An enumerator is a 2-tape TM with an attached printer and a set $P \subseteq Q$ of "print" states. Whenever the enumerator enters a print state, the contents of its second tape are sent to the printer. A language L is enumerable if there exists an enumerator M such that, on input ε (the empty string), M prints every element of L (possibly with repetitions). Note that if L is infinite, then M will never halt.

1. Prove that Σ^* is enumerable.

Solution: Number the symbols of Σ in some arbitrary order: $\sigma_1, \sigma_2, \ldots, \sigma_m$. The enumerator will print all strings of length 0, then all strings of length 1, and so on. For a particular value k, the method for going through the strings is analogous to incrementing a base-k counter. Initially M prints ε (the empty string), and then writes σ_1 on its print tape. It then does the following:

1. start at the left-most non-blank symbol
2. while the current symbol is σ_m do
3.   replace the current symbol with σ_1 and move right
4. if the current symbol is σ_i (not $\sqcup$) then
5.   replace the current symbol with σ_{i+1} and print
6.   goto 1
7. if the current symbol is $\sqcup$ then
8.   replace the current symbol with σ_1 and print
9.   goto 1

2. Prove that a language is recognizable if and only if it is enumerable.

Solution: ("if") Suppose L is enumerable, and let M be the corresponding enumerator. We will define a TM M' that recognizes L. Let M' be a TM that does the following: on input w, M' runs M on input ε. Every time M prints a string (i.e. enters a print state), M' compares the printed string with w, and accepts if they are identical. If $w \in L$ then M will eventually print w (since M enumerates L), and so M' will eventually accept w. On the other hand, if $w \notin L$ then M will never print w, so M' will never accept w. So $L(M) = L$.

("only if") Suppose L is recognizable, i.e. $L = L(M)$ for some TM M. Let $s_1, s_2, s_3, \ldots$ be an enumeration of Σ^* (Note: we showed in part 1 that Σ^* is enumerable, so what really mean here is that we can run through every string in Σ^* by running an enumerator for Σ^* and using each of the strings that it prints). Define an enumerator E for L as follows.

1. for $t \leftarrow 1, 2, 3, \ldots, \infty$ do
2.   for $j \leftarrow 1, 2, \ldots, t$ do
3.   simulate M on s_j for t steps
4.   if M accepts then print s_j

Since the amount of work done for each value of t is finite, every value of t will eventually be reached. Suppose $w \in L = L(M)$, and let i be the minimum index such that $w = s_i$. We must show that E eventually outputs w. Let $t_M(w)$ be the number of steps M takes to accept w, and let $T = \max\{t_M(w), i\}$. Then in the iteration where t takes the value T, E will simulate M on $w = s_i$ for at least $t_M(w)$ steps, and thus E will print w in this iteration. On the other hand, E will never print a string not in L, since it only prints a string after M has accepted it.