Dr. Rajni Mohana

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

END SEMESTER EXAMINATION-2015

B.Tech VIth Semester

COURSE CODE: 10B11CI612 MAX. MARKS: 45

COURSE NAME: Compiler Design

COURSE CREDITS: 04 MAX. TIME: 3 HRS

Note: *All questions are compulsory. Carrying of mobile phone during examinations will be treated as case of unfair means.*

Attempt all questions of a section together.

Section A [9 x 1 = 9 Marks]

1. _____ verifies if the tokens are properly sequenced in accordance with the grammar of the language.
   a. Syntax analyser b. Lexical analyser c. Code generator d. a & b
2. Which of the following is equivalent to A →Aa | b
   a. A → A' A' →aA' | ϵ b. A → A' b A'→A' a | ϵ c. A →aA' A'→bA' | ϵ d. A →bA' | ϵ A'→aA'
3. i) Semantic errors are detected only at compile time.
   ii) LL(1) and LR(1) parser announce errors using valid prefix property.
   a. true, true b. true, false c. false, true d. false, false
4. If attributes of the parent node depends on its children, then it is _____ attributes.
   a. TAC b. synthesized c. inherited d. directed
5. Identify the odd statement in the list
   a. a = b + c b. b[i] = a c. a = &b d. a = b + *c
6. Generate AST for the following code:
   if a < b then m = p * -q + p * p * -q
7. Generate triples for
   if i< j then m[i+1] = x
   else m[j+1] = x
8. Compute the type value using appropriate type constructors associated with a 2D array of pointer to characters whose rows are indexed from 1 to 15 and columns are indexed from 1 to 5.
9. Every S-attribute grammar is also a L-attribute grammar. (True/False). Justify.

Section B [3x4.5 = 13.5 Marks]

1. Generate target code for the expression **d = a / b + c** for a machine language that has only the following instructions MOV, ADD, SUB, JMP, CJ<, CJ>, CJ=

2. Write a L-attributed Syntax Directed Translation to convert infix to prefix.

   $E \rightarrow E + T \mid E - T \mid T$

   $T \rightarrow T * F \mid F$

   $F \rightarrow (E) \mid id$

3. Consider 2 basic data types *integer* and *boolean*. Additionally, consider the basic types *type_error* and *void*. A language allows for type constructors of *arrays* and *pointers*. The language also allows implicit type casting from *boolean* to *integer* for the operators '+' and '–' and *integer* to *boolean* for the operators '*and*' and '*or*'. Pointers can only be created for integers. Construct a syntax directed definition (SDD) for type checking for the given language over the following grammar:

   $S \rightarrow$ if E then E

   $E \rightarrow E + E \mid E - E \mid *E \mid E[E] \mid E$ *and* $E \mid E$ *or* $E \mid$ id $\mid$ TRUE $\mid$ FALSE $\mid$ DIGIT

   (where, TRUE, FALSE and DIGIT are terminal values.)

Section C

1. Generate the CLR parsing table and parse the input string ***ace***, for the following grammar

   $S \rightarrow aAd \mid bBd \mid aBe \mid bAe$

   $A \rightarrow c$

   $B \rightarrow c$ [7.5 Marks]

2. Generate the 3 address intermediate code using backpatching technique and optimize it by building the basic blocks, for the following code.
   (Consider every array index requires 2 bytes of memory.) [15 Marks]

```
begin
        read n;
        do
                if a[i] = n then
                        a[j] = true;  j = j + 1;
                else
                        a[k] = true;
                        do
                                k = k + 1;  j = j + 1;
                                a[j] = a[k];
                        while ( 2 * k < n )
                end if
                a[i] = a[k] + a[j];  i = i+ 1;
        while i<n;
        s = k + j;
        print a[s];
end
```