**Description of language tokens and its regular definitions**

There were only 5 complex tokens which are as:

1. Identifier:

Id - > letter(letter | digit )\*

letter -> a|b|c|d|….|z|A|B||C|D|….|Z

digit -> 0|1|2|3|4…|9

**Description:** Identifiers have more than one lexemes therefore it’s a com[;ex token. Identifier starts with a any letter from a-z or A-Z then after that any numbers of letters in the range or a-z and A-Z can come. Also any no of digits can also come in the range of 0-9 after first letter.

1. Num constant:

Num -> OS (digit) (digit)\*

OS -> +|-|^

digit -> 0|1|2|3|4…|9

**Description:** Num have more than one lexemes therefore it’s a com[;ex token. Num can start with a sign it’s a optional sign a num may or may not have any sign. After that a first digit comes in the range of 0-9. And you can have any no digits after the first digit.

1. Literal constant:

literal - > ‘ letter ’

letter -> a|b|c|d|….|z|A|B||C|D|….|Z

**Description:** Literals are enclosed in the single quote and come in the range of the a|b|c|d|….|z|A|B||C|D|….|Z.

1. Relational operators:

Relation -> RO equal

RO -> **< | > | =**

Equal -> = | ^

**Description:** This is a complex token because it has ore then one lexemes. Here any of the relational operator can come and after that equal is optional it can may or may not comein the string.

1. Comments:

Comments-> hashOther

Hash -> #

Other-> 0|1|2|3|4…|9|a|b|c|d|….|z|A|B||C|D|….|Z|+|-|^|**<| > | =**

**Description:** Comment can be considered as a complex token as it is producing different lexemes.