1 Regular Grammar for CUTE Lang

This chapter explains what regular grammar is used in CUTE Lang.

1.1 EBNF

To explain without any complexity, grammar is explained by EBNF (Extended BNF).

EBNF for Regular Grammar

```
program                 = { whites | lexeme } .

whites                  = white .
white                   = whitechar | comment .

whitechar               = newline | vtab | tab | space .
newline                 = cr lf | cr | lf | ff .
cr                      = ? CARRIAGE RETURN ? .
lf                      = ? LINE FEED ? .
ff                      = ? FORM FEED ? .
vtab                    = ? VERTICAL TAB ? .
tab                     = ? HORIZONTAL TAB ? .
space                   = ? SPACE ? .

comment                 = sl com | ml com .
sl com                  = sl beg, { sl any }, [ newline ] .
sl beg                  = "//" .
sl any                  = visible | space | tab .
ml com                  = ml beg, ml anys, { ml com, ml anys }, ml end .
ml beg                  = "/*" .
ml end                  = "*/" .
ml anys                 = { ml any }
                          - ( { ml any }, ( ml beg | ml end ), { ml any } ) .
ml any                  = printable .

lexeme                  = bi sym | un sym | special | res sym | res id
                        | literal | id .
special                 = "{" | "}" | "," | ";" | "[" | "]" | "(" | ")" .
literal                 = byte | integer | float
                        | char | string .

integer                 = "0", ( "b" | "B" ), binary
                        | "0", ( "o" | "O" ), octal
                        | [ "0", ( "d" | "D" ) ], decimal
                        | "0", ( "x" | "X" ), hexadecimal .
binary                  = binit, { binit } .
octal                   = octit, { octit } .
decimal                 = digit, { digit } .
hexadecimal             = hexit, { hexit } .
binit                   = "0" | "1" .
octit                   = binit
                        | "2" | "3" | "4" | "5" | "6" | "7" .
digit                   = octit
                        | "8" | "9" .
hexit                   = digit
                        | "a" | "b" | "c" | "d" | "e" | "f"
                        | "A" | "B" | "C" | "D" | "E" | "F" .
```

```
float                = decimal, ".", decimal, [ exponent ]
                     | decimal, exponent .
exponent             = ( "e" | "E" ), [ "+" | "-" ], decimal .

char                 = "'" ( visible - ( "'" | "\" ) | space | escape ) "'" .
string               = '"' { visible - ( '"' | "\" ) | space | escape } '"' .
printable            = visible | whitechar
visible              = lower | upper | bi sym | un sym | digit | special .
lower                = "a" | "b" | "c" | "d" | "e" | "f" | "g"
                     | "h" | "i" | "j" | "k" | "l" | "m" | "n"
                     | "o" | "p" | "q" | "r" | "s" | "t" | "u"
                     | "v" | "w" | "x" | "y" | "z" .
upper                = "A" | "B" | "C" | "D" | "E" | "F" | "G"
                     | "H" | "I" | "J" | "K" | "L" | "M" | "N"
                     | "O" | "P" | "Q" | "R" | "S" | "T" | "U"
                     | "V" | "W" | "X" | "Y" | "Z" .
```