Contrapositive of last thm: Suppose \( A \leq_m B \). Then:

- If \( A \) is undecidable then \( B \) is undecidable.
- If \( A \) is not c.e., then \( B \) is not c.e.
- If \( A \) is not co-c.e., then \( B \) is not co-c.e.

Suppose \( A \leq_m C \) where \( A \) is not co-c.e. and also
\( D_R \leq_m C \) (we know \( D_R \) is not c.e.)

\[ E_{TM} = \{ \langle M \rangle : \exists \langle c \rangle \text{ s.t. } C \text{ is not a valid accepting trace of a computation of } M \text{ on input } x \} \]

\[ N_{TM} = \{ \langle M \rangle : \exists \langle c \rangle \text{ s.t. } C \text{ is an accepting trace of } M \text{ on input } x \} \]

We might expect to be able to do \( A_{TM} \leq_m N_{TM} \), but not \( A_{TM} \leq_m E_{TM} \).

\( N_{TM} \) is c.e.: A nondet TM \( N \) can guess an \( x \) and a \( c \), and feed \( \langle M, x, c \rangle \) to a universal TM computation checker.

\[ N_{TM} = L(N), \text{ and } N \text{ can be simulated by a DTM.} \]

Show \( N_{TM} \) is undecidable by \( \bar{A} \) arguing as in §5.1 or \( \bar{B} \) by reduction.
Suppose we had a decider $R$ for $\text{NE}_{TM}$. (or for $\text{ETM}$)
then we could build a decider $S$ for $\text{A_{TM}}$ as follows:

1. Modify $M$'s code to a TM $M'$ that first does a test for whether its own input $w$ equals $x$:

   \[ M' = \begin{cases} \text{accept} & \text{if } w = x \\ \text{reject} & \text{if } w \neq x \end{cases} \]

   (Step 1 can be done computably effectively)

2. Feed $\langle M' \rangle$ to our decider $R$ for $\text{NE}_{TM}$, and copy back the same answer.

3. If $R$ accepts, accept; else reject.

Given that $R$ is total, $S$ is total because steps 1 and 2 (and 3) always halt.

Then $S$ would decide $\text{A_{TM}}$ since $\langle M' \rangle \in \text{A_{TM}} \iff M' \text{ accepts } x \iff M' \text{ accepts } x \iff \text{LM}' = \{w \in \{0,1\}^* \mid w = x \} \neq \emptyset \iff \text{LM}' \in \text{NE}_{TM}$.

But since $\text{A_{TM}}$ is undecidable, $S$ cannot exist. \( \therefore R \) cannot exist; \( \therefore \text{NE}_{TM} \) is undecidable.

\[ \text{5.3: To show $A_{TM} \leq_{m} \text{NE}_{TM}$, } \langle M, x \rangle \xrightarrow{f} M' \leq_{m} \text{NE}_{TM} \]

Define a computable function $f$ by: instance of $A_{TM}$

$f$ is computable since it only does a code translation. The reduction $f$ is correct because $\langle M, x \rangle \in A_{TM} \iff f(\langle M, x \rangle) = M' \in \text{NE}_{TM}$ as above.
ATM \leq_m \{ M : L(M) = \Sigma^* \} \equiv ALLTM. Via reduction.

Wrong fact: interlaces:

$$\forall x \in \Sigma^n \exists c (c \text{ is a valid acc trace of } M(x))$$

Let's try a variant of the last reduction: $$\langle M, x \rangle \xrightarrow{c} M' =$$

\[\begin{array}{c}
\text{Does } W = x? \\
\downarrow \\
\text{yes} \\
\text{accept}
\end{array}
\]

Clearly this \( f \) is computable—just makes one change to the previous \( f \).

Analysis:

\[\langle M, x \rangle \in \text{ATM} \iff M' \text{ accepts all strings } W \text{ including } x \iff f(M, x) = M' \in \text{ALLTM}.\]

\[\begin{array}{c}
\text{All tm} \subseteq \text{ATM} \text{ so ALLTM is undecidable, indeed not co-ce.}
\end{array}\]

Does \( DTM \leq_m \text{ALLTM} \) as well?

Yes, by a "waiting" code construction.

\[\langle M \rangle \xrightarrow{g} M'' = \]

Instance \( f \) \( DTM \equiv \{ M : M \text{ does not accept } M' \} \)

Then \( g \) is also a computable code translation.

Correctness: \( \langle M \rangle \in DTM \iff M \text{ does not accept } \langle M \rangle \)

\[\iff \text{the diamond never rejects any } W \equiv L(M'') = \Sigma^* \]

\[\iff f(M) \in \text{ALLTM}.\]

\[\iff \text{since DTM is not-ce, ALLTM is not-ce either.} \]

\[\iff \text{ALLTM is neither ce nor co-ce.} \]

\[\downarrow \text{input } W.\]

\[\begin{array}{c}
\text{Compute } n = 1\text{wl}.
\end{array}\]

\[\downarrow \text{Simulate } M(\langle \langle M \rangle \rangle) \text{ for up to } n \text{ steps.} \]

\[\begin{array}{c}
\text{Did } M \text{ accept } \langle \langle M \rangle \rangle \text{ in that time?}
\end{array}\]

\[\begin{array}{c}
\text{no} \rightarrow \text{reject } W.
\end{array}\]

\[\begin{array}{c}
\text{yes} \rightarrow \text{accept } W, \langle \langle M \rangle \rangle.
\end{array}\]

\[\begin{array}{c}
\text{If } M \text{ does accept accept-W, } \langle \langle M \rangle \rangle \text{ then it does so in some number } n \text{ of steps so in same number of steps all } W \text{ with } 1\text{wl2n get rejected.} \]

\[\text{Indeed, } L(\langle M \rangle') \supseteq \Sigma^*.\]