied, we see that either both Alice and Bob are knights, are both are knaves. But it is impossible to determine which case it is with the information have.

What if Alice says "At least one of us is a knight," and Bob says "We're both the same type." Using the same propositions P and Q, we know have

$$P \implies P \vee Q, \text{ i.e. Alice is a knight implies at least one of them is a knight,}$$
$$Q \implies (P \wedge Q) \vee (\neg P \wedge \neg Q), \text{ i.e. Bob is a knight implies both are knights or both are knaves,}$$
$$\neg P \implies \neg(P \vee Q), \text{ i.e. Alice is a knave implies neither of them is a knight,}$$
$$\neg Q \implies \neg((P \wedge Q) \vee (\neg P \wedge \neg Q)), \text{ i.e. Bob is a knave implies they can't both be knights or knaves.}$$

P	Q	$P \lor Q$	$P \implies P \lor Q$	$\neg P \implies \neg(P \lor Q)$	$Q \implies P \lor Q$	$\neg Q \implies \neg(P \lor Q)$
$False$	$False$	$False$	$True$	$True$	$True$	$True$
$False$	$True$	$True$	$True$	$False$	$True$	$True$
$True$	$False$	$True$	$True$	$True$	$True$	$False$
$True$	$True$	$True$	$True$	$True$	$True$	$True$

Table 1: The truth table for the first knights and knaves problem.

P	Q	$P \lor Q$	$P \land Q$	$\neg P \land \neg Q$	$(P \land Q) \lor (\neg P \land \neg Q)$	$P \implies P \lor Q$	$Q \implies (P \land Q) \lor (\neg P \land \neg Q)$	$\neg P \implies \neg(P \lor Q)$	$\neg Q \implies \neg((P \land Q) \lor (\neg P \land \neg Q))$
$False$	$False$	$False$	$False$	$True$	$True$	$True$	$True$	$True$	$False$
$False$	$True$	$True$	$False$	$False$	$False$	$True$	$False$	$False$	$True$
$True$	$False$	$True$	$False$	$False$	$False$	$True$	$True$	$True$	$True$
$True$	$True$	$True$	$True$	$False$	$True$	$True$	$True$	$True$	$True$

Table 2: The truth table for the second knights and knaves problem.

Again, we do a proof by enumeration. The truth table for the above propositions is in table 2. Looking up the rows in which all givens are satisified, we see that Alice must be a knight, but Bob can be either a knight or a knave.

It is also possible to do the above two problems using logical equivalences and inference rules, but since you don't have to know most of the required equivalences, we will not do so here.

5 Quantifiers

Given a proposition $P(x)$, suppose we want to say something like "There are exactly three distinct integers x for which $P(x)$ is true," or "there are less than two distinct integer x for which $P(x)$ is false." We can do so using just quantifiers.

Let's start with the statement "There are at least three distinct integers x for which $P(x)$ is true." With quantifiers, the equivalent statement is $\exists a \in \mathbf{Z}.\ \exists b \in \mathbf{Z}.\ \exists c \in \mathbf{Z}.\ a \neq b \land a \neq c \land b \neq c \land P(a) \land P(b) \land P(c)$. The first three clauses in the conjunction are needed to require there to be three distinct integeres, and the last three require the proposition to be satisfied by each of them.

Now in order to require that there not be more than three distinct integers that satisfy $P(x)$, we must specify that any other integer does not satisfy it. This is equivalent to saying that if an integer does satisfy $P(n)$, it must be either a, b, or c. Thus our statement becomes $\exists a \in \mathbf{Z}.\ \exists b \in \mathbf{Z}.\ \exists c \in \mathbf{Z}.\ a \neq b \land a \neq c \land b \neq c \land P(a) \land P(b) \land P(c) \land (\forall d \in \mathbf{Z}.\ P(d) \implies d = a \lor d = b \lor d = c)$.

This technique can easily be generalized to values other than 3. So what we have done is essentially defined relational operators with quantifiers. Since we defined the $\geq$ and $=$ operators, we can compose any other operator from them. For example, the $\leq$ operator can be defined as the disjunction of $\neg \geq$ and $=$.

Of course, in order to do so, we'll need to negate a quantified statement. Negating quantifiers is simple: flip the quantifier, and move the negation to the right of the quantifier. For example, the negation of the statement $\forall x.\ P(x)$, $\neg\forall x.\ P(x)$, is $\exists x.\ \neg P(x)$. Similarly, the negation of the statement $\exists x.\ P(x)$, $\neg\exists x.\ P(x)$, is $\forall x.\ \neg P(x)$. Intuitively, this is because the statement "$P(x)$ is not true for all x" is equivalent to "there is an x for which $P(x)$ is false." And the statement "$P(x)$ is not true for any x" is equivalent to "$P(x)$ is false for all x." In general, it is preferable to only have negations to the right of a quantifier.