**ASSIGNMENT # 04 (CLO-3) B.S.(C.S.)-7**

**Date of Submission: 10-06-2021 Marks (10 points)**

**Q.1**

Write an attribute grammar for the floating point value of a decimal number given by the following grammar. (Hint: use a ***count*** attribute to count the number of digits to the right of the decimal point.)

**dnum num.num**

**num num digit | digit**

**digit 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9**

**Q. 2**

Consider an expression grammar as it would be written for a predictive parser and write an attribute grammar for the value of an expression given by this grammar.

exp🡪 term exp'

exp'🡪 + term exp' | - term exp' |

term🡪 factor term'

term'🡪\* factor term' |

factor🡪 (exp) | ***number***

**Q.3**

Consider the following grammar for integer binary trees (in linearized form):

*btree* **( number** *btree**btree* **) | nil**

Write an attribute grammar to check that a binary tree is ordered, that is, that the values

of the numbers of the first subtree are ≤ the value of the current number and the values

of all the numbers of the second subtree are > the value of the current number.

For example, (2 (1 nil nil) (3 nil nil)) is ordered, but (1 (2 nil nil) (3 nil nil)) is not.

**Q.4**

suppose that we have a production ***A* *BCD***. Each of the four non-terminals A, B, C and D have two attributes: s is a synthesized attribute, and i is an inherited attribute. For each of the sets of rules below tell whether:

( i ) the rules are consistent with an **S-**attributed definition.

( ii ) the rules are consistent with an **L**-attributed definition.

a) A*.s* = B.*i* + C.*s*

b) A*.s* = B.*i* + C.*s* and D.i = A.i + B.s

c) A*.s* = B.s + D.*s*

**Q. 5**

Consider the following grammar.

decl 🡪 var-list : type

var-list🡪 var-list , **id** | **id**

type 🡪 **integer | real**

1. Write an attribute grammar for the type of a variable.
2. Rewrite it so that the type of a variable can be defined as a purely synthesized attribute, and give a new attribute grammar for the type that has this property.

**Q.6**

Draw dependency graphs corresponding to each grammar rule for the following attribute grammar and draw a dependency graph for the string 3\*(4+5)\*6.

exp🡪 term exp’

exp’🡪 + term exp’ | - term exp’ |

term🡪 factor term’

term’🡪\* factor term’ |

factor🡪 (exp) | ***number***

**Q.7**

Show that, given the attribute grammar,

**Grammar rule semantic rules**

decl 🡪 type var-list *var-list.dtype= type.dtype*

type 🡪 **int** *type.dtype= integer*

type 🡪 **float** *type.dtype= real*

var-list1 🡪 **id ,** var-list2 ***id****. dtype= var-list1.dtype*

*var-list2. dtype= var-list1.dtype*

var-list🡪 **id *id****.dtype= var-list.dtype*

If the attribute *type.dtype* is kept on the value stack during an LR parse, then this value can not be found at a fixed position in the stack when reductions by the rule var-list**id** occur.

-----------------------------------------------------------------------------------------------------------------------

**Q. 8**

Determine whether the following syntax directed definition is **L-attributed** or not.

|  |  |
| --- | --- |
| **PRODUCTION** | **SEMANTIC RULES** |
| 1. T F T ' 2. T ' F T1' 3. T ' 4. F **digit** | T '.inh = F.val  T.val = T’. syn  T1'.inh = T '.inh F.val  T’. syn = T1'.syn  T’.syn = T’. inh  F.val = **digit.** Lexval |

**---------------------------------------------------------------------------------------------------**

**Q.9**

Determine whether the syntax-directed definition containing following production and rules is an L-attributed grammar?

PRODUCTION SEMANTIC RULES

A *B C* A.*s* = B.*b* ;

B.*i* = *f* ( C.*c*, A.*s*)

---------------------------------------------------------------------------------------------------------------

**Q.10**

Consider the following attribute grammar:

|  |  |
| --- | --- |
| **Grammar Rule** | **Semantic Rule** |
| SABC | B.u=S.u  A.u=B.v + C.v  S.v=A.v |
| Aa | A.v = 2 \* A.u |
| B b | B.v = B.u |
| C c | C.v = 1 |

1. Draw the parse tree for the string abc (the only string in the language) and draw the dependency graph for the associated attributes. Describe a correct order for the evaluation of the attributes.
2. Suppose that S.u is assigned the value 3 before attribute evaluation begins. What is the value S.v when evaluation has finished?
3. Suppose the attribute equations are modified as follows:

|  |  |
| --- | --- |
| Grammar Rule | Semantic Rule |
| SABC | B.u=S.u  C.u= A.v  A.u=B.v + C.v  S.v=A.v |
| Aa | A.v = 2 \* A.u |
| B b | B.v = B.u |
| C c | C.v = C.u – 2 |

What value does S.v have after attribute evaluation, if S.u=3 before evaluation begins?

**Q.11**

1. Draw dependency graphs for the following grammar rules.

S → Exp

Exp → Exp / exp | **Num | num.num**

1. Describe the two passes required to compute the attributes on the syntax tree of **5/2/2.0**.

**-----------------------------------------------------------------------------------------------------**

**Q.12**

**i) Ex. 5.1.1.**

|  |  |  |
| --- | --- | --- |
|  | **Productions** | **Semantic Rules.** |
| 1 | L E n | L.val = E.val |
| 2 | E E1+T | E.val = E1.val+ T.val |
| 3 | E T | E.val = T.val |
| 4 | T T1 \* F | T.val = T1.val F.val |
| 5 | T F | T.val = F.val |
| 6 | F ( E ) | F.val = E.val |
| 7 | F **digit** | F.val = **digit**.lexval |

For the Syntax-Directed Definition of the above Figure, give annotated parse tree for the following expression. **a) (3+4) \* (5+6) n. b) 1\*2\*3\*(4+5)n.**

**ii) Ex. 5.1.2.**

|  |  |  |
| --- | --- | --- |
|  | **Productions** | **Semantic Rules.** |
| **1** | T F T' | T′.inh = F. val  T.val = T'.syn |
| **2** | T′ \* F T1' | T1′.inh = T'.inh F.val  T′. syn = T1'. Syn |
| **3** | T' | T′. syn = T1'. Inh |
| **4** | F **digit** | F.val = digit . lexval |

**An SDD based on a grammar suitable for top-down parsing**.

For the Syntax-Directed Definition of the above Figure, give annotated parse tree for the following expression. **a) (3+4) \* (5+6) n. b) 1\*2\*3\*(4+5)n.**

**Q.13**

Consider the following grammar where numbers may be octal or decimal that is indicated by one character suffix **o** (for octal) or **d** (for decimal).

based-num🡪num basechar

basechar🡪 o | d

num🡪num digit | digit

digit🡪 0|1|2|3----------|8|9

1. Write the attribute grammar / semantic rules w.r.t. each production of the grammar.
2. Draw a parse tree showing attribute computations of the above grammar for the string **3450** .
3. Draw the dependency graph for the above string **3450** .

**Q.14**

Consider the following syntax-directed definition over the grammar defined by G = ({S, A, Sign}, S, {‘,’, ‘-‘, ‘+’, ‘n’}, P) with P the set of production and the corresponding semantic rules depicted below. There is a special terminal symbol “n” that is lexically matched by any string of one numeric digit and whose value is the numeric value of its decimal representation. For the non-terminal symbols in G we have defined two attributes, ***sign***and ***value***. The non-terminal A has these two attributes whereas **S** only has the ***value***attribute and **Sign** only has the ***sign***attribute.

S → A Sign || S.val = A.val; A.sign = Sign.sign; print(A.val);

Sign → + || Sign.sign = 1

Sign → - || Sign.sign = 0

A → n || A.val = value(n)

A → A1 , n || A1.sign = A.sign;

if(A.sign = 1) then

A.val = min (A1.val,value(n));

else

A.val = max(A1.val,value(n));

**a)** Explain the overall operation of the above syntax-directed definition and indicate (with a brief explanation) which of the attributes are either synthesized or inherited.

**b)** Give an attributed parse tree for the source string “5,2,3-“ and evaluate the attributes in the attributed parse tree depicting the order in which the attributes need to be evaluated (if more than one order is possible indicate one.)

**c)** Suggest a modified grammar and actions exclusively using synthesized attributes. Explain its basic operation.

----------------------------------------------THE END ------------------------------------------