Gramática Linguagem L

S → DB

D → final id “=“ [“-”] const”;” |

Tipo id [“=“ [“-”] const] {“,” id [“=“ [-] const}\* “;”

Tipo → int | byte | string | boolean

B → begin {C}\* end

C → id = Exp “;” | while Exp ( C | B ) |

If Exp ( C | B ) [else ( C | B ) ] | “;” |

readln”,” id “;” | (write | writeln)”,” Exp {“,” Exp }\* “;”

Exp → ExpS [ ( “<“ | “>“ | “<=“ | “>=“ | “==“ | “!=“ ) ExpS ]

ExpS → [ “+” | “-” T { ( “+” | “-” | or ) T }\*

T → F { ( “\*” | “/” | and ) F }\*

F → “(“ Exp “)” | not F | const | id