Autumn-22

Ans to ques no: 1(a)

CFG for regular expression $(0+1)^*01^*$:

$L = \{0, 001, 101, 00111, 10111, \ldots\}$

CFG:

$S \rightarrow A0B$

$A \rightarrow 0A \mid 1A \mid \varepsilon$

$B \rightarrow 1B \mid \varepsilon$

Generate 00111:

$S \rightarrow A0B$
$\rightarrow 0A0B$
$\rightarrow 0\varepsilon 0B$
$\rightarrow 001B$
$\rightarrow 0011B$
$\rightarrow 00111B$
$\rightarrow 00111\varepsilon \rightarrow 00111$. Ans:

Or

CFG for regular expression $0^*1(0+1)^*$

$L = \{1, 10, 010, 0011, 00011, \ldots\}$

CFG:

$S \rightarrow A1B$

$A \rightarrow 0A \mid \varepsilon$

$B \rightarrow 0B \mid 1B \mid \varepsilon$.

Generate 00010:

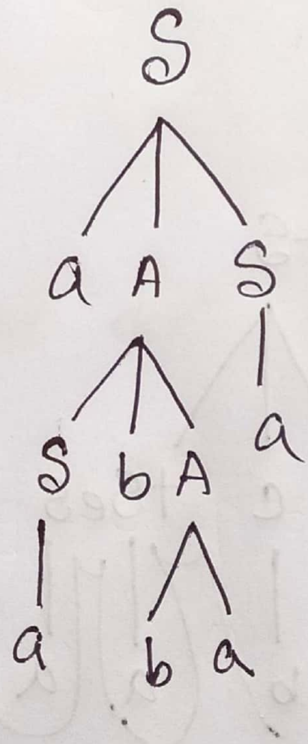
$S \rightarrow A1B$
$\rightarrow 0A1B$
$\rightarrow 00A1B$
$\rightarrow 000A1B$
$\rightarrow 000\varepsilon 1B$
$\rightarrow 00010B$
$\rightarrow 00010\varepsilon \rightarrow 00010$: Ans

Ans to the ques no: 1(b).

$S \rightarrow aAS \mid a$

$A \rightarrow SbA \mid SS \mid ba$.

Show that $S \Rightarrow aabbaa$ by constructing a derivation tree ~~tool~~ by rightmost derivation, whose yeild is aabbaa



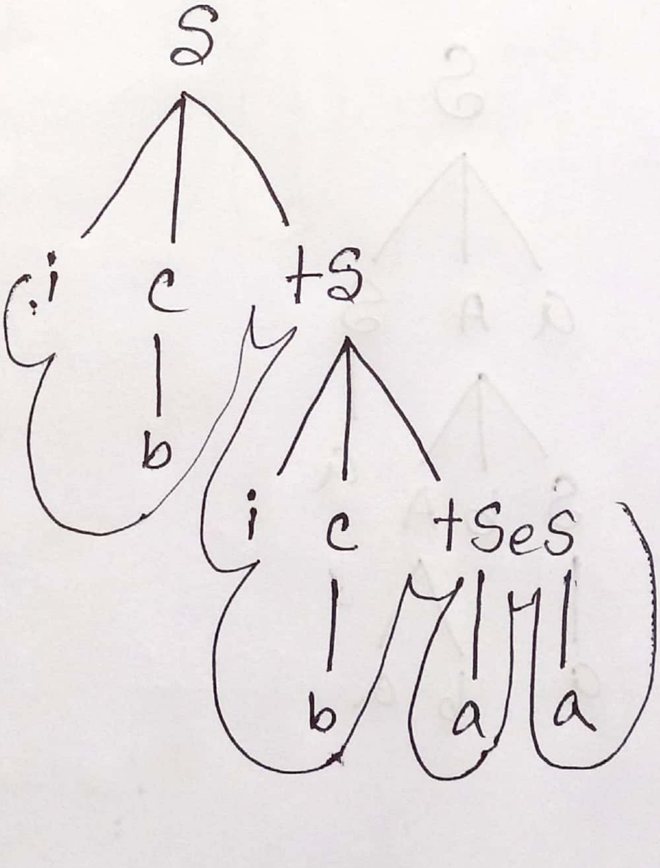
Ans to ques no : 1(C).

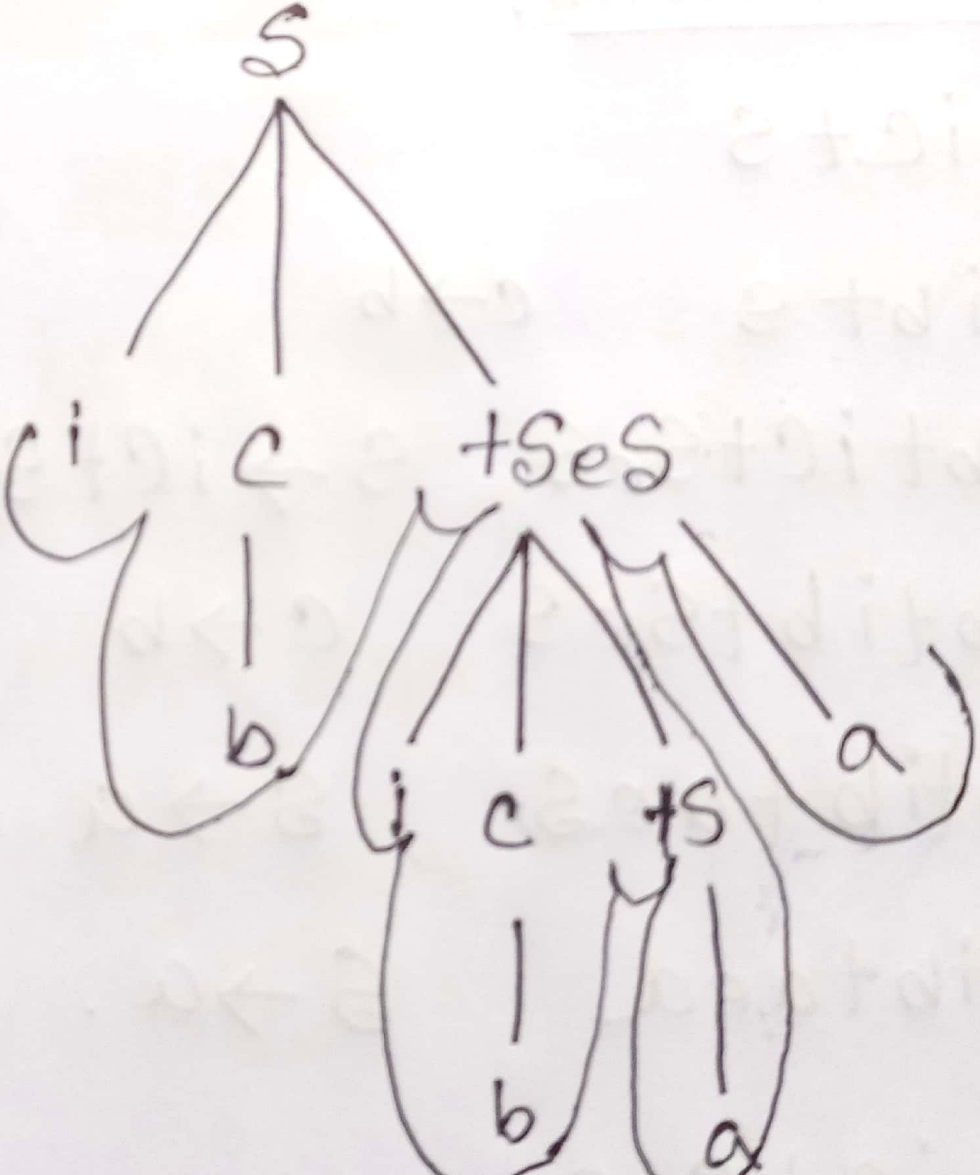
$S \rightarrow iCtS \mid iCtSeS \mid a$

$C \rightarrow b$

Show in particular that the string 'ibtibtaea' has two:

i) parse trees:




S
i
c
tSeS
b
i
c
tS
a
b
a

Leftmost derivation:

11)

1) $S \rightarrow iCtS$

$\rightarrow ibtS \quad C \rightarrow b$

$\rightarrow ibtiCtSeS \quad S \rightarrow iCtSeS$

$\rightarrow ibtibtSeS \quad C \rightarrow b$

$\rightarrow ibtibtaeS \quad S \rightarrow a$

$\rightarrow ibtibtaea \quad S \rightarrow a.$

2) $S \rightarrow iCtSeS$

$\rightarrow ibtSeS \quad C \rightarrow b$

$\rightarrow ibtiCtSeS \quad S \rightarrow iCtS$

$\rightarrow ibtibtSeS \quad C \rightarrow b$

$\rightarrow ibtibtaeS \quad S \rightarrow a$

$\rightarrow ibtibtaea \quad S \rightarrow a.$

iii) Rightmost derivations:

1) $S \to ictS$

$\to ictictSeS$ $\quad S \to ictSeS$

$\to ictictSea$ $\quad S \to a$

$\to ictictaea$ $\quad S \to a$

$\to ictibtaea$ $\quad c \to b$

$\to ibtibtaea.$ $\quad c \to b$

2) $S \to ictSeS$

$\to ictSea$ $\quad S \to a$

$\to ictictSea$ $\quad S \to ictS$

$\to ictictaea$ $\quad S \to a$

$\to ictibtaea$ $\quad c \to b$

$\to ibtibtaea$ $\quad c \to b$

2(a)or,

A regular language can be described using a context free grammar.

I can represents a regular language with a context-free grammar. Here's an example:

Let, Regular language that accepts string of a's and b's where the number of a's is equal to the the number of b's. i.e.,

$$L = \{a^n b^n \mid n \geq 0\}$$

context-free grammar:

- ~~$S \to \varepsilon / aS$~~
- $S \to \varepsilon \mid aSb$

The production rules allow you to generate strings in which a's and b's are balanced, such as $\{\varepsilon, ab, aabb, aaabbb \ldots\}$

2) b) Can you give a CFG for the following languages over the alphabet Σ ={a,b}

all strings in the languages, L = {a^n b^2n c^4n | n ≥ 0}

if you can not, justify the reason.

Sol:

No, it is not possible to define the language L = {a^n b^2n c^4n | n ≥ 0} using a context-free grammar (CFG).

Context-free grammars are not powerful enough to generate languages that have a direct relationship between the counts of different symbols. In this case, the number of 'a's is related to the number of 'b's and 'c's in a specific pattern (n, 2n, 4n). A CFG can generate languages where the counts of two symbols are related (e.g., a^n b^n), but it cannot handle relationships between three or more symbols.

To generate the language L = {a^n b^2n c^4n | n ≥ 0}, you would need a more powerful grammar formalism like a context-sensitive grammar or a Turing machine.

2(c)

Construct a CFG for $L = \{a^n b^{2n} \mid n \geq 0\}$

$L = \{\varepsilon, abb, aabbbb, aaabbbbbb, \ldots\}$

$S \rightarrow \varepsilon$

$S \rightarrow aSbb$

Now,

Using $R = \{S \rightarrow aSbb \mid \varepsilon\}$ derive "aabbbb"

$S \Rightarrow aSbb$

$\Rightarrow a\underline{aSbb}\,bb$

$\Rightarrow aabbbb$

All non-empty strings that read the same from left or right.

$$\underbrace{a(a+b)^*}_{A} a + b \underbrace{(a+b)^*}_{A} b + a + b$$

$S \longrightarrow aAb \mid bAb \mid a \mid b$

$A \longrightarrow aA \mid bA$

OR

Context-free grammar can be made simpler by removing all the extraneous symbols while yet preserving a converted grammar that is equivalent to the original grammar.

1. Each variable (i.e, non-terminal) and terminal of G is used to derive some word in language L.
2. There should not be any production like $X \to Y$ when X and Y are non-terminal.
3. If ε is not in L, then the production $X \to \varepsilon$ is unnecessary.

Procedure for eliminating unit productions from a CFG. are given below –

Step 1 : To remove $X \to Y$, add production $X \to a$ to the grammar rule whenever $Y \to a$ occurs in the grammar.

Step 2 : Now delete $X \to Y$ from the grammar.

Step 3 : Repeat step 1 and 2 until all unit Productions are removed.

Remove the unit productions from the following grammar :

$S \to AB$

$A \to a$

$B \to C \mid b$

$C \to D$

$D \to E$

$E \to a$

Step - 1 : Remove unit production,

$E \rightarrow a$

As, $D \rightarrow E$

$D \rightarrow a$

As, $C \rightarrow D$

$C \rightarrow a$

As, $B \rightarrow C$

$B \rightarrow a$

~~Now~~, So, $A \rightarrow AB$

$A \rightarrow a$

$B \rightarrow a \mid b$

$C \rightarrow a$

$D \rightarrow a$

$E \rightarrow a$

Step - 2 : Remove unreachable state,

$A \rightarrow AB$

$A \rightarrow a$

$B \rightarrow a \mid b$

Ans. to the Q. No – 5(a)

$S \to aSb \mid bY \mid Ya$

$Y \to bY \mid aY \mid c \mid \varepsilon$

Step : 1

$S_0 \to S$

$S \to aSb \mid bY \mid Ya$

$Y \to bY \mid aY \mid c \mid \varepsilon$

Step : 2 Remove the Null Production.

$Y \to \varepsilon$

Removing $Y \to \varepsilon$.

$S_0 \to S$

$S \to aSb \mid bY \mid Ya \mid b \mid a$

$Y \to bY \mid aY \mid c \mid b \mid a$.

Step : 3 Remove the Unit Production,

$S_0 \to S$

Removing, $S_0 \to S$

$S_0 \to aSb \mid bY \mid Ya \mid b \mid a$

$S \to aSb \mid bY \mid Ya \mid b \mid a$

$Y \to bY \mid aY \mid c \mid b \mid a$

Step : 4 $\begin{cases} A \rightarrow a \\ A \rightarrow AB \end{cases}$

$S_0 \rightarrow aSb \mid bY \mid Ya \mid b \mid a$

$S \rightarrow aSb \mid bY \mid Ya \mid b \mid a$

$Y \rightarrow bY \mid aY \mid c \mid b \mid a$

Step : 5

Let,

$P \rightarrow ab$

$Q \rightarrow b$

$D \rightarrow a$

$S_0 \rightarrow PS \mid QY \mid YD \mid b \mid a$

$S \rightarrow PS \mid QY \mid YD \mid b \mid a$

$Y \rightarrow QY \mid DY \mid c \mid b \mid a$

$P \rightarrow ab$

$D \rightarrow a$

$Q \rightarrow b$

OR,

Ans. to. the. Q. No - 5(a) OR

$S \to aXbX$

$X \to aY | bY | \varepsilon$

$Y \to X | c$

Step-1

$S \to aXbX$

$X \to aY | bY | \varepsilon$

$Y \to X | c$

Step-2 Remove the null production,

$X \to \varepsilon$

$Y \to \varepsilon$

Removing $X \to \varepsilon$,

$S \to aXbX | abX | aXb | ab$

$X \to aY | bY$

$Y \to X | c | \varepsilon$

Removing $Y \to \varepsilon$,

$S \to aXbX | abX | aXb | ab$

$X \to aY | bY | a | b$

$Y \to X | c$

Step: 3 Remove the Unit Production,

$Y \to X$,

$S \to aXbX \mid abX \mid aXb \mid ab$

$X \to aY \mid bY \mid a \mid b$

$Y \to aY \mid bY \mid a \mid b \mid c$.

Step: 4: $A \to a$, $B \to b$,

$S \to AXBX \mid ABX \mid AXB \mid AB$

$X \to AY \mid BY \mid a \mid b$

$Y \to AY \mid BY \mid a \mid b \mid c$

$A \to a$

$B \to b$.

Step: 5: let, $D \to AX$, $E \to BX$,

$S \to DE \mid AE \mid DB \mid AB$

$X \to AY \mid BY \mid a \mid b$

$Y \to AY \mid BY \mid a \mid b \mid c$

$A \to a$

$B \to b$

$D \to AX$

$E \to BX$.