LEX and Yacc are used to achieve the following tasks.

This is similar to Calculator, except it is used with text strings.

Text processor: Let op1 and op2 are operands whose values are text (contains alphanumeric characters).

op1 * op2 is a concatenation of the two operands.

op1~op2 is true if op1 is a prefix of op2 (Eg: abc ~ abcd has the value true).

op1@op2 is true if op1 is a suffix of op2

op1#op2 is true if op1 is a substring of op2

op1 = op2 is true if both operands are equal.

op1 <> op2 is true if op1 is not equal to op2.

op1^numb, where numb is a non-negative integer is, op1 concatenated to itself numb number of times.

op1%numb is the prefix of op1 whose length is numb.

op1&numb is the suffix of op1 whose length is numb.

?op1 gives the length of op1.

Operations can be nested using braces ( ).

Precedence & associativity rules

Highest to lowest precedence, the operators are: (),?,^.*, %,&,~,@,#,=

Associativity is from left to right (except for ? and ^ which are right to left).

An appropriate CFG for this and used.

Example:

Input: abc^2 ===> Output: abcabc

Input: abc^?xyz ===> Output: abcabcabc

concatenate%6&3 ===> cat

concatenate%6&3 = cat ===> true