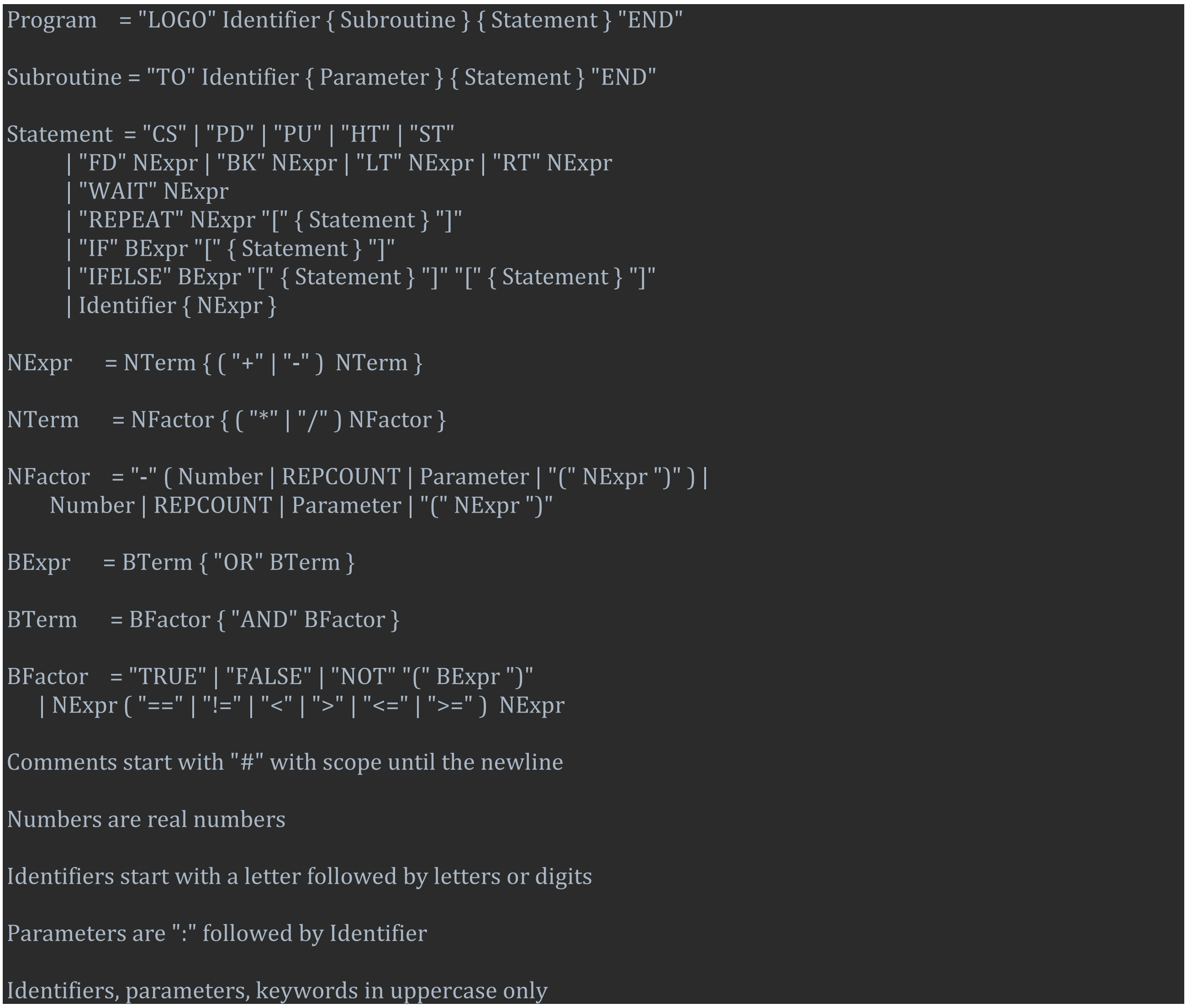
Report Logo

Grammar to use:



```
Program    = "LOGO" Identifier { Subroutine } { Statement } "END"

Subroutine = "TO" Identifier { Parameter } { Statement } "END"

Statement  = "CS" | "PD" | "PU" | "HT" | "ST"
         | "FD" NExpr | "BK" NExpr | "LT" NExpr | "RT" NExpr
         | "WAIT" NExpr
         | "REPEAT" NExpr "[" { Statement } "]"
         | "IF" BExpr "[" { Statement } "]"
         | "IFELSE" BExpr "[" { Statement } "]" "[" { Statement } "]"
         | Identifier { NExpr }

NExpr      = NTerm { ( "+" | "-" )  NTerm }

NTerm      = NFactor { ( "*" | "/" ) NFactor }

NFactor    = "-" ( Number | REPCOUNT | Parameter | "(" NExpr ")" ) |
      Number | REPCOUNT | Parameter | "(" NExpr ")"

BExpr      = BTerm { "OR" BTerm }

BTerm      = BFactor { "AND" BFactor }

BFactor    = "TRUE" | "FALSE" | "NOT" "(" BExpr ")"
    | NExpr ( "==" | "!=" | "<" | ">" | "<=" | ">=" )  NExpr

Comments start with "#" with scope until the newline

Numbers are real numbers

Identifiers start with a letter followed by letters or digits

Parameters are ":" followed by Identifier

Identifiers, parameters, keywords in uppercase only
```

Introduction:

22.05.2020

Installed and set “ant” and “java8”.
First I looked at how the Logo.jj file works.
Once I understood this, I started by adding the two methods for subroutines and statements.

I started translating the grammar for subroutines and statements and also for all other voids like nexpr.

24.05.2020

The first problem I had was when I had to add the number. Because I was able to write all the expressions but I was stuck on the Factor expressions.

In fact, I couldn't print the number here. Once I was able to solve the problem, I noticed that the function was called twice to print the number.

So, I wrote a function so that I only had to call this one for the times I needed it and not write it all down every time.

I also used this method for other functions that were repeated in the code, so as not to have useless double code.

26.05.2020

Solved this problem, I was able to run some logo files and I was able to do some tests.

Another difficulty was for the Repeat in fact, I was able to do the function for a normal for, but now I'm stuck for a nested for, I'm trying to figure out how to add a variable so that when the function is called the second time the variable does not overwrite.

02.06.2020

I've created a test file, to test +,-,*,/

.

```
LOGO FACTOR
   FD 1 + 2
   FD 1 * 1
   FD 1 - 1
   FD 1 / 1
END
```

After I went out to find a solution, I tested the functions on the polygon.logo file,
I had some problems in the polygon.logo file, when I couldn't print the split.
Before:

```
private void poly(double SIDES){
  for(int i = 0; i <= SIDES; i++){
    logo.fd(50);
    logo.rt(360SIDES);
  }
}
```

After:

```
logo.rt(360/SIDES);
```

Everything works.

09.06.2020

I had trouble with the REPEAT part:

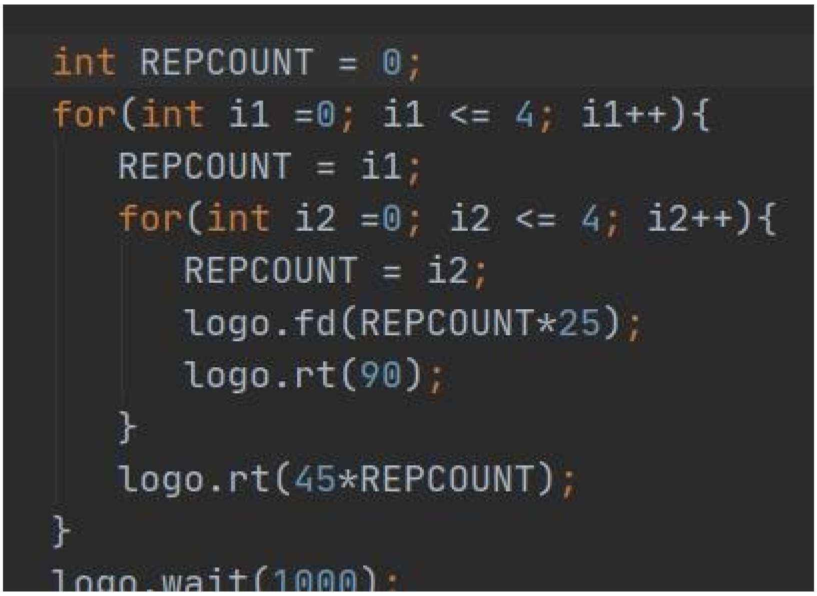
```
<REPEAT>{indent(); pw.print("for(int i = 0; i <= ");} nexpr() {pw.println("; i++){");numIndent++;} <LBRA>
(statement())* <RBRA> {numIndent--; indent();pw.println("}");}
```

This was my first test to see if it worked, and for single cycles it works great.

The problem is with nested cycles.

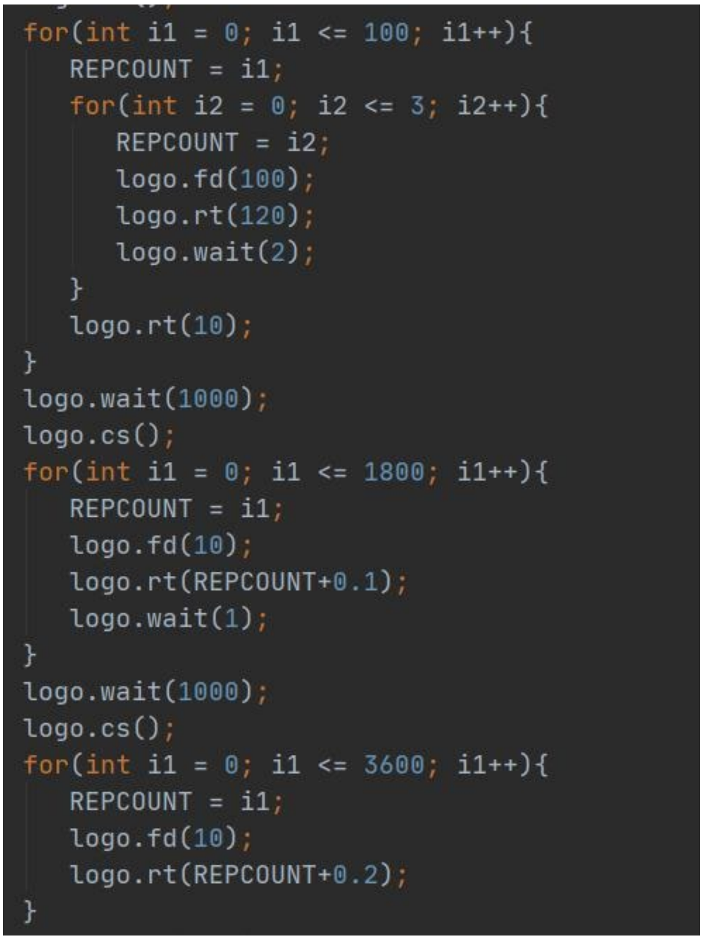
I declared a REPCOUNT variable:

I applied a counter to add a number:



```
int REPCOUNT = 0;
for(int i1 =0; i1 <= 4; i1++){
    REPCOUNT = i1;
    for(int i2 =0; i2 <= 4; i2++){
        REPCOUNT = i2;
        logo.fd(REPCOUNT*25);
        logo.rt(90);
    }
    logo.rt(45*REPCOUNT);
}
logo.wait(1000);
```

My problem at the moment is that I can't decrease REPCOUNT correctly, while the counter is no problem:



```
for(int i1 = 0; i1 <= 100; i1++){
    REPCOUNT = i1;
    for(int i2 = 0; i2 <= 3; i2++){
        REPCOUNT = i2;
        logo.fd(100);
        logo.rt(120);
        logo.wait(2);
    }
    logo.rt(10);
}
logo.wait(1000);
logo.cs();
for(int i1 = 0; i1 <= 1800; i1++){
    REPCOUNT = i1;
    logo.fd(10);
    logo.rt(REPCOUNT+0.1);
    logo.wait(1);
}
logo.wait(1000);
logo.cs();
for(int i1 = 0; i1 <= 3600; i1++){
    REPCOUNT = i1;
    logo.fd(10);
    logo.rt(REPCOUNT+0.2);
}
```

I added a line for REPCOUNT, that decrease the count.

Demo.logo works.

12.06.2020

Last day.. even though the demo worked, I noticed an error in REPCOUNT.
The decrease was wrong, a wrong value came back to me, even if I didn't notice it immediately, because the logo files worked.

I solved it by simply indicating REPCOUT as the variable in the for.
Since I was already increasing and decreasing automatically, I didn't have to worry.

```
void repcount():{Token r;}{
<REPCOUNT> {pw.print("i" + counter + "");}
}
```

Everything works!
Finito

Giorgio Bakhiet Derias