6.15

- Theorem: For any language A, there exists a language B, s.t. $A \leq_T B$ and $B \nleq_T A$
- First, notice $A \leq_T A$, so finding B s.t. $B \nleq_T A$ is the interesting part
- $B \nleq_T A$ means B is harder than A: even with an oracle for A, we can't decide B
- Special case: suppose A is decidable, then any undecidable language works, say A_{TM}
- Generalize: T_A = { M : M is a Turing machine with access to an A-oracle }
- A^A_{TM} = { <M, w> : M $\in T_A$ and M accepts w }
- Show $A \leq_T A^A_{TM}$ (easy: w $\in$ A iff <N,w> $\in A^A_{TM}$, where N is a TM for A)
- and $A^A_{TM} \nleq_T A$ (by diagonalization, just like we showed A_{TM} undecidable)

A^{A}_{TM} is Undecidable

- Theorem: A^{A}_{TM} is undecidable (relative to A).
- Recall: A^{A}_{TM} = { <M, w> : M $\in$ T_A and M accepts w }
- Proof: Suppose there exists a TM H that decides A^{A}_{TM} relative to A. Then, for any input <M,w>, where M $\in$ $T_{A,}$ H accepts if M accepts w and rejects otherwise.
- Consider a TM D that takes an input <M>, the description of M, and takes the following steps.
  - Run H on <M,<M>>
  - If H accepts, reject
  - If H rejects, accept
- Since H is a decider, D is also a decider.
- D on <D> = accept

  iff {def. D} H <D, <D>> = reject

  iff {def. H} D on <D> = reject (Go both directions!) ⇲