Math Logic 6021, week # 1

Assignment for Wednesday: 1. Read Chapter 0

2. Come to class with questions about whatever you don't understand. (Please make your questions as specific as you can.)

Mathematical Induction:

¿Why am I going to be so pedantic about induction?

1. Logic treads very close to inconsistencies, and one must be careful to (try to) avoid them.
2. Think of an application in CS: Formal methods are used to find subtle errors, and for that we need precision.

A basic tool: Proof by mathematical induction (on the natural numbers, called $\mathbb{N}$.)

"Ordinary" induction: Common first example: Show that

$$0 + 1 + 2 + \cdots + n \;=\; \sum_{i=0}^{i=n} i \;=\; \frac{n(n+1)}{2}.$$

It's clear from context that it covers all natural numbers, n: 0, 1, 2, ...

Proof:

Base case, $n = 0$**:** For $n = 0$, the left-hand-side is 0, and the right-hand side is $\frac{0(0+10}{2} = 0$, so the equation is satisfied.

Inductive Case:

Assume that $\sum_{i=0}^{i=n} i = \frac{n(n+1)}{2}$,

and prove that $\sum_{i=0}^{i=n+1} i = \frac{(n+1)(n+2)}{2}$

$$\begin{aligned}\sum_{i=0}^{i=n+1} i &= (\sum_{i=0}^{i=n} i) + (n+1) \\ &= \frac{n(n+1)}{2} + (n+1) \\ &= \frac{n(n+1}{2} + \frac{2(n+1)}{2} \\ &= \frac{(n+1)(n+2)}{2}\end{aligned}$$

So the theorem follows by the **Principle of Mathematical Induction.**

Principle of Mathematical Induction ("Original form")

If a statement S is true of 0, and if whenever it's true for a natural number n it's also true for $n+1$, then S is true of all natural numbers.[1]

[1]We could elaborate the statement. Say we want to prove "*for all natural numbers* $n \geq$ (say) *13 S(n)" Now the base case is* $n = 13$. *For the inductive case we assume that* $S(n)$ *holds* and *that* $n \geq 13$ *and then prove that* $S(n+1)$ *holds. I don't state it that way here simply because the next form of induction gets complicated to state with that addition. But I can prove this elaborated form from the other.* Define $S'(n)$ *to be* $S(n+13)$. *Now prove by ordinary induction that* $S'(n)$ *holds. Notice that I'm proving exactly my elaboration above; all I'm doing is changing what number I call "n".*

Arguing carefully from explicit assumptions: Giuseppe Peano (1858 - 1932) realized that a lot of arithemtical our arithmetical knowledge of natural numbers could be inferred from simple definitions plus induction.

Start out with the natural numbers:

- We have 0,
- and we can apply function S ("successor" function) to any natural number to get another natural number.

Underlying assumptions here:

- S is a *function*: for each object x, there is exactly one object $S(x)$.
- There are no other natural numbers.

Special names: $1 = S(0), \quad 2 = S(1) = S(S(0)), \ldots$.

Now define, for natural numbers x, y,

$$x < S(y) \;\longleftrightarrow\; (x < y \vee x = y)$$
$$x + 0 \;=\; x \qquad \text{and} \qquad x + S(y) \;=\; S(x + y).$$

And prove some familiar "facts":

Addition is associative: For all natural numbers x, y, z,

$$(x + y) + z \ = \ x + (y + z).$$

Proof by induction — but **¿induction on which one number?**

I choose z.

But I also carefull choose what I prove by induction:

Prove, by induction on z, that for all x, y, $(x + y) + z = x + (y + z)$.

Base step, $z = 0$: Show that $(x + y) + 0 \ = \ x + (y + 0)$.

By the recursive definition of $+$,

- $(x + y) + 0 \ = \ x + y$, and
- $y + 0 \ = \ y$, so, by substitution $x + (y + 0) \ = \ x + y$ — which equals $(x + y) + 0$.

Inductive step:

Inductive Assumption: *for all u and v*

$$(u + v) + z = u + (v + z)^{2}$$

And prove that: *for all x and y*

$$(x + y) + S(z) = x + (y + S(z)).$$

$$\begin{array}{lll} (x+y)+S(z) & & \\ = & S((x+y)+z) & \text{by recursive definition of } + \\ = & S(x+(y+z)) & \text{by inductive hypo. and defn. of } < \\ = & x+S(y+z) & \text{by recursive definition of } + \\ = & x+(y+S(z)) & \text{by recursive definition of } + \end{array}$$

[2] I switched to u and v to avoid double use of variables x, y — the two different "occurrences" of x, y were "quantified by" two different phrases "for all x, y." The important point below is that I am not limiting the u, v to any particular x, y I might be discussing.

Problem to try: Prove, *by induction*, that, for all natural numbers n,

$$0 \cdot 2^0 + 1 \cdot 2^1 + 2 \cdot 2^2 + \cdots + n \cdot 2^n = (n-1)2^{n+1} + 2.$$

Homework problem A: Prove, *by induction*, that, for all natural numbers n,

$$n^5 - 5n^3 + 4n \text{ is divisble by } 120.$$

A bad example: Fibonacci numbers: $f_0 = 0$, $f_1 = 1$, and, for $n \geq 2$, let $f_n = f_{n-1} + f_{n-2}$.

Prove that f_n **is** $\begin{cases} \text{even} & \text{if } n \text{ is divisible by } 3 \\ \text{odd} & \text{otherwise.} \end{cases}$

Base case, $n = 0$**:** 0 is divisible by 3, $f_0 = 0$, and 0 is even, as desired.

Inductive Step: *Assume the result for* n*, and then prove it for* $n+1$.

Case I, $n = 0$**:** Then $n + 1 = 1$, 1 is not divisible by 3, and $f_1 = 1$ (by definition) is odd, as desired.

Case II, $n = 1$**:** Then $n + 1 = 2$, 2 is not divisible by 4, and $f_2 = f_0 + f_1 = 1$ which is odd, as desired.

Case III, $n > 1$**,** n **divisible by 3:** So $n + 1$ is not divisible by 3. By inductive hypothesis, f_n is even and f_{n-1} is odd, so $f_{n+1} = f_n + f_{n-1}$ is odd, as desired.

Case IV: $n > 1$**,** $n \bmod 3 = 2$**:** So $n + 1 \bmod 3 = 0$.
Since n is not divisible by 3, by inductive hypothesis, f_n is odd.
And $(n-1) \bmod 3 = 1$, so f_{n-1} is odd.
And thus $f_n + f_{n-1}$ is even, as desired.

Case V: $n > 1$**,** $n \bmod 3 = 1$**:** Analogous.

Math Logic, CS 6021, week # 1, Woden's Day

****Important**:** Turn in your claims **now** on problem A

From last time:

1. ¿Any questions on proviing that

$$0 \cdot 2^0 + 1 \cdot 2^1 + 2 \cdot 2^2 + \cdots + n \cdot 2^n = (n-1)2^{n+1} + 2 \quad ?$$

2. I'll draft someone to prove, by induction, that for all natural numbers n,

$$n^5 - 5n^3 + 4n \text{ is divisble by } 120.$$

3. **Reminder:** Fibinacci numbers.

   Origin: Early mathematical biology. Nicolo of Pisa, filius Bonacci, was trying to predict the population of a colony of rabbits.

   Number f_n refers to the number of *pairs* of rabbits at time n.

   Definition: $f_0 = 0$, $f_1 = 1$, and, for $n \geq 2$, let $f_n = f_{n-1} + f_{n-2}$.

   Oversimplifications include:

   - Rabbits never die.
   - Each birth consists of 1 male & 1 female.

Time	# Pairs	**Explanation**
$n = 0$	$f_0 = 0$	You have no rabbits.
$n = 1$	$f_1 = 1$	You buy one pair of babies (kits
$n = 2$	$f_2 = 2$	Your original kits have matured & are regular to bear more
$n = 3$	$f_3 = 2$	Original pair still alive and has had 1 pair of kits.
$n = 4$	$f_4 = 3$	Original pair has had another set of kits.
$n = k+2$	$f_{k_2} = f_{k+1} + f_k$	f_{k+1} still alive from time $k+1$ and f_k of those were old enough to have more kits.

Theorem:

$$f_i \text{ is } \begin{cases} \text{even} & \text{if } n \text{ is divisible by } 3 \\ \text{odd} & \text{otherwise.} \end{cases}$$

"Proof":

Base case, $i = 0$: 0 is divisible by 3, $f_0 = 0$, and 0 is even, as desired.

Inductive Step: *<u>Assume</u> the result for $i = n$, and then <u>prove</u> it for $i = n + 1$.*

Case I, $n = 0$: Then $n + 1 = 1$, 1 is not divisible by 3, and $f_1 = 1$ (by definition) is odd, as desired.

Case II, $n = 1$: Then $n + 1 = 2$, 2 is not divisible by 4, and $f_2 = f_0 + f_1 = 1$ which is odd, as desired.

Case III, $n > 1$, n divisible by 3: So $n + 1$ is not divisible by 3. By inductive hypothesis, f_n is even and f_{n-1} is odd, so $f_{n+1} = f_n + f_{n-1}$ is odd, as desired.

Case IV: $n > 1$, $n \bmod 3 = 2$: So $n + 1 \bmod 3 = 0$.
Since n is not divisible by 3, by inductive hypothesis, f_n is odd.
And $(n - 1) \bmod 3 = 1$, so f_{n-1} is odd.
And thus $f_n + f_{n-1}$ is even, as desired.

Case V: $n > 1$, $n \bmod 3 = 1$: Analogous.

- **¿What's wrong with that "proof"?**

- **¿What can we do to fix it?**

I asked you to reach Chapter 0.

¿ Are there any questions about that material ?

There are various equivalent principles to the principle of mathematical induction. In proofs, I frequently use:

- **The principle of well-ordering of the natural numbers:**

  Every non-empty set of natural numbers has a least element.

  How can we use that to prove theorem result about Fibonacci numbers?

Reading Assignment: **§§1.0-1.2** And come to class with questions.

Warnings regarding Table II:

1. There are many possible choices of boolean operators.

   C's designers chose `!`, `&&`, `||`, `<`, `<=`, `==`, `>`, `>=`, `!=`

   Enderton chose $\neg$, $\wedge$, $\vee$, $\rightarrow$, and $\leftrightarrow$

   Other authors make other choices — but we'll stick with Enderton's.

2. The remarks give *approximate* translations into English.

   But one can argue *all the translations* are approximate. Besides, English can be ambiguous.

   Translations into C are easier, but — for example

   There is even a subtle *difference between C's "`&&`" and Enderton's "$\wedge$."*

Problem Assignment: **§1.1 ## 2, 3**.

We *might* get to these on Friday (Frea's Day): that depends upon how fast we go through other material. You may continue to "claim" each problem (pass me notes at the start of class) until somebody presents it in class (or I give you a solution)

Frea's Day:

I am emailing to you a note on induction: (1) a sample of some English that I think is clear, and (2) two new homework problems, B and C.

Finish proof of Exercise A.

Enderton gives syntax for formulas of sentential (propositional) logic. On page 17 he shows how a formula corresponds to its "ancestral tree." This tree is basically the same as the parse trees used in formal language theory and compiler theory.

1. Other authors use slightly different syntax. The important point is that we could write a parser program that could reconstruct the tree from the formula.

   (Traditional computer languages are far more generous in their syntax — *e.g.*, they allow expressions such as $x + 2 * y - 3 - z/5$ — making parsing more difficult.)

   Skim §1.3's material on the parsing algorithm — that's material you've probably seen, or will see, in other classes. If you haven't seen it, read about Polish notation.

2. *Informally*, we leave out many of the parentheses — see Enderton's discussion of that (also in §1.3). But please *don't leave out more parentheses* than Enderton suggests.

   Also, *informally*, we use other symbols than A_i as sentence symbols.

   Note 2 synonyms that I'm likely to use:

Enderton's term	**Synonym**
sentence symbol	*proposition letter*
sentential logic	*propositional logic*

Questions on §§1.0-1.2?

Reminder: What do the following mean?

- A truth assignment ν *satisfies* a formula φ?
- A formula φ is *satisfiable*?
- A formula φ is a *tautology*?
- A set Σ of formulas *tautologically implies* a formula φ — written — in the context of sentential logic — $\Sigma \models \varphi$?

Also skim §**1.6,** *Switching circuits*. It's not part of this class, but I recommend you read that section for your background knowledge and to help you relate this course to others.

Read: §§**1.4-1.5** and prepare questions as needed.

Suggestion: If you find the discussion of induction and recursion in §1.5 difficult to read, try to think of 2 additional examples and trace them as examples as well. (If, as you trace, you find they weren't good examples, that has also helped you understand the discussion.)

Homework Problems: §**1.4 #2,** §**1.5 ## 2, 4, 7, 9**