Theorem: For any language $A$, there exists a language $B$, s.t. $A \leq_T B$ and $B \not\leq_T A$

First, notice $A \leq_T A$, so finding $B$ s.t. $B \not\leq_T A$ is the interesting part

$B \not\leq_T A$ 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 $\}$

$A_{TM}^A = \{ <M, w> : M \in T_A$ and $M$ accepts $w \}$

Show $A \leq_T A_{TM}^A$ (easy: $w \in A$ iff $<N, w> \in A_{TM}^A$, where $N$ is a TM for $A$)

and $A_{TM}^A \not\leq_T A$ (by diagonalization, just like we showed $A_{TM}$ undecidable)
$A^A_{TM}$ is Undecidable

- Theorem: $A^A_{TM}$ is undecidable (relative to A).

- Recall: $A^A_{TM} = \{ \langle M, w \rangle : M \in T_A \text{ and } M \text{ accepts } w \}$

- 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 $<D> = \text{accept}$
  
  iff {def. D} H $<D, <D>> = \text{reject}$

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