Complier Design Assignment

on Semantic Analysis

Submitted By:
Habibur Rahman
17BCS071

Submitted To:
Mr. Sarfaraz Masood

B-Tech VIth Semester (Computer Science)

Department of Computer Engineering,
Faculty of Engineering and Technology,
Jamia Millia Islamia ,
New Delhi
Session: 2019-20

Question 1) :- what is an attribute ? Explain the various types of Attributes and their uses.

Attributes grammer is a special form of content-free grammer where some additional information are appended to one or more of its non-terminals in order to provide content. Sensetive information values, such as integer., float, Character, String and expressions.

There are two types of Attributes :—

① Inherited :→ An attribute is said to be inherited attribute if its parse tree node value is determined by the attributes value at parent or seblings node. The Production must have non-terminal as a symbol in its body. An inherited attribute at Node N defined only in terms of attributes value N's parent, N itself and N. seblings.

② Synthesized :→

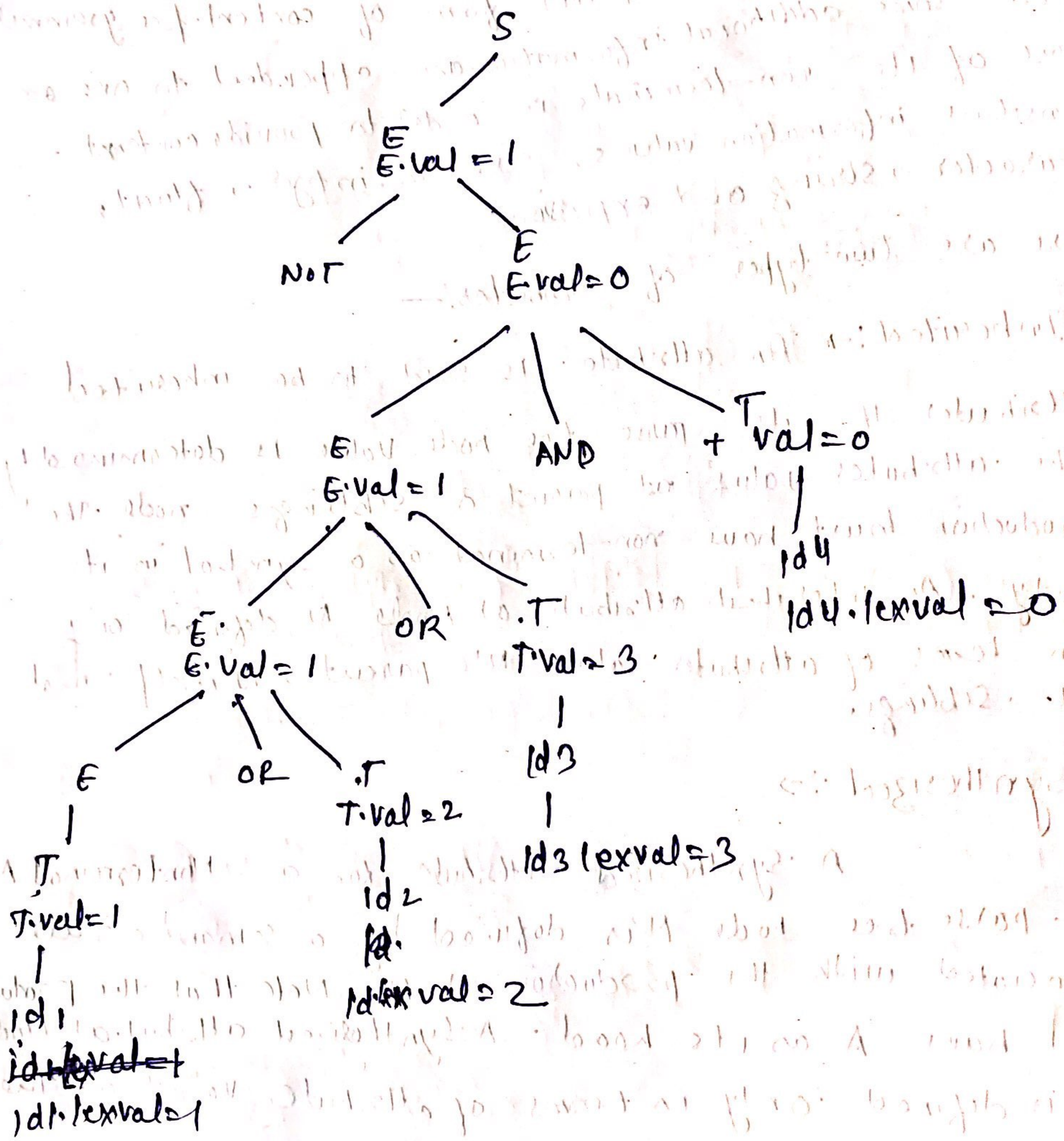
A Synthesized attribute for a Nonterminal A at parse tree node N is defined by a semantic rule associated with the production at N. Note that the production must have A as its head. A Synthesized attribut at Node N is defined only in terms of attribute values at the children of N and at itself.

Production	Semantic Rules
$S \rightarrow E$	check (E.val)
$E \rightarrow E_1$ or T	E.val = E_1.val 'OR' T.val
$E \rightarrow E_1$ AND T	E.val = E_1.val 'AND' T.val
$E \rightarrow T$	E.val = T.val
$T \rightarrow id$	T.val = id.lexval

E → NOT E   E.val = ~E.val.

Exomple :- NOT (id1 OR id2 OR ·id3 AND id4)

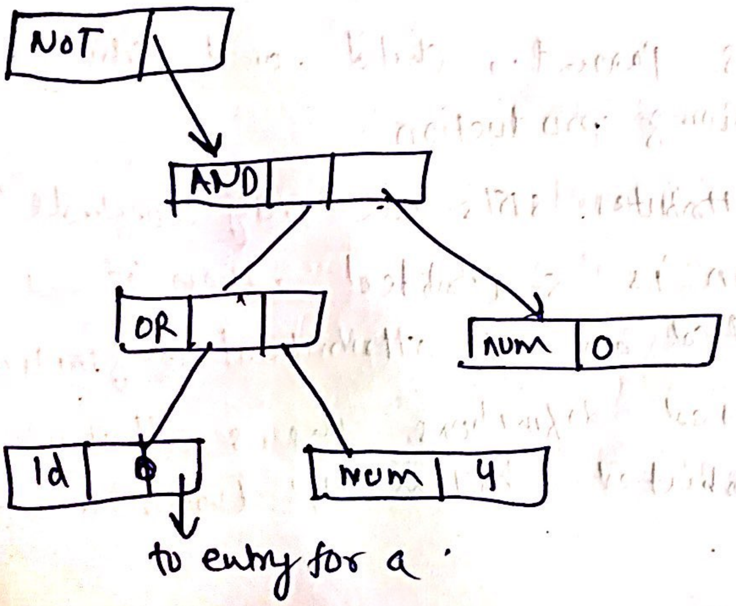
NOT (1 OR 2 OR 3 AND 0)



Q2:- Design the syntax Directed Definition for the constructing Syntax tree for the boolean expressions use Suitable exampels. to create a tree for any boolean expression from their selected CFG.

E → E1·OR T	E.nptr = mknode("OR" E1.nptr, T.nptr)
E → E1 AND T	mknode("AND", E1.nptr, T.nptr)
E → Not E	mknode("NOT", E1.nptr)
E → T	T.nptr
T → id	mk leaf(id, id.lexval)
T → num	mk leaf(num, num.val)

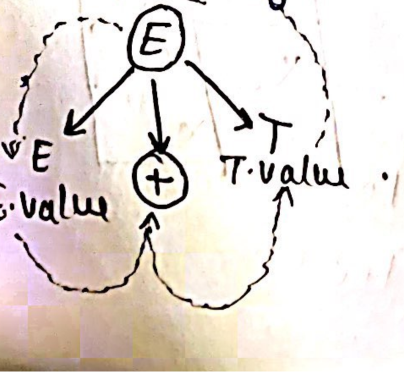
NOT (a OR 4 AND 0)



Question 3:- Difference b/w S-Attributes Definition & L attributes Definition Also explain how Sementic rule are ·

Synthesized attributes it is called S-attributed SDT. These attributes are evaluated using S-attributed SDTs that have their semantic actions written after the production.

E.val = E.value + T.value



attributes in S-attributed SDTs are ~~a~~ evaluated in bottom up parsing as values of the pattern node depend upon the values of the child nodes.

L- attributed SDT :- This ~~S~~ form of SDT uses inherited attributes with ~~instruction~~ restriction of not taking value from right siblings.

In L- attributed SDTs, a non-terminal can get values from its parent, child and sibling nodes. as in the following production.

L-attributes SDTs we may conclude that if a definition is S-attributed, then it is also L-attributed as L-attributed definition encloses S-attributed definitions the S attributed definition are also L-attributed bottom up Evaluation of S-attributed definition.

① Maintain the values of the synthesized attributes of the grammar symbols into a parallel stack.

② The evaluation of values of the attributes are done during reduction operation.

Z	Z.Z
Y	Y.Y
X	X.X
x	x

IF A → XYZ

After Reduction

A	A.a
X	X.

Q:-4:- Along with mentioning appropriate rules and annoted parse tree perform type checking on the given expression :-

```
if ( a! = b)
  {
    if (i= J)
     {
       a=b+c ;
     }
  }
```

// Note a, b, c , i and j are of integer
//type.

S → if (E) {S1}

S → id := E

S → S1 ; S2

E → E1 + E2

E → F != E ,

E → F == E

E → F

F → id

F → num

S.type = (if E.type = Boolean then S1.type. else type_error)

S.type = (if id.type = E.type then void . else typeerror)

S.type = (if S1.type = void end S2.type = void then void else type_error)

E.type = (if E1.type = E2.type. then E1.type else type-error)

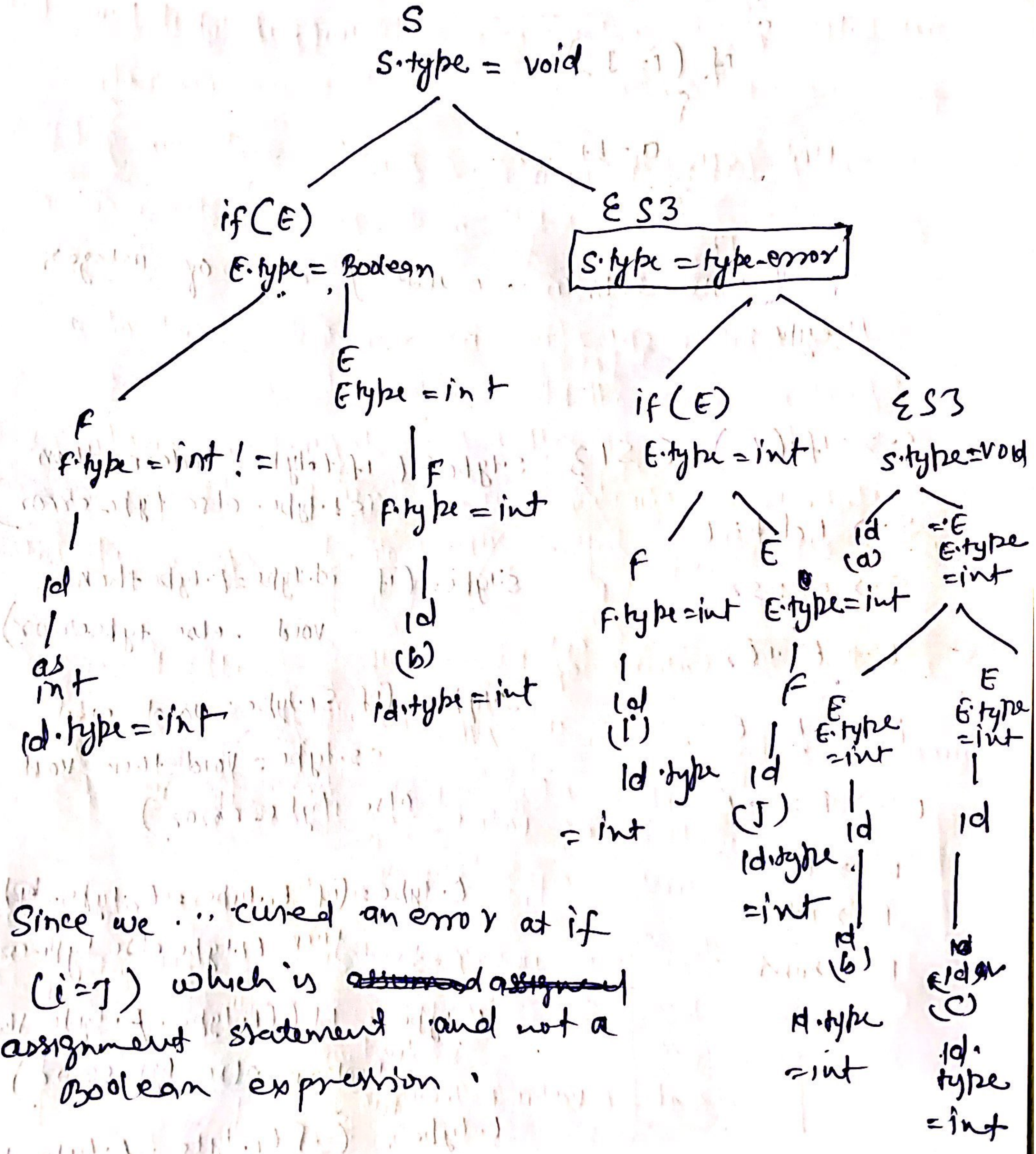
E.type = if (E1.type = E.type then Boolean else typeerror)

E.type = (if E1.type = F.type then Boolean else typezerror)

E = F.type

F.type = lookup (id. entry)

F.type = integer.



Since we .. cured an error at if (i=J) which is ~~assumed~~ ~~assigned~~ assignment statement and not a Boolean expression.