Enumerating Sets

A language is said to be **recursive** if there is a TM which halts on all inputs and accepts the language. The notions corresponding to r.e. sets and recursive sets for functions are partial recursive and total recursive functions resp.

A central notion is enumerating a set (or making a list of its elements). This is not usually formally defined, but nevertheless very important. I will give a formal definition here. First, note that an infinite set is countable iff it is in 1-1 correspondence with the integers. Countability is thus a necessary condition for enumerability. We will also require that the process of making the list should not get into cycling. More precisely, we say that a set $S$ can be enumerated (or “we can make a list of the set”) iff there is a **total recursive function** $f : \mathbb{N} \to S$ which is onto. So, the “list” $\{f(1), f(2), f(3), \ldots\}$ (i) contains all elements of $S$ (Note: an element may appear more than once !) and (ii) does not contain any elements not in $S$. The fact that $f$ is total recursive means that there is an enumeration process, which makes any finite initial piece of the list in finite time. An equivalent (why ?) definition for infinite sets $S$ is: $S$ can be enumerated iff there is a TM which prints out strings, so that (i) any element in $S$ is printed out at some finite time and (ii) only strings in $S$ are ever printed out. Some important enumerable sets are:

(a) $\{(i, j) : i, j \in \mathbb{N}\}$ can be enumerated.

(b) For any finite alphabet $\Sigma$, $\Sigma^*$ is enumerable.

(c) Any r.e. set is enumerable: Run through a list of pairs $(i, j)$ (see a) and for each pair, if the $i$ th string - $w_i$ (see b) is accepted by the TM in exactly $j$ steps, then “print out” $w_i$.

(d) The set of all TM’s is enumerable.

(e) The set of all polynomial time bounded TM’s is enumerable. (We will see this later).