Grammar Rules:

<exp> ::=	**<term> { op <term> | (<exp>) } | (<exp>)**
<term> ::=	**relop <int> | <int> - <int>**
<int> ::=	**0-9999999**
<relop> ::=	**> | < | >= | <= | == | not**
<-> ::=	**-**

Token Categories:

Logical Operators:

- **and** is the AND operator
- **or** is the OR operator
- **nand** is the NAND operator
- **xor** is the XOR operator
- **xnor** is the XNOR operator

Rule List:

-Connect two terms like _ and _ to compare results

-cannot exist by itself

-the logical operator will use the truth table of binary to determine output

-the two terms it can connect must comprise of one or more the following:

    -a result to a relation operation (>)

    -a dash operator result

    -parenthesis of a result

Dash Operator:

- Only use the short **en-dash -** symbol from the ASCII standard.

Rule List:

-Connects two integers, and the form must follow “int- int” (underscore means an integer)

-cannot exist by itself

-Is invalid if there is a missing int on either side of the dash

Relational Operator:

- <
- >
- <=
- >=
- ==
- !=
- **not**

Rule List:
-expresses a range of integer values, these values must be positive and they must be real

-the operator must be next to an integer to be valid, cannot exist on its own

Integer:
“9”
Rule List:
-must be a number like 3 or 42, integer only, non-negative, real.

Expression Syntax Patterns:
Expression:
term {op term}
Rule List:
-essentially connects terms using an operator discussed earlier
-can include a non-terminal symbol
-expression is defined in terms of a term, which is defined in terms of a factor, and one of the alternatives for factor is a term
-a chain of calls from expression to a term to a factor and back to an expression must always consume at least one token from the input statement

>

2 - - 4

- 7

- 7 -
- = 6
- (!= 5) and
- 2 - 4 *and* >< 300
- >= 5) xnor < 10