We are currently getting ready to prove the Gödel Completeness Theorem.The form of the Completeness Theorem that we prove is due to Leon Henkin. He wrote an interesting article:

The discovery of my completeness proofs

in the Bulletin of Symbolic Logic 2 (1996) that describes how he found his proof.

Read:

1. *Enderton*, Section 2.4 and 2.5.
2. *Notes on the Completeness Theorem*

These are notes to accompany Enderton and the class lectures on the Gödel Completeness Theorem and its consequences.

Do the following exercises from: *Enderton*. **p 129-130: 2,4, 6. Due Monday, January 28.**

Non-standard Models Exercises: 0.1, 0.5, 0.6, 0.10 **due Monday, February 18** If you can, do Exercise 0.1 in two different ways: one using the result of Exercise 0.6, and one directly, without using Exercise 0.6. (Note that the languages for the two exercises are not the same.)

Solutions to homework 2 *Solutions to Homework 2*

Exercises: 0.7, 0.8, 0.2, 0.13 **due Monday, February 25**

Due March 3:

1. Exercise 5.10 from *Notes on the Completeness Theorem* (also Enderton, Exercise 4, p. 163)
2. Exercise 4, page 202 from Enderton.

Read *Enderton*, Section 2.6.

We are skipping *Enderton*, Section 2.7 and 2.8.

Read *Enderton* Section 3.1.

Read *Algorithms and Computability* pp 21-34.
And keep reading to keep up with lectures pp24-∞.

Due March 12:

From *Algorithms and Computability*: Exercises 6.14, 6.15,6.16,6.17.

Caution: A Turing Machine is simply a special kind of finite, ordered, list of quadruples. It doesn't *know* anything. So in doing Exercise 6.16 you cannot use reasoning of the form: *No such Turing Machine exists because whenever it finds a 1 it wouldn't know if there is still another 1 further to the left. So it would never halt.*

Hour Exam: Friday, March 28 You should know all the definitions of important concepts, the statements of all the theorems (As examples: the Extended Completeness Theorem; Compactness Theorem; statements and results about elimination of quantifiers, completeness of a theory; definition of a Turing Machine; definition of a Turing Computable number-theoretic function, definition of a primitive recursive function, etc., etc., etc., and be able to do some fairly simple things.)

due Monday, March 31: From *Algorithms and Computability*: Exercises 7.7 (not collected),7.10(not collected), 7.11, 8.5,8.7,8.8(a).

due Monday, April 14 starting with page 41:
9.11(not collected),9.12,10.8,2.10(p49),2.12,3.8,4.6 *For 10.8 on page 44 it is not necessary to erase each time. So the machine can just print to the right* b0b1b10b11b101b110...

due Wednesday, April 23: 7.10 (p62), 1.10 p67) just for f_1, 2.6(Selection),2.7

Homework Solutions for 4-23-08 *Solutions*

due Wednesday, May 30 From the one page handout: For each of the four pairs $\mathfrak{A}$, $\mathfrak{B}$ of structures:

(1) Find the largest m such that $\mathfrak{A} \equiv_m \mathfrak{B}$. (In doing this you don't have to *prove* that this is the largest m, but you should show that $\mathfrak{A} \not\equiv_{m+1} \mathfrak{B}$.
(2) Find the largest m such that Duplicator has a winning strategy for the game $G_m(\mathfrak{A}, \mathfrak{B})$. Again, you don't need to *prove* that you have the right m, but it should *be* the right one!
(3) Make a wonderful conjecture based on your results from (1) and (2).

Definition 0.1. Let A and B be subsets of $\mathbb{N}$. $A \leq_m B$ (A is *many-one reducible to* B) if there is a recursive function f such that for all

$x \in \mathbb{N}$,

$$x \in A \iff f(x) \in B\ .$$

We showed in class

Proposition 0.2. *Suppose $A \leq_m B$. Then:*

(1) *B r.e. $\implies$ A r.e.;*
(2) *B recursive $\implies$ A recursive.*

Definition 0.3. Let A and B be subsets of $\mathbb{N}$.

$$A \equiv_m B \iff A \leq_m B \ \&\ B \leq_m A\ .$$

Then it is easy to see that $\equiv_m$ is an equivalence relation. Since A r.e. and $A \equiv_m B$ implies B is r.e., the equivalence classes give a partition of the r.e. subsets of $\mathbb{N}$.

Exercise 0.4. Show that there is a subset A of $\mathbb{N}$ such that

(1) A is r.e., and
(2) $B \leq_m A$ for every r.e. subset B of $\mathbb{N}$.

We say that A is a *complete* r.e. set.

Hint. It might not be a bad idea to think about the Enumeration Theorem (Proposition 2.17 on page 76 of *Algorithms and Computability*).

Exercise 0.5. How many different equivalence classes of recursive sets are there? (Give reasons!)

Hint. You might perhaps think about Exercise 0.4, and Exercise 1.38 (page 20 of *Algorithms and Computability*).

Exercises 0.4 and 0.5 are **due ?**

Start Reading about the theory CnA_E.
The following are **Notes on Second-Order Logic and Games**.
Games In Finite Model Theory