TD series n°2

Exn°1

- First Grammar : an Unrestricted Grammar (Type 0)
- Second " : Regular Grammar (Type 3)
- Third " : Context - Free Grammar (Type 2)

(You can explain why? using definitions and rules).

G1: $S \xRightarrow{①} abcA$ or $S \xRightarrow{3} Aabc$, both words $\notin L(G)$ it contain non terminal symbol (derivation in 1 step)

- derivation in 2-steps: $S \xRightarrow{1} abcA \xRightarrow{3} abc \in L(G)$

The Same Thing for G2 and G3

$\varepsilon \notin L(G)$ for The Three grammars.

- 2 derivation of The first grammar for generating the word abcabcabc

a) $S \xRightarrow{①} abcA \xRightarrow{5} abcS \xRightarrow{2} abcAabc \xRightarrow{5} abcSabc$
$\xRightarrow{①} abcabcAabc \xRightarrow{3} abcabcabc.$

b) $S \xRightarrow{2} Aabc \xRightarrow{4} Sabc \xRightarrow{1} abcAabc \xRightarrow{5} abcSabc$
$\xRightarrow{①} abcabcAabc \xRightarrow{3} abcabcabc.$

Specification textual (natural lge) of L(G):

- All the strings in abc in any order (power)

formally $L(G) = \{(abc)^n, n \geq 1\}$

- All the string of 1 possible terminated by 0, or strip 0

$L(G) = \{0, 1^n, 1^n0, n \geq 1\}$

- All of arithmetic expressions use + and * on variable a in prefixe notation (notation polonaise)

$L(G) = \{(*+)^n a^{n+1}, n \geq 0\}$