

Terminals (lexers output):

- id
- const
- lit
- punct: { , } () : , ,
- op: * + / - =
- rel
- datatype
- keyword: while if else return

Grammar rules:

1. $S \rightarrow \text{datatype } S' \mid \lambda$
2. $S' \rightarrow \text{id ARGUMENTS FUNCTION } S$
3. $\text{ARGUMENTS} \rightarrow (\text{VALUES})$
4. $\text{VALUES} \rightarrow \text{datatype id VALUES'}$
5. $\text{VALUES}' \rightarrow \lambda \mid , \text{VALUES}$
6. $\text{FUNCTION} \rightarrow \{ \text{LINES RETURN} \}$
7. $\text{LINES} \rightarrow ((\text{DECLARE} \mid \text{MODIFY} \mid \text{while V_FUNCTION} \mid \text{if V_FUNCTION ELSE}) \text{LINES}) \mid \lambda$
8. $\text{RETURN} \rightarrow \text{return OPERATION};$
9. $\text{DECLARE} \rightarrow \text{datatype id DECLARE'}$
10. $\text{DECLARE}' \rightarrow \lambda \mid = \text{OPERATION}$
11. $\text{MODIFY} \rightarrow \text{id} = \text{OPERATION};$
12. $\text{V_FUNCTION} \rightarrow (\text{COMPARE}) \{ \text{LINES} \}$
13. $\text{ELSE} \rightarrow \lambda \mid \text{else } \{ \text{LINES} \}$
14. $\text{OPERATION} \rightarrow \text{S_VALUE } (\lambda \mid (+ \mid - \mid * \mid /) \text{S_VALUE})$

- 15. COMPARE \rightarrow OPERATION rel OPERATION
- 16. S_VALUE \rightarrow id C_FUNCTION | const | lit
- 17. C_FUNCTION \rightarrow D_ARGS | λ
- 18. D_ARGS \rightarrow (E_VALUES)
- 19. E_VALUES \rightarrow S_VALUE E_VALUES'
- 20. E_VALUES' \rightarrow λ | , E_VALUES