Dr B R Ambedkar National Institute of Technology, Jalandhar
B.Tech. (CSE) VII Sem
CSPC-401- System Programming and Compiler Design

1. Indicate each of the following statements whether true or false, Justify your answer.
   a) **True,** the semantic analyzer verifies the parse tree to ensure that it is meaningful and consistent with the rules of the programming language. It checks for semantic errors, such as type mismatches, undeclared variables, or incorrect function arguments, which cannot be detected during syntactic analysis.
   b) **True,** every SLR(1) grammar is LR(0), but the reverse is not necessarily true. SLR(1) grammars consider follow sets to resolve conflicts, making them more powerful than LR(0) grammars, which do not. Therefore, an LR(0) grammar may not always be SLR(1).
   c) **True,** Nondeterministic Finite Automata (NFA) and Deterministic Finite Automata (DFA) are equivalent in terms of the languages they can recognize. For every NFA, there exists a corresponding DFA that accepts the same language. This is established by the subset construction algorithm, which can convert an NFA to an equivalent DFA.
   d) **True,** A DFA cannot count arbitrarily large numbers, including matching pairs of parentheses beyond a fixed limit. While it is possible to construct a DFA for a fixed number of pairs of parentheses (e.g., exactly three pairs).
   e) **True,** this is because NFAs recognize regular languages, while context-free grammars can describe more complex languages, such as those with nested structures (e.g., balanced parentheses). Regular languages, which NFAs recognize, are a subset of context-free languages, but not all context-free languages are regular.

2. **True,** it groups the characters into meaningful tokens like keywords, identifiers, operators, or literals based on predefined rules. White spaces, operators, and special symbols act as boundaries that help the lexer separate tokens.
   a) 19 different tokens
   b) A grammar is typically defined as a 4-tuple: G = (N, Σ, P, S)
      i. **N**: A finite set of **non-terminal symbols** (variables).
      ii. **Σ**: A finite set of **terminal symbols.**
      iii. **P**: A finite set of **production rules**.
      iv. **S**: A **start symbol**
   c)
      i) The leftmost derivation for the string **"aaabbabbba"** is:

      S⇒aB
      ⇒aaBB
      ⇒aaaBBB
      ⇒aaabBB
      ⇒aaabbB
      ⇒aaabbaBB

⇒aaabbabB
⇒aaabbabbS
⇒aaabbabbbA
⇒aaabbabbba

ii) The rightmost derivation for the string **"aaabbabbba"** is:

S ⇒aB

⇒aaBB

⇒aaBbS

⇒aaBbbA

⇒aaBbba

⇒aaaBBbba

⇒aaaBbSbba

⇒aaaBbaBbba

⇒aaaBbabbba

⇒aaabbabbba

3. **Eliminate the left recursion from grammar(G1):**
   G1:
   E → EOT / T
   O → + / -
   T → TPF / F
   P → *
   F → (E) / id

   **Step 1:** Eliminate Left Recursion for E
   E→TE′
   E′→OTE′|ϵ
   **Step 2:** Eliminate Left Recursion for T
   T→FT′
   T′→PFT′|ϵ

**Final Transformed Grammar**

After eliminating left recursion, we have the following grammar:

E→TE′

E′→OTE′ | ϵ

O→+ | −

T→FT′

T ′→PFT′|ϵ

P→∗

F→(E) | id

**Eliminate the left factoring from the grammar(G2):**

G2:

S → f / fc / fcd / fcde

**Solution**: After eliminating left factoring, the resulting grammar G2 is:

1. S→fS′
2. S′→ϵ | cS″
3. S″→ ϵ | dS"'
4. S''' → ϵ | e

**Ques(4): Rules for first set**

1. If X is terminal, then FIRST(X) is {X}.
2. If X => Ɛ is a production, then add Ɛ to FIRST(X).
3. If X is nonterminal and X => Y1 Y2 ... Y*k.* is a production, then place *a* in FIRST(X) if for some *i, a* is in FIRST(Y*i*), and Ɛ is in all of FIRST(Y1), ... , FIRST(Y*i*-1); that is, Y1, ... ,Y*i*-1 => Ɛ. If Ɛ is in FIRST(Y*j*) for all *j* = 1, 2, ... *, k,* then add Ɛ to FIRST(X).

For example, everything in FIRST(Y1) is surely in FIRST(X). If Y1 does not derive Ɛ, then we add nothing more to FIRST(X), but if Y1 => Ɛ, then we add FIRST(Y2) and so on.

**Rules for follow set**

1. Place $ in FOLLOW(S), where S is the start symbol and $ is the input right end marker.
2. If there is a production A => α Bβ, then everything in FIRST(β), except for Ɛ, is placed in FOLLOW(B).
3. If there is a production A => α B, or a production A => α Bβ where FIRST(β) contains Ɛ (i.e., β => Ɛ), then everything in FOLLOW(A) is in FOLLOW(B).

First and Follow for the Grammar G1.

G1:

E → EOT / T

O → + / -

T → TPF / F

P → *

F → (E) / id

**First Sets:**

- First(E)={(,id}
- First(O)={+,−}
- First(T)={(,id}
- First(P)={∗}
- First(F)={(,id}

**Follow Sets:**

- Follow(E)={+,-,$,)}
- Follow(O)={(, id}
- Follow(T)={+,−,∗,),$}
- Follow(P)={(, id}
- Follow(F)={+,−,∗,),$}

# Agumented Grammar $G'$

$E' \longrightarrow E$
① $E \longrightarrow EOT$
② $E \longrightarrow T$
③ $O \longrightarrow +$
④ $O \longrightarrow -$
⑤ $T \longrightarrow TPF$
⑥ $T \longrightarrow F$
⑦ $P \longrightarrow *$
⑧ $F \longrightarrow (E)$
⑨ $F \longrightarrow id$

**$I_0$**
$E' \to \cdot E$
$E \to \cdot EOT$
$E \to \cdot T$
$T \to \cdot TPF$
$T \to \cdot F$
$F \to \cdot (E)$
$F \to \cdot id$

**$I_1$** Goto $(I_0, E)$
$E' \to E \cdot$
$E \to E \cdot OT$
$O \to \cdot +$
$O \to \cdot -$

**$I_2$** Goto $(I_0, T)$
$E \to T \cdot$
$T \to T \cdot PF$
$P \to \cdot *$

**$I_3$** Goto $(I_0, F)$
$T \to F \cdot$

**$I_4$** Goto $(I_0, ()$
$F \to (\cdot E)$
$E \to \cdot EOT$
$E \to \cdot T$
$T \to \cdot TPF$
$T \to \cdot F$
$F \to \cdot (E)$, $F \to \cdot id$

**$I_5$** Goto $(I_0, id)$
$F \to id \cdot$

**$I_6$** Goto $(I_1, O)$
$E \to EO \cdot T$
$T \to \cdot TPF$
$T \to \cdot F$
$F \to \cdot (E)$
$F \to \cdot id$

**$I_7$** Goto $(I_1, +)$
$O \to + \cdot$

**$I_8$** Goto $(I_1, -)$
$O \to - \cdot$

**$I_9$** Goto $(I_2, P)$
$T \to TP \cdot F$
$F \to \cdot (E)$
$F \to \cdot id$

**$I_{10}$** Goto $(I_2, *)$
$P \to * \cdot$

**$I_{11}$** Goto $(I_4, E)$
$F \to (E \cdot)$
$E \to E \cdot OT$
$O \to \cdot +$
$O \to \cdot -$

**$I_2$** Goto $(I_4, T)$
$E \to T \cdot$
$T \to T \cdot PF$
$P \to \cdot *$

Goto $(I_4, F)$
**$I_3$** $T \to F \cdot$

Goto $(I_4, ()$
**$I_4$**
$F \to (\cdot E)$
$E \to \cdot EOT$
$E \to \cdot T$
$T \to \cdot TPF$
$T \to \cdot F$
$F \to \cdot (E)$
$F \to \cdot id$

**$I_5$** Goto $(I_4, id)$
$F \to id \cdot$

**$I_{12}$** Goto $(I_6, T)$
$E \to EOT \cdot$
$T \to T \cdot PF$
$P \to \cdot *$

**$I_3$** Goto $(I_6, F)$
$T \to F \cdot$

Goto $(I_6, ()$
**$I_4$**

Goto $(I_6, id)$
**$I_5$**

**$I_{13}$** Goto $(I_9, F)$
$T \to TPF \cdot$

Goto $(I_9, ()$ = **$I_4$**

Goto $(I_9, id)$ = **$I_5$**

Goto $(I_{11}, ))$
**$I_{14}$** $F \to (E) \cdot$

Goto $(I_{11}, O)$
**$I_6$**
$E \to EO \cdot T$
$T \to \cdot TPF$
$T \to \cdot F$
$F \to \cdot (E)$
$F \to \cdot id$

Goto $(I_{11}, +)$ = **$I_7$**

Goto $(I_{11}, -)$ = **$I_8$**

Goto $(I_{12}, P)$
**$I_9$**
$T \to TP \cdot F$
$F \to \cdot (E)$
$F \to \cdot id$

Goto $(I_{12}, *)$ = **$I_{10}$**

Total Item Sets from $I_0$ to $I_{14}$ → 15

| | + | − | * | id | ( | ) | $ | E | O | T | P | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $I_0$ | | | | $S_9$ | $S_8$ | | | 1 | | 2 | | 3 |
| $I_1$ | $S_3$ | $S_4$ | | | | | Accept | | 6 | | | |
| $I_2$ | $r_2$ | $r_2$ | $S_7$ $r_2$ | $r_2$ | $r_2$ | $r_2$ | $r_2$ | | | | 9 | |
| $I_3$ | $r_6$ | $r_6$ | $r_6$ | | | $r_6$ | $r_6$ | | | | | |
| $I_4$ | | | | $S_9$ | $S_8$ | | | 11 | | 2 | | 3 |
| $I_5$ | $r_9$ | $r_9$ | $r_9$ | | | $r_9$ | $r_9$ | | | | | |
| $I_6$ | | | | $S_9$ | $S_8$ | | | | | 12 | | 3 |
| $I_7$ | | | | $r_3$ | $r_3$ | | | | | | | |
| $I_8$ | | | | $r_4$ | $r_4$ | | | | | | | |
| $I_9$ | | | | $S_9$ | $S_8$ | | | | | | 13 | |
| $I_{10}$ | | | | $r_7$ | $r_7$ | | | | | | | |
| $I_{11}$ | $S_3$ | $S_4$ | | | | $S_8$ | | 6 | | | | |
| $I_{12}$ | $r_1$ | $r_1$ | $S_7$ $r_1$ | | | $r_1$ | $r_1$ | | | 9 | | |
| $I_{13}$ | $r_5$ | $r_5$ | $r_5$ | | | $r_5$ | $r_5$ | | | | | |
| $I_{14}$ | | | | $r_8$ | $r_8$ | | | | | | | |

Given Grammar is not LR(0) ∵ there are two S/r conflict

But $G_1$ is SLR(1)

6(a):- $E \rightarrow T+E \mid T$

$T \rightarrow a$

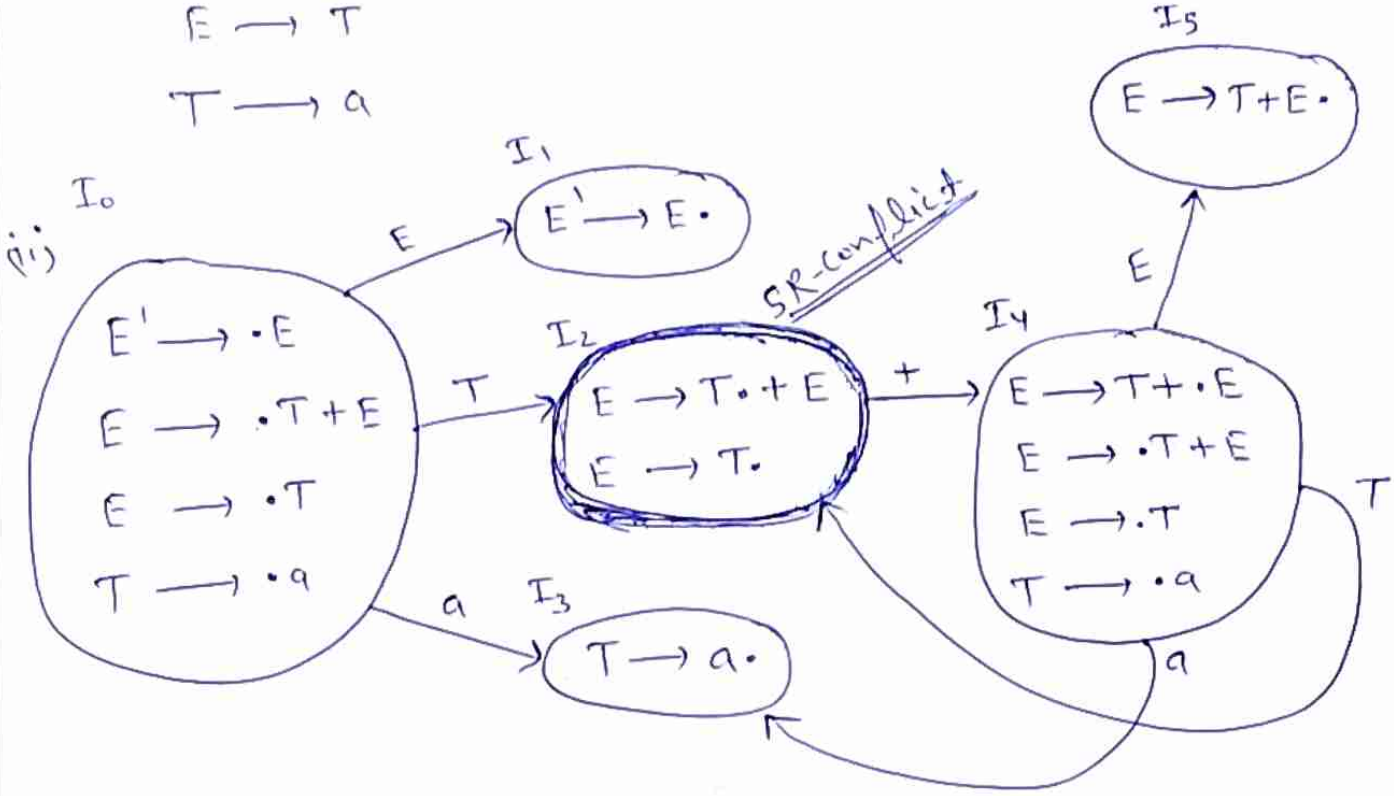
(i) $E' \rightarrow E$

$E \rightarrow T+E$

$E \rightarrow T$

$T \rightarrow a$

(ii)




$I_2$ : Contain SR conflict so it is not LR(0) grammar.

6(b):- $E \rightarrow E+T \mid T$

$T \rightarrow a$

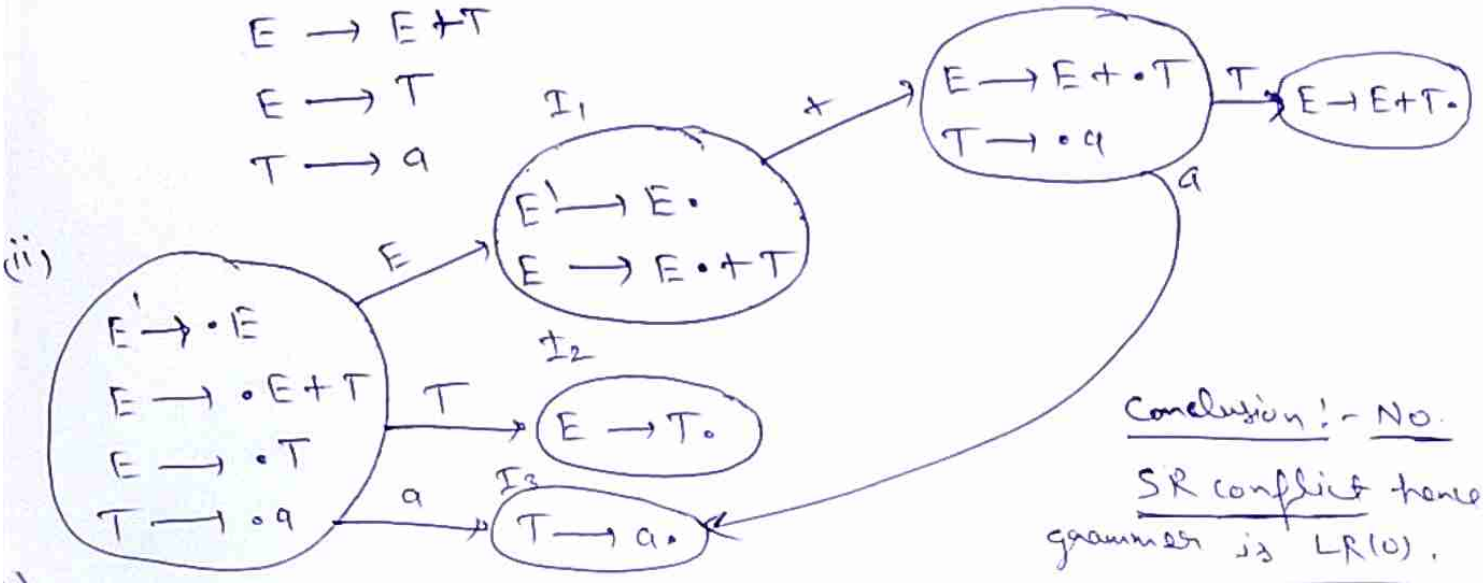
(i) $E' \rightarrow E$

$E \rightarrow E+T$

$E \rightarrow T$

$T \rightarrow a$

(ii)




Conclusion :- No.

SR conflict hence grammar is LR(0).