Plan for the rest of semester:

Today: Halting problem

Friday: Equivalence of languages gen by unrestricted grammars are TA and every TA lang has a unrestricted grammar.

Next week: review

Halting problem: We have a way of encoding TM, and corresponding to that, encoding the input to our TM (we discussed this for the universal TM.) We have the language

\[ L = \{ \text{w} \in \{0,1\}^* \text{ where } w \text{ is an encoding of a TM } M \text{ followed by an encoding of input } I \text{ to that TM where } M \text{ halts on input } I \} \]
Thm: $L$ is not TD.

Note: $L$ is TA/TE - just run it is accepted by the universal Turing machine (modified to first check if we have a valid encoding, and also modified so that every halting state is an accept state).

Short but uninformative proof:

Suppose $L$ were TD for contradiction. Then we will prove that every TA/TE language is TD. How?

If we have a TA lang. $X$, there is a TM $M$ that accepts $X$. We construct a TM $\hat{M}$ that decides $X$, $\hat{M}$ first feeds $M$ and $w$ ($w$ is our candidate for being in $X$) to the TM that decides $L$. 
If this TM says \( M \) doesn't halt on \( w \),
\( \hat{M} \) should say no.
If this TM says \( M \) does halt on \( w \),
\( \hat{M} \) should run \( M \) and return
\( M \)'s answer.

No matter what, \( \hat{M} \) halts, and it always
gives the right answer.
So \( \hat{M} \) decides \( X \), so \( X \) is TD.

We know that not every TA lang is
tD, so our assumption that \( L \) is TD
must be false.
Longer, more informative proof:

Assume for contradiction that \( L \) is TD. Then there is a TM \( H \) that decides \( L \).

**Method:** We will construct from \( H \) a TM that has paradoxical behavior. There will be a string which it should both accept and reject (should neither reject nor accept).

0) From \( H \), construct \( H' \) that will halt only in one of two states - one final state \( q_f \) and one non-final state \( q_n \).

1) From \( H' \) construct \( H'' \) so that,
   a) When \( H' \) enters \( q_f \), \( H'' \) goes into an infinite loop. (add transitions \((q_f, \text{anything}) \rightarrow (q_f, \text{same}, R)\))
   b) The behavior on entering \( q_n \) is untouched.
\( H'' \) does the following:
- its input is a TM \( M \), followed by input to \( M \) (called \( w \)), all suitably encoded.
- if \( M \) halts on \( w \), then \( H'' \) runs forever.
- if \( M \) doesn't halt on \( w \), then \( H'' \) halts. (in \( q_n \))

2) Construct a TM \( \hat{H} \) that first copies its input, and (so if \( \hat{H} \) starts with input \( w \), \( \hat{H} \) puts \( \text{WOW} \) on the tape), and moves the head all the way back to the beginning, then runs \( H'' \).

\( \hat{H} \) is a TM. So we can encode \( \hat{H} \) as a string and feed it as input to \( \hat{H} \).
What happens?

$
\hat{A}
$
on any input should either run forever, or halt in state $q_n$.

Suppose $\hat{A}$ runs forever, given the string for $\hat{A}$ as input. Then $\hat{A}$ actually does the following:

It duplicates the input to $\hat{A} \hat{O} \hat{A}$ (as strings) and feeds answer to $H''$. Since $\hat{A}$ doesn't halt on the input $\hat{A}$ (as a string), $H''$ halts in $q_n$.

Suppose $\hat{A}$ halts given $\hat{A}$ as input, then $\hat{A}$ feeds $\hat{A} \hat{O} \hat{A}$ to $H''$, and runs forever.

This is nonsense, so $\hat{A}$ can't exist, so $H''$ and $H' \prime$ and $H$ can't exist.