Rice’s Theorem. Let \( \mathcal{L} \) be any set of languages that satisfies the following conditions:

- There is a Turing machine \( Y \) such that \( \text{Accept}(Y) \in \mathcal{L} \).
- There is a Turing machine \( N \) such that \( \text{Accept}(N) \notin \mathcal{L} \).

The language \( \text{AcceptIn}(\mathcal{L}) := \{ \langle M \rangle \mid \text{Accept}(M) \in \mathcal{L} \} \) is undecidable.

Prove that the following languages are undecidable using Rice’s Theorem:

1. \( \text{AcceptRegular} := \{ \langle M \rangle \mid \text{Accept}(M) \text{ is regular} \} \)

   \[ \text{Solution:} \] Here and in all following solutions, let \( M_{\text{Accept}} \) be any Turing machine that accepts every input, and let \( M_{\text{Reject}} \) be any Turing machine that rejects every input.

   Let \( M \) be any Turing machine that decides our favorite non-regular language \( \{0^n1^n \mid n \geq 0 \} \). Then \( \text{Accept}(M_{\text{Reject}}) = \emptyset \) is regular, but \( \text{Accept}(M) \) is not. \( \blacksquare \)

2. \( \text{AcceptILLINI} := \{ \langle M \rangle \mid M \text{ accepts the string ILLINI} \} \)

   \[ \text{Solution:} \] \( \text{Accept}(M_{\text{Accept}}) = \Sigma^* \) contains the string ILLINI, but \( \text{Accept}(M_{\text{Reject}}) = \emptyset \) does not. \( \blacksquare \)

3. \( \text{AcceptPalindrome} := \{ \langle M \rangle \mid M \text{ accepts at least one palindrome} \} \)

   \[ \text{Solution:} \] Again, \( \text{Accept}(M_{\text{Accept}}) = \Sigma^* \) contains at least one palindrome, but \( \text{Accept}(M_{\text{Reject}}) = \emptyset \) does not. \( \blacksquare \)

4. \( \text{AcceptThree} := \{ \langle M \rangle \mid M \text{ accepts exactly three strings} \} \)

   \[ \text{Solution:} \] Let \( M \) be any Turing machine that accepts the strings IL, LI, and NI, and nothing else. \( \text{Accept}(M) \) contains exactly three strings, but \( \text{Accept}(M_{\text{Reject}}) = \emptyset \) does not. \( \blacksquare \)

5. \( \text{AcceptUndecidable} := \{ \langle M \rangle \mid \text{Accept}(M) \text{ is undecidable} \} \)

   \[ \text{Solution:} \] Let \( \text{AcceptAcceptEmpty} \) be any Turing machine that accepts the language \( \{ \langle M \rangle \mid M \text{ accepts the empty string } \varepsilon \} \). (Given \( \langle M \rangle \), simulate \( M \) on input \( \varepsilon \); if \( M \) accepts \( \varepsilon \), then accept.) Then \( \text{Accept}(\text{AcceptAcceptEmpty}) \) is undecidable, but \( \text{Accept}(M_{\text{Reject}}) = \emptyset \) is not. \( \blacksquare \)
To think about later. Which of the following are undecidable? How would you prove that?

1. **Accept\{\{\epsilon\}\} := \{(M) \mid M \text{ accepts only the string } \epsilon; \text{ that is, } \text{Accept}(M) = \{\epsilon\}\}**

**Solution: Undecidable by Rice’s theorem.** Let \( \mathcal{L} = \{\{\epsilon\}\} \) — the set containing one language, which contains one string, which is empty. Let \( M_\epsilon \) be a Turing machine with the transitions

\[
\delta(\text{start}, \square) = (\text{accept}, \cdot, \cdot)
\]

\[
\delta(\text{start}, a) = (\text{reject}, \cdot, \cdot) \quad \text{for all } a \in \Sigma.
\]

Clearly \( \text{Accept}(M_\epsilon) = \{\epsilon\} \in \mathcal{L} \). On the other hand, \( \text{Accept}(M_{\text{Reject}}) = \emptyset \notin \mathcal{L} \). ■

2. **Accept\{\emptyset\} := \{(M) \mid M \text{ does not accept any strings; that is, } \text{Accept}(M) = \emptyset\}**

**Solution: Undecidable by Rice’s theorem.** Let \( \mathcal{L} = \{\emptyset\} \) — the set containing one language, which contains no strings. We immediately have \( \text{Accept}(M_{\text{Reject}}) = \emptyset \in \mathcal{L} \) but \( \text{Accept}(M_{\text{Accept}}) = \Sigma^* \notin \mathcal{L} \). ■

3. **Accept=Reject := \{(M) \mid \text{Accept}(M) = \text{Reject}(M)\}**

**Solution: Undecidable by definition-chasing.** \( \text{Accept}(M) = \text{Reject}(M) \) if and only if \( M \) diverges on every input string. Thus, \( \text{Accept}=\text{Reject} = \text{NeverHalt} \), which is proved undecidable in the notes. ■

4. **Accept\neqReject := \{(M) \mid \text{Accept}(M) \neq \text{Reject}(M)\}**

**Solution: Undecidable by closure properties.** \( \text{Accept}(M) \neq \text{Reject}(M) \) if and only if \( M \) halts on at least input string. Thus, \( \text{NeverHalt} = \text{TMEncodings} \setminus \text{Accept}\neq\text{Reject} \), where \( \text{TMEncodings} \) is the language of all Turing machine encodings. \( \text{TMEncodings} \) is decidable, but \( \text{NeverHalt} \) is not. Thus, Corollary 3(d) in the undecidability notes implies that \( \text{Accept}\neq\text{Reject} \) is undecidable. ■

5. **Accept\cup\text{Reject} := \{(M) \mid \text{Accept}(M) \cup \text{Reject}(M) = \Sigma^*\}**

**Solution: Undecidable by definition-chasing.** \( \text{Accept}(M) \cup \text{Reject}(M) = \Sigma^* \) if and only if \( M \) halts on every input string. Thus, \( \text{Accept}\cup\text{Reject} = \text{NeverDiverge} \), which is proved undecidable in the notes. ■