AUTOMATA AND FORMAL LANGUAGES UACS012

TUTORIAL - 6

[BHAGYA VINOD RANA]

CFG

U19CS012

1) What is Context Free Grammer (CFG)? List down the applications of CFG.

1) CFG stands for Context Free Grammer : It is a Formal grammer which is used to generate all possible patterns of strings in given formal language.

Context-Free Grammer G can be defined by four tuples as:

$G = (V, T, P, S)$, where G: Grammer, which consist of set of production rule [used to generate string] ($V \cap T = \phi$)

T = finite ~~final~~ set of terminal symbol. [denoted by lower case letters]

V = finite set of Non-terminal symbol [Capital Letters]

P = Set of Production Rules (Substitution Rules) which is used for replacing non-terminals symbols (left side of prod^n) in string with other terminal or non-terminal symbols. (right side)

S = start symbol which is used to derive the string.

Type 2 Grammer: $\alpha \longrightarrow \beta$, α ↓ only one variable, β $(V \cup T)^*$

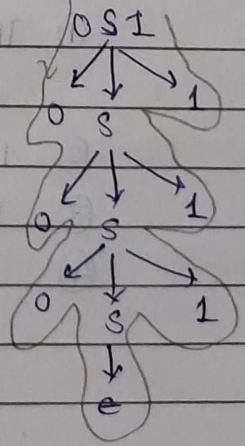
Eg: $S \rightarrow 0S1 \mid \epsilon$ CFG

It has generated $\{0^n 1^n, n \geq 0\}$ language.

$0S1$
↓
$0S1$
↓
$0S1$
↓
$0S1$
↓
ϵ

$\Rightarrow 00001111$

Tree



Applications of C.F.G.

(a) For construction of compilers
(b) For defining and translation of programming languages
(c) For parsing the program by constructing syntax tree
(d) For describing arithematic expression.
(e) Development of XML (Extensible Markup Language)
(DTD (Document type Definition) → CFG

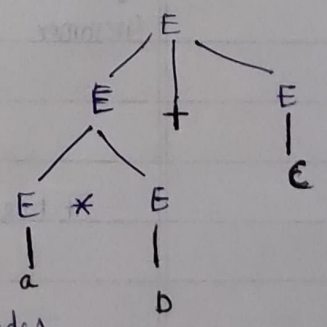
2.) Define the following terms:

a) Derivation
The process of deriving a string is called derivation.

b) Parse Tree (derivation tree)
The geometric representation of a derivation is called parse tree.
parse tree follows precedence of operators.
The deepest sub-tree is traversed first (i.e. operator in parent node has less precedence over the operator in sub-tree).

(E = a|b|c)

⊛ Properties: Rules: E=E+E, E=E×E, Eg: a × b + c



① The Root node is always a node indicating start symbol.
② The derivation is read from left to right.
③ The leaf nodes is always terminal nodes.
④ The interier nodes are always non-terminal nodes.

3.)

$E \rightarrow E + T \mid E - T \mid T$

$T \rightarrow T \times F \mid T \div F \mid F$

$F \rightarrow G \uparrow F \mid G$

$G \rightarrow I$

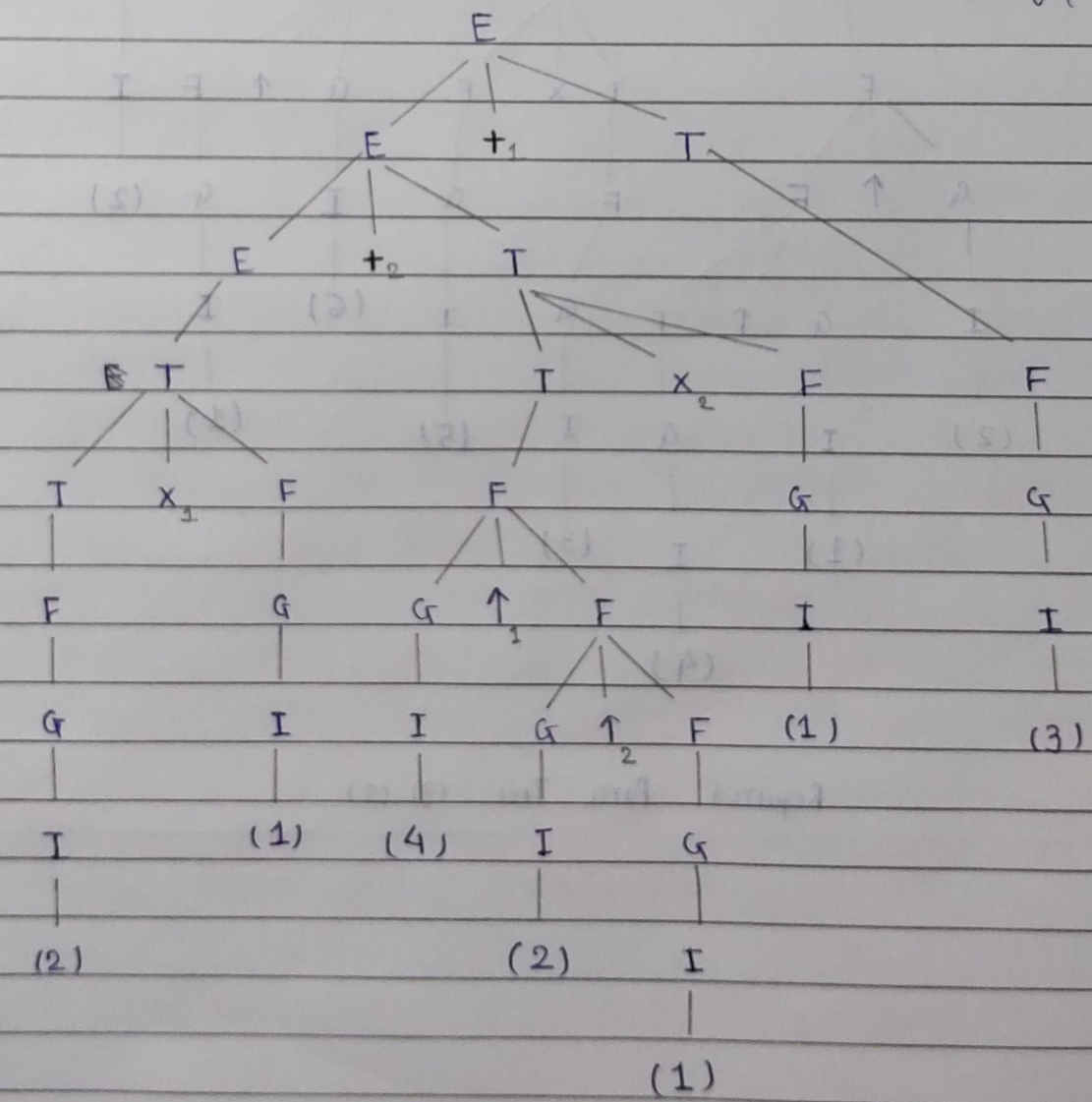
$I \rightarrow 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

} Grammer

Draw the parse tree for the string:

1.) $\left((2 \times_1 1) +_2 \left((4 \uparrow_1 (2 \uparrow_2 1)) \times_2 1\right)\right) +_1 3$

priority $\{I > \uparrow > (\times / \div) > (+$

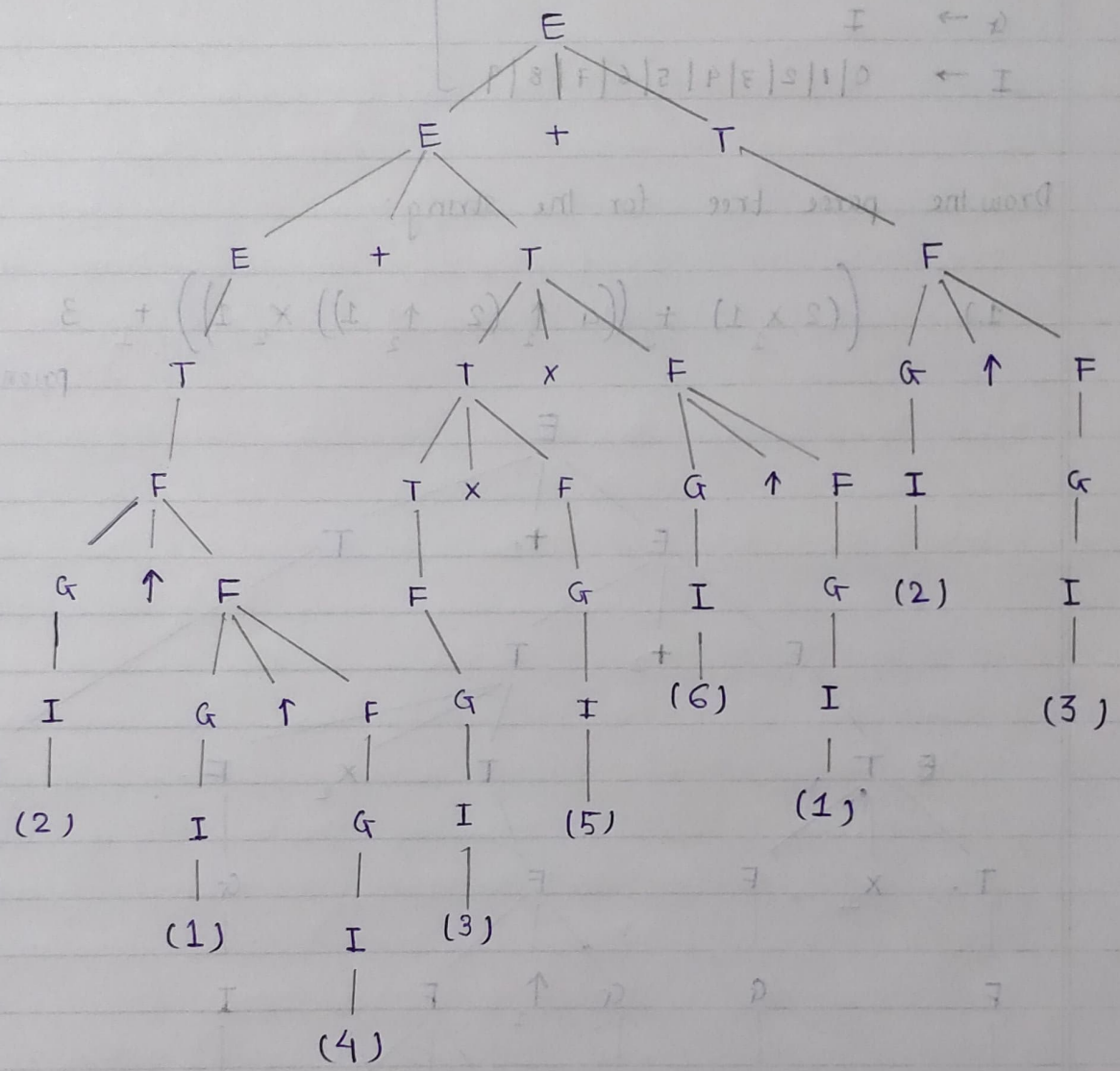


Required Parse Tree (3.(a))

U19CS012

2) $2 \uparrow 1 \uparrow 4 + 3 \times 5 \times 6 \uparrow 1 + 2 \uparrow 3$

$= ((2 \uparrow (1 \uparrow 4)) + ((3 \times 5) \times (6 \uparrow 1)) + (2 \uparrow 3)$



Required Parse Tree (3).(2)

4.) $E \rightarrow E+T \mid T$

$T \rightarrow F \times T \mid F$

$F \rightarrow I$

$I \rightarrow 0|1|2|3|4|5|6|7|8|9$

Derive the string : $2+3\times5\times6+2$

	Using
$E \rightarrow E+T$	$E \rightarrow E+T$
$\rightarrow \overbrace{E+T}+T$	$E \rightarrow E+T$
$\rightarrow (\widetilde{T})+(\widetilde{F\times T})+T$	$E \rightarrow T$, $T \rightarrow F\times T$
$\rightarrow (\widetilde{F})+F\times \widetilde{F\times T}+F$	$T \rightarrow F$, $T \rightarrow F\times T$, $I \rightarrow F$
$\rightarrow F+F\times F\times F+F$	$I \rightarrow F$
$\rightarrow I+I\times I\times I+I$	$F \rightarrow I$
$\rightarrow 2+3\times5\times6+2$	$I \rightarrow 2$, $I \rightarrow 3$, $I \rightarrow 5$, $I \rightarrow 6$

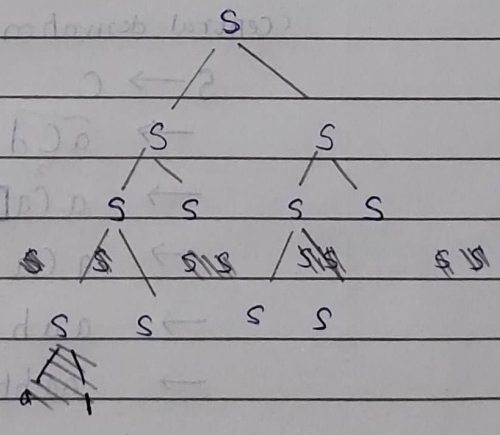
Hence, the given string is derived using given grammer.

5.) What are different possible strings for the grammer

1) $S \rightarrow SS$

$S \rightarrow a$

$S \rightarrow b$



$S \rightarrow SS$ (even length)

$\rightarrow (SS)S$ (odd length)

Any length string can be formed ($\gamma = 2$)

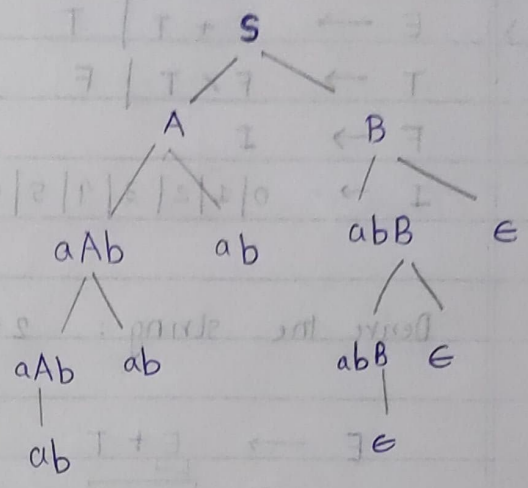
S→ can be either a or b. $L = (a+b)^2 (a+b)^*$ (Regular Exp)

∴ This grammer can generate every possible combination of string comprising of letters (a or b) of length greater than or equal to 2 Eg: L={aa, ab, ba, bb, aba, abb, ... }

2) $S \to A \mid B$

$A \to aAb \mid ab$

$B \to abB \mid \in$



$A \to$ aaa... bbb...

$a^n b^n$, $n \geq 1$

$B \to (ab)^n$, $n \geq 0$

This grammer can generate string of type

[$a^n b^n$, $(n \geq 1)$ OR $(ab)^n$ where $(n >= 0)$]

---

6) Give the left most derivation of string $w = aabbccdd$

$S \to AB \mid C$

$A \to aAb \mid ab$

$B \to cBd \mid cd$

$C \to aCd \mid aDd$

$D \to bDc \mid bc$

(Central derivation)

$S \to C$

$\to aCd$

$\to a(aDd)d$

$\to a(a(bDc)d)d$

$\to aab(bc)cdd$

$\to aabbccdd$

Using

$S \to C$

$C \to aCd$

$C \to aDd$

$D \to bDc$

$D \to bc$

Left most derivation ⇒

$S \to AB$

$\to aAb\,B$

$\to aabb\,B$

$\to aabb\,cBd$

$\to aabbccdd$

Ans

Using

$S \to AB$

$A \to aAb$

$A \to ab$

$B \to cBd$

$C \to cd$