Top Down Parsing

Q1. Consider the following grammar and test whether the grammar is LL(1) or not.

S→aBDh , B→cC, C→bC|ε, D→EF, E→g|f, F→f|ε.

Q2. Generate Top Down Parser for following Grammar

E -->E + T | T

T -->id | id[] | id[X]

X -->E , E | E

Q3. Generate top down parser for following grammar and parse string aa+a*

S -> SS+ | SS* | a

Q4. Create M table of following grammar

S-> Pt
P-> C | B
B-> H : T
H->Cd | H : d | epsilon
T-> Se | S : T
C->bT | epsilon

Q.5 Find the FIRST and FOLLOW set for the following Grammar? Check whether the grammar is LL(1) or not. If yes, then construct a Predictive parsing table for the grammar.

$S \rightarrow [SX]/[X]/S[X]$ $X \rightarrow \in/id, id/id,$ $Y \rightarrow \in/id$	$S \rightarrow iEtSS_1/a$ $S_1 \rightarrow eS/\in$ $E \rightarrow b$	$S \rightarrow AaAb/BbBa$ $A \rightarrow \in$ $B \rightarrow \in$
$S \rightarrow aBDh$ $B \rightarrow cC$ $C \rightarrow bC/\in$ $D \rightarrow EF$ $E \rightarrow g/\in$ $F \rightarrow f/\in$ Parse the string: acbgfh	$S \rightarrow 1AB/\in$ nhu $A \rightarrow 1AC/0C$ $B \rightarrow 0S$ $C \rightarrow 1$	$S \rightarrow aABb$ $A \rightarrow c/\in$ $B \rightarrow d/\in$
S->ABCDE A->a/ϵ B->b/ϵ C->c D->d/ϵ E->e/ϵ	$D \rightarrow L:T$ $L \rightarrow L, id/id$ $T \rightarrow int$	$S' \rightarrow S\#$ $S \rightarrow qABC$ $A \rightarrow a/bbD$ $B \rightarrow a/\in$ $C \rightarrow b/\in$ $D \rightarrow c/\in$

Other Questions:

Q1. For the grammar A→(A)A|ε. Compute FIRST and FOLLOW set of A.

Q2. Check whether the following grammars are ambiguous or not? If yes, write its equivalent unambiguous grammar.

$S \rightarrow A$ $A \rightarrow A + A / B + +$ $B \rightarrow y$	S→ $aSbS/bSaS/\in$	bExp→bExp or bExp bExp→bExp and bExp bExp→not bExp bExp→true bExp→false
$R \rightarrow R + R$ $R \rightarrow R.R$ $R \rightarrow R*$ $R \rightarrow a$ $R \rightarrow b$ $R \rightarrow c$		

Q.3: Eliminate left recursion and (or) then left factoring (if any) from the following grammar:

$Q \rightarrow QED/q$ $E \rightarrow f$ $D \rightarrow DfAfA/nFA/d$ $N \rightarrow DfA/n$ $A \rightarrow a$	$E \rightarrow EE +$ $E \rightarrow EE*$ $E \rightarrow id$	$S \rightarrow (T) / a$ $T \rightarrow T,S / S$
$S \rightarrow iEtS/iEtSeS/a$ $E \rightarrow b$	$S \rightarrow bSSaaS/bSSaSb/bSb/a$	

Operator Precedence Parsing

Q1. Consider the following grammar and generate operator precedence relation table by finding leading and trailing. **Where terminals are { '+' 'int' '(' ')' 'id' '=' '/' } and non terminals are {S,V,A,E,F}**

S -> VAE

E-> F | E+F

F->V | int | (E)

V-> id

A-> = | /

Q2: Check whether the following grammars are operator grammar or not? If yes, then disambiguate the above grammar and generate a operator-precedence parser, using Leading and Trailing method.

$E \rightarrow EAE/(E)/id$ $X \rightarrow +/-/*/\%$	$E \rightarrow E+T/T$ $T \rightarrow T*F/F$ $F \rightarrow (E)\ /\ id$	