Agenda

• **Yesterday**
  - Two undecidability proofs
• **Today**
  - Reductions (Section 5.3)
• **Tomorrow**
  - Section 6.3 & part of 6.4
Announcements

• Remember to let me know if you plan to come to my house tomorrow evening
Reductions and decidability

• To prove a language is decidable, we have converted it to another language and used the decidability of that language
  - Example - use decidability of $E_{DFA}$ to determine decidability of $NOINT_{DFA}$
Reductions and undecidability

• To prove a language is undecidable, we have assumed it’s decidable and found a contradiction
  - Example - assume decidability of $\text{HALT}_{TM}$ and show $A_{TM}$ is decidable which is a contradiction

• In each case, we have to do a computation to convert one problem to another problem
  - What kind of computations can we do?
TM’s and computation

• TM’s can do more than just accept and reject strings
  - They can perform functions

**Definition:** A function $f : \sum^* \rightarrow \sum^*$ is a computable function if there is some TM $M$ that, on every input $w$, halts with $f(w)$ on the tape
Examples

• The copying TM discussed several weeks ago
  - Start with $w$ on the tape, halt with $ww$ on the tape

• Finding intersection of two DFA’s
  - Start with $<A,B>$ on the tape, where $A$ and $B$ are DFA’s, halt with $<C>$ on the tape, where $L(C) = L(A) \cap L(B)$
Mapping reducibility

**Definition:** Language $A$ is mapping reducible to language $B$, written $A \leq_m B$, if there is a computable function $f : \Sigma^* \rightarrow \Sigma^*$, where for every $w$,

$$w \in A \text{ iff } f(w) \in B$$