

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

MID SEMESTER EXAMINATION-2015

B.Tech VI<sup>th</sup> Semester

COURSE CODE: 10B11CI612

MAX. MARKS: 30

COURSE NAME: Compiler Design

COURSE CREDITS: 04

MAX. TIME: 2 HRS

*Note: All questions are compulsory.*

*Attempt the questions of a section together.*

Section A

(Marks: 6x1 = 6)

1. Write a grammar for Boolean expressions that includes the constants *true* and *false*, the operators *and*, *or*, and *not*, and parentheses. Be sure to give *or* a lower precedence than *and* and *and* a lower precedence than *not*.
2. Show that the following grammar is LL(1)  
$$S \rightarrow AaAb \mid BbBa$$
$$A \rightarrow \epsilon$$
$$B \rightarrow \epsilon$$
3. Consider a semi-reduced input string "id [ E , E , id ]". Identify the next 2 handles for the grammar:  
$$L \rightarrow id [ E ]$$
$$E \rightarrow id$$
$$E \rightarrow E , id$$
4. Show the complete process followed by a compiler for compiling a given source code in the form of a flowchart.
5. State the difference between the *shift-shift* and *shift-reduce* conflicts.
6. When do we require a stack of symbol tables?

Section B

(Marks: 3x3 = 9)

1. Compute FIRST(X) and FOLLOW(X) for all non-terminals X of the following grammar.

$$\begin{aligned} Stmt &\rightarrow Stmt ; Stmt \\ &\quad | id := Exp \\ &\quad | print ( ExpList ) \\ Exp &\rightarrow Stmt , Exp \\ &\quad | Exp Binop Exp \\ &\quad | num \\ &\quad | id \\ ExpList &\rightarrow Exp , ExpList \\ &\quad | Exp \\ Binop &\rightarrow + | - | * | / \end{aligned}$$

2. Show that the following grammar is ambiguous for the input “(( x ) , ( y , x ) )”.  
Reconstruct the grammar after removing the ambiguity.

$$\begin{aligned} P &\rightarrow ( B , P ) | B \\ B &\rightarrow B | ( P ) \\ B &\rightarrow x | y | z \end{aligned}$$

3. Compute the canonical collection of sets of LR(0) items for the grammar:

$$\begin{aligned} S &\rightarrow aAS | bA \\ A &\rightarrow cA | d \end{aligned}$$

Section C

(Marks: 3x5 = 15)

1. Construct NFA, DFA and minimized DFA for the regular expression that represents single line comments (starting with //) with characters from the alphabet {a,b}
2. Parse the input string **abab** using the SLR parser for the grammar:

$$\begin{aligned} S &\rightarrow AS | b \\ A &\rightarrow SA | a \end{aligned}$$

3. Perform LL(1) parsing on the input string  $c = I + 5$ ; for the grammar

$$\begin{aligned} Stmt &\rightarrow id = Exp; \\ Exp &\rightarrow constant Rest \\ &\quad | id Rest \\ Rest &\rightarrow operator id Rest \\ &\quad | operator constant Rest \end{aligned}$$

Where identifier, constant and operator are standard tokens.