

编译原理 - 作业(4) : 语义分析

截至时间 : 2022.5.19/周四 上课前 (14:20)

提交方式 : 超算习堂 (<https://easyhpc.net/course/144>)

Q1: (P309, Exercise 5.1.1) For the SDD below, give annotated parse trees for the following expressions:

PRODUCTIONS	SEMANTIC RULES
1) $L \rightarrow E \mathbf{n}$	$L.val = E.val$
2) $E \rightarrow E_1 + T$	$E.val = E_1.val + T.val$
3) $E \rightarrow T$	$E.val = T.val$
4) $T \rightarrow T_1 * F$	$T.val = T_1.val \times F.val$
5) $T \rightarrow F$	$T.val = F.val$
6) $F \rightarrow (E)$	$F.val = E.val$
7) $F \rightarrow \text{digit}$	$F.val = \text{digit.lexval}$

- (1) $(3 + 4) * (5 + 6) \mathbf{n}$
- (2) $(9 + 8 * (7 + 6) + 5) * 4 \mathbf{n}$

Q2: (p323, Exercises 5.3.1) Below is a grammar for expressions involving operator + and integer or floating-point operands. Floating-point numbers are distinguished by having a decimal point:

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

$$L \rightarrow \text{num} . \text{num} \mid \text{num}$$

Give an SDD to determine the type of each term T and expression E .

Q3: (p317, Exercises 5.2.4) This grammar generates binary numbers with a “decimal” point:

$$S \rightarrow L . L \mid L$$

$$L \rightarrow L B \mid B$$

$$B \rightarrow 0 \mid 1$$

- (1) Design an L-attributed SDD to compute $S.val$, the decimal number value of an input string. For example, the translation of string 101.101 should be the decimal number 5.625. Hint: use an inherited attribute $L.side$ that tells which side of the decimal point a bit is on.
- (2) Draw the annotated parse tree of 101.101.

Q4: For the code snippet below :

```

1: int x = 0;
2: float y = 0.0;
3: while (x < 10) {
4:   int y, z;
5:   y = x;
6:   z = 0;
7:   while (y < 10) {
8:     z = z + y;
9:     y = y + 1;
10:  }
11: }
12: y = x * 1.0;

```

Regarding the semantic analysis of variable type, we consider the following simplified grammar and syntax-directed translation (SDT):

- 1) $D \rightarrow T \{ L.type = T.type \} L$
- 2) $T \rightarrow \text{int} \{ T.type = \text{int} \}$
- 3) $T \rightarrow \text{float} \{ T.type = \text{float} \}$
- 4) $L \rightarrow \{ L_1.type = L.type \}$
 $L_1, \text{id} \{ addtype(\text{id.entry}, L.type)$
- 5) $L \rightarrow \text{id} \{ addtype(\text{id.entry}, L.type)$

- (1) In the above SDT, both T and L have attribute 'type'. The type attribute is synthesized or inherited? Please explain.
- (2) For Line 4 of the code snippet : `int y, z;` Construct the annotated parse tree based on the above SDT.
- (3) For Lines 3, 7 and 11 of the code snippet, list the valid variables (name and type) in symbol table.