\( L\emptyset = \{ \langle M \rangle \in \Sigma^* \mid L(M) = \emptyset \} \)

\( L_{\neq \emptyset} = \{ \langle M \rangle \in \Sigma^* \mid L(M) \neq \emptyset \} \)

Using a pair generator and trying all strings in some canonical order we showed that \( L_{\neq \emptyset} \) is R.E.

How about \( L\emptyset \)?

Note that if \( L\emptyset \) is R.E. \( \Rightarrow L\emptyset \) is recursive, recursive / decidable / same computable
Claim \( L \Phi \) is non-r.e.

Consider some known non-r.e. lang \( L \) and \( L \leq f \Phi \)

\[ \{ L_d, L_w \} \]

\( L_w \leq L \Phi \langle M \rangle \)

\( \langle M, w \rangle \)

\( \Xi^* \)

\( \Xi^* \)

If \( M \) doesn't accept \( w \) then \( f(M, w) = \langle M' \rangle \) must satisfy \( \langle M' \rangle \notin L \Phi \)

If \( M \) accepts \( w \) then \( \langle M' \rangle \notin L \Phi \)
Currying function

g(x, y) converted to \( g_x(y) \)

(transform a 2-input to a 1-input function by fixing one parameter)

\( g_x(y) : g_{xy} \) with no input

\[ f(M, w) \text{ can be transformed to } f_{xy}(w) \]
and can be further transformed to

\[ f_{M, w} \]

Can be done using a program (Turing machine)

Initially \( f \) is similar the function for \( M_n \) and then transformed into \( \overline{M_n^{M, w}} \) : code 1
Next we stitch another code:

**code 2**: If \( M \) accept \( w \) then read the input and accept it

**Code 1**

**Code 2** : \( \langle M' \rangle \)

\( M' \) : 1. It initially ignores the input
2. Runs \( M \) on \( w \)
3. If \( M \) accepts \( w \), it accepts \( x \)
4. If \( M \) doesn't accept \( w \), then \( x \) is not accepted

\( M \) accepts \( w \) \( \Rightarrow \) \( L(M') = \Sigma^* \)

\( M \) doesn't accept \( w \) \( \Rightarrow \) \( L(M') = \emptyset \)
Property of r.e. languages

\( \langle M \rangle \) empty
\( \langle M \rangle \) non-empty
\( \langle M \rangle \) \( \not\in \mathbb{E}^* \)
\( \langle M \rangle \) non-receivable

Any non-trivial property of r.e. languages is undecidable

Rice's Theorem

Not all machines satisfy the property
Some \( \langle M \rangle \) should satisfy and some \( \langle M \rangle \) shouldn't satisfy