Rice's Theorem

Rice's theorem helps explain one aspect of the pervasiveness of undecidability. Here is the theorem and its proof, following the needed definition.

A **property of languages** is a predicate P: $\mathcal{P}(\Sigma^*) \to \{\mathsf{false}, \mathsf{true}\}$ for some alphabet Σ. That is, the input to P is a language and the output is a truth value. The value $P(L) = \mathsf{true}$ (we can write "$P(L)$") means that L *has* the property P; the value $P(L) = \mathsf{false}$ (we can write "$\overline{P(L)}$") means that L does not have property P. Example properties are: is finite, is infinite, is regular, is r.e., contains the empty string, contains the string 1011, contains some palindrome, contains only palindromes.

A **nontrivial property of r.e. languages** is a property of languages P such that $P(L_0)$ for some r.e. language L_0 and $\overline{P(L_1)}$ for some r.e. language L_1. In English, some r.e. language has the property and some r.e. language does not. All the example properties we listed above are nontrivial with the exception of "is r.e.".

Theorem [Rice]: If P is a nontrivial property of r.e. languages then

$$L_P = \{\langle M \rangle :\ P(L(M))\}$$

is undecidable. More specifically, (1) if $P(\emptyset)$ then L_P is not r.e., and (2) if $\overline{P(\emptyset)}$ then L_P is not co-r.e..

Proof: We prove the second claim; the first is similar. So we are assuming that $\emptyset$ does *not* have property P: $P(\emptyset) = \mathsf{false}$. We show $A_{\mathrm{TM}} \leq_{\mathrm{m}} L_P$. To show this, we must exhibit a Turing computable function f for which $\langle M' \rangle = f(\langle M, w \rangle)$ is a machine accepting a language with property P iff M accepts w. Let the behavior of M' on input x to be:

> Run M on w.
>
> If M rejects, *reject*.
>
> Run M_1 on x where M_1 is a (fixed) machine for which $P(\langle M_1 \rangle) = 1$. We know such an M_1 exists because P is a *nontrivial* property of r.e. languages.
>
> If M_1 accepts, *accept*; if M_1 rejects, *reject*."

Clearly M' is Turing computable from M and w. Observe that

(1) if M accepts w then $L(M') = L(M_1)$, which is a language with property P.

(2) if M does not accept w, then $L(M') = \emptyset$ which, by assumption, is a language that does not have property P.

Now try to do case (2) on your own.