I. Parsing Stack

Given grammar:

1) S→(S) S| ε

Show the *parsing stack*, the *input*, and the *actions* of an LL(1) parser at each step in the recognition of the input string **().**

Parsing stack	Input	Action
$ S	( ) $	**Replace** S→(S) S
$ S) S (	( ) $	**Match**
$ S) S	) $	**Replace** S→ε
$ S)	) $	**Match**
$ S	$	**Replace** S→ε
$	$	**Accept**

2)

Show the *parsing stack*, the *input*, and the *actions* of an LL(1) parser at each step in the recognition of the input string **x-2*y$**.

E → T E'
E' → + T E' | - T E' | ε
T → F T'
T' → * F T' | / F T' | ε
F → num | id

Parsing stack	Input	Action
$ E	**x-2*y$**	**Replace E → T E'**
$ *E'T*	**x-2*y$**	**Replace T → F T'**
$ *E'T'F*	**x-2*y$**	**Replace F → id**
$ *E'T'id*	**x-2*y$**	**Match**
$ *E'T'*	**-2*y$**	**Replace** T'→ε
$ *E'*	**-2*y$**	**Replace E' → - T E'**
$ *E'T* -	**-2*y$**	**Match**
$ *E'T*	**2*y$**	**Replace T → F T'**
$ *E'T' F*	**2*y$**	**Replace F → num**
$ *E'T' num*	**2*y$**	**Match**
$ *E'T'*	***y$**	**Replace T' → * F T'**
$ *E'T' F**	***y$**	**Match**
$ *E'T' F*	**y$**	**Replace F → id**

$ *E'T' id*	**y$**	**Match**
$ *E'T'*	**$**	**Replace T' → ε**
$ *E'*	**$**	**Replace E' → ε**
$	**$**	**Accept**

II. First and Follow Sets

Grammar	First Set	Follow Set
S → A B | z **A → x | ε** **B → y | ε**	First(*S*) = { x, y, z, ε } First(*A*) = { x, ε } First(*B*) = { y, ε }	Follow(S) = {$} Follow(A) = {y,$} Follow(B) = {$}

Grammar	First Set	Follow Set
S→ABCDE	First(S) ={a,b,c}	Follow(S) ={$}
A→a|ε	First(*A*) ={a, ε }	Follow(A)= {b,c}
B→b|ε	First(*B* ={b, ε }	Follow(B)= {c}
C→c	First(C)= {c}	Follow(C)={d,e,$}
D→d|ε	First(D) ={d, ε }	Follow(D)= {e,$}
E→e|ε	First(E)= {e, ε }	Follow(E) ={$}

Grammar	First Set	Follow Set
S→AB|BC| ε|aSABC	First(S) ={a,b, c,ε }	Follow(S) ={$,a,b,c}
A→aAa|ε	First(*A*) ={a, ε }	Follow(A)= {a,b,c,$}
B→bB|ε	First(*B* ={b, ε }	Follow(B) ={$,a,b,c}
C→cC|ε	First(C)= {c, ε}	Follow(C)= {$,a,b,c}

Grammar	First Set	Follow Set
E→TE'	First(E)={id,(}	Follow(E)= {$,)}
E'→ +TE'|ε	First(E')= {+, ε }	Follow(E') ={$,)}
T→FT'	First(T)= {id,(}	Follow(T) ={+,$,)}
T'→*FT'|ε	First(T')= {*, ε }	Follow(T')= {+,$,)}
F→id|(E)	First(F) ={id,(}	Follow(F) ={*,+,$,) }

III. Parsing Tables

1) Given the following grammar, first set, and follow set construct LL(1)Parsing Table

Grammar	First Set	Follow Set
E→TE’	First(E)={id,(}	Follow(E)= {$,)}
E’→ +TE’|ε	First(E’)= {+, ε }	Follow(E’) ={$,)}
T→FT’	First(T)= {id,(}	Follow(T) ={+,$,)}
T’→*FT’|ε	First(T’)= {*, ε }	Follow(T’)= {+,$,)}
F→id|(E)	First(F) ={id,(}	Follow(F) ={*,+,$,) }

Solution:

	id	+	*	(	)	$
E	E→TE’			E→TE’		
E’		E’→ +TE’			E’→ ε	E’→ ε
T	T→FT’			T→FT’		
T’		T’→ε	T’→*FT’		T’→ε	T’→ε
F	F→id			F→(E)		

2) Given the following grammar and first set and follow set construct LL(1)Parsing Table

Grammar	First Set	Follow Set
E →numE’ E’→E A|ε A→+E’|*E’	First(E)={num} First(E’)={num,ε} First(A)={+,*}	Follow(E)= {$,+,*} Follow(E’)= {$,+,*} Follow(A)= {$,+,*}

	Num	+	*	$
E	E →numE’			
E’	E’→E A	E’→ε	E’→ε	E’→ε
A		A→+E’	A→*E’	

IV. IS the following grammar LL(1)

1) S→aSbS |bSaS |ε
   NO.It's not LL(1)
   Because:
   1. it is ambiguous grammar
   2. Multiple entry in LL(1) Parsing Table
      First (S)={a,b,ε} Follow(S)={a,b,$}.
      We have two productions rule in T[S,a] and in T[S,b]

2)

S→aABb — One alternative no problem
A→c|ε — {c}{d,b}
B→d|ε — {d}{b}
Yes, it is LL(1)

3)

S→ A|a
A→a

NO.It's not LL(1)
Multiple entry in LL(1) Parsing Table
We have two productions rule in T[S,a]
First(S)={a} and {a}