

**Advance Software Engineering (SE-406)**

**LAB A1-G3**

**Laboratory Manual**



**Department of Software Engineering**

**DELHI TECHNOLOGICAL UNIVERSITY(DTU)**

Shahbad Daulatpur, Bawana Road, Delhi-110042

**Submitted to: -**

Ms. Parul Sharma

**Submitted by:-**

Name: ASHISH KUMAR

Roll number: 2K18/SE/041

## **EXPERIMENT 4**

- ASHISH KUMAR

- 2K18/SE/041

**Aim:-** To count Halstead metrics on a selected source code(searching/sorting).

### **Introduction:-**

Halstead's metrics are included in a number of current commercial tools that count software lines of code. By counting the tokens and determining which operators are and which are operands, the following base measures can be collected:

$n1$  = Number of distinct operators.

$n2$  = Number of distinct operands.

$N1$  = Total number of occurrences of operators.

$N2$  = Total number of occurrences of operands.

In addition to the above, Halstead defines the following:

$n1^*$  = Number of potential operators.

$n2^*$  = Number of potential operands.

Halstead refers to  $n1^*$  and  $n2^*$  as the minimum possible number of operators and operands for a module and a program respectively.

### **Halstead metrics are:**

1. Halstead Program Length – The total number of operator occurrences and the total number of operand occurrences.

$$N = N1 + N2$$

And estimated program length is,  $N^* = n1 \log_2 n1 + n2 \log_2 n2$

2. Halstead Vocabulary – The total number of unique operator and unique operand occurrences.

$$n = n1 + n2$$

3. Program Volume – Proportional to program size represents the size, in bits, of space necessary for storing the program. This parameter is dependent on specific algorithm implementation. The properties  $V$ ,  $N$ , and the number of lines in the code are shown to be linearly connected and equally valid for measuring relative program size.

$$V = N * \log_2 (n)$$

4. Program Difficulty – This parameter shows how difficult to handle the program is.

$$D = (n1 / 2) * (N2 / n2)$$

As the volume of the implementation of a program increases, the program level decreases and the difficulty increases. Thus, programming practices such as redundant usage of operands, or the failure to use higher-level control constructs will tend to increase the volume as well as the difficulty.

5. Programming Effort – Measures the amount of mental activity needed to translate the existing algorithm into implementation in the specified program language.

$$E = D * V = \text{Difficulty} * \text{Volume}$$

Advantages of Halstead Metrics:

- It is simple to calculate.
- It measures overall quality of the programs.
- It predicts the rate of error.
- It does not require the full analysis of programming structure.
- It is useful in scheduling and reporting projects.
- It can be used for any programming language.

Disadvantages of Halstead Metrics:

- It depends on the complete code.
- It has no use as a predictive estimating model.

### Source Code (in C++):

```
int sort (int x[ ], int n)

{
    int i, j, save, im1;
    /*This function sorts array x in ascending order */
    If (n< 2) return 1;
    for (i=2; i< =n; i++)
    {
        im1=i-1;
        for (j=1; j< =im1; j++)
            if (x[i] < x[j])
            {
                Save = x[i];
                x[i] = x[j];
                x[j] = save;
            }
    }
}
```

```

    }
}
return 0;
}

```

### **Result:-**

operators	occurrences	operands	occurrences
int	4	sort	1
()	5	x	7
,	4	n	3
[]	7	i	8
if	2	j	7
<	2	save	3
;	11	iml	3
for	2	2	2
=	6	1	3
–	1	0	1
<=	2	–	–
++	2	–	–
return	2	–	–
{}	3	–	–
<b>n1=14</b>	<b>N1=53</b>	<b>n2=10</b>	<b>N2=38</b>

[Table: List of the operators and operands]

Here N1=53 and N2=38.

#### **I. The program length $N=N1+N2$**

$$= 53+38=91$$

#### **II. Vocabulary of the program $n=n1+n2$**

$$=14+10=24$$

#### **III. Volume $V= N * \log_2 N$**

$$= 91 * \log_2 24 = 417 \text{ bits.}$$

**IV. Purity Ratio =  $N^{\wedge} / N$**

$$\text{Where, } N^{\wedge} = n1 \log_2 (n1) + n2 \log_2 (n2)$$

$$N^{\wedge} = 14 \log_2 (14) + 10 \log_2 (10) = 86.498$$

$$\text{So, Purity ratio} = 86.498 / 91 = 0.95$$

**V. Program Difficulty,  $D = (n1 / 2) * (N2 / n2)$**

$$= (14 / 2) * (38 / 10) = 26.6$$

**VI. Effort,  $E = D * V$**

$$= 26.6 * 417 = 11092.2 \text{ PM}$$

**Learning from experiment:-** We have successfully learnt about Halstead Metrics and we compute the metrics of a given source code written in C++ language. We listed out the advantages and disadvantages of Halstead Metrics as well.