

Compulsory Question 1:**Question 1. [10 marks]**

Consider the following grammar:

$S \rightarrow \text{Session } \$$

$\text{Session} \rightarrow \text{Facts Question}$

$\text{Session} \rightarrow (\text{Session}) \text{Session}$

$\text{Facts} \rightarrow \text{Fact Facts}$

$\text{Facts} \rightarrow$

$\text{Fact} \rightarrow ! \text{string}$

$\text{Question} \rightarrow ? \text{string}$

- Calculate nullable, FIRST and FOLLOW for this grammar: [5]
- Construct the LL(1) parsetable. [4]
- Why is this grammar a LL(1) grammar? Give reasons for your answer by referring to your LL(1) parsetable. [1]

Answer:

- nullable, FIRST and FOLLOW:

	nullable	FIRST	FOLLOW	
S	no	(? !		✓
Session	no	(? !)\$	✓
Fact	no	!	! ?	✓
Facts	yes	!	!	✓
Question	no	?)\$	✓

- LL(1) parsing table

	()	\$!	?	string
S	$S \rightarrow \text{Session } \$$			$S \rightarrow \text{Session } \$$	$S \rightarrow \text{Session } \$$	
Session	$\text{Session} \rightarrow (\text{Session}) \text{Session}$			$\text{Session} \rightarrow \text{Facts Question}$	$\text{Session} \rightarrow \text{Facts Question}$	
Fact				$\text{Fact} \rightarrow ! \text{string}$		
Facts				$\text{Facts} \rightarrow \text{Fact Facts}$ $\text{Facts} \rightarrow$		
Question					$\text{Question } ? \text{string}$	

1/2 mark for each correct entry, max 4 marks.

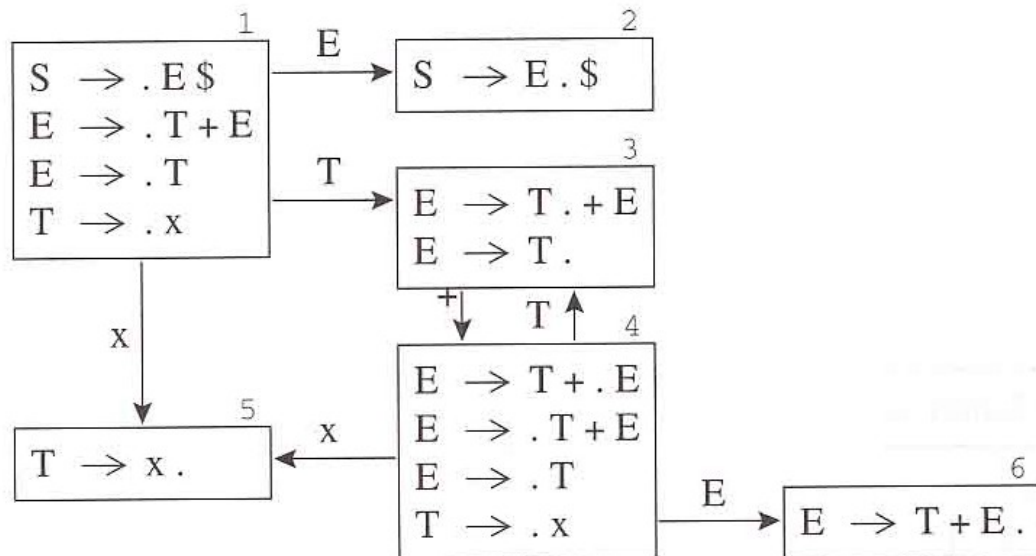
c) This is not a LL(1) grammar because there are duplicate entries in the LL(1) parsing table. (1mark)

Choose between Question 2 and Question 3 below:

Question 2 [10 marks]

a) Construct the LR(0) and SLR parsetables for the following grammar and its given LR(0)-DFA: [10]

$0 \quad S \rightarrow E \$$
 $2 \quad E \rightarrow T$
 $1 \quad E \rightarrow T + E$
 $3 \quad T \rightarrow x$



Answer:

LR(0) Parsetable

	x	+	\$	<i>E</i>	<i>T</i>
1	s5			g2	g3
2			a		
3	r2	s4,r2	r2		
4	s5			g6	g3
5	r3	r3	r3		
6	r1	r1	r1		

One mark for each correct row (excluding accept)

SLR Parse Table

	x	+	\$	<i>E</i>	<i>T</i>
1	s5			g2	g3
2			a		
3		s4	r2		
4	s5			g6	g3
5		r3	r3		
6			r1		

One mark for each correct row (excluding accept)

Question 3. [10 marks]

- 1) What is the difference between a Concrete Parse Tree and an Abstract Parse Tree?
[2]

Answer : Concrete Parse Tree – represents the concrete syntax of the source language. Abstract Syntax – conveys the phrase structure of the source program with all parsing issues resolved but without any semantic interpretation.

- 2) Consider the grammar below describing the abstract syntax of expressions.

$E \rightarrow E + E$

$E \rightarrow E - E$

$E \rightarrow E * E$

$E \rightarrow E / E$

$E \rightarrow \text{id}$

$E \rightarrow \text{num}$

Describe the Visitor Pattern. Supply the java code for the grammar above to illustrate this design pattern [8]

Answer: A visitor implements an interpretation – an object that contains a visit method for each syntax tree class. Each syntax tree class contains an accept method. An accept method serves as a hook for interpretations. It is called by a visitor class, and passes control back to an appropriate method in the visitor class. It allows the addition of a new interpretation without editing and recompiling existing classes, providing that each of the appropriate classes has an accept method, etc

Mark code on discretion.