

INDIAN INSTITUTE OF TECHNOLOGY, KHARAGPUR

Date: Oct 31, 2023

Time: 1 Hrs

Full Marks: 20

Deptt: Computer Science & Engineering

Class Test 2

Sub No: CS31003

Subject Name: Compilers

3rd Year B. Tech

Answer all the questions. Take and state suitable assumptions, if needed

1. Consider the CFG below, where P is the start symbol

$P \rightarrow D$

$D \rightarrow T \text{ id}; D$

$T \rightarrow B \mid \text{struct } \{ D \}$

$B \rightarrow \text{int}$

$B \rightarrow \text{float}$

$D \rightarrow \epsilon$

(a) Add suitable semantic actions at the suitable places, such that the syntax directed translation (SDT) enables the grammar to (i) Derive the type expressions for all the variables, (ii) to compute the relative address of all the variables, and to (iii) populate the symbol table. Assume that the symbol table contains the following fields (variable name, data type, relative address).

You may consider the availability of the suitable functions, which you may invoke to insert/update the symbols table entries.

(b) Consider the declaration statements of a piece of source code segment bellow

`int x;`

`struct {int x; float y;} q;`

Apply your developed SDT and demonstrate how does this code fragment populate the symbol table entries. In this process, suitably annotate the relevant parse tree.

(c) Explain how does the translation process discriminate the same variable name x as a structure member and an integer variable?

[2+2+1]

2. Consider the following grammar (Start symbol S)

$S \rightarrow \text{while } B \text{ do } S$

$\mid \text{begin } L \text{ end}$

$\mid A$

$L \rightarrow L S$

$\mid S$

$A \rightarrow \text{id} = E$

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

$B \rightarrow E_1 \text{ relop } E_2$

$B \rightarrow (B)$
 $B \rightarrow B1 \ \&\& \ B2$
 $B \rightarrow B1 \ || \ B2$
 $B \rightarrow \text{true}$
 $B \rightarrow \text{false}$
 $E \rightarrow \text{id}$

Note that, the **relop** indicates the relational operators, such as $<$, $>$ etc

- Augment the above grammar with suitable marker nonterminals at suitable places of the production, such that it can handle backpatching.
- Write the semantic actions to design a suitable syntax directed translator (SDT) to generate three address code of the above code snippet. Note that the semantic actions should handle backpatching to generate the three address code.
- Apply your SDT to translate the following code snippet to the respective three address code. Assume that the address generation starts from the address 300. Draw the suitably annotated parse tree and clearly show the backpatching steps (say, the *goto Label* statements before and after the backpatching).

```

begin
    while a > b && x < y do
        begin
            x = y + z
        end
        x = y - z
    end
end

```

[2+4+4]

3. Consider the following grammar, where S is the start symbol.

$S \rightarrow \text{id} = E \mid L = E$
 $E \rightarrow E + E \mid \text{id} \mid L$
 $L \rightarrow \text{id} [E] \mid L [E]$

- Design a syntax directed translator to generate three address code for the array references.
- Using your SDT, translate the following statement to a three-address code. Show the annotated parse tree in the process.

$A[i+1][j] = B[p][q+1]$

[2+3]