

MiniC Language Manual

Name: Swetanjali Dutta

Roll Number: 20171077

31/08/2020

1 Macro Syntax Specification using Context Free Grammars

1.1 Meta Notation:

- $\langle \text{foo} \rangle$ means foo is a non terminal.
- **foo**(in bold font) means foo is a terminal i.e a token.
- $[x]$ means zero or one occurrence of x i.e x is optional. Note that *brackets in quotes* i.e '[' and ']' are terminals.
- x^* means zero or more occurrences of x .
- x^+ means one or more occurrences of x .
- $x^+,$ means a comma separated list of one or more x s.
- $\{\}$ i.e large braces are used for grouping. Note that *braces in quotes* i.e '{' and '}' are terminals.
- $|$ separates alternatives.

1.2 Production Rules:

1. $\langle \text{program} \rangle \rightarrow \langle \text{decl} \rangle^+$
2. $\langle \text{decl} \rangle \rightarrow \langle \text{var_decl} \rangle \mid \langle \text{method_decl} \rangle$
3. $\langle \text{var_decl} \rangle \rightarrow \langle \text{type} \rangle \langle \text{identifier} \rangle^+, ;$
4. $\langle \text{method_decl} \rangle \rightarrow \{ \langle \text{type} \rangle \mid \mathbf{VOID} \} \mathbf{ID} ([\{ \langle \text{type} \rangle \langle \text{identifier} \rangle^+,]) \langle \text{block} \rangle$
5. $\langle \text{block} \rangle \rightarrow \{ ' \langle \text{var_decl} \rangle^* \langle \text{statement} \rangle^* \}$
6. $\langle \text{type} \rangle \rightarrow \mathbf{INT} \mid \mathbf{UINT} \mid \mathbf{BOOL} \mid \mathbf{CHAR}$
7. $\langle \text{statement} \rangle \rightarrow \langle \text{assignment} \rangle^+, ;$
| $\langle \text{method_call} \rangle;$
| $\mathbf{IF} (\langle \text{expr} \rangle) \langle \text{block} \rangle [\mathbf{ELSE} \langle \text{block} \rangle]$
| $\mathbf{FOR} ([\langle \text{assignment} \rangle^+,]; [\langle \text{expr} \rangle]; [\langle \text{expr} \rangle]) \langle \text{block} \rangle$
| $\mathbf{WHILE} (\langle \text{expr} \rangle) \langle \text{block} \rangle$
| $\mathbf{BREAK};$
| $\mathbf{CONTINUE};$
| $\langle \text{block} \rangle$
| $\mathbf{RETURN} [\langle \text{expr} \rangle];$
| $\mathbf{PRINT} (\langle \text{expr} \rangle);$
8. $\langle \text{assignment} \rangle \rightarrow \langle \text{identifier} \rangle \mathbf{ASSIGN} \langle \text{expr} \rangle$
9. $\langle \text{method_call} \rangle \rightarrow \mathbf{ID} ([\langle \text{expr} \rangle^+,])$

10. $\langle \text{expr} \rangle \rightarrow \langle \text{identifier} \rangle$
 $\quad \quad \quad \langle \text{expr} \rangle \langle \text{arithmetic_op} \rangle \langle \text{expr} \rangle$
 $\quad \quad \quad \langle \text{expr} \rangle \langle \text{relational_op} \rangle \langle \text{expr} \rangle$
 $\quad \quad \quad \langle \text{expr} \rangle \langle \text{conditional_op} \rangle \langle \text{expr} \rangle$
 $\quad \quad \quad \langle \text{expr} \rangle \langle \text{equality_op} \rangle \langle \text{expr} \rangle$
 $\quad \quad \quad \langle \text{expr} \rangle \textbf{THEN} \langle \text{expr} \rangle \textbf{OTHERWISE} \langle \text{expr} \rangle$
 $\quad \quad \quad \langle \text{literal} \rangle$
 $\quad \quad \quad \langle \text{method_call} \rangle$
 $\quad \quad \quad \textbf{NOT} \langle \text{expr} \rangle$
 $\quad \quad \quad \textbf{NEGATE} \langle \text{expr} \rangle$
 $\quad \quad \quad (\langle \text{expr} \rangle)$
11. $\langle \text{identifier} \rangle \rightarrow \textbf{ID} \mid \textbf{ID}\{ '[' \langle \text{expr} \rangle ']' \}^*$
12. $\langle \text{literal} \rangle \rightarrow \textbf{INT_LIT} \mid \textbf{FLOAT_LIT} \mid \textbf{CHAR_LIT} \mid \langle \text{bool_lit} \rangle$
13. $\langle \text{bool_lit} \rangle \rightarrow \textbf{TRUE} \mid \textbf{FALSE}$
14. $\langle \text{arithmetic_op} \rangle \rightarrow \textbf{ADD} \mid \textbf{SUB} \mid \textbf{MUL} \mid \textbf{DIV} \mid \textbf{MOD}$
15. $\langle \text{relational_op} \rangle \rightarrow \textbf{LT} \mid \textbf{GT} \mid \textbf{LE} \mid \textbf{GE}$
16. $\langle \text{conditional_op} \rangle \rightarrow \textbf{AND} \mid \textbf{OR}$
17. $\langle \text{equality_op} \rangle \rightarrow \textbf{EQ} \mid \textbf{NE}$

1.3 Start Symbol:

- program

2 Micro Syntax Specification using Regular Expressions

2.1 Meta Notation:

- Token Type \rightarrow Lexeme

2.2 Rules:

1. FALSE \rightarrow false
2. TRUE \rightarrow true
3. NOT \rightarrow !
4. NEGATE \rightarrow -
5. VOID \rightarrow void
6. INT \rightarrow int
7. UNINT \rightarrow uint
8. CHAR \rightarrow char
9. BOOL \rightarrow bool
10. THEN \rightarrow ?
11. OTHERWISE \rightarrow :
12. FOR \rightarrow for
13. WHILE \rightarrow while

- 14. IF \rightarrow if
- 15. ELSE \rightarrow else
- 16. BREAK \rightarrow break
- 17. RETURN \rightarrow return
- 18. ADD \rightarrow +
- 19. SUB \rightarrow -
- 20. MUL \rightarrow *
- 21. DIV \rightarrow /
- 22. MOD \rightarrow %
- 23. LT \rightarrow <
- 24. GT \rightarrow >
- 25. LE \rightarrow <=
- 26. GE \rightarrow >=
- 27. AND \rightarrow &&
- 28. OR \rightarrow ||
- 29. EQ \rightarrow ==
- 30. NE \rightarrow !=
- 31. ASSIGN \rightarrow =
- 32. PRINT \rightarrow print
- 33. INT_LIT \rightarrow [0-9][0-9]*
- 34. FLOAT_LIT \rightarrow [0-9][0-9]*.[0-9][0-9]*
- 35. CHAR_LIT \rightarrow '[a-zA-Z0-9]'
- 36. ID \rightarrow [a-zA-Z][a-zA-Z0-9_]*

3 Lexical Considerations

4 Semantic Checks