Claim: Let $L$ be an RE language such that $\overline{L}$ is also RE. Then $L$ is recursive.

Proof sketch:

- Let $L(M) = L$.
- Let $L(M') = \overline{L}$.
- We construct $M''$ to simulate $M$ and $M'$ in parallel using two tapes.
- Because either $M$ or $M'$ must accept $x$, $M''$ halts on all inputs.

Defn: A language $L$ is said to be co-RE if $\overline{L}$ is RE.

Corollary: If $L$ is a recursive language, then $L$ is co-RE.
Claim: For any language $L$, $L \leq L$.

Claim: Let $L_1$, $L_2$, and $L_3$ be languages such that $L_1 \leq L_2$ and $L_2 \leq L_3$. Then $L_1 \leq L_3$.

Defn: $L_1$ and $L_2$ are said to be many-one equivalent ($L_1 \equiv L_2$) if $L_1 \leq L_2$ and $L_2 \leq L_1$.

Rice’s Theorem

Defn: Let $RE(\Sigma)$ denote the set of RE languages over $\Sigma$. A property of $RE(\Sigma)$ is any subset of $RE(\Sigma)$. $\emptyset$ and $RE(\Sigma)$ are trivial properties; all other properties are nontrivial.

Defn: Let $\mathcal{P} \subseteq RE(\{0,1\})$. Then $L_{\mathcal{P}} = \{w \in \{0,1\}^* \mid L(M(w)) \in \mathcal{P}\}$.

Theorem 9.11: (Rice’s Theorem) If $\mathcal{P}$ is a nontrivial property of $RE(\{0,1\})$, then $L_{\mathcal{P}}$ is not recursive.
Case 1: $\emptyset \not\in \mathcal{P}$.

- We will show that $L_U \leq L_\mathcal{P}$.
- Because $\mathcal{P}$ is a nontrivial property of $RE\{0,1\}$, there exist $L \in \mathcal{P}$, and a TM $M_L$ such that $L(M_L) = L$.
- We will construct a TM $M$ that, on input $x$, produces an output $y$ such that $x \in L_U$ iff $L(M(y)) \in \mathcal{P}$.

$M(y)$ will be a 2-track TM operating as follows on input $w$:

1. Write $x$ to track 2, leaving $w$ on track 1.
2. Return to the left end of $x$ and $w$.
3. Simulate $U$ on $x$ using track 2.
4. If $U$ accepts $x$, simulate $M_L$ on $w$ using track 1.
5. If $M_L$ accepts $w$, accept.

If $x \in L_U$, then $w \in L(M(y))$ iff $w \in L$; i.e., $L(M(y)) = L \in \mathcal{P}$. If $x \not\in L_U$, then $L(M(y)) = \emptyset \not\in \mathcal{P}$. 
How \( M \) computes \( y \) given \( x \):

- Only step 1 of \( M(y) \) depends on \( x \); hence the remainder of \( y \) can be stored in the finite control of \( M \).
- \( M \) only needs to insert the states and transitions needed to write \( x \) on track 2.

Case 2: \( \emptyset \in \mathcal{P} \). By Case 1, \( \overline{\mathcal{P}} \) is not recursive.

\[
\overline{\mathcal{P}} = \{w \in \{0, 1\}^* \mid L(M(w)) \in \overline{\mathcal{P}}\} \\
= \{w \in \{0, 1\}^* \mid L(M(w)) \notin \mathcal{P}\} \\
= \{w \in \{0, 1\}^* \mid L(M(w)) \in \mathcal{P}\} \\
= \overline{\mathcal{P}}.
\]

Because \( \overline{\mathcal{P}} \) is not recursive, \( \mathcal{P} \) is not recursive.