Gramática Linguagem NOVA (EBNF)

<program>::
<function> <main>

<function>:: <function_declaration>
| empty

<function_declaration>::
<type> ID AB_PAR <parameters> FEC_PAR <scope> SP1

<type>:: PR_INT
| PR_FLOAT
| PR_BOOL
| PR_STRING
| PR_VOID

<parameters>::
<type> ID
| <type> ID SP2 <parameters>
| empty

<scope>:: AB_CH <commands> FEC_CH SP1

<commands>::
<cmd> SP1 <commands>
| empty

<cmd>:: <declaration>
| <attribution>
| <function_call>
| <printout>
| <readin>
| <ifelse>
| <while>
| <for>
| <shoot>

<declaration>::
<type> ID SP1

<attribution>::
ID OP_ATR <value>

<value>:: <array>
| <expression>

<array>:: AB_CH <elements> FEC_CH
| AB_CH FEC_CH

<elements>:: <constant>
| <constant> SP2 <elements>

<constant>:: CTE_FLOAT
| CTE_INT
| CTE_STR
| CTE_BOOL

<expression>::
<logical_expression> OP_ATR <expression>
| <logical_expression>

<logical_expression>::
<eq_expression> OP_AND <logical_expression>
| <eq_expression> OP_OR <logical_expression>
| <eq_expression>

<eq_expression>::
<comparative_exp> OP_IG <eq_expression>
| <comparative_exp> OP_DIF <eq_expression>
| <comparative_exp>

<comparative_exp>::
<add_exp> OP_MEQ <comparative_exp>
| <add_exp> OP_MAQ <comparative_exp>
| <add_exp> OP_MEIGQ <comparative_exp>
| <add_exp> OP_MAIGQ <comparative_exp>
| <add_exp>

<add_exp>::
<mult_exp> OP_AD <add_exp>
| <mult_exp> OP_SUB <add_exp>
| <mult_exp>

<mult_exp>::
<neg_exp> OP_MULT <mult_exp>
| <neg_exp> OP_DIV <mult_exp>
| <neg_exp> OP_MOD <mult_exp>
| <neg_exp>

<neg_exp>::

OP_NEG <exp_aux>
| <exp_aux>

<exp_aux>::

AB_PAR <atom_exp> FEC_PAR
| <atom_exp>

<atom_exp>::

ID
| CTE_INT
| CTE_FLOAT
| CTE_STRING
| PR_TRUE
| PR_FALSE

<function_call>::

ID AB_PAR <parameters_call> FEC_PAR

<parameters_call>::

<parameter_item>
| <parameter_item> SP2 <parameters_call>

<parameter_item>::

<constant>
| ID

<printout>::

PRINTOUT AB_PAR <message> FEC_PAR

<message>::

CTE_STR
| CTE_STR OP_AD <message>

<readin>::

READ_IN AB_PAR <type> SP2 ID FEC_PAR

<ifelse>::

<if> <else> SP1

<if>::

PR_IF AB_PAR <expression> FEC_PAR AB_CH <commands> FEC_CH

<else>::

PR_ELSE AB_CH <commands> FEC_CH
| empty

<while>::

PR_WHILE AB_PAR <expression> FEC_PAR AB_CH <commands> FEC_CH SP1

<for>::

PR_FOR AB_PAR <for_steps> FEC_PAR AB_CH <commands> FEC_CH SP1

<for_steps>::

<for_index_declaration> SP1 <for_limit_declaration> SP1 <for_step_declaration>

<for_index_declaration>::

PR_INT ID OP_ATR CTE_INT

<for_limit_declaration>::

ID OP_MEQ CTE_INT
| ID OP_MAQ CTE_INT
| ID OP_MEIGQ CTE_INT
| ID OP_MAIGQ CTE_INT

<for_step_declaration>::

ID OP_ATR ID OP_AD CTE_INT

<shoot>::

PRSHOOT <constant>
| PRSHOOT ID

Gramática Linguagem NOVA em LL(1) (EBNF)

<prøgram>::
<function> <main>

<function>:: <function_declaration>
| empty

<function_declaration>::
<type> ID AB_PAR <parameters> FEC_PAR <scope> SP1

<type>:: PR_INT
| PR_FLOAT
| PR_BOOL
| PR_STRING
| PR_VOID

<parameters>::
<type> ID <parameters_aux_1>

<parameters_aux_1>::
SP2 <parameters>
| empty

<scope>:: AB_CH <commands> FEC_CH SP1

<commands>::
<cmd> <commands_aux>

<commands_aux>::
SP1 <commands>
| empty

<cmd>:: <declaration>
| <attribution>
| <function_call>
| <printout>
| <readin>
| <ifelse>
| <while>
| <for>
| <shoot>

<declaration>::

<type> ID SP1

<attribution>::
ID OP_ATR <value>

<value>:: <array>
| <expression>

<array>:: AB_CH <elements> FEC_CH
| AB_CH FEC_CH

<elements>::
<constant> <elements_aux>

<elements_aux>::
SP2 <elements>
| empty

<constant>:: CTE_FLOAT
| CTE_INT
| CTE_STR
| CTE_BOOL

<expression>::
<logical_expression> <expression_aux_1>

<expression_aux_1>::
OP_ATR <expression>
| empty

<logical_expression>::
<eq_expression> <logical_expression_aux_1>

<logical_expression_aux_1>::
OP_AND <logical_expression>
| OP_OR <logical_expression>
| empty

<eq_expression>::
<comparative_exp> <eq_expression_aux>
<eq_expression_aux>::
OP_IG <eq_expression>
| OP_DIF <eq_expression>
| empty

<comparative_exp>::

<add_exp> <comparative_exp_aux>

<comparative_exp_aux>::
<comparative_exp>
| OP_MAQ <comparative_exp>
| OP_MEIGQ <comparative_exp>
| OP_MAIGQ <comparative_exp>
| empty

<add_exp>::
<mult_exp> <add_exp_aux>

<add_exp_aux>::
OP_AD <add_exp>
| OP_SUB <add_exp>
| empty

<mult_exp>::
<neg_exp> <mult_exp_aux>

<mult_exp_aux>::
OP_MULT <mult_exp>
| OP_DIV <mult_exp>
| OP_MOD <mult_exp>
| empty

<neg_exp>::
OP_NEG <exp_aux>
| <exp_aux>

<exp_aux>::
AB_PAR <atom_exp> FEC_PAR
| <atom_exp>

<atom_exp>::
ID
| CTE_INT
| CTE_FLOAT
| CTE_STRING
| PR_TRUE
| PR_FALSE

<function_call>::
ID AB_PAR <parameters_call> FEC_PAR

<parameters_call>::

<parameter_item> <parameters_call_aux>

<parameters_call_aux>::
SP2 <parameters_call>
| empty

<parameter_item>::
<constant>
| ID

<printout>::
PRINTOUT AB_PAR <message> FEC_PAR

<message>::
CTE_STR
| CTE_STR OP_AD <message>

<readin>::
READ_IN AB_PAR <type> SP2 ID FEC_PAR

<ifelse>::
<if> <else> SP1

<if>::
PR_IF AB_PAR <expression> FEC_PAR AB_CH <commands> FEC_CH

<else>::
PR_ELSE AB_CH <commands> FEC_CH
| empty

<while>::
PR_WHILE AB_PAR <expression> FEC_PAR AB_CH <commands> FEC_CH SP1

<for>::
PR_FOR AB_PAR <for_steps> FEC_PAR AB_CH <commands> FEC_CH SP1

<for_steps>::
<for_index_declaration> SP1 <for_limit_declaration> SP1 <for_step_declaration>

<for_index_declaration>::
PR_INT ID OP_ATR CTE_INT

<for_limit_declaration>::

ID OP_MEQ CTE_INT
| ID OP_MAQ CTE_INT
| ID OP_MEIGQ CTE_INT
| ID OP_MAIGQ CTE_INT

<for_step_declaration>::

ID OP_ATR ID OP_AD CTE_INT

<shoot>::

PRSHOOT <constant>
| PRSHOOT ID