DHARMSINH DESAI UNIVERSITY, NADIAD
FACULTY OF TECHNOLOGY
B.TECH. SEMESTER VI [INFORMATION TECHNOLOGY]
SUBJECT: (IT 608) LANGUAGE TRANSLATOR

Examination	**: Second Sessional**	**Seat No.**	**:**
Date	**: 15/02/2016**	**Day**	**: Monday**
Time	**: 12.30 to 1.45**	**Max. Marks**	**: 36**

INSTRUCTIONS:
1. Figures to the right indicate maximum marks for that question.
2. "^" indicates null , "|" is a rule separator , other symbols used carry their usual meanings
3. Assume suitable data, if required & mention them clearly.
4. Draw neat sketches wherever necessary.

Q.1 **Do as directed.**

(a) Undeclared and multiple identifiers are examples of **[1]**
1.semantic errors 2.declartion errors 3.transposition errors 4.none of these

(b) During which type of analysis the computer tries to detect construction that have the right syntactic structure but no meaning to the operation involved **[1]**
1. Syntax analysis 2.semantic analysis 3.both (1) and (2) 4. neither (1) or (2)

(c) which of the following is a semantic action **[1]**
1.checking semantic validity of constructs in synthesis phase
2.determining the meaning of synthesis phase
3.constructing an intermediate representation
4.all of the above

(d) Consider following SDT:- **[3]**
S→ XY { X.val=2 , Y.val=4; printf (X.val , Y,val); }
X→ a { printf (X.val); }
Y→ b {printf (Y.val);
Is above SDT valid, give reason? If valid, give annotated parse tree for the input "ab". Otherwise give the correct equivalent SDT and then show the annotated parse tree on the modified SDT for input "ab".

(e) Explain the advantage of precedence function table over the precedence relation table in operator precedence parser in detail. **[2]**

(f) Consider the following expression grammar. The semantic rules for expression calculation are stated next to each grammar production. **[2]**

CFG	semantic Action
E → number	**E.val = number. val**
| E '+' E	**E(1).val = E(2).val + E(3).val**
| E '×' E	**E(1).val = E(2).val × E(3).val**

The above grammar and the semantic rules are fed to a yacc tool (which is an LALR (1) parser generator) for parsing and evaluating arithmetic expressions. What problem is detected by the yacc? How it resolves that problem?
Note:- possible problems are :left recursion problem /reduce-reduce action conflict/shift-reduce conflict/ not left factored.

(g) From following which are properties of LR parser. **[2]**
A) This kind of parsing is applicable to only small class of grammars
B)The class of grammar that can be parsed by LR parser is a superset of class of grammars that can be parsed using predictive parsers.
C)They can be constructed to recognize most of the programming languages for which the context free grammar can be written.
D)It is deterministic parser and it Expands the non-terminals.
E)It is a non-recursive parser and does shift-reduce actions.
F)It starts with an empty stack. And Ends with the root nonterminal on the stack.

Q.2 Do as directed. **[12]**

(a) Is the given grammar "S" suitable for SLR parser? Justify your answer. [8]
Grammar: S → qABC A→a|bbD B→ a | ^ C→b | ^ D→x | ^.
(i) Generate canonical LR(0) item set. 2
(ii) Compute First and Follow function. 2
(iii) Construct parse table. 2
(iv)Show trace of parser on input **'qbbxa'**. 2

(b) Design operator precedence parser for following grammar: [4]
Grammar: E → E + T | T T→ T * F | F F→ a.
Show precedence relation table, function table and trace of your parser on input **a+a*a.**

OR

(b) Explain types of errors which are detected in different phases of compiler. Also explain error response and error recovery strategies. [4]

Q.3 (a) Explain with example, what special care is needed in symbol table, if language needs support of block structuring. Also, Explain at least two symbol table organization techniques possible in a block structured language, with **example**. **[6]**

(b) Give SDT to print prefix form of given infix expression. Grammar to describe the valid infix expressions is given below. **[6]**

S→ E # **E → E +T | T** **T → T*F | F** **F → 0 | 1 | ….| 9**

Also show the annotated parse tree for "4+7*3"

OR

Q.3 (a) A robot is to be moved to a unit step in a direction specified as a command given to it. The robot moves in the direction North, South, East, West on receiving N, S, E, W command respectively & in the direction North-East , North-West, South-East, South-West on receiving A, B, C, D commands respectively. The CFG rules are as follows:- **[8]**

Seq →Start **Seq → Seq Move** **Move→ N |S |E |W |A |B |C |D**.

The current position of the robot is initialized to (0, 0) Cartesian coordinates on receiving command Start. Give SDT for knowing position of a robot after receiving a sequence of commands. Using the SDT , draw annotated parse tree for following sequence:

Start N N A A C C N

(b) For following variables encountered in some program P, show the logical organization of symbol table if the organization is → a) unordered b) tree structured **[4]**

variables are :- " x,t,a,b,x,z,w,a,b,c"

Note:-language does not support blocks.