

$L(G) = (S, P, V_n, V_t):$

$V_n = \{ \langle \text{program} \rangle, \langle \text{identifier} \rangle, \langle \text{function} \rangle, \langle \text{fun_return} \rangle, \langle \text{params} \rangle, \langle \text{body} \rangle, \langle \text{data_type} \rangle, \langle \text{letter} \rangle, \langle \text{alpha_numeric} \rangle, \langle \text{digit} \rangle, \langle \text{var_init} \rangle, \langle \text{instruction} \rangle, \langle \text{var_unary} \rangle, \langle \text{return} \rangle, \langle \text{assign} \rangle, \langle \text{operation} \rangle, \langle \text{sign} \rangle, \langle \text{value} \rangle, \langle \text{number} \rangle, \langle \text{string} \rangle, \langle \text{char} \rangle, \langle \text{assign_sign} \rangle, \langle \text{float} \rangle, \langle \text{integer} \rangle, \langle \text{for} \rangle, \langle \text{while} \rangle, \langle \text{if} \rangle, \langle \text{comparison_sign} \rangle, \langle \text{for_end} \rangle, \langle \text{loop_body} \rangle, \langle \text{comb_sign} \rangle \},$

$V_t = \{ \text{function, void, int, float, string, a,b...z, A,B...Z, _, 0,1...9, boolean, true, false, ++, --, return, +, -, *, /, ", ., print, (,), \{, \}, \{, break, =, ==, >=, <=, <, >, !, !=, +=, -=, for, ;, \&\&, | | \},$

$P = \{$

$\langle \text{program} \rangle \rightarrow \langle \text{function} \rangle +$

$\langle \text{function} \rangle \rightarrow \text{function } \langle \text{fun_return} \rangle \langle \text{identifier} \rangle (\langle \text{params} \rangle^*) \{ \langle \text{body} \rangle \}$

$\langle \text{fun_return} \rangle \rightarrow \text{void} \mid \langle \text{data_type} \rangle$

$\langle \text{data_type} \rangle \rightarrow \text{int} \mid \text{float} \mid \text{string} \mid \text{boolean}$

$\langle \text{identifier} \rangle \rightarrow \langle \text{letter} \rangle \langle \text{alpha_numeric} \rangle^*$

$\langle \text{alpha_numeric} \rangle \rightarrow \langle \text{letter} \rangle \mid \langle \text{digit} \rangle$

$\langle \text{letter} \rangle \rightarrow \text{a} \mid \text{b} \mid \dots \mid \text{z} \mid \text{A} \mid \text{B} \mid \dots \mid \text{Z} \mid$

$\langle \text{digit} \rangle \rightarrow 0 \mid 1 \mid \dots \mid 8 \mid 9$

$\langle \text{params} \rangle \rightarrow \langle \text{data_type} \rangle \langle \text{identifier} \rangle$

$\langle \text{body} \rangle \rightarrow \langle \text{instruction} \rangle^*$

$\langle \text{instruction} \rangle \rightarrow \langle \text{flow_control} \rangle \mid \langle \text{print} \rangle \mid \langle \text{assign} \rangle \mid \langle \text{return} \rangle \mid \langle \text{var_init} \rangle \mid \langle \text{var_unary} \rangle;$

$\langle \text{var_unary} \rangle \rightarrow \langle \text{identifier} \rangle ++ \mid \langle \text{identifier} \rangle --$

$\langle \text{return} \rangle \rightarrow \text{return } \langle \text{identifier} \rangle;$

$\langle \text{assign} \rangle \rightarrow \langle \text{identifier} \rangle = \langle \text{identifier} \rangle; \mid \langle \text{identifier} \rangle = \langle \text{operation} \rangle +;$

$\langle \text{operation} \rangle \rightarrow \langle \text{identifier} \rangle \langle \text{sign} \rangle \langle \text{identifier} \rangle$

$\langle \text{sign} \rangle \rightarrow + \mid - \mid * \mid /$

$\langle \text{var_init} \rangle \rightarrow \langle \text{data_type} \rangle \langle \text{identifier} \rangle; \mid \langle \text{data_type} \rangle \langle \text{identifier} \rangle = \langle \text{value} \rangle; \mid \langle \text{data_type} \rangle \langle \text{identifier} \rangle \langle \text{assign_sign} \rangle \langle \text{identifier} \rangle$

<assign_sign> -> -= | +=

<value> -> true | false | <number> | <string>

<number> -> <float> | <integer>

<float> -> <digit>+ . <digit>+

<integer> -> <digit>+

<string> -> "<char>*"

<char> -> <letter> | <digit>

<print> -> print (<string>) | print(<string> + <identifier>);

<flow_control> -> <for> | <while> | <if>

<for> -> for(int <identifier> = <integer>; <identifier> <comparison_sign> <identifier>;

<for_end>) { <body> }

<comparison_sign> -> > | < | >= | <= | == | !=

<for_end> -> <var_unary> | <assign>

<if> -> if(condition) { <body> } | if(condition) {<body>} else { <body> }

<while> -> while(<condition>){<loop_body>}

<loop_body> -> <body> break; | <body>

<condition> -> <identifier> <comparison_sign> <identifier> | <identifier> <comparison_sign>

<number> | !<condition> |

<condition> -> <condition> <comb_sign> <condition>

<comb_sign> -> && | ||

}