KADI SARVA VISHWAVIDHYALAYA

B.E. Semester-VII Examination- November-2016

Subject Code :-CE 701 **Subject Name:-Compiler Design**

Date:- 8/11/2016 **Time:-10:30 AM to 1:30PM** **Total Marks:-70**

Instructions:

1. Answer each section in separate Answer sheet.
2. Use of scientific calculator is permitted.
3. All questions are **Compulsory**.
4. Indicate **clearly,** the options you attempt along with its respective question number.
5. Use the last page of main supplementary of **rough work.**

Section - I

Q-1 (All compulsory)

(A) Explain the working of front end of the compiler with suitable example. [5]

(B) Justify with example: left recursive and non left factored grammars cannot be LL(1) grammars. [5]

(C) Justify the use of symbol table by different phases of the compiler. [5]

OR

(C) Explain the error recovery strategies of parser. [5]

Q-2 Answer the following questions.

(A) Construct a minimum state DFA for the given RE using subset construction: [5]
(a|b)*a(a|b)a*#

(B) Explain Handle and Handle pruning with suitable example. [5]

OR

(A) Construct a minimum state DFA for the given RE using subset construction: [5]
(a|bc*)c? a(a|b)a*#

(B) Explain Input Buffering and sentinels. [5]

Q-3 Answer the following questions.

(A) Draw minimized DFA for given expression using syntax tree method. [5]
a*(a|b)*(a|b)#

(B) Test whether the following grammar is LL(1) or not? Also, construct Predictive Parsing table for the same. [5]

S → ACB | cbB | Ba
A → da | BC
B → g | ϵ
C → h | ϵ

OR

(A) Draw minimized DFA for given expression using syntax tree method. [5]

c*a*(a|b)(a|b)*c#

(B) Test whether given grammar is LL(1) or not? Also, draw predictive parsing table for the same. [5]

P → 1PB | ϵ
Q → 1PC | &C
B → &S
C → 1

Section – II

Q-4 (All compulsory)

(A) Prepare operator precedence table for grammar, [5]

S -> (L) | a
L -> L, S|S

Also, parse the string (a,a).

(B) What is a shift-reduce parser? Explain in detail the conflicts that may occur during shift- reduce parsing. [5]

(C) Explain synthesis and inherited attributes. [5]

OR

(C) What are intermediate code representations? Explain triple, quadruple and Indirect triple with suitable example. [5]

Q-5 Answer the following questions.

(A) Explain any two code optimization techniques with examples. [5]

(B) Check whether following grammar is LALR or not: [5]

S → Aa | bAc | Bc | bBd
A → d
B → d

OR

(A) Construct SLR parsing table for following grammar and state whether it is SLR or not. [5]

S → aBc | bCc | aCd | bBd
B → e
C → e

(B) Explain: i) Copy Propagation ii) Dead-code Elimination [5]

Q-6 Answer the following questions.

(A) Draw DAG and Abstract Syntax Tree for following expression: [5]

c=a*b+d*(-b)/a*b

(B) Construct 3A code and draw flow graph for the following code segment: [5]

```
prod=0; n=5;
for( i=0;i<=n; i++)
{
prod+=a[i]*b[i];
i++;
}
```

OR

(A) Explain activation tree and activation record. [5]

(B) Construct 3A code for following code: [5]

```
prod=0; n=5;
p=10;
for( i=0;i<=n; i++)
{
if(a[i]<p)
{
prod+=a[i]*b[i];
}
else{
prod+=a[i];
}
i++;
}
```

-------- All the Best----------

KADI SARVA VISHWAVIDHYALAYA

B.E. Semester-VII Examination-November-2015

Subject Code:-CE 701 **Subject Name:-Compiler Design**

Date:-20/11/2015 **Time:-10:30 AM to 1:30PM** **Total Marks:-70**

Instructions:

1. Answer each section in separate Answer sheet.
2. Use of scientific calculator is permitted.
3. All questions are **Compulsory**.
4. Indicate **clearly**, the options you attempt along with its respective question number.
5. Use the last page of main supplementary of **rough work**.

Section - I

Q-1 (All compulsory)

(A) **Select the correct option(each carry one mark):** [10]

1. **Which of the following suffices to convert an arbitrary CFG to an LL(1) grammar?**

(a) Removing left recursion alone
(b) Factoring the grammar alone
(c) Removing left recursion and factoring the grammar
(d) None of the single above

2. **Consider a program P that consists of two source modules M1 and M2 contained in two different files. If M1 contains a reference to a function defined in M2 the reference will be resolved at**

a) Edit time
b) Compile time
c) Link time
d) Load time

3. **Given the following expression grammar:**

E -> E * F | F+E | F
F -> F-F | id

which of the following is true?

(a) * has higher precedence than +
(b) – has higher precedence than *
(c) + and — have same precedence
(d) + has higher precedence than *

4. **Pick the machine independent phase of Compiler.**

(a) Lexical analysis
(b) Syntax analysis
(c) Semantic analysis
(d) All of the above

5. **Which of the following is the most powerful parser?**
   (a) SLR
   (b) CLR
   (c) LALR
   (d) Operator Precedence
6. **Predictive parsing can be:**
   (a) Recursive
   (b) Non Recursive
   (c) Constructive
   (d) Both a and b
7. **The type of Conflicts LR(0) parsing can have:**
   (a) Shift-Shift
   (b) Shift-Reduce
   (c) Reduce-Reduce
   (d) Both b and c
8. **Can regular grammar be ambiguous?**
   (a) Always
   (b) Not always, depends on grammar
   (c) Never
9. **Which of the following statements is false?**
   (a) An unambiguous grammar has same left most and right most derivation
   (b) LL(1) parser is a top-down parser
   (c) LALR is more powerful than SLR
   (d) An ambiguous grammar can never be LR(K) for any k
10. **Dynamic linking can cause security concerns because**
   (a) Security is dynamic
   (b) The path for searching dynamic libraries is not known till run time.
   (c) Linking is insecure
   (d) Cryptographic procedures are not available for dynamic linking

(C) What does parsing mean in the context of compilers? Explain with suitable example. [5]

OR

(C) Which data structure is suitable to implement Symbol table? Justify your answer.. [5]

Q-2 Answer the following questions.

(A) Construct a minimum state DFA for the given RE using subset construction: b(a|b)*a(a|b)a*# [5]

(B) Explain the elimination of left recursion and left factoring with suitable example. [5]

OR

(A) Construct a minimum state DFA for the given RE using subset construction: (a|b)? a(a|b)a*# [5]

(B) Explain Error recovery strategy for Lexical analyzer. [5]

Q-3 Answer the following questions.

(A) Draw minimized DFA for given expression using syntax tree method. [5]
(a|b)*a(a|b)*(a|b)#

(B) Construct Predictive Parsing table for following grammar. [5]
S' → S #
S → aXYZ
X → q | bbD
Y → q | ϵ
Z → b | ϵ
D → c | ϵ

OR

(A) Draw minimized DFA for given expression using syntax tree method. [5]
c*a(a|b)*(a|b)#

(B) Test whether given grammar is LL(1) or not? Construct Predictive Parsing table for it. [5]
bexpr → bexpr or bterm | bterm
bterm → bterm and bfactor | bfactor
bfactor → not bfactor | (bexpr) | true | false

Section – II

Q-4 (All compulsory)

(A) Prepare operator precedence table for grammar, [5]
S → xAy | xBy | xAz
A → aS | q
B → q

(B) Test whether given grammar is LL(1) or not? Justify. [5]
S → 1AB | ϵ
A → 1AC |.&C
B → &S
C → 1

(C) Compare the Syntax directed definition with Syntax directed Translation Scheme. Highlight the uses of it. [5]

OR

(C) What are three address code representations? Also explain record structures used for three address code. [5]

Q-5 Answer the following questions.

(A) What is the necessity of optimization in compilation? Can human being optimize a program better than automated compiler? Justify your answer. [5]

(B) What is activation record? What type of information kept in it? Explain its structure. [5]

OR

(A) Explain peephole optimization with example. [5]

(B) Write short note on: Code optimization techniques. [5]

Q-6 Answer the following questions.

(A) Construct SLR parsing table for following grammar. [5]

S → PP
P → pP
P → d

(B) Construct 3A code and draw flow graph for the following code segment: [5]

```
Prod=0;
i=1;
do{
prod+=a[i]*b[i];
i++;
}while(i<=20);
```

OR

(A) Construct CLR parsing table for following grammar. [5]

Stat →LHS=RHS
Stat → RHS
LHS→*RHS
LHS→id
RHS→ LHS

(B) Draw DAG and Abstract Syntax Tree for following expression: [5]

c=a*b+d*(-b)/a*b

-------- All the Best----------