TUTORIAL – 3B

(TURING MACHINES AND UNRESTRICTED GRAMMARS)

Problem-1

Design Turing Machines (TMs) as well as unrestricted grammars for the following languages:

$L_1 = \{ w \in \{a,b,c\}^* \mid \#a(w) = \#b(w) = \#c(w) \}$

$L_2 = \{ ww \mid w \in \{a,b\}^* \}$

$L_3 = \{ a^i b^j c^k d^l \mid i=k \text{ and } j=l \}$

$L_a = \{ a^n w c^n \mid w \in \{a,b,c\}^*, n \geq 0, \text{ and } \#a(w) = n \}$

$L_b = \{ a^n w c^n \mid w \in \{a,b,c\}^*, n \geq 0, \text{ and } \#b(w) = n \}$

$L_c = \{ a^n w c^n \mid w \in \{a,b,c\}^*, n \geq 0, \text{ and } \#c(w) = n \}$

Problem-2

Consider the unrestricted grammar over the singleton alphabet $\Sigma = \{a\}$, having the start symbol S, and with the following productions.

$S \rightarrow AS \mid aT, \quad Aa \rightarrow aaaA, \quad AT \rightarrow T, \quad T \rightarrow \varepsilon$

What is the language generated by this unrestricted grammar? Justify.

Problem-3

Prove that, a language L is recursive if and only if there is an enumeration machine enumerating the strings of L in a non-decreasing order of length (string of the same length may be arranged in lexicographic order). For example, strings of {0, 1}* would be arranged as 0, 1, 00, 01, 10, 11, 000, 001, 010, 011, 100, 101, 110, 111, ...

Problem-4

Prove that any grammar, defined over non-terminals N and terminals Σ, can be converted to an equivalent grammar with rules of the form, $\alpha A \gamma \rightarrow \alpha\beta\gamma$ for $A \in N$ and $\alpha, \beta, \gamma \in (\Sigma \cup N)^*$.

THANK YOU !