Key Definition: A language $A$ \{many-one reduces\} mapping-reduces to a language $B$, written $A \leq_m B$, if there is a total computable function $f : \Sigma^* \rightarrow \Sigma^*$ such that:

$$\text{for all } x \in \Sigma^*, \ x \in A \iff f(x) \in B.$$ 

Note: This is the same as $x \in A \iff f(x) \in \bar{B}$, so $A \leq_m B \iff \bar{A} \leq_m \bar{B}$. “Mapping Reduction are Mirror.”

Visual Convention

$A \leq_m B$ is indicated by making the angle from $A$ up to $B$ steeper than “45°.”

Key Lemma: For all languages $A, B$:

- If $A \leq_m B$ and $B$ is decidable, then $A$ is decidable.
- If $A \leq_m B$ and $B \in \text{RE}$, then $A \in \text{RE}$, i.e., $A$ is recognizable.
- If $A \leq_m \bar{B}$ and $B \in \text{co-RE}$, then $A \in \text{co-RE}$.

Proof of (c) first: If $A \leq_m B$ and $B \in \text{co-RE}$, then $\bar{B} \in \text{RE}$ and assuming (\text{\textcircled{\textcircled{1}}}) $\bar{A} \leq_m B$: $\bar{A} \leq_m B$ by (\text{\textcircled{\textcircled{1}}}), $\bar{A} \in \text{RE}$, so finally $A \in \text{co-RE}$. 

Note $A \leq_m K 

\iff \bar{A} \leq_m \bar{K}$. 

since $D = \sim K$. 

Proof of (a): Take any total TM $M_B$ s.t. $L(M_B) = B$. Take a total TM $T$ that computes $f$. Build $M_A$ as follows:

compute $y = T(x)$

Input $x \in \Sigma^*$

$M_A$

Output $y$

Correctness: $M_A$ is total and for all $x$,

$M_A$ accepts $x \iff M_B$ accepts $y$

by construction

by reduction $\iff L(M_B) = B \iff f(x) \in B$

by construction $\iff y \in B, \Rightarrow L(M_A) = A.$

For (b), even if $M_B$ merely recognizes $B$, $M_A$ still recognizes $A$. $\blacksquare$

For (b), even if $M_B$ merely recognizes $B$, $M_A$ still recognizes $A$. $\blacksquare$

Remark $\Rightarrow$ Chapter 7 (Thm 7.31)

In case (a), you might think the total running time $t(n)$ for $M_A$ equals the sum of the runtime $t_1(n)$ for $y = T(x)$ and the $(n=1x1)$ runtime $t_2(2)$ for $M_B$. But note: $t(n) = 1-1!$ which might be $t(n) = 1!$. Best estimate for the time by $M_A$ is $t_2(t_1(n))$. Still polynomial, $t_2(n) = n^{o(1)}$.

Corollary — The Contraposible: If $A \leq_m B$, then:

(a'): if $A$ is undecidable then $B$ is undecidable.

(b'): if $A$ is not r.e. (recognizable) then $B$ is not r.e. either.

(c'): if $A \in \text{co-R.E.}$, then $B \in \text{co-R.E.}$.

Part (a') is how we use reductions to show undecidability. And if $A \in \text{co-R.E.}$ and $A \leq_m B$ and $B \leq_m B$ too, then $B$ is in the intersection of two "upward reducibility cones" so $B$ is not r.e. nor r.e. nor co-r.e. nor co-r.e.
Examples of reductions: A simple "f" first:

1. \( K \subseteq \text{ATM} \) via the function \( f(u) = \langle u, u \rangle \).

Correct since \( K = \{ u : u \) is the code of a TM \( M_u \) such that \( M_u \) accepts \( u \} \)

so \( u \in K \iff M_u \text{ accepts } u \iff \langle M_u, u \rangle \in \text{ATM} \)

Annoying issue again: what if \( u \) is not a valid code?

OK, then \( \langle u, u \rangle \) isn't valid either, so \( u \notin K \) and \( \langle u, u \rangle \notin \text{ATM} \).

2. \( \text{ATM} \leq \text{HALT}_{\text{TM}} \)

\[ \langle M, x \rangle \quad \langle M', x' \rangle \]

make = \( x \) \hspace{1cm} \text{Construction} \hspace{1cm} \text{space} \hspace{1cm} \text{for \text{at least one \text{HALT}_{\text{TM}}}} \]

\[ \text{computability} \quad f(\langle M, x \rangle) = \langle M', x' \rangle \text{ where } x' = x \]

and \( M' \) is computed by adding \( \langle c, c, s, \text{why} \rangle \) to the code of \( M \).

This is a "lexical transformation of code" and is (easily) computable.

Correctness: \( \langle M, x \rangle \in \text{ATM} \iff M(x) \) accepts \( \iff M'(x) \) halts, because \( M' \) doesn't halt when \( M \) doesn't accept. \( \iff \langle M', x \rangle \in \text{HALT}_{\text{TM}} \)

Thus \( \text{ATM} \leq \text{HALT}_{\text{TM}} \).

3. Also \( \text{HALT}_{\text{TM}} \leq \text{ATM} \). Thinking of correctness first, we need \( f' \)

defined on instances \( \langle M, x \rangle \) of the \( \text{HALT}_{\text{TM}} \) problem so that \( \langle M, x \rangle \) is in the \( \text{HALT}_{\text{TM}} \) language.

\[ M(x) \iff M' \text{ accepts } x \iff \langle M', x \rangle \in \text{ATM} \text{ where } \langle M', x \rangle = f'(\langle M, x \rangle). \]
Since \( \text{ATM} \leq_m \text{HALT}_{TM} \) and \( \text{HALT}_{TM} \leq_m \text{ATM} \), we write \( \text{ATM} \equiv_m \text{HALT}_{TM} \) and say they are mapping-equivalent.

\[
\text{ATM} \leq_m \text{NB}_{TM}:
\begin{align*}
\langle M, x \rangle & \xrightarrow{C} M' = \\
\text{NE}_{TM} & : \text{INST: } M \\
& \text{Guess: } \exists M' \neq \emptyset
\end{align*}
\]

Visually, ATM does not mapping reduce to BPM:

**Correctness:** This \( f \) is compatible because it takes \( M \) and \( x \) and inserts it as a "\((0,0)" into the code of \( M' \):
\[
\langle M, x \rangle \in \text{ATM} \iff M \text{ accepts } x \iff \text{for all } w, M' \text{ accepts } w \Rightarrow L(M') = \Sigma^* \\
\Rightarrow L(M') \neq \emptyset \Rightarrow \langle M' \rangle \in \text{NE}_{TM}
\]

\[
\langle M, x \rangle \in \text{ATM} \iff M \text{ does not accept } x \iff \text{for all } w, M' \text{ does not accept } w \\
\Rightarrow L(M') = \emptyset \Rightarrow \langle M' \rangle \in \text{NB}_{TM}.
\]

\( \text{ATM} \equiv_m \text{NB}_{TM} \).

\( \text{ATM} \equiv_m \text{E}_{TM} \), since \( \text{E}_{TM} \equiv_m \text{NB}_{TM} \), so \( \text{ATM} \equiv_m \text{NB}_{TM} \).

**Corollary:** \( \tilde{\text{ATM}} \equiv_m \tilde{\text{E}}_{TM} \) since \( \tilde{\text{E}}_{TM} \equiv_m \tilde{\text{NB}}_{TM} \), so we get that since \( \tilde{\text{ATM}} \) is not recognizable, \( \tilde{\text{E}}_{TM} \) is not recognizable either.

**Added:** By the same construction and analysis, we can show \( \tilde{\text{E}}_{TM} \equiv_m \text{ALL}_{TM} \) too:
\[
M \in \text{E}_{TM} \Rightarrow \text{for all } w, M'(w) \text{ never finds an accept } \Rightarrow L(M') = \Sigma^* \\
M \notin \text{E}_{TM} \Rightarrow \text{for some long enough } w, M'(w) \text{ finds accept } \Rightarrow L(M') \neq \Sigma^*.
\]

**Else:** \( M \) is accepting, \( w \) is not accepted.