Reductions: if we can reduce (transform) problem $A$ into a problem $B$, then solving problem $B$ gives solution to problem $A$.

Example: $HALT_{TM} = \{ <M,w> | M \text{ is a TM that halts on } w \}$

Thm 5.1: $HALT_{TM}$ is undecidable.

Note: $HALT_{TM}$ is the halting problem, $A_{TM}$ is the acceptance problem.
Thm 5.1: $E_{TM}$ is undecidable, where

$$E_{TM} = \{ <M> \mid M \text{ is a TM and } L(M) = \emptyset \}$$

reduce a known undecidable problem to $E_{TM}$ (i.e. solve a known undecidable problem using a line for $E_{TM}$)

i.e. $A_{TM}$ or $HALT_{TM}$

assume we have decide $\text{Emptyness}(M)$

want: decide $\text{Acceptance}(M, w)$

create $M'$ as shown on the right:

$L(M') = \{ \emptyset \}$ if $w$ is not accepted by $M$ (rejected or infinite computation)

if $w$ is accepted by $M$

$M'$: 1. erase the input $u$
    2. write down $w$ and move to the first symbol on the tape
    3. jump to go of $M$
Thm 5.3: \( \text{REGULAR}_{\text{TM}} \) is undecidable, where 

\( \text{REGULAR}_{\text{TM}} = \{ <M> | M \text{ is a TM and } L(M) \text{ is regular} \} \)  

(e.g. \( \text{ATM}, \text{HALT}_{\text{TM}}, \text{E}_{\text{TM}} \))

want: find a known undecidable problem, assume a fnc decideRegular, use it to solve the undecidable problem

assume fnc decideRegular exists (by contradiction)
then write own decideAcceptance, using decideRegular as a black box
but then, since decideAcceptance cannot exist (the problem \( \text{ATM} \) is undecidable), decideRegular cannot exist (if it did, decideAcceptance would exist). hence, \( \text{REGULAR}_{\text{TM}} \) is undecidable

fnc decide Acceptance \( (M, w) \):

create \( M' \) as described
if decideRegular \( (M) \) = true
then \( M \) did not accept \( w \) and we return FALSE
else return TRUE

Notice:
\( L(M') = \begin{cases} \emptyset & \text{if } M \text{ does not accept } w \\ \{a^k b^k | k \geq 0\} & \text{if } M \text{ accepts } w \end{cases} \)

0) go past \( u \) and write a special new symbol after \( u \) (\#)
1) write \( w \) after \( \# \) and move to the first position of \( w \)
2) run \( M \) on \( w \), not moving to \( \# \)
3) if \( M \) accepts, check if \( w = a^k b^k \) for some \( k \), if you accept, else reject
Thm 5.4: $E_{TM}$ is undecidable, where

$$E_{TM} = \{ <M_1, M_2> \mid M_1, M_2 \text{are TM's and } L(M_1) = L(M_2) \}$$
Thm 5.4: \( \text{ALL}_{\text{CFG}} \) is undecidable, where

\[
\text{ALL}_{\text{CFG}} = \{ \langle G \rangle \mid G \text{ is a CFG and } L(G) = \Sigma^* \}
\]

Idea: reduce from \( \text{ATM} \), i.e. we will show \( \text{ATM} \leq_m \text{ALL}_{\text{CFG}} \)

Create a CFG \( G \) s.t. it generates all strings over \( \Gamma \cup \{\#\} \) that are not a valid path of this form:

\[
x = \#u_1w_1q_1q_2u_2q_3w_3 \# \text{ or } \#u_1q_1u_2q_2w_2 \#
\]

where \( u_1 \) is a valid state, \( u_2 \) is yielded from \( \text{previous state} \), and \( w \) is a valid string (every second symbol reversed i.e. CFG)
Rice’s Thm [Problem 5.28]:

Let \( p \) be a language property. If \( p \) holds for some but not all languages, then the following language is undecidable:

\[
R = \{ <M> \mid M \text{ is a TM and } L(M) \text{ satisfies } p \}
\]