RECURSIVE ENUMERABLE LANGUAGES

Recursive Enumerable (RE) or Type -0 Language

RE languages or type-0 languages are generated by type-0 grammars. An RE language can be accepted or recognized by Turing machine which means it will enter into final state for the strings of language and may or may not enter into rejecting state for the strings which are not part of the language. It means TM can loop forever for the strings which are not a part of the language. RE languages are also called as Turing recognizable languages.

Recursive Language (REC)

A recursive language (subset of RE) can be decided by Turing machine which means it will enter into final state for the strings of language and rejecting state for the strings which are not part of the language. e.g.; $L= \{a^n b^n c^n | n >= 1\}$ is recursive because we can construct a turing machine which will move to final state if the string is of the form $a^n b^n c^n$ else move to non-final state. So the TM will always halt in this case. REC languages are also called as Turing decidable languages. The relationship between RE and REC languages can be shown in Figure 1.
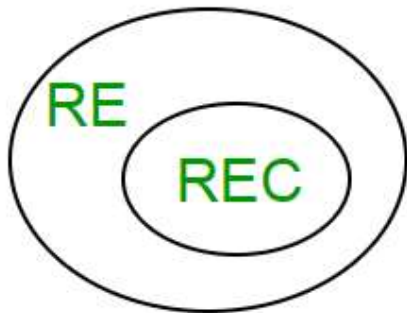

Figure 1

Closure Properties of Recursive Languages

- **Union**: If L1 and If L2 are two recursive languages, their union L1∪L2 will also be recursive because if TM halts for L1 and halts for L2, it will also halt for L1∪L2.
- **Concatenation:** If L1 and If L2 are two recursive languages, their concatenation L1.L2 will also be recursive. For Example:
- $L1= \{a^n b^n c^n | n >= 0\}$
- $L2= \{d^m e^m f^m | m >= 0\}$
- L3= L1.L2
- $= \{a^n b^n c^n d^m e^m f^m | m >= 0 \text{ and } n >= 0\}$ is also recursive.

L1 says n no. of a's followed by n no. of b's followed by n no. of c's. L2 says m no. of d's followed by m no. of e's followed by m no. of f's. Their concatenation first matches no. of a's, b's and c's and then matches no. of d's, e's and f's. So it can be decided by TM.

- **Kleene Closure:** If L1is recursive, its kleene closure L1* will also be recursive. For Example:

     L1= $\{a^n b^n c^n | n >= 0\}$

     L1*= $\{a^n b^n c^n || n >= 0\}$* is also recursive.

- **Intersection and complement**: If L1 and If L2 are two recursive languages, their intersection L1 ∩ L2 will also be recursive. For Example:

- L1= $\{a^n b^n c^n dm | n >= 0 \text{ and } m >= 0\}$

- L2= $\{a^n b^n c^n d^n | n >= 0 \text{ and } m >= 0\}$

- L3=L1 ∩ L2

- = $\{a^n b^n c^n d^n | n >= 0\}$ will be recursive.

L1 says n no. of a's followed by n no. of b's followed by n no. of c's and then any no. of d's.

L2 says any no. of a's followed by n no. of b's followed by n no. of c's followed by n no. of d's. Their intersection says n no. of a's followed by n no. of b's followed by n no. of c's followed by n no. of d's. So it can be decided by turing machine, hence recursive.

Similarly, complement of recursive language L1 which is ∑*-L1, will also be recursive.