Project 1 Report

Jacob Sherrill
The University of Tulsa
CS4013 – Compiler Construction
September 2016

Introduction

For this project, I have created a lexical analyzer for a subset of the Pascal programming language. This lexical analyzer will later be invoked by a parser. This program produces a listing file and a token file from an input pascal source file and a reserved word file.

Methodology

I began this project by creating a main 'driver' that requires lines of Pascal source code. After that, I drew all finite state machines that would be needed to analyze the tokens. That gave way to implementing the finite state machines in code, using an if-else structure format. Returning and writing to token and listing files as well as a token list was also achieved.

Implementation

I used the Java programming language to create the lexical analyzer. I am using Git for version control of my code.

Discussion and Conclusions

Without first illustrating the finite state machines of a lexical analyzer by hand, I would not have been able to complete this portion of the compiler. Keeping track of all tokens and their respective integers is important for the projects that will come after this.

References

Aho, Alfred et al. *Compilers – Principles, Techniques, and Tools*. Addison-Wesley, 1986. p. 746.

Appendix I: Sample Inputs and Outputs

Input: “Source.txt” (Aho, 746.)

```
program example(input, output);
var x, y: integer;
var c, d: real;
function gcd(a, b: integer): integer;
begin
       if b = 0 then gcd := a
       else gcd := gcd(b, a mod b)
end;

begin
       array[3 .. 4]
       if b <> 0 then a := 122.38888E-22
       if b < 0 then a := 2999999999
       if b > 0 then a := 2.00005
       if b >= 0 or b <= -2 then a := ((2 + 5) - 3) * 7
       if b <= 9 and b > 1 then a := 2
end;

begin
       read(x, y);
       write(gcd(x, y));
end.
```

Input: “Reserved.txt”

```
div    3 3
mod    3 4
and    3 5
or     2 3
begin 5 0
end    6 0
program7 0
var    8 0
function 9 0
if 10 0
then 11 0
else 12 0
integer13 0
array 14 0
of 15 0
real 16 0
procedure 17 0
while 18 0
do 19 0
not    20 0
```

Output: “Source.txt”: Token file

```
Line No.              Lexeme        TOKEN-TYPE     ATTRIBUTE
3                     program       7 (RES)               0
3                     example               25 (ID)               0 (NULL)
3                     (                     4 (CATCHALL)  3 (LEFTPAREN)
3                     input                 25 (ID)               0 (NULL)
3                     ,                     4 (CATCHALL)          7 (COMMA)
3                     output                25 (ID)               0 (NULL)
3                     )                     4 (CATCHALL)  4 (RIGHTPAREN)
3                     ;                     4 (CATCHALL)  5 (SEMICOLON)
4                     var           8 (RES)               0
4                     x                     25 (ID)               0 (NULL)
4                     ,                     4 (CATCHALL)          7 (COMMA)
4                     y                     25 (ID)               0 (NULL)
4                     :                     4 (CATCHALL)          6 (COLON)
4                     integer       13 (RES)                      0
4                     ;                     4 (CATCHALL)  5 (SEMICOLON)
5                     var           8 (RES)               0
5                     c                     25 (ID)               0 (NULL)
5                     ,                     4 (CATCHALL)          7 (COMMA)
5                     d                     25 (ID)               0 (NULL)
5                     :                     4 (CATCHALL)          6 (COLON)
5                     real          16 (RES)                      0
5                     ;                     4 (CATCHALL)  5 (SEMICOLON)
6                     function              9 (RES)               0
6                     gcd                   25 (ID)               0 (NULL)
6                     (                     4 (CATCHALL)  3 (LEFTPAREN)
6                     a                     25 (ID)               0 (NULL)
6                     ,                     4 (CATCHALL)          7 (COMMA)
6                     b                     25 (ID)               0 (NULL)
6                     :                     4 (CATCHALL)          6 (COLON)
6                     integer       13 (RES)                      0
6                     )                     4 (CATCHALL)  4 (RIGHTPAREN)
6                     :                     4 (CATCHALL)          6 (COLON)
6                     integer       13 (RES)                      0
6                     ;                     4 (CATCHALL)  5 (SEMICOLON)
7                     begin         5 (RES)               0
8                     if            10 (RES)                      0
8                     b                     25 (ID)       loc9 (ptr to sym tab)
8                     =                     1 (RELOP)     1 (EQ)
8                     0                     24 (INT)      0 (NULL)
8                     then          11 (RES)                      0
8                     gcd                   25 (ID)       loc7 (ptr to sym tab)
8                     :=                    21 (ASSIGNOP) 1 (ASSIGN)
8                     a                     25 (ID)       loc8 (ptr to sym tab)
9                     else          12 (RES)                      0
9                     gcd                   25 (ID)       loc7 (ptr to sym tab)
9                     :=                    21 (ASSIGNOP) 1 (ASSIGN)
9                     gcd                   25 (ID)       loc7 (ptr to sym tab)
9                     (                     4 (CATCHALL)  3 (LEFTPAREN)
9                     b                     25 (ID)       loc9 (ptr to sym tab)
9                     ,                     4 (CATCHALL)          7 (COMMA)
9                     a                     25 (ID)       loc8 (ptr to sym tab)
9                     mod           3 (RES)               4
9                     b                     25 (ID)       loc9 (ptr to sym tab)
9                     )                     4 (CATCHALL)  4 (RIGHTPAREN)
10                    end           6 (RES)               0
10                    ;                     4 (CATCHALL)  5 (SEMICOLON)
```

```
12                   begin          5 (RES)              0
13                   array          14 (RES)                    0
13                   [                     4 (CATCHALL)  1 (LEFTBRACK)
13                   3                     24 (INT)      0 (NULL)
13                   ..                    4 (CATCHALL)  9 (DOTDOT)
13                   4                     24 (INT)      0 (NULL)
13                   ]                     4 (CATCHALL)  2 (RIGHTBRACK)
14                   if             10 (RES)                    0
14                   b                     25 (ID)       loc9 (ptr to sym tab)
14                   <>                    1 (RELOP)     2 (NE)
14                   0                     24 (INT)      0 (NULL)
14                   then           11 (RES)                    0
14                   a                     25 (ID)       loc8 (ptr to sym tab)
14                   :=                    21 (ASSIGNOP) 1 (ASSIGN)
14                   122            99 (LEXERR)   2 (REALLONG)
14                   122.38888E-22         23 (LONGREAL) 0 (NULL)
15                   if             10 (RES)                    0
15                   b                     25 (ID)       loc9 (ptr to sym tab)
15                   <                     1 (RELOP)     3 (LT)
15                   0                     24 (INT)      0 (NULL)
15                   then           11 (RES)                    0
15                   a                     25 (ID)       loc8 (ptr to sym tab)
15                   :=                    21 (ASSIGNOP) 1 (ASSIGN)
15                   2999999999                   24 (INT)      0 (NULL)
16                   if             10 (RES)                    0
16                   b                     25 (ID)       loc9 (ptr to sym tab)
16                   >                     1 (RELOP)     6 (GT)
16                   0              99 (LEXERR)   7 (INTLONG)
16                   0                     24 (INT)      0 (NULL)
16                   then           11 (RES)                    0
16                   a                     25 (ID)       loc8 (ptr to sym tab)
16                   :=                    21 (ASSIGNOP) 1 (ASSIGN)
16                   2              99 (LEXERR)   2 (REALLONG)
16                   2.00005        22 (REAL)            0 (NULL)
17                   if             10 (RES)                    0
17                   b                     25 (ID)       loc9 (ptr to sym tab)
17                   >=                    1 (RELOP)     5 (GE)
17                   0                     24 (INT)      0 (NULL)
17                   or             2 (RES)              3
17                   b                     25 (ID)       loc9 (ptr to sym tab)
17                   <=                    1 (RELOP)            4 (LE)
17                   -                     2 (ADDOP)            2 (MINUS)
17                   2                     24 (INT)      0 (NULL)
17                   then           11 (RES)                    0
17                   a                     25 (ID)       loc8 (ptr to sym tab)
17                   :=                    21 (ASSIGNOP) 1 (ASSIGN)
17                   (                     4 (CATCHALL)  3 (LEFTPAREN)
17                   (                     4 (CATCHALL)  3 (LEFTPAREN)
17                   2                     24 (INT)      0 (NULL)
17                   +                     2 (ADDOP)            1 (PLUS)
17                   5                     24 (INT)      0 (NULL)
17                   )                     4 (CATCHALL)  4 (RIGHTPAREN)
17                   -                     2 (ADDOP)            2 (MINUS)
17                   3                     24 (INT)      0 (NULL)
17                   )                     4 (CATCHALL)  4 (RIGHTPAREN)
17                   *                     3 (MULOP)            MULT
17                   7                     24 (INT)      0 (NULL)
18                   if             10 (RES)                    0
18                   b                     25 (ID)       loc9 (ptr to sym tab)
18                   <=                    1 (RELOP)            4 (LE)
18                   9                     24 (INT)      0 (NULL)
```

```
18                   and           3 (RES)              5
18                   b                    25 (ID)       loc9 (ptr to sym tab)
18                   >                    1 (RELOP)     6 (GT)
18                   1                    24 (INT)      0 (NULL)
18                   then          11 (RES)                    0
18                   a                    25 (ID)       loc8 (ptr to sym tab)
18                   :=                   21 (ASSIGNOP) 1 (ASSIGN)
18                   2                    24 (INT)      0 (NULL)
19                   end           6 (RES)              0
19                   ;                    4 (CATCHALL)  5 (SEMICOLON)
21                   begin         5 (RES)              0
22                   read                 25 (ID)              0 (NULL)
22                   (                    4 (CATCHALL)  3 (LEFTPAREN)
22                   x                    25 (ID)       loc3 (ptr to sym tab)
22                   ,                    4 (CATCHALL)         7 (COMMA)
22                   y                    25 (ID)       loc4 (ptr to sym tab)
22                   )                    4 (CATCHALL)  4 (RIGHTPAREN)
22                   ;                    4 (CATCHALL)  5 (SEMICOLON)
23                   write                25 (ID)              0 (NULL)
23                   (                    4 (CATCHALL)  3 (LEFTPAREN)
23                   gcd                  25 (ID)       loc7 (ptr to sym tab)
23                   (                    4 (CATCHALL)  3 (LEFTPAREN)
23                   x                    25 (ID)       loc3 (ptr to sym tab)
23                   ,                    4 (CATCHALL)         7 (COMMA)
23                   y                    25 (ID)       loc4 (ptr to sym tab)
23                   )                    4 (CATCHALL)  4 (RIGHTPAREN)
23                   )                    4 (CATCHALL)  4 (RIGHTPAREN)
23                   ;                    4 (CATCHALL)  5 (SEMICOLON)
24                   end           6 (RES)              0
24                   .                    4 (CATCHALL)  8 (DOT)
                     98 (EOF)                    0 (NULL)
```

Output: “Source.txt”: Listing File

```
1
2
3             program example(input, output);
4             var x, y: integer;
5             var c, d: real;
6             function gcd(a, b: integer): integer;
7             begin
8                    if b = 0 then gcd := a
9                    else gcd := gcd(b, a mod b)
10            end;
11
12            begin
13                   array[3 .. 4]
14                   if b <> 0 then a := 122.38888E-22
LEXERR:Real first part too long:    122
15                   if b < 0 then a := 2999999999
16                   if b > 0 then a := 2.00005
LEXERR:Int too long:  0
LEXERR:Real first part too long:    2
17                   if b >= 0 or b <= -2 then a := ((2 + 5) - 3) * 7
18                   if b <= 9 and b > 1 then a := 2
19            end;
20
21            begin
22                   read(x, y);
23                   write(gcd(x, y));
24            end.
```

Input: “SourceErrors.txt”: File With Errors

```
reallylongword

555556.01;
4444.555556;
00;
01;
1.;
12345678901;
10101010101;
1.0000001;
5.26EE82;
9.99E222;

# @ ! $ %

9programme example(input, output);
var x, y: integer;
var c, d: realaaaaaaa;
function gcd(a, b: integer): integer;
begin
       if b = 01 then gcd := a
       else gcd := gcd(b, a mod b)
end;
```

Input: “Reserved.txt”

```
div    3 3
mod    3 4
and    3 5
or     2 3
begin 5 0
end    6 0
program7 0
var    8 0
function 9 0
if 10 0
then 11 0
else 12 0
integer13 0
array 14 0
of 15 0
real 16 0
procedure 17 0
while 18 0
do 19 0
not    20 0
```

Output: “SourceErrors.txt”: Token File

```
Line No.            Lexeme        TOKEN-TYPE    ATTRIBUTE
2                   reallylongword 99 (LEXERR)          6 (LONGWORD)
4                   555556        99 (LEXERR)   2 (REALLONG)
4                   555556.01            22 (REAL)             0 (NULL)
4                   ;                    4 (CATCHALL)   5 (SEMICOLON)
5                   4444.555556          99 (LEXERR)    3 (RLLONGSCND)
5                   4444.555556          22 (REAL)             0 (NULL)
5                   ;                    4 (CATCHALL)   5 (SEMICOLON)
6                   0             99 (LEXERR)   5 (LEADZERO)
6                   00                   24 (INT)       0 (NULL)
6                   ;                    4 (CATCHALL)   5 (SEMICOLON)
7                   0             99 (LEXERR)   5 (LEADZERO)
7                   01                   24 (INT)       0 (NULL)
7                   ;                    4 (CATCHALL)   5 (SEMICOLON)
8                   1.            22 (REAL)             0 (NULL)
8                   ;                    4 (CATCHALL)   5 (SEMICOLON)
9                   12345678901                 24 (INT)       0 (NULL)
9                   ;                    4 (CATCHALL)   5 (SEMICOLON)
10                  10101010101          99 (LEXERR)    7 (INTLONG)
10                  10101010101                 24 (INT)       0 (NULL)
10                  ;                    4 (CATCHALL)   5 (SEMICOLON)
11                  1             99 (LEXERR)   2 (REALLONG)
11                  1.0000001            99 (LEXERR)    3 (RLLONGSCND)
11                  1.0000001            22 (REAL)             0 (NULL)
11                  ;                    4 (CATCHALL)   5 (SEMICOLON)
12                  5.26E         23 (LONGREAL)  0 (NULL)
12                  E82                  25 (ID)               0 (NULL)
12                  ;                    4 (CATCHALL)   5 (SEMICOLON)
13                  9.99E222             99 (LEXERR)    4 (EXPLONG)
13                  9.99E222             23 (LONGREAL)  0 (NULL)
13                  ;                    4 (CATCHALL)   5 (SEMICOLON)
15                  #                    99 (LEXERR)    1 (UNRECOGSYM)
15                  @                    99 (LEXERR)    1 (UNRECOGSYM)
15                  !                    99 (LEXERR)    1 (UNRECOGSYM)
15                  $                    99 (LEXERR)    1 (UNRECOGSYM)
15                  %                    99 (LEXERR)    1 (UNRECOGSYM)
17                  9                    24 (INT)       0 (NULL)
17                  programme                   25 (ID)               0 (NULL)
17                  example              25 (ID)               0 (NULL)
17                  (                    4 (CATCHALL)   3 (LEFTPAREN)
17                  input                25 (ID)               0 (NULL)
17                  ,                    4 (CATCHALL)          7 (COMMA)
17                  output               25 (ID)               0 (NULL)
17                  )                    4 (CATCHALL)   4 (RIGHTPAREN)
17                  ;                    4 (CATCHALL)   5 (SEMICOLON)
18                  var           8 (RES)               0
18                  x                    25 (ID)               0 (NULL)
18                  ,                    4 (CATCHALL)          7 (COMMA)
18                  y                    25 (ID)               0 (NULL)
18                  :                    4 (CATCHALL)          6 (COLON)
18                  integer       13 (RES)                     0
18                  ;                    4 (CATCHALL)   5 (SEMICOLON)
19                  var           8 (RES)               0
19                  c                    25 (ID)               0 (NULL)
19                  ,                    4 (CATCHALL)          7 (COMMA)
19                  d                    25 (ID)               0 (NULL)
19                  :                    4 (CATCHALL)          6 (COLON)
19                  realaaaaaaa   99 (LEXERR)          6 (LONGWORD)
```

```
20                   function              9 (RES)              0
20                   gcd                   25 (ID)              0 (NULL)
20                   (                     4 (CATCHALL)  3 (LEFTPAREN)
20                   a                     25 (ID)              0 (NULL)
20                   ,                     4 (CATCHALL)         7 (COMMA)
20                   b                     25 (ID)              0 (NULL)
20                   :                     4 (CATCHALL)         6 (COLON)
20                   integer       13 (RES)                     0
20                   )                     4 (CATCHALL)  4 (RIGHTPAREN)
20                   :                     4 (CATCHALL)         6 (COLON)
20                   integer       13 (RES)                     0
20                   ;                     4 (CATCHALL)  5 (SEMICOLON)
21                   begin         5 (RES)              0
22                   if            10 (RES)                     0
22                   b                     25 (ID)       loc11 (ptr to sym tab)
22                   =                     1 (RELOP)     1 (EQ)
22                   0             99 (LEXERR)   5 (LEADZERO)
22                   01                    24 (INT)      0 (NULL)
22                   then          11 (RES)                     0
22                   gcd                   25 (ID)       loc9 (ptr to sym tab)
22                   :=                    21 (ASSIGNOP) 1 (ASSIGN)
22                   a                     25 (ID)       loc10 (ptr to sym tab)
23                   else          12 (RES)                     0
23                   gcd                   25 (ID)       loc9 (ptr to sym tab)
23                   :=                    21 (ASSIGNOP) 1 (ASSIGN)
23                   gcd                   25 (ID)       loc9 (ptr to sym tab)
23                   (                     4 (CATCHALL)  3 (LEFTPAREN)
23                   b                     25 (ID)       loc11 (ptr to sym tab)
23                   ,                     4 (CATCHALL)         7 (COMMA)
23                   a                     25 (ID)       loc10 (ptr to sym tab)
23                   mod           3 (RES)              4
23                   b                     25 (ID)       loc11 (ptr to sym tab)
23                   )                     4 (CATCHALL)  4 (RIGHTPAREN)
24                   end           6 (RES)              0
24                   ;                     4 (CATCHALL)  5 (SEMICOLON)
                     98 (EOF)                     0 (NULL)
```

Output: “SourceErrors.txt”: Listing File

```
1
2             reallylongword
LEXERR:Word too long:reallylongword
3
4             555556.01;
LEXERR:Real first part too long:    555556
5             4444.555556;
LEXERR:Real second part too long:   4444.555556
6             00;
LEXERR:Leading zero:  0
7             01;
LEXERR:Leading zero:  0
8             1.;
9             12345678901;
10            10101010101;
LEXERR:Int too long:  10101010101
11            1.0000001;
LEXERR:Real first part too long:    1
LEXERR:Real second part too long:   1.0000001
12            5.26EE82;
13            9.99E222;
LEXERR:Exponent too long:    9.99E222
14
15            # @ ! $ %
LEXERR:Unrecognized Symbol:  #
LEXERR:Unrecognized Symbol:  @
LEXERR:Unrecognized Symbol:  !
LEXERR:Unrecognized Symbol:  $
LEXERR:Unrecognized Symbol:  %
16
17            9programme example(input, output);
18            var x, y: integer;
19            var c, d: realaaaaaaa;
LEXERR:Word too long:realaaaaaaa
20            function gcd(a, b: integer): integer;
21            begin
22                   if b = 01 then gcd := a
LEXERR:Leading zero:  0
23                   else gcd := gcd(b, a mod b)
24            end;
```

Appendix II: Program Listings