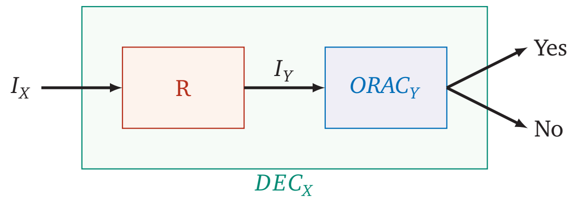
Proving that a language L is undecidable by reduction requires several steps, but you should begin by determining the reduction you want to do: $X \leq Y$ and draw out the reduction diagram:



Then you got to fill in the missing pieces:

- Choose a language X that you already know is undecidable (because we told you so in class). The simplest choice is usually the standard halting language
- Describe an algorithm that decides X, using an algorithm that decides Y as a black box (A_Y in the diagram). Typically your reduction will have the following form:
- Prove that your algorithm is correct. This proof almost always requires two separate steps:
  - Prove that if $x \in L'$ then $y \in L$.
  - Prove that if $x \notin L'$ then $y \notin L$.

Very important: Name every object in your proof, and *always* refer to objects by their names. Never refer to "the Turing machine" or "the algorithm" or "the input string" or (gods forbid) "it" or "this", even in casual conversation, even if you're "just" explaining your intuition, even when you're just *thinking* about the reduction.

Prove that the following languages are undecidable.

1. $E_{TM} := \{\langle M \rangle \mid M \text{ is a TM and } L(M) = \emptyset\}$
2. $EQ_{TM} := \{\langle M_1, M_2 \rangle \mid M_1 \text{ and } M_2 \text{ are TMs and } L(M_1) = L(M_2)\}$
3. $INF_{TM} := \{\langle M \rangle \mid M \text{ is a TM and } L(M) \text{ is an infinite language}\}$
4. $ALL_{TM} := \{\langle M \rangle \mid M \text{ is a TM and } L(M) = \Sigma^* \}$
5. $REG_{TM} := \{\langle M \rangle \mid M \text{ is a TM and } L(M) \text{ is a regular language}\}$