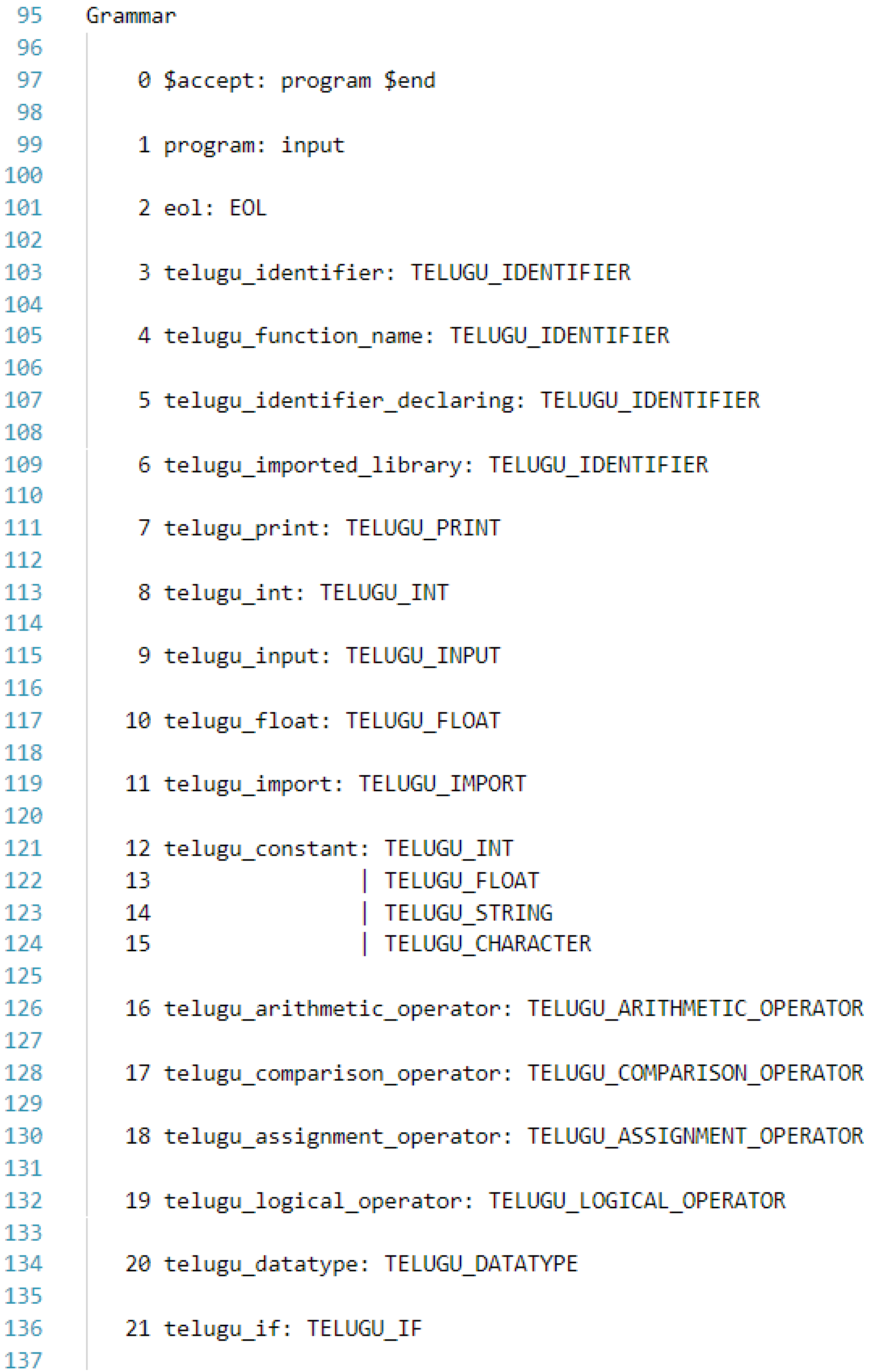
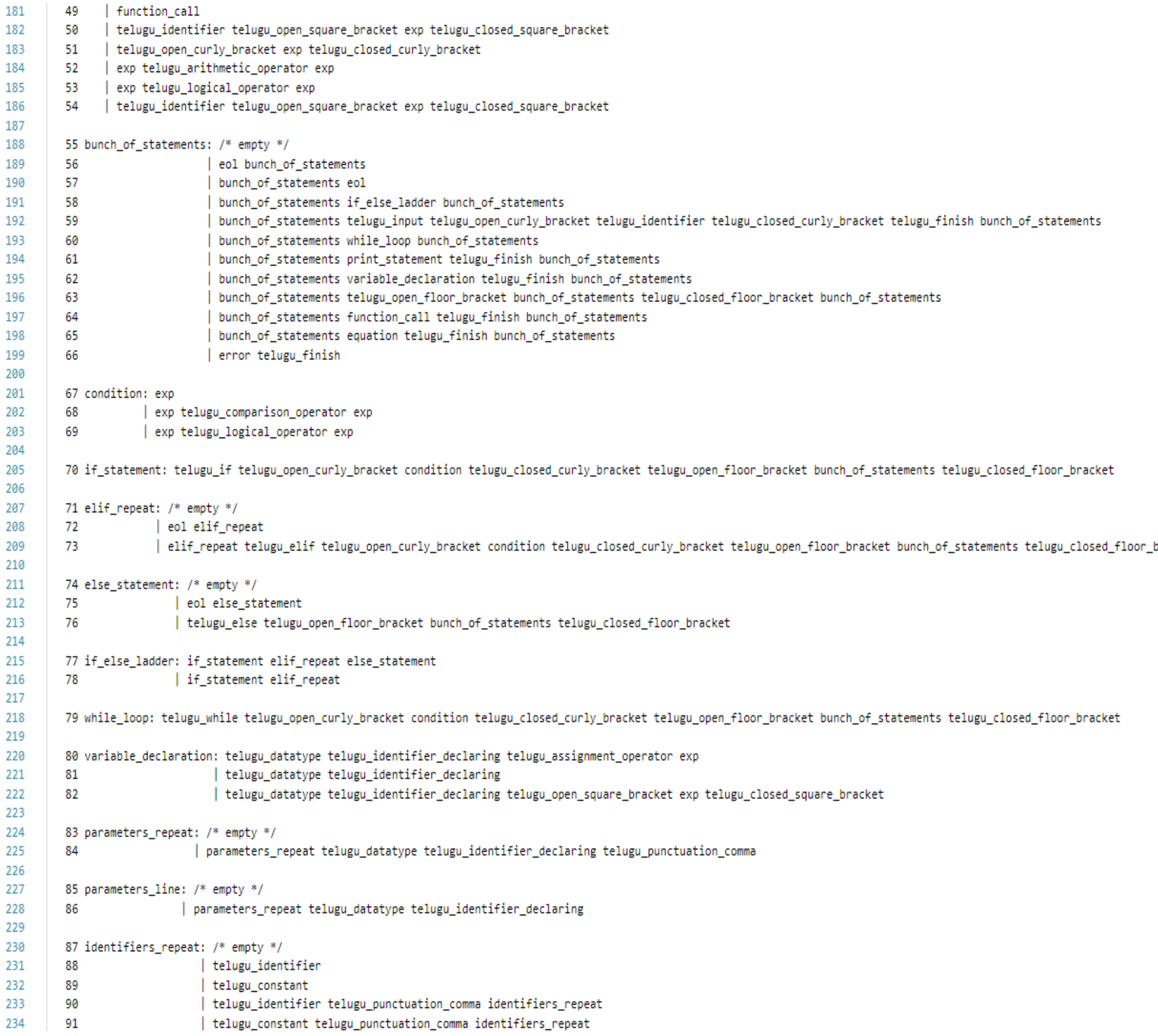
TELUGU GRAMMAR RULES DEFINITION

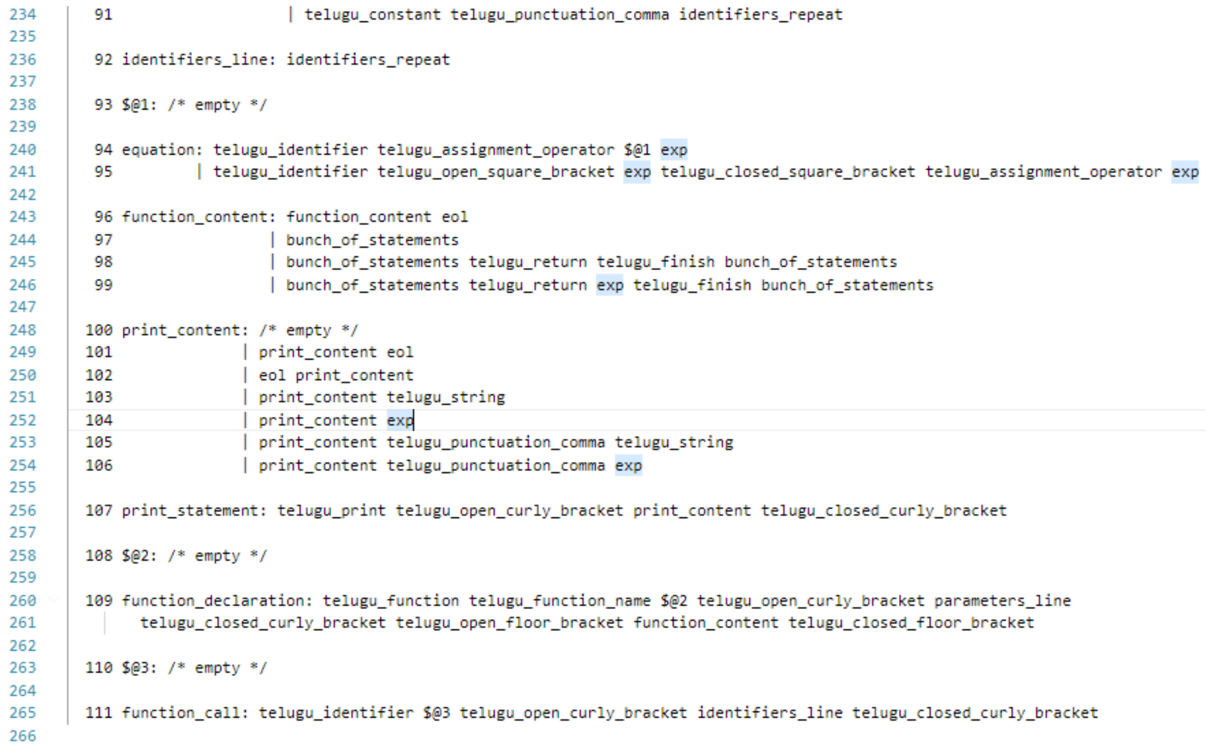


```
Grammar

    0 $accept: program $end

    1 program: input

    2 eol: EOL

    3 telugu_identifier: TELUGU_IDENTIFIER

    4 telugu_function_name: TELUGU_IDENTIFIER

    5 telugu_identifier_declaring: TELUGU_IDENTIFIER

    6 telugu_imported_library: TELUGU_IDENTIFIER

    7 telugu_print: TELUGU_PRINT

    8 telugu_int: TELUGU_INT

    9 telugu_input: TELUGU_INPUT

   10 telugu_float: TELUGU_FLOAT

   11 telugu_import: TELUGU_IMPORT

   12 telugu_constant: TELUGU_INT
   13                | TELUGU_FLOAT
   14                | TELUGU_STRING
   15                | TELUGU_CHARACTER

   16 telugu_arithmetic_operator: TELUGU_ARITHMETIC_OPERATOR

   17 telugu_comparison_operator: TELUGU_COMPARISON_OPERATOR

   18 telugu_assignment_operator: TELUGU_ASSIGNMENT_OPERATOR

   19 telugu_logical_operator: TELUGU_LOGICAL_OPERATOR

   20 telugu_datatype: TELUGU_DATATYPE

   21 telugu_if: TELUGU_IF
```

```
22 telugu_elif: TELUGU_ELIF

23 telugu_else: TELUGU_ELSE

24 telugu_while: TELUGU_WHILE

25 telugu_string: TELUGU_STRING

26 telugu_open_curly_bracket: TELUGU_OPEN_CURLY_BRACKET

27 telugu_closed_curly_bracket: TELUGU_CLOSED_CURLY_BRACKET

28 telugu_open_square_bracket: TELUGU_OPEN_SQUARE_BRACKET

29 telugu_closed_square_bracket: TELUGU_CLOSED_SQUARE_BRACKET

30 telugu_open_floor_bracket: TELUGU_OPEN_FLOOR_BRACKET

31 telugu_closed_floor_bracket: TELUGU_CLOSED_FLOOR_BRACKET

32 telugu_punctuation_comma: TELUGU_PUNCTUATION_COMMA

33 telugu_finish: TELUGU_FINISH

34 telugu_function: TELUGU_FUNCTION

35 telugu_return: TELUGU_RETURN

36 telugu_character: TELUGU_CHARACTER

37 input: /* empty */
38      | input eol
39      | eol input
40      | input telugu_import telugu_imported_library telugu_finish
41      | telugu_import telugu_imported_library telugu_finish input
42      | input bunch_of_statements input
43      | input function_declaration input

44 exp: telugu_int
45    | telugu_float
46    | telugu_character
47    | telugu_string
48    | telugu_identifier
49    | function call
```



```
49    | function_call
50    | telugu_identifier telugu_open_square_bracket exp telugu_closed_square_bracket
51    | telugu_open_curly_bracket exp telugu_closed_curly_bracket
52    | exp telugu_arithmetic_operator exp
53    | exp telugu_logical_operator exp
54    | telugu_identifier telugu_open_square_bracket exp telugu_closed_square_bracket

55 bunch_of_statements: /* empty */
56                    | eol bunch_of_statements
57                    | bunch_of_statements eol
58                    | bunch_of_statements if_else_ladder bunch_of_statements
59                    | bunch_of_statements telugu_input telugu_open_curly_bracket telugu_identifier telugu_closed_curly_bracket telugu_finish bunch_of_statements
60                    | bunch_of_statements while_loop bunch_of_statements
61                    | bunch_of_statements print_statement telugu_finish bunch_of_statements
62                    | bunch_of_statements variable_declaration telugu_finish bunch_of_statements
63                    | bunch_of_statements telugu_open_floor_bracket bunch_of_statements telugu_closed_floor_bracket bunch_of_statements
64                    | bunch_of_statements function_call telugu_finish bunch_of_statements
65                    | bunch_of_statements equation telugu_finish bunch_of_statements
66                    | error telugu_finish

67 condition: exp
68          | exp telugu_comparison_operator exp
69          | exp telugu_logical_operator exp

70 if_statement: telugu_if telugu_open_curly_bracket condition telugu_closed_curly_bracket telugu_open_floor_bracket bunch_of_statements telugu_closed_floor_bracket

71 elif_repeat: /* empty */
72            | eol elif_repeat
73            | elif_repeat telugu_elif telugu_open_curly_bracket condition telugu_closed_curly_bracket telugu_open_floor_bracket bunch_of_statements telugu_closed_floor_b

74 else_statement: /* empty */
75               | eol else_statement
76               | telugu_else telugu_open_floor_bracket bunch_of_statements telugu_closed_floor_bracket

77 if_else_ladder: if_statement elif_repeat else_statement
78               | if_statement elif_repeat

79 while_loop: telugu_while telugu_open_curly_bracket condition telugu_closed_curly_bracket telugu_open_floor_bracket bunch_of_statements telugu_closed_floor_bracket

80 variable_declaration: telugu_datatype telugu_identifier_declaring telugu_assignment_operator exp
81                     | telugu_datatype telugu_identifier_declaring
82                     | telugu_datatype telugu_identifier_declaring telugu_open_square_bracket exp telugu_closed_square_bracket

83 parameters_repeat: /* empty */
84                  | parameters_repeat telugu_datatype telugu_identifier_declaring telugu_punctuation_comma

85 parameters_line: /* empty */
86                | parameters_repeat telugu_datatype telugu_identifier_declaring

87 identifiers_repeat: /* empty */
88                   | telugu_identifier
89                   | telugu_constant
90                   | telugu_identifier telugu_punctuation_comma identifiers_repeat
91                   | telugu_constant telugu_punctuation_comma identifiers_repeat
```



```
 91                  | telugu_constant telugu_punctuation_comma identifiers_repeat

 92 identifiers_line: identifiers_repeat

 93 $@1: /* empty */

 94 equation: telugu_identifier telugu_assignment_operator $@1 exp
 95         | telugu_identifier telugu_open_square_bracket exp telugu_closed_square_bracket telugu_assignment_operator exp

 96 function_content: function_content eol
 97                 | bunch_of_statements
 98                 | bunch_of_statements telugu_return telugu_finish bunch_of_statements
 99                 | bunch_of_statements telugu_return exp telugu_finish bunch_of_statements

100 print_content: /* empty */
101              | print_content eol
102              | eol print_content
103              | print_content telugu_string
104              | print_content exp
105              | print_content telugu_punctuation_comma telugu_string
106              | print_content telugu_punctuation_comma exp

107 print_statement: telugu_print telugu_open_curly_bracket print_content telugu_closed_curly_bracket

108 $@2: /* empty */

109 function_declaration: telugu_function telugu_function_name $@2 telugu_open_curly_bracket parameters_line
      telugu_closed_curly_bracket telugu_open_floor_bracket function_content telugu_closed_floor_bracket

110 $@3: /* empty */

111 function_call: telugu_identifier $@3 telugu_open_curly_bracket identifiers_line telugu_closed_curly_bracket
```

Verbal Grammar Definition

Variable Declaration:

sankhya a=3;

multiple variables declaration in the same line is not allowed like: sankhya a=3,b=4;

Headers:

thechko library_name;

multiple imports and renaming libraries like:

thechko lib1,lib2;   // not allowed

thechko lib1 as my_lib;   // not allowed

Function Declaration/Call:

```
function function_name(condition){
    // bunch of statements
    // function declarations not allowed in here
    //can have return statement like ‘return;’ or ‘return exp;’
}
```

Input/Output Commands:

```
chupi(var1_value);
chupi(“any_string”);
chupi(“string1”, var1_value, “string2”, var2_value);
theesko(var1);
```

If-Else Ladder:

```
okavela(condition_1){   //flower braces are must
    //bunch_of_statements
}
lekaokavela(condition_2){       // elseIf can repeat 0 or more number of times
    //bunch_of_statements
}
lekapothe{                      // need not be
    //bunch_of_statements
}
```

Loop:

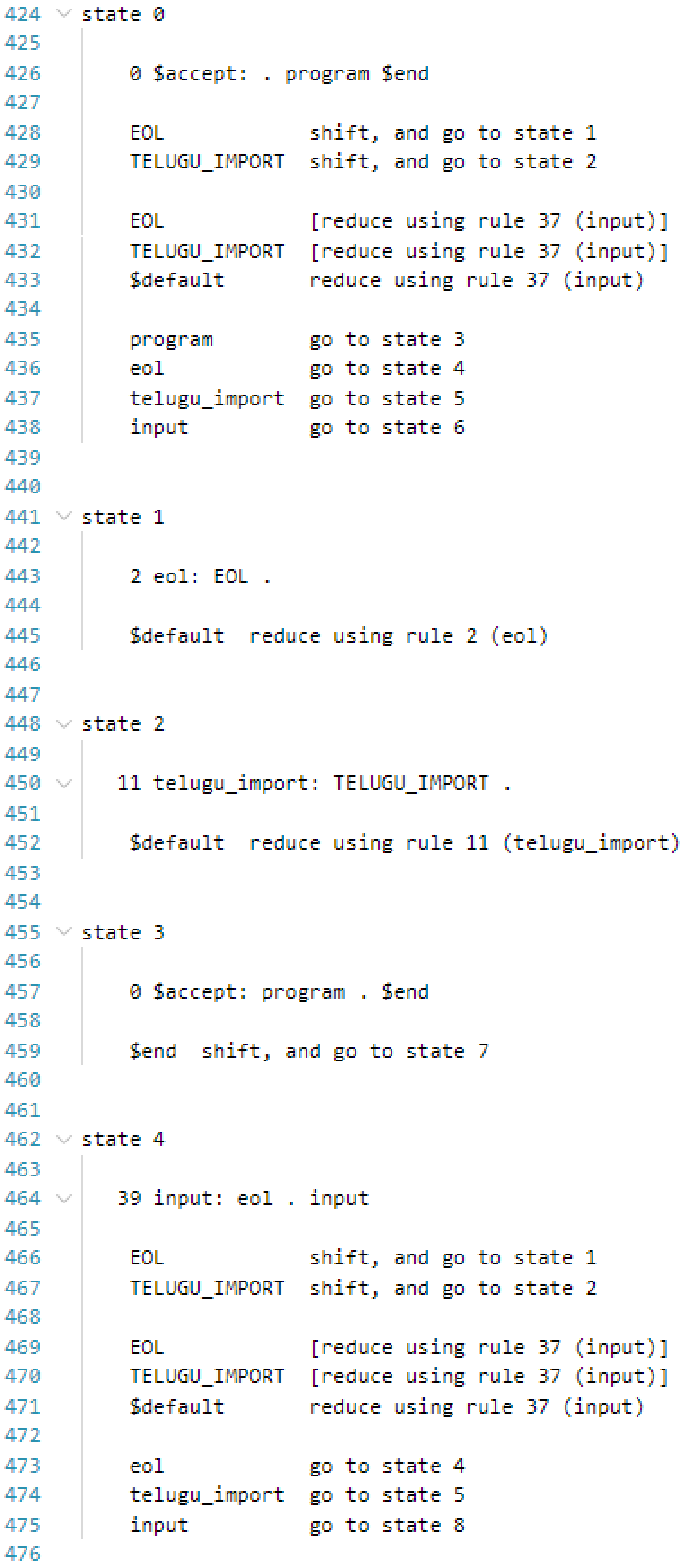
```
aithaunte(condition){    // only while loop allowed
    //bunch_of_statements
}
```

Array Indexing:

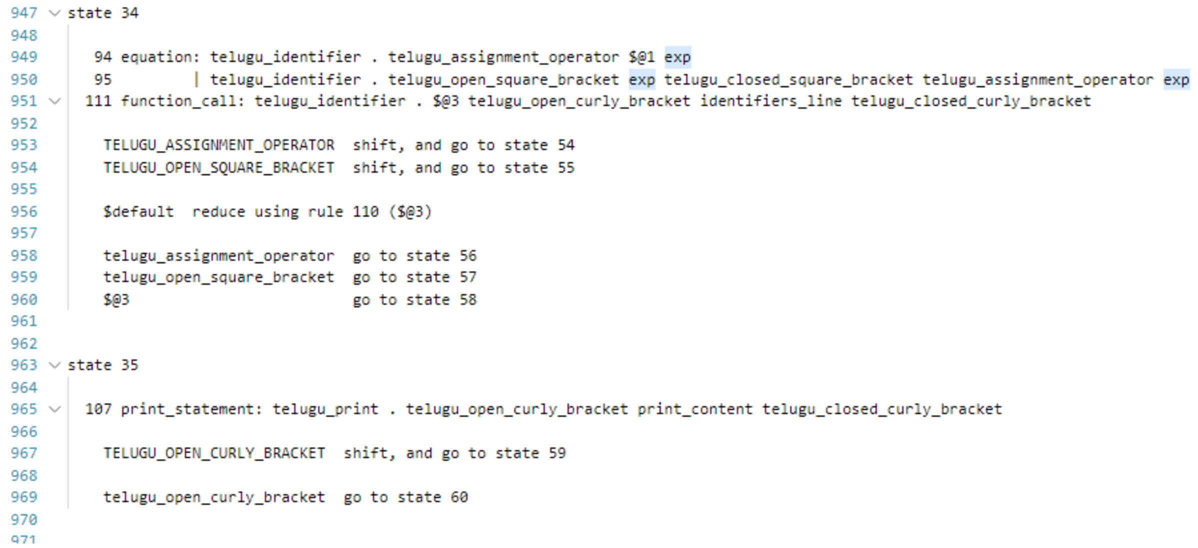
```
sankhya arr[15];
arr[3];
arr[2+1*(x)+4];     // arr[any valid expression] is allowed
```

PARSE TABLE:

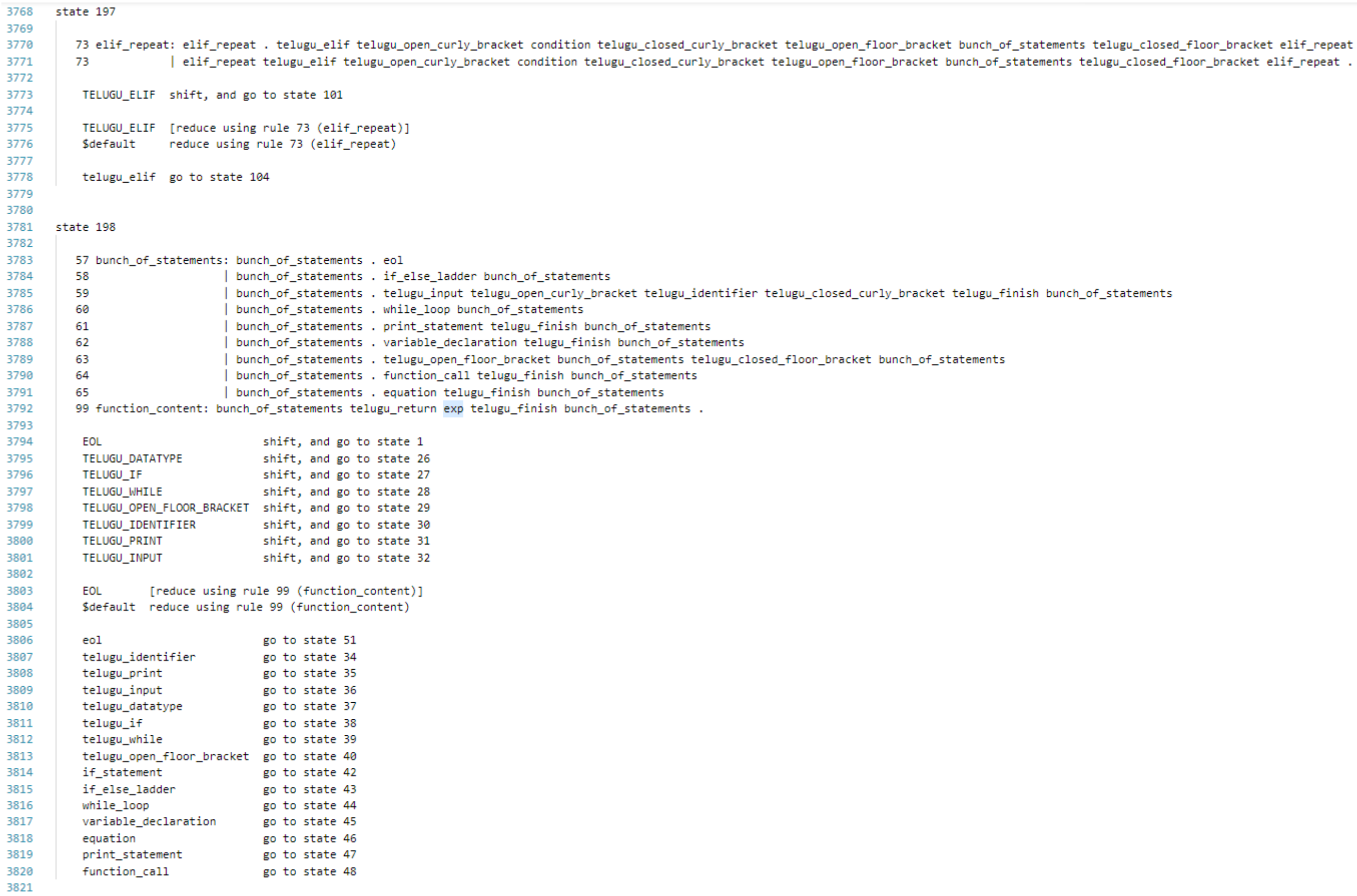
(Only few lines of the parse table are shown below)



```
state 0

    0 $accept: . program $end

    EOL            shift, and go to state 1
    TELUGU_IMPORT  shift, and go to state 2

    EOL            [reduce using rule 37 (input)]
    TELUGU_IMPORT  [reduce using rule 37 (input)]
    $default       reduce using rule 37 (input)

    program        go to state 3
    eol            go to state 4
    telugu_import  go to state 5
    input          go to state 6


state 1

    2 eol: EOL .

    $default  reduce using rule 2 (eol)


state 2

   11 telugu_import: TELUGU_IMPORT .

    $default  reduce using rule 11 (telugu_import)


state 3

    0 $accept: program . $end

    $end  shift, and go to state 7


state 4

   39 input: eol . input

    EOL            shift, and go to state 1
    TELUGU_IMPORT  shift, and go to state 2

    EOL            [reduce using rule 37 (input)]
    TELUGU_IMPORT  [reduce using rule 37 (input)]
    $default       reduce using rule 37 (input)

    eol            go to state 4
    telugu_import  go to state 5
    input          go to state 8

```



```
state 34

  94 equation: telugu_identifier . telugu_assignment_operator $@1 exp
  95         | telugu_identifier . telugu_open_square_bracket exp telugu_closed_square_bracket telugu_assignment_operator exp
 111 function_call: telugu_identifier . $@3 telugu_open_curly_bracket identifiers_line telugu_closed_curly_bracket

    TELUGU_ASSIGNMENT_OPERATOR  shift, and go to state 54
    TELUGU_OPEN_SQUARE_BRACKET  shift, and go to state 55

    $default  reduce using rule 110 ($@3)

    telugu_assignment_operator  go to state 56
    telugu_open_square_bracket  go to state 57
    $@3                         go to state 58


state 35

 107 print_statement: telugu_print . telugu_open_curly_bracket print_content telugu_closed_curly_bracket

    TELUGU_OPEN_CURLY_BRACKET  shift, and go to state 59

    telugu_open_curly_bracket  go to state 60

```



```
state 197

   73 elif_repeat: elif_repeat . telugu_elif telugu_open_curly_bracket condition telugu_closed_curly_bracket telugu_open_floor_bracket bunch_of_statements telugu_closed_floor_bracket elif_repeat
   73            | elif_repeat telugu_elif telugu_open_curly_bracket condition telugu_closed_curly_bracket telugu_open_floor_bracket bunch_of_statements telugu_closed_floor_bracket elif_repeat .

    TELUGU_ELIF  shift, and go to state 101

    TELUGU_ELIF  [reduce using rule 73 (elif_repeat)]
    $default     reduce using rule 73 (elif_repeat)

    telugu_elif  go to state 104


state 198

   57 bunch_of_statements: bunch_of_statements . eol
   58                    | bunch_of_statements . if_else_ladder bunch_of_statements
   59                    | bunch_of_statements . telugu_input telugu_open_curly_bracket telugu_identifier telugu_closed_curly_bracket telugu_finish bunch_of_statements
   60                    | bunch_of_statements . while_loop bunch_of_statements
   61                    | bunch_of_statements . print_statement telugu_finish bunch_of_statements
   62                    | bunch_of_statements . variable_declaration telugu_finish bunch_of_statements
   63                    | bunch_of_statements . telugu_open_floor_bracket bunch_of_statements telugu_closed_floor_bracket bunch_of_statements
   64                    | bunch_of_statements . function_call telugu_finish bunch_of_statements
   65                    | bunch_of_statements . equation telugu_finish bunch_of_statements
   99 function_content: bunch_of_statements telugu_return exp telugu_finish bunch_of_statements .

    EOL                        shift, and go to state 1
    TELUGU_DATATYPE            shift, and go to state 26
    TELUGU_IF                  shift, and go to state 27
    TELUGU_WHILE               shift, and go to state 28
    TELUGU_OPEN_FLOOR_BRACKET  shift, and go to state 29
    TELUGU_IDENTIFIER          shift, and go to state 30
    TELUGU_PRINT               shift, and go to state 31
    TELUGU_INPUT               shift, and go to state 32

    EOL       [reduce using rule 99 (function_content)]
    $default  reduce using rule 99 (function_content)

    eol                        go to state 51
    telugu_identifier          go to state 34
    telugu_print               go to state 35
    telugu_input               go to state 36
    telugu_datatype            go to state 37
    telugu_if                  go to state 38
    telugu_while               go to state 39
    telugu_open_floor_bracket  go to state 40
    if_statement               go to state 42
    if_else_ladder             go to state 43
    while_loop                 go to state 44
    variable_declaration       go to state 45
    equation                   go to state 46
    print_statement            go to state 47
    function_call              go to state 48

```