Recursively Enumerable Languages

A TM accepts a string w if the TM halts in a final state. A TM rejects a string w if the TM halts in a nonfinal state or *the TM never halts*.

A language L is *recursively enumerable* if some TM accepts it.

A language L is *recursive* if some TM accepts it *and* halts on every input. Note: the complement of a recursive language is also recursive.

L is recursive implies L is recursively enumerable.

Diagonalization

Suppose that we have a 2-D table of bits with an infinite number of rows and columns.

$$\begin{array}{ccccc} \boxed{1} & 0 & 1 & 1 & \cdots \\ 0 & \boxed{1} & 1 & 1 & \cdots \\ 1 & 0 & \boxed{0} & 0 & \cdots \\ 1 & 0 & 1 & \boxed{0} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{array}$$

We can construct a row that is not in the table by inverting the diagonal elements.

Recursively Enumerable $\neq$ All Languages

Theorem: If S is an infinite countable set, then its powerset is not countable.

Proof: Let $S = \{x_1, x_2, x_3, \ldots\}$ be an infinite countable set.

Let $S_1, S_2, S_3, \ldots$ be any sequence of subsets of S.

Consider the subset $S' = \{x_i : x_i \notin S_i\}$.

Diagonalization. Let $A[i][j] = 1$ iff $x_j \in S_i$. Let $B[i] = 1$ iff $A[i][i] \neq 1$.

S' differs from every S_i, therefore $S_1, S_2, S_3, \ldots$ cannot enumerate all subsets of S.

We can construct a set like S' for any sequence $S_1, S_2, S_3, \ldots$; therefore, the powerset of S is uncountable.

Theorem: Some languages are not recursively enumerable.

Proof: The set of strings is an infinite countable set.

The set of languages is not countable because it is the powerset of the set of strings.

Recursively enumerable languages are countable because TMs are countable.

Therefore, recursively enumerable languages $\subset$ all languages.

Recursive $\neq$ Recursively Enumerable

Theorem: There exists a recursively enumerable language that is not recursive.

Proof: Let $M_1, M_2, M_3, \ldots$ be an enumeration of TMs.

Let $x_1, x_2, x_3, \ldots$ be an enumeration of inputs.

Consider the language:
$L = \{x_i : x_i \text{ is accepted by } M_i\}$

Note that $\overline{L}$ (the complement of L) is a diagonalization.

L is recursively enumerable (we enumerate M_i and run the UTM on M_i and x_i).

$\overline{L}$ is not accepted by any TM, so $\overline{L}$ is not recursively enumerable.

L is not recursive because $\overline{L}$ is not recursive.