Solution 2: Show that the following language is undecidable:

\[ L = \{\langle M \rangle : M \text{ is a Turing machine and } M \text{ accepts exactly the strings in } \Sigma^* \text{ whose length is a power of } 2 \}. \]

Assume for contradiction that \( L \) is decidable by a TM \( R \). We construct TM \( S \) to decide \( A_{\text{TM}} \).

\[ S = \text{"On input } \langle M, w \rangle \text{:"} \]

1. Construct \( M' \) as described below.
2. Run \( R \) on \( \langle M' \rangle \).
3. If \( R \) accepts then \text{accept}, otherwise \text{reject}."

\( M' = \text{"On input } x \text{:"} \)

1. Run \( M \) on \( w \).
2. If \( M \) accepts \( w \) then check if the length of \( x \) is a power of 2. If it is then \text{accept}, otherwise \text{reject}."

We see that when \( M \) accepts \( w \) then \( R \) will accept \( M' \), otherwise \( R \) will not accept \( M' \). Therefore \( S \) decides \( A_{\text{TM}} \), but we know that \( A_{\text{TM}} \) is undecidable, a contradiction.

By mapping reducibility, we reduce \( A_{\text{TM}} \) to \( L \) (\( A_{\text{TM}} \leq_{m} L \)). By corollary 5.23, because \( A_{\text{TM}} \) is undecidable, \( L \) is undecidable:

\[ F = \text{"On input } \langle M, w \rangle \text{:"} \]

1. Construct the following machine
   \( M' = \text{"On input } x \text{:"} \)
   1. Run \( M \) on \( w \).
   2. If \( M \) accepts \( w \) then check if the length of \( x \) is a power of 2.
      If it is then \text{accept}, otherwise \text{reject}."
2. Output \( \langle M' \rangle \"

Solution 3: Show that the following language is undecidable:

\[ EQ_{\text{TM}} = \{(M, N) | M \text{ and } N \text{ are TMs such that } L(M) = L(N)\} \]

Reduce from both \( E_{\text{TM}} \) and \( A_{\text{TM}} \). Recall that \( EQ_{\text{DFA}} \) was decidable.

In class we saw how to reduce \( E_{\text{TM}} \) to \( EQ_{\text{TM}} \). Here we will reduce from \( A_{\text{TM}} \) to prove that \( EQ_{\text{TM}} \) is undecidable.

Let \( D \) be a TM that decides \( EQ_{\text{TM}} \). We could then construct a decider \( S \) for \( A_{\text{TM}} \) as follows.

\[ S = \text{"On input } \langle M, w \rangle \text{, an encoding of a TM } M \text{ and a string } w, \]

1. Construct TM \( R_1 \) from \( M \) and \( w \) and TM \( R_2 \) as detailed below.

5: Solution Sketches-1
2. Run $D$ on $⟨R_1, R_2⟩$.
3. If $D$ accepts, reject; otherwise, accept.”

$R_1$ = “On input $x$,
1. Run $M$ on $w$.
2. If $M$ accepts, accept”

Notice that $R_2$ is the TM that we constructed when we proved $EQ_{TM}$ was undecidable by reducing from $E_{TM}$ (i.e., $L(R_1) = ∅$).

$R_2$ = “On input $x$,
1. reject.”

Thus, we contrive that $L(R_1) = ∅$ if and only if $M$ rejects $w$, while $L(R_2) = ∅$ always. Since, by assumption, we have a decider $D$ that tells us if these two machines recognize the same language, we know that if $D$ rejects $R_1$ and $R_2$, then this implies that $M$ accepts $w$.

**Solution 4:** Give yourself the following test, then check your answers on the back of the handout. Classify each of the following problems as either
- (D) decidable,
- (R) recognizable but not decidable,
- (C) co-recognizable but not decidable, or
- (N) neither recognizable nor co-recognizable,
and indicate which undecidable examples follow from Rice’s Theorem.

1. $EQ_{NFA}$, the Equivalence problem for NFA’s.
   D; recall the $EQ_{DEA}$ algorithm from textbook.
2. $\{⟨M⟩| M$ is a Turing Machine that runs for at least $n$ steps when started with a blank input tape, where $n$ is the length of the string $⟨M⟩\}$.
   D; just simulate $M$ for up to $|⟨M⟩|$ steps.
3. $\{⟨M⟩| M$ is a Turing Machine that accepts at least two inputs\}.
   R; Undecidable by Rice’s Theorem; Recognizable by running the TM in parallel (i.e., using the dovetailing technique from class) on all input strings until it accepts two strings.
4. $\{⟨M⟩| M$ is a Turing Machine that does not halt on input $⟨M⟩\}$.
   C; Undecidable by diagonalization. Co-recognizable by simulating $M$ on input $⟨M⟩$ and accepting when $M$ halts.
5. $\{⟨M⟩| L(M)$ is regular\}
   N; Undecidable by Rice’s Theorem; Neither recognizable nor co-recognizable from class.
6. $EQ_{TM}$
   N; Undecidable by Rice’s Theorem; Neither recognizable nor co-recognizable from textbook.

**Solution 5:** (Mapping Reducibility) Answer the following True or False:

5: Solution Sketches-2
1. $A_{TM}$ is mapping reducible to $E_{TM}$.
   False; $A_{TM}$ is recognizable, $E_{TM}$ is not. See Corollary 5.23.

2. $A_{TM} \leq \text{m } 0^*1^*$.
   False; $0^*1^*$ is decidable, $A_{TM}$ is not. See Theorem 5.22.

**Solution 6:** (Rice’s Theorem and Mapping Reducibility)
Consider the problem of testing whether a Turing machine $M$ accepts any binary string with an odd number of zeros.

1. Formulate this problem as a language; call it $ODDZ$.
   $ODDZ = \{ \langle M \rangle \mid L(M) \cap 1^*01^*(01^*0)^*1^* \neq \emptyset \}$

2. Show that $ODDZ$ is undecidable.
   Use Rice’s Theorem, show hypotheses are satisfied.

   Yes, by running the TM in parallel (i.e., using the dove-tailing technique from class) on all inputs strings with an odd number of zeros until it accepts.

   no, undecidable, but recognizable (if it also were co-recognizable then it would be decidable).