Objective:
basic operators (R-arithmetics, comparison) // seems like need to add “,” as we need to do concatenation
iterative operators (priority “()”, concatenation “{}”)
unary operators (“-”) // need to find in paper

These can be found in that LPSAT paper's appendix.

Also with constants:
constants in definition: decide upper limit (maximum value) for variables
constants in expression: combined with basic operators // need to find in paper

Algorithm (only for wire, assign sentences)
for linear constrains, we need variables and constants
variables: already extracted to @var
operators: build word list like qw (+ - * % ….)
1. find one basic operator, check immediate former element for “)””}” and later one for “(””{”, if clear then do LP generation and replace this unit expression with temp variable _tmp00, but remember to keep “()””{}” if there is more than 1 // it seems there'll be only “,” in “{}”
2. go on iteration, repeat step 1, until no iterative operators // remember to replace _tmp00 with desired variable on left side of “=”

translate constants

So, we have functions to realize below:

First, iteration steps. Basically
```
While (find basic operators)
{
        if (judge unit == FALSE) {next;}
        Write linear constrains to output file.
        Find immediate (){}, remove them, store into new element $new;
        replace substr $items[i-1,i,i+1] with $new;  // Important function need be down independently
}
```
OK, first than first, we need to take “,” out as basic operator.
```
While ($items[i] /,/)  // we can use $_ = $items[i]
{
        @tmp = split /,/;
        my @new;
        $new[0..2] = $tmp[0..1];
        $new[1] = “,”;
        replace $items[i] with @new;
}
```
Second(or we can do it firstly?), translate constants and set variable limits from definition
```
While ($items[i], /[|]/) {
        remove [], inner part 28:0 => var < 2^29
}
```
2b'10 => 2