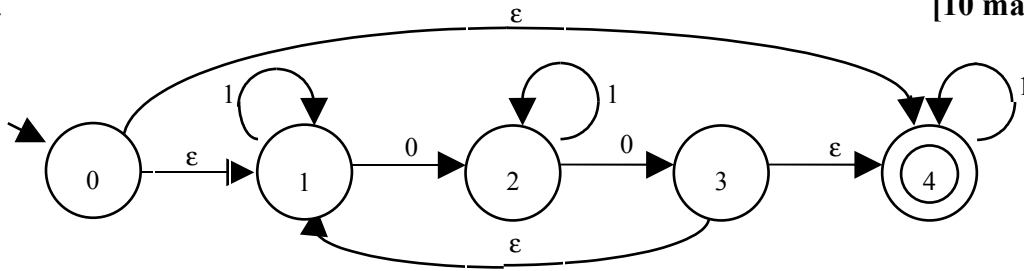


CA448 Compiler Construction 1 Repeat Examination Answers 2008

1. [10 marks]

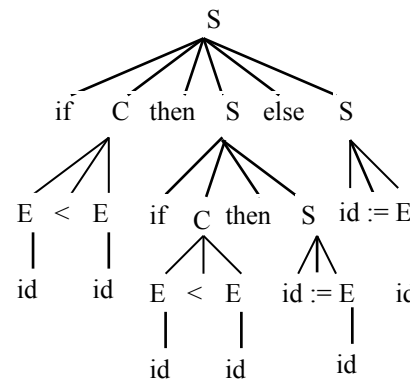
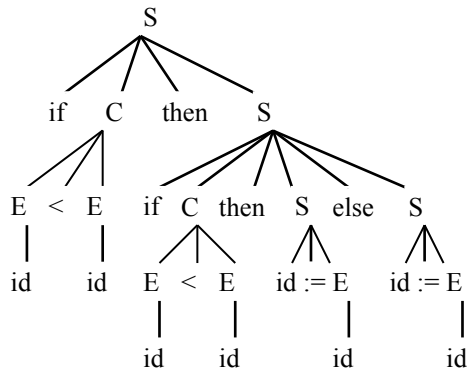


2. [10 marks]

State	0	1
A = {0,1,4}	B	C
B = {2}	D	B
C = {1,4}	B	C
D = {1,3,4}	B	C

The start state is A and the accepting states are A, C and D.

3. The following parse trees are produced for the given expression: [10 marks]



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

$$\begin{aligned}
 \text{LOOKAHEAD}(A \rightarrow aA) &= \text{FIRST}(a) \\
 &= \{a\}
 \end{aligned}$$

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

$$\begin{aligned}
 \text{LOOKAHEAD}(B \rightarrow bB) &= \text{FIRST}(b) \\
 &= \{b\}
 \end{aligned}$$

$$\begin{aligned}
 \text{LOOKAHEAD}(B \rightarrow \epsilon) &= (\text{FIRST}(\epsilon) - \{\epsilon\}) \cup \text{FOLLOW}(B) \\
 &= \{\} \cup \{c, \$\} = \{c, \$\}
 \end{aligned}$$

$$\begin{aligned}
 \text{LOOKAHEAD}(C \rightarrow cC) &= \text{FIRST}(c) \\
 &= \{c\}
 \end{aligned}$$

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

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

5. The converted grammar is as follows:

[10 marks]

Stat \rightarrow with Desig do Stat end
Stat \rightarrow Desig := Exp
Desig \rightarrow id Desig'
Desig' \rightarrow . id Desig'
Desig' \rightarrow ϵ
Exp \rightarrow Desig Exp'
Exp' \rightarrow + Exp
Exp' \rightarrow ϵ

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

[10 marks]

0 : E' \rightarrow •E\$
E \rightarrow •E+T
E \rightarrow •T
T \rightarrow •T*F
T \rightarrow •F
F \rightarrow •id
F \rightarrow •(E)
1 : E' \rightarrow E•\$
E \rightarrow E•+T
2 : E' \rightarrow E\$•
3 : E \rightarrow E+•T
T \rightarrow •T*F
T \rightarrow •F
F \rightarrow •id
F \rightarrow •(E)
4 : T \rightarrow F •
5 : F \rightarrow id•
6 : F \rightarrow (•E)
E \rightarrow •E+T
E \rightarrow •T
T \rightarrow •T*F
T \rightarrow •F
F \rightarrow •id
F \rightarrow •(E)
7 : E \rightarrow T•
T \rightarrow T•*F
8 : T \rightarrow T*•F
F \rightarrow •id
F \rightarrow •(E)
9 : T \rightarrow T*F•
10 : F \rightarrow (E)•
11 : E \rightarrow E+T•
T \rightarrow T•*F

$$12 : F \rightarrow (E\bullet)$$

$$E \rightarrow E\bullet+T$$

There are shift-reduce conflicts in states 7 and 11, so the grammar is not LR(0).

7. FOLLOW(E) = {+,),}\$} **[10 marks]**
 FOLLOW(T) = {+,*,),}\$}
 FOLLOW(T) = {+,*,),}\$}

The shift-reduce conflicts in states 7 and 11 can therefore be resolved: reduce on +, - and \$ and shift on *. The grammar is therefore SLR(1).

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

- 1: S \rightarrow \bullet Ab, \$
 S \rightarrow \bullet B, \$
 A \rightarrow \bullet aB, b
 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
 B \rightarrow a \bullet , \$
 B \rightarrow a \bullet A, \$
 B \rightarrow \bullet a, b
 B \rightarrow \bullet aA, b
 A \rightarrow \bullet aB, \$
 6: A \rightarrow aB \bullet , b
 7: B \rightarrow aA \bullet , \$
 8: A \rightarrow a \bullet B, \$
 B \rightarrow a \bullet , b
 B \rightarrow a \bullet A, b
 B \rightarrow \bullet a, \$
 B \rightarrow \bullet aA, \$
 A \rightarrow \bullet aB, b
 9: A \rightarrow aB \bullet , \$
 10: B \rightarrow aA \bullet , b

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

9. The attributes in the grammar could be evaluated in the following sequence: **[10 marks]**

1. T.type
2. I₁.name
3. I₂.name
4. L.in
5. L₁.in
6. I₁.type
7. I₂.type

10. The three-address code is as follows:

[10 marks]

```
t1 := a + b
t2 := c + d
t3 := e - t2
t4 := t1 - t3
```