Nome: Adryan da Silva Freitas / Kalebe Nascimento Sousa
Email: adryan.freitas@gsuite.iff.edu.br / kalebe.s@gsuite.iff.edu.br
Matrícula: 202011250126 / 202011250134
Curso: Engenharia da Computação

A1 Compiladores

Gramática BNF

<letter>::= a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z

<digit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

<special> ::= "@" | "!" | "#" | "$" | "%" | "^" | "&" | "*" | "(" | ")" | "/" | "\" | "." | ">" | "<" | "?" | ";" | ":" | "-" | "=" | "+" | "_" | "}" | "{" | "~" | "'" | "`" | "[" | "]" | "“" | "”" | "|"

<number>::= <digit> | <digit> <number> | <number> **.** <number>

<string>::= " (<letter> | <number> | <digit> | <special>)* "

<id> ::= { <letter> }({ <letter> } | { <digit> } | "_")*

<program>::= <program> <statements> | <statements>

<statements> ::= <decl_var> | <attr> ; | <expressions> | <commands>

<decl_var> ::= <tipo> <id> ;

<type> ::= int | float | char | string

<expressions>::= <expr_art> ; | <expr_art> | <expr_rel> | <expr_rel> ; | <expr_log> ; | <expr_log>

<expr_art> ::= <number> | <expr_art> + <expr_art> | <expr_art> - <expr_art> | <expr_art> / <expr_art> | <expr_art> * <expr_art> | <expr_art> % <expr_art> | - <expr_art> | (<expr_art>);

<attr> ::= <id> = <string> | <id> = <expr_art> | <id> = <letter> | <id> ++ | <id> - -

<value> ::= <expr_art> | <id> | <string> | <letter> | true | false

<expr_rel_operator> ::= == | != | > | < | >= | <=

<expr_rel> ::=<value> <expr_rel_operator> <value> | (<expr_rel>)

<expr_log_operator> ::= && | "||'

<expr_log>::= <expr_rel> <expr_log_operator> <expr_rel> | <value> <expr_log_operator> <value> | ! <expr_log> | ! | ! <value> | (<expr_log>)

<start_block> :== **{**

<end_block> :== **}**

<block> ::= <start_block> <program> <end_block>

<expr_if> :: <expr_log> | <expr_rel>

<commands>::= printf (<value>) ; | printf (<expressions>) ;

| scanf (<string>, &<id>); | scanf (<string>, <id>) ;

| if (<expr_if>) <block> | if (expr_if) <block> else <block>

| for (<attr> ; <expr_rel> ; <attr>) <block>

| while (<expressions>) <block>