***Updated Grammar***

**<language>::=<functions> <Main\_body>**

**<Main\_body>::=BEGIN <Statements> END**

**<statement>::= <statement> <statement> | <statement\_w\_endl> NEWL | <statement\_wo\_endl>| ε**

**<statement\_w\_endl>::=ID[<function\_call> | <assignmentArray> |<arith\_op>]| <return>**

**<statement\_wo\_endl>::= <loop > | <if\_statement>**

**<functions>::=<functionRE>| ε**

**<functionRE>::= DEFINE ID <parse\_functionCall> OPCURL <statements> CLCURL**

**<parse\_functionCall>::= ID OPPARENT <parse\_argument\_list>**

* Functions:

**<parse\_argument\_list>:== <parse\_expression> <variable> CLPARENT**

**<variables> :== <value> <variable> | ε**

**<variable> :== COMMA <parse\_expression>**

**<value>::= NUM | ID**

* Conditions

**<Condition> ::= OPPARENT <comparisons><optional\_logic> CLPARENT**

**<optional\_logic> ::= <Logic\_op> <comparisons> <optional\_logic> | ε**

**<comparisons> ::= <value><comparison\_op><value>**

**<comparison\_op>::=EQU | SMALL | SMALLQUI | NOTEQUI**

**<Logic\_op>::= AND| OR| NOT**

* Arithmetic operations:

**<arith\_op> ::= <check> ASSIGN <arith\_expr>**

**<arith\_expr> ::= <term> <arith\_expr\_tail>**

**<arith\_expr\_tail> ::= <add\_op> <term> <arith\_expr\_tail> | ε**

**<term> ::= ID | NUM**

**<add\_op> ::= ADD | SUB | DIV | MUL**

assignment:

**<assignmentArray>::= ID<array> = <zorto>**

**<zorto>::= ID CONCA <array>| ID**

**<Parse\_expression> ::= ID |NUM| WU |AG |PT |GO |BR |GL |ST| |SO| NO | WE | EA |SC | BU | EM | ε**

* Array Declaration:

**<array> ::= <size><size>**

**<size>::= OPBRACK NUM CLBRACKET**

* Repetition:

**<loop>::=LOOP<condition>OPCURL <statements>CLCURL**

* Selection:

**<if\_statement>::=IF<condition>OPCURL<Statements>CLCURL<else\_statement>**

**<else\_statement>::=ELSE OPCURL<statements>CLCURL | ε**

Return:

<return> ::= RETURN ID