
Problem 3.18. Show that a language is decidable iff some enumerator enumerates the language in the standard string order.

The proof is in two parts.

Part a. If a language is decidable, then some enumerator enumerates the language in the standard string order.

Proof. For any decidable language L, let M be the **TM**s that decides it. We construct an enumerator E that enumerates L in the standard string order:

$E =$ "Ignore the input:

1. Repeat the following for $i = 1, 2, 3, \cdots$
2. Generate all possible strings $s_1, s_2, s_3, \cdots$ of length i, where each $s_k \in \Sigma^*$.
3. Sort the strings in standard string order.
4. Run M on each s_k. If M accepts s_k, *print* s_k."

□

Part b. If some enumerator enumerates a language in the standard string order, then the language is decidable.

Proof. For any Turing-recognizable language L, let E be the enumerator that enumerates it in the standard string order. We construct a **TM** M that decides it:

"On input string w:

1. Run the enumerator E.
2. Repeat the following for each string s printed by E.
3. If $w = s$, *accept*.
4. If w precedes s in the standard string order, then continue. Otherwise *reject*.
5. If E halts, *reject*."

□