D_Tm = D = \{ \langle M \rangle : M \text{ does not accept } \langle M \rangle \} \\
\text{Complement is} \\
K_Tm = K = \{ \langle M \rangle : M \text{ does accept } \langle M \rangle \}

K is undecidable because D is undecidable. But K is c.e. because K is a specially marked subset of

A_Tm = \{ \langle M, w \rangle : M \text{ accepts } w \} \text{, Universal Turing Machine}

which is the language of the Turing Kit, or of any one.

The Ktau problem is the special case of the ATM problem.

A_Tm: INST: A TM M and an input \( w \in \Sigma^* \)

QUES: Does \( M \) accept \( w \)?

K_Tm: INST: Just M.

QUES: Does \( M \) accept \( \langle M \rangle \)?

\( \langle M \rangle \in K_Tm \text{ } \Leftrightarrow \langle M, M \rangle \in A_Tm \)

Theorem: \( A_Tm \) is c.e. but not decidable.

Goal: Get further such conclusions from observing that the function \( f(\langle M \rangle) = \langle M, M \rangle \) satisfies the relation

\( f(x) = \langle x, x \rangle \text{, } x \in K_Tm \Leftrightarrow \text{ for } \langle x \rangle \in A_Tm \)
**Defn:** Given any languages $A, B \subseteq \Sigma^*$, we write $A \leq_m B$ and say $A$ maps-one-reduces to $B$ if there is a computable function $f: \Sigma^* \to \Sigma^*$ such that for all $x \in \Sigma^*$, $x \in A \iff f(x) \in B$.

**Previous Example:** $A = K_m$, $B = A_m$.

**Theorem:** Suppose $A \leq_m B$. Then
1. If $B$ is decidable then $A$ is decidable.
2. If $B$ is c.e., then $A$ is c.e.
3. If $B$ is co-c.e., i.e., $\overline{B}$ is c.e., then $A$ is 10-c.e.

**Proof:**
1. Take a total TM $M_a$ such that $L(M_a) = B$ and a total TM $T$ computing $f$. View $\overline{\text{"Transducer"}}$ them as solid boxes.

Then $M_A$ is total, and for all $x$:
- $M_A$ accepts $x \iff M_B$ accepts $y = f(x) \iff y \in B \iff x \in A$

So $M_A$ decides $A$. 

- $L(M_B) = B$
- $x \in A \iff f(x) \in B$ since $f$ is a reduction.

\[ \text{If } \text{acc, } \text{then } \text{accept } x \text{ else reject } x \]
In (b), by B is c.e., we are only given that \( L(M_B) = \overline{B} \), not that \( M_B \) is total. We draw a "fuzzy box" for \( M_B \):

Then \( M_A \) represents executable code and for all \( x \), \( x \in A \iff \forall y : y \in B \iff M_B \) accepts \( y \iff M_A \) accepts \( x \). \( \therefore L(M_A) = A \), so A is c.e.

For (c), we have \( A \leq_m B \) and \( B \) is c.e., i.e. \( \overline{B} \) is c.e.

We claim \( \overline{A} \leq_m \overline{B} \) because \( x \in A \iff \forall y : y \in B \iff x \notin \overline{A} \iff \forall y : y \notin \overline{B} \iff x \in \overline{A} \).

Since \( \overline{B} \) is c.e., we get \( \overline{A} \) is c.e. by part (b), so \( A \) is co-c.e.

By symmetry of REC, \( A \) decidable \( \iff \overline{A} \in \text{REC} \)

Diagram also conveys Theorem 3: \( A \) is decidable \( \iff \text{REC} \cap \text{co-RE} = \text{REC} \).

\( A \leq_m B \) means the angle from \( A \) up to \( B \) is at least as steep as the diagonal walls.
Proof: Suppose \( B \) and \( \tilde{B} \) are both c.e. Then we can take TMs \( M_1, M_2 \) (not necessarily total) st \( L(M_1) = B \) and \( L(M_2) = \tilde{B} \). Build \( M_3 \) to execute the following flowchart loop:

Since \( L(M_1) \) and \( L(M_2) \) are complementary on any input \( x \), exactly one of them will eventually accept. Thus \( M_3(x) \) always exits the loop (by) then, so \( M_3 \) is total and \( L(M_3) = B \), so \( B \) is RE.

Contrapositive of Theorem 2: Suppose \( A \leq_m B \) then:

(a) If \( A \) is undecidable then \( B \) is undecidable
(b) If \( A \) is not c.e. then \( B \) is not c.e.
(c) If \( A \) is not co-c.e. then \( B \) ditto. \( A_m \leq_m \operatorname{NE}_m \)

Example: We can prove \( \operatorname{NE}_m \) is undecidable by showing goal: Build a compatible function \( f \) st. \( \langle M, w \rangle \mapsto \langle \text{form, w} \rangle \) \( f(M, w) \) will be the code of a single machine \( M' \) such that \( L(M') \neq \emptyset \implies M \) accepts \( w \). Correctness of the reduction Proof is mainly constructing \( M' \) from \( M \) and \( w \).
\((M,w) \xrightarrow{f} M'\)

We can build the code of \(M'\) given that of \(M\) and \(w\).

And for correctness:

If \(M\) accepts \(w\), then for all \(y\), \(M'\) accepts \(y\), so \(L(M') = \Sigma^*\), so \(L(M') \neq \emptyset\).

\(\therefore (M,w) \in \text{A}_{\text{TM}} \implies (M') \in \text{NE}_{\text{TM}}\).

But if \(M\) does not accept \(w\), then for all \(y\), \(M'\) never gets to accept \(y\), so \(L(M') = \emptyset\).

\(\therefore (M,w) \in \text{A}_{\text{TM}} \implies (M') \in \text{NE}_{\text{TM}}\) (by instantiation).

\(\therefore (M,w) \in \text{A}_{\text{TM}} \implies f(M,w) = (M') \in \text{NE}_{\text{TM}}\)

and \(f\) is computable, so \(\text{A}_{\text{TM}} \leq_m \text{NE}_{\text{TM}}\), and \(\text{A}_{\text{TM}}\) is undecidable, so \(\text{NE}_{\text{TM}}\) is undecidable.

Note, however, that \(\text{NE}_{\text{TM}}\) is c.e. → Thursday.