Github link:

Lang.y used:

```
%{
    #include <stdio.h>

    #include <stdlib.h>

    #define YYDEBUG 1
%}
```

```
%token BREAK

%token CASE

%token CHAR

%token CONST

%token FINAL

%token DEFAULT

%token DO

%token WHILE

%token IF

%token THEN

%token ELSE

%token DOUBLE

%token FLOAT

%token INT

%token LONG

%token SHORT

%token FOR

%token PRINTF

%token RETURN

%token SWITCH

%token VOID

%token TRY
```

```
%token CATCH

%token VAR

%token OPERATOR

%token SEPARATOR

%token SPACE

%token READ

%token WRITE

%token '+'

%token '-'

%token ';'

%token '{'

%token '}'

%token '['

%token ']'

%token ':'

%token '='

%token '*'

%token '/'

%token '%'

%token '^'

%token '&'

%token '>'

%token '<'

%token EQUALGREATER

%token EQUALMINOR

%token RELATION

%token nr

%token IDENTIFIER

%token BOOLEAN


%%
```

```
program: VAR decllist ';' ;

decllist: decl | decl ';' decllist;

stmtlist: stmt | stmt ';' {stmt};

stmt: decl|simplestmt|structstmt;

decl: IDENTIFIER ':' type;

type: primarytypes | arraytype;

primarytypes: INT | LONG | CHAR | BOOLEAN | DOUBLE | SHORT;

arraytype: primarytypes '[' nr ']';

simplestmt: assignstmt | iostmt;

assignstmt: IDENTIFIER '=' expression;

iostmt: READ | WRITE '('IDENTIFIER')';

structstmt: cmpdstmt | ifstmt | whilestmt;

cmpdstmt: '{'stmtlist'}';

operation: '+' | '-' | '*' | '/' | '%' | '^' | '&';

ifstmt: IF condition THEN stmt '['ELSE stmt']';

whilestmt: WHILE condition DO stmt;

expression: expression '+' term | term;

term: IDENTIFIER | nr | CHAR;

relation: '<' | EQUALMINOR | EQUALGREATER | '>';

condition: expression relation expression;



%%


yyerror(char *s)
{
  printf("%s\n", s);
}


extern FILE *yyin;
```

```c
main(int argc, char **argv)
{
  if (argc > 1)
    yyin = fopen(argv[1], "r");
  if ( (argc > 2) && ( !strcmp(argv[2], "-d") ) )
    yydebug = 1;
  if ( !yyparse() )
    fprintf(stderr,"No errors detected\n");
}
```