For a given TM, we hope to check if there exists an equivalent finite automaton. The problem can be formulated as follows:

\[ \text{REGULAR}_{\text{TM}} = \{ \langle M \rangle \mid M \text{ is a TM and} \]
\[ L(M) \text{ is a regular language} \}\]

As before, assume this language is decidable and has a decider \( R \).

We construct a decider \( S \) for \( A_{\text{TM}} \) as follows.
Design a TM $M_2$ such that it recognizes

\[
\begin{align*}
\text{a regular language} & \quad \text{if } M \text{ accepts } w \\
\text{a non-regular language} & \quad \text{if } M \text{ rejects } w
\end{align*}
\]

(1)

Run $R$ on input $\langle M_2 \rangle$

If $R$ accepts, accept; if $R$ rejects, reject

Then we have

\[
\begin{align*}
S \text{ accepts} & \quad \text{if } M \text{ accepts } w \\
S \text{ rejects} & \quad \text{if } M \text{ rejects } w
\end{align*}
\]
Thus $S$ is a decider for $A_{TM}$, a contradiction

Now we give a specific design of $M_2$ so that (1) holds

We let $M_2$ recognize

$$
\begin{cases}
\Sigma^* & \text{if } M \text{ accepts } w \\
0^n1^n, \forall n \geq 0 & \text{if } M \text{ rejects } w
\end{cases}
$$

Note that $\Sigma^*$ is a regular language, but $0^n1^n, \forall n \geq 0$ is not

The implementation is as follows
On input $x$:

1. If $x$ has the form $0^n1^n$, accept
2. If $x$ does not have the form $0^n1^n$, run $M$ on input $w$ and accept if $M$ accepts $w$

- We see that if $M$ accepts $w$, then any $x \in \Sigma^*$ is accepted
- On the other hand, if $M$ does not accept $w$, only $0^n1^n, \forall n \geq 0$ are accepted
So far, our strategy for proving languages undecidable involves a reduction from $A_{TM}$.

Sometimes reducing from some other undecidable language is more convenient.

Here we show an example by considering

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

Assume $EQ_{TM}$ is decidable and has a decider $R$.

We construct a decider $S$ for $E_{TM}$ as follows.
For input $\langle M \rangle$, where $M$ is a TM:

1. Run $R$ on input $\langle M, M_1 \rangle$, where $M_1$ is a TM such that

$$L(M_1) = \emptyset$$

2. If $R$ accepts, accept; if $R$ rejects, reject

For $M_1$, we simply let it reject any input string. Recall we learned how to design an NFA for $\emptyset$

But $E_{TM}$ is undecidable by an earlier proof. Thus $EQ_{TM}$ is undecidable