Prove that the following languages are undecidable.

1. $E_{TM} := \{\langle M \rangle \mid M \text{ is a TM and } L(M) = \emptyset\}$

Solution: $E_{TM}$ is the problem of determining whether the language of a TM is empty. We will reduce $\text{Decide}A_{TM}$ to $\text{Decide}E_{TM}$.

\[
\begin{array}{c}
\langle M \rangle \quad \text{R} \quad \text{DecideE}_{TM} \\
\langle M' \rangle \quad \text{ DecideE}_{TM} \\
\langle M' \rangle \quad \text{Yes} \quad \text{No} \\
\langle M' \rangle \quad \text{Yes} \quad \text{No} \\
\langle M' \rangle \quad \text{Yes} \quad \text{No} \\
\end{array}
\]

$M'(x)$:
- if $x \neq w$ REJECT
- else Run $M$ on input $w$ and accept iff $M$ accepts $w$

$\text{DecideA}_{TM}(\langle M, w \rangle)$:
- Construct $M'$ using $M$ and $w$
- Run $\text{DecideE}_{TM}$ on $\langle M' \rangle$
- if $\text{DecideE}_{TM}(\langle M' \rangle)$ reject
- else accept

If $\text{DecideE}_{TM}$ were a Decider for $E_{TM}$, then $\text{DecideA}_{TM}$ is a Decider on $A_{TM}$. But a decider for $A_{TM}$ can not exist, and hence $E_{TM}$ is undecidable.

2. $EQ_{TM} := \{\langle M_1, M_2 \rangle \mid M_1$ and $M_2$ are TMs and $L(M_1) = L(M_2)\}$

Solution: $EQ_{TM}$ is the problem of determining whether the languages of two TMs are the same. Let us assume that one of the languages is $\emptyset$, we end up with the problem of determining whether the language of the other machine is empty—that is, problem $1(E_{TM})$. Let’s do a reduction from $E_{TM}$.

The reduction is as follows. Let $\text{Decide}_{EQ_{TM}}$ decide $EQ_{TM}$ and we construct $\text{Decide}_{ETM}$ to decide $E_{TM}$ as follows.

\[
\begin{array}{c}
\langle M \rangle \quad \text{R} \quad \text{Decide}_{EQ_{TM}} \\
\langle M' \rangle \quad \text{Decide}_{ETM} \\
\langle M' \rangle \quad \text{Yes} \quad \text{No} \\
\langle M' \rangle \quad \text{Yes} \quad \text{No} \\
\langle M' \rangle \quad \text{Yes} \quad \text{No} \\
\end{array}
\]
Decide $E_M$: Let $M$ be a TM that rejects all inputs ($L(M) = \emptyset$).

If $\text{Decide}_{E_{\text{ETM}}} (\langle M, M' \rangle)$

return $\text{TRUE}$

else

return $\text{FALSE}$

3. $\text{INF}_{TM} := \{ \langle M \rangle \mid M \text{ is a TM and } L(M) \text{ is an infinite language} \}$

Solution: Let’s do a reduction from the accept language:

$$A_{TM} \Rightarrow \text{INF}_{TM}$$

The reduction is as follows. On input $\langle M, w \rangle$ we encode the following machine:

$M'(x)$:
run $M$ on input $w$ and return $\text{TRUE}$ if $M$ accepts $w$
otherwise return false

In this case, if $\text{ORAC}_{\text{INF}_{TM}}$ output yes, you know that the language $M'$ represents is infinite which is only possible if $M$ accepts $w$. If the oracle returns not true, you know $M$ must not accept $w$.

4. $\text{ALL}_{TM} := \{ \langle M \rangle \mid M \text{ is a TM and } L(M) = \Sigma^* \}$

Solution: Let’s do a reduction from $A_{TM}$.

$$A_{TM} \Rightarrow \text{ALL}_{TM}$$

The reduction is as follows. On input $\langle M, w \rangle$ we encode the following machine:
**DEC1_{ATM}(w):**

Let M’ be a TM that runs w on M and returns TRUE if M accepts w
if DEC2_{ALL_{TM}}( < M’ > )
return TRUE
else
return FALSE

If DEC2_{ALL_{TM}} outputs yes, M accepts w and L(M’) = Σ∗ and decides for ALL_{TM}.
If DEC1_{ALL_{TM}} decides ALL_{TM}, then DEC2_{ATM} decides A_{TM}. But A_{TM} is undecidable,
so DEC1_{ALL_{TM}} cannot exist and hence ALL_{TM} also must be undecidable. ■

5. **REG_{TM} := \{ ⟨M⟩ | M is a TM and L(M) is a regular language \}**

**Solution:** Let’s do a reduction from the accept language:

\[ A_{TM} \Rightarrow REG_{TM} \]

The reduction is as follows. On input \( ⟨M, w⟩ \) we encode the following machine:

\[ M’(x); \]
if x is of the form 0^n1^n
accept x
elseif M accepts w
accept x
else
reject x

This means: If the original M accepts w, then M’ will accept every string, this is regular. If the original M rejects w, then M’ will only accepts strings 0^n1^n, this is not regular.

So on the input \( ⟨M’⟩ \), if REG_{TM} returns TRUE then M accepts w and if REG_{TM} returns FALSE then M rejects w.

Therefore REG_{TM} must be undecidable. ■