

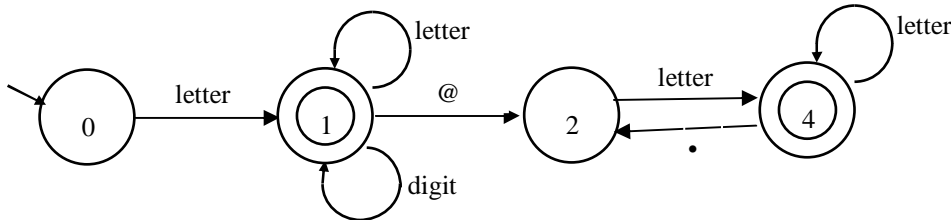
CA448 Compiler Construction 1 Repeat Examination Answers 2009

1. $\text{letter (letter | digit)^* ((@ \text{letter letter}^* (. \text{letter letter}^*)^*) | \epsilon)$

[10 marks]

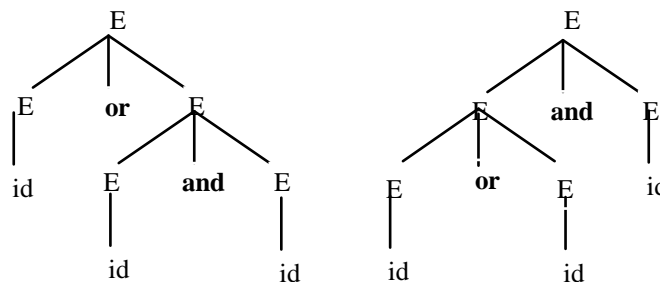
2

[10 marks]



3. The following parse trees are produced for the expression: id or id and id:

[10 marks]



Therefore the grammar is ambiguous.

4.

[10 marks]

$$\begin{aligned} \text{LOOKAHEAD}(S \rightarrow ABC) &= (\text{FIRST}(A) - \{\epsilon\}) \cup (\text{FIRST}(B) - \{\epsilon\}) \cup \text{FIRST}(C) \cup \text{FOLLOW}(S) \\ &= \{a, c\} \cup \{b\} \cup \{c\} \cup \{\$\} = \{a, b, c, \$\} \end{aligned}$$

$$\text{LOOKAHEAD}(A \rightarrow aA) = \text{FIRST}(a) = \{a\}$$

$$\text{LOOKAHEAD}(A \rightarrow C) = (\text{FIRST}(C) - \{\epsilon\}) \cup \text{FOLLOW}(A) = \{c\} \cup \{b, c, \$\} = \{b, c, \$\}$$

$$\text{LOOKAHEAD}(B \rightarrow bB) = \text{FIRST}(b) = \{b\}$$

$$\text{LOOKAHEAD}(B \rightarrow \epsilon) = \text{FOLLOW}(B) = \{c, \$\}$$

$$\text{LOOKAHEAD}(C \rightarrow cC) = \text{FIRST}(c) = \{c\}$$

$$\text{LOOKAHEAD}(C \rightarrow \epsilon) = \text{FOLLOW}(C) = \{b, c, \$\}$$

The grammar is not LL(1) since c predicts both $C \rightarrow cC$ and $C \rightarrow \epsilon$

5. Left factoring gives:

[10 marks]

```

VarDec → var Declist
Declist → Decl Declist'
Declist' → Declist
Declist' → ε
Decl → IdList : id
IdList → IdList , id
IdList → id
    
```

Eliminating left-recursion gives:

VarDec \rightarrow var Declist
Declist \rightarrow Dec Declist '
Declist ' \rightarrow Declist
Declist ' \rightarrow ϵ
Dec \rightarrow IdList : id
IdList \rightarrow id IdList '
IdList ' \rightarrow , id IdList '
IdList ' \rightarrow ϵ

6. The following LR(0) items are produced for this grammar:

[10 marks]

1: S \rightarrow \bullet Ab
S \rightarrow \bullet B
A \rightarrow \bullet aB
B \rightarrow \bullet a
B \rightarrow \bullet aA
2: S \rightarrow A \bullet b
3: S \rightarrow Ab \bullet
4: S \rightarrow B \bullet
5: A \rightarrow a \bullet B
B \rightarrow a \bullet
B \rightarrow a \bullet A
B \rightarrow \bullet a
B \rightarrow \bullet aA
A \rightarrow \bullet aB
6: A \rightarrow aB \bullet
7: B \rightarrow aA \bullet

There is a shift-reduce conflict in state 5, so the grammar is not LR(0).

7. The following LR(0) items are produced for this grammar:

[10 marks]

1: S \rightarrow \bullet A
A \rightarrow \bullet Ba
A \rightarrow \bullet C
B \rightarrow \bullet aC
B \rightarrow \bullet b
C \rightarrow \bullet B
2: S \rightarrow A \bullet
3: A \rightarrow B \bullet a
C \rightarrow B \bullet
4: A \rightarrow C \bullet
5: B \rightarrow a \bullet C
C \rightarrow \bullet B
B \rightarrow \bullet aC
B \rightarrow \bullet b

- 6: $B \rightarrow b\bullet$
- 7: $A \rightarrow Ba\bullet$
- 8: $B \rightarrow aC\bullet$
- 9: $C \rightarrow B\bullet$

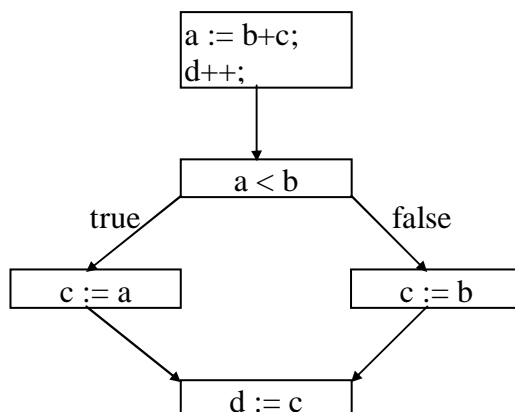
$FOLLOW(C) = \{a, \$\}$. There is therefore a shift-reduce conflict in state 3, so the grammar is not SLR(1).

8. The following LR(1) items are produced for this grammar: [10 marks]

- 1: $S \rightarrow \bullet A, \$$
 $A \rightarrow \bullet Ba, \$$
 $A \rightarrow \bullet C, \$$
 $C \rightarrow \bullet B, \$$
 $B \rightarrow \bullet aC, a\$$
 $B \rightarrow \bullet b, a\$$
- 2: $S \rightarrow A\bullet, \$$
- 3: $A \rightarrow B\bullet a, \$$
 $C \rightarrow B\bullet, \$$
- 4: $A \rightarrow C\bullet, \$$
- 5: $B \rightarrow a\bullet C, a\$$
 $C \rightarrow \bullet B, a\$$
 $B \rightarrow \bullet aC, a\$$
 $B \rightarrow \bullet b, a\$$
- 6: $B \rightarrow b\bullet, a\$$
- 7: $A \rightarrow Ba\bullet, \$$
- 8: $B \rightarrow aC\bullet, a\$$
- 9: $C \rightarrow B\bullet, a\$$

There are no conflicts, so the grammar is LR(1).

9. [10 marks]



10. The attributes in the grammar could be evaluated in the following sequence:

[10 marks]

1. Bitlist₀.pos
2. Bitlist₁.pos
3. Bit₀.pos
4. Bitlist₂.pos
5. Bit₁.pos
6. Bitlist₃.pos
7. Bit₂.pos
8. Bit₃.pos
9. Bit₃.val
10. Bitlist₃.val
11. Bit₂.val
12. Bitlist₂.val
13. Bit₁.val
14. Bitlist₁.val
15. Bit₀.val
16. Bitlist₀.val
17. Num.val