34 Recall that \( w^R \) denotes the reversal of string \( w \); for example, \( TURING^R = GNIRUT \). Prove that the following language is undecidable.

\[
\text{RevAccept} := \{ \langle M \rangle \mid M \text{ accepts } \langle M \rangle^R \}
\]

Note that Rice's theorem does not apply to this language.

35 Let \( M \) be a Turing machine, let \( w \) be an arbitrary input string, and let \( s \) be an integer. We say that \( M \) accepts \( w \) in space \( s \) if, given \( w \) as input, \( M \) accesses only the first \( s \) (or fewer) cells on its tape and eventually accepts.

35.A. Sketch a Turing machine/algorithm that correctly decides the following language:

\[
\{ \langle M, w \rangle \mid M \text{ accepts } w \text{ in space } |w|^2 \}
\]

35.B. Prove that the following language is undecidable:

\[
\{ \langle M \rangle \mid M \text{ accepts at least one string } w \text{ in space } |w|^2 \}
\]

36 Consider the language \( \text{SometimesHalt} = \{ \langle M \rangle \mid M \text{ halts on at least one input string} \} \). Note that \( \langle M \rangle \in \text{SometimesHalt} \) does not imply that \( M \) accepts any strings; it is enough that \( M \) halts on (and possibly rejects) some string.

36.A. Prove that \( \text{SometimesHalt} \) is undecidable.

36.B. Sketch a Turing machine/algorithm that accepts \( \text{SometimesHalt} \).

37 For each of the following languages, either prove that the language is decidable, or prove that the language is undecidable.

37.A. \( L_0 = \{ \langle M \rangle \mid \text{given any input string, } M \text{ eventually leaves its start state} \} \)

37.B. \( L_1 = \{ \langle M \rangle \mid M \text{ decides } L_0 \} \)

37.C. \( L_2 = \{ \langle M \rangle \mid M \text{ decides } L_1 \} \)

37.D. \( L_3 = \{ \langle M \rangle \mid M \text{ decides } L_2 \} \)

37.E. \( L_4 = \{ \langle M \rangle \mid M \text{ decides } L_3 \} \)