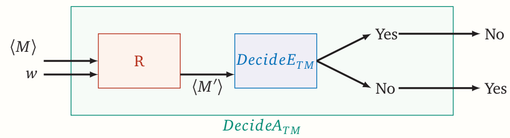
Prove that the following languages are undecidable.

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 lanuage of a TM is empty. We will reduce $DecideA_{TM}$ to $DecideE_{TM}$.




```
M'(x):
  if x != w
      REJECT
  else
      Run M on input w and accept iff M accepts w
```

```
DecideA_TM(⟨M,w⟩):
  Construct M' using M and w
  Run DecideE_TM on ⟨M'⟩
  if DecideE_TM(⟨M'⟩)
      reject
  else
      accept
```

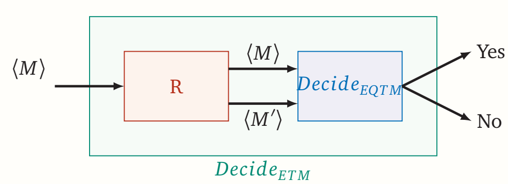
If $DecideE_{TM}$ were a Decider for E_{TM}, then $DecideA_{TM}$ is a Decider on A_{TM}. But a decider for A_{TM} can not exist, and hence E_{TM} is undecidable.

■

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

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 $Decide_{EQTM}$ decide EQ_{TM} and we construct $Decide_{ETM}$ to decide E_{TM} as follows.



$\underline{Decide_{ETM}(\langle M \rangle)}$:
Let M' be a TM that rejects all inputs($L(M') = \emptyset$).
if $Decide_{EQTM}(\langle M, M' \rangle)$
return TRUE
else
return FALSE

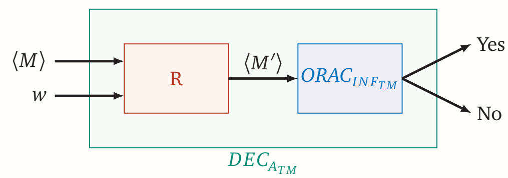
If $Decide_{EQTM}$ decides EQ_{TM}, $Decide_{ETM}$ decides E_{TM}. But E_{TM} is undecidable as we proved in problem 1, so EQ_{TM} also must be undecidable.

■

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

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

$$A_{TM} \Rightarrow INF_{TM}$$



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

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

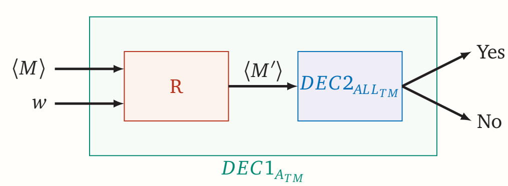
In this case, if $ORAC_{INF_{TM}}$ output yes, you know taht the language M' represents is infinite which is only possible if M accepts w. If the oracle returns not true, you know M must not accept w

■

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

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

$$A_{TM} \Rightarrow ALL_{TM}$$



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

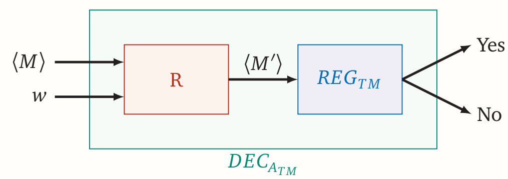
$\underline{DEC1_{ATM}(w)}$:
Let M' be a TM that runs w on M and returns TRUE if M accepts w
if $DEC2_{ALL_{TM}}(< M' >)$
  return TRUE
else
  return FALSE

If $DEC2_{ALL_{TM}}$ outputs yes, M accepts w and $L(M') = \Sigma^*$ and decides for ALL_{TM}. If $DEC1_{ALL_{TM}}$ decides ALL_{TM}, then $DEC2_{A_{TM}}$ decides A_{TM}. But A_{TM} is undecidable, so $DEC1_{ALL_{TM}}$ cannot exist and hence ALL_{TM} also must be undecidable. ■

5. $REG_{TM} := \{\langle M \rangle \mid M \text{ is a TM and } L(M) \text{ is a regular language}\}$

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

$$A_{TM} \Rightarrow REG_{TM}$$



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

$\underline{M'(x)}$:
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 $\langle M'\rangle$, 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. ■