Proving that a language \( L \) is undecidable by reduction requires several steps. (These are the essentially the same steps you already use to prove that a problem is NP-hard.)

- Choose a language \( L' \) that you already know is undecidable (because we told you so in class). The simplest choice is usually the standard halting language
  \[
  \text{HALT} := \{ (M, w) \mid M \text{ halts on } w \}
  \]

- Describe an algorithm that decides \( L' \), using an algorithm that decides \( L \) as a black box. Typically your reduction will have the following form:

  Given an arbitrary string \( x \), construct a special string \( y \), such that \( y \in L \) if and only if \( x \in L' \).

In particular, if \( L = \text{HALT} \), your reduction will have the following form:

  Given the encoding \( \langle M, w \rangle \) of a Turing machine \( M \) and a string \( w \), construct a special string \( y \), such that \( y \in L \) if and only if \( M \) halts on input \( w \).

- 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 ever refer to “the Turing machine” or “the algorithm” or “the code” 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 to yourself.

Prove that the following languages are undecidable.

1. \( \text{AcceptILLINI} := \{ (M) \mid M \text{ accepts the string } \text{ILLINI} \} \)
2. \( \text{AcceptThree} := \{ (M) \mid M \text{ accepts exactly three strings} \} \)
3. \( \text{AcceptPalindrome} := \{ (M) \mid M \text{ accepts at least one palindrome} \} \)
4. \( \text{AcceptOnlyPalindromes} := \{ (M) \mid \text{Every string accepted by } M \text{ is a palindrome} \} \)

A solution for problem 1 appears on the next page; don’t look at it until you’ve thought a bit about the problem first.
Solution (for problem 1): For the sake of argument, suppose there is an algorithm \textsc{DecideAcceptIllini} that correctly decides the language \textit{AcceptIllini}. Then we can solve the halting problem as follows:

\begin{center}
\textsc{DecideHalt}(\langle M, w \rangle): \\
\text{Encode the following Turing machine } M': \\
\qquad M'(x): \\
\qquad \langle \langle \text{ignore the input string } x \rangle \rangle \\
\qquad \text{run } M \text{ on input } w \\
\qquad \langle \langle \text{ignore the output of } M \rangle \rangle \\
\qquad \text{return True} \\
\text{if } \textsc{DecideAcceptIllini}(\langle M' \rangle) \\
\quad \text{return True} \\
\text{else} \\
\quad \text{return False}
\end{center}

We prove this reduction correct as follows:

\begin{itemize}
  \item \[\Rightarrow\] Suppose \(M\) halts on input \(w\).
    \begin{itemize}
      \item Then \(M'\) accepts every input string \(x\).
      \item In particular, \(M'\) accepts the string \textit{ILLINI}.
      \item So \textsc{DecideAcceptIllini} accepts the encoding \(\langle M' \rangle\).
      \item So \textsc{DecideHalt} correctly accepts the encoding \(\langle M, w \rangle\).
    \end{itemize}
  \item \[\Leftarrow\] Suppose \(M\) does not halt on input \(w\).
    \begin{itemize}
      \item Then \(M'\) diverges on every input string \(x\).
      \item In particular, \(M'\) does not accept the string \textit{ILLINI}.
      \item So \textsc{DecideAcceptIllini} rejects the encoding \(\langle M' \rangle\).
      \item So \textsc{DecideHalt} correctly rejects the encoding \(\langle M, w \rangle\).
    \end{itemize}
\end{itemize}

In both cases, \textsc{DecideHalt} is correct. But that’s impossible, because \textsc{Halt} is undecidable. We conclude that the algorithm \textsc{DecideAcceptIllini} does not exist. \[\blacksquare\]

As usual for undecidability proofs, this proof invokes \textit{four} distinct Turing machines:

\begin{itemize}
  \item The hypothetical algorithm \textsc{DecideAcceptIllini}.
  \item The new algorithm \textsc{DecideHalt} that we construct in the solution.
  \item The arbitrary machine \(M\) whose encoding is part of the input to \textsc{DecideHalt}.
  \item The special machine \(M'\) whose encoding \textsc{DecideHalt} constructs (from the encoding of \(M\) and \(w\)) and then passes to \textsc{DecideAcceptIllini}.
\end{itemize}