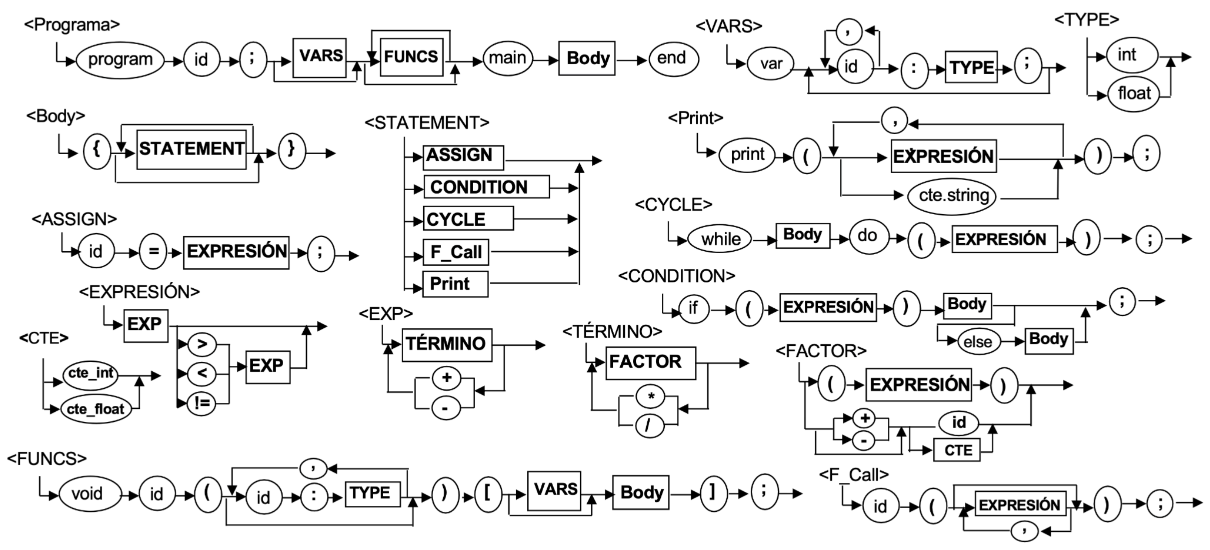
A00573134 - Macías Romero Jorge Humberto

TC3002B: Desarrollo de aplicaciones avanzadas de Ciencias Computacionales

Módulo: Compiladores **Mini Proyecto INDIVIDUAL :Little_Duck** **Abril2024**



Expresiones regulares

id → ^\$[^$]+ Así como en PHP, los IDs se declaran con un signo de $.
cte_string → “[.*]”
cte_int → [0-9]+
cte_float → [0-9]+\.[0-9]+

Lista de Tokens

program, main, void, end, var, int, float, print, while, do, if, else
( ) { } [ ] , : ;
+ - * / = > < !=

Context Free Grammar

Unset

```
<PROGRAM> → program id ; <has_vars> <has_funcs> main <BODY> end
<has_vars> → <VARS> | ε
<has_funcs> → <FUNCS> <has_funcs> | ε

<VARS> → var <var_complement>
<var_complement> → <id_complement> : <TYPE> ; <var_complement> | ε
<id_complement> → id | , id <id_complement> | ε

<TYPE> → int | float

<BODY> → { <body_complement> }
<body_complement> → <STATEMENT> <body_complement> | ε

<STATEMENT> → <ASSIGN> | <CONDITION> | <CYCLE> | <F_CALL> | <PRINT>
```

A00573134 - Macías Romero Jorge Humberto

```
<PRINT> → print ( <print_complement> ) ;
<print_complement> → <EXPRESSION> <expression_aux>
                   | cte_string <expression_aux>
<expression_aux> → , <print_complement> | ε

<ASSIGN> → id = <EXPRESSION> ;

<CYCLE> → while <BODY> do ( <EXPRESSION> ) ;

<CONDITION> → if ( <EXPRESSION> ) <BODY> <condition_else> ;
<condition_else> → else <BODY> | ε

<CTE> → cte_int | cte_float

<EXPRESSION> → <EXP> <expression_aux>
<expression_aux> → <expression_logics> <EXP> | ε
<expression_logics> → > | < | !=

<EXP> → <TERM> <exp_aux>
<exp_aux> → <exp_operation> <TERM> <exp_aux> | ε
<exp_operation> → + | -

<TERM> → <FACTOR> <term_aux>
<term_aux> → <term_operation> <TERM> <term_aux> | ε
<term_operation> → * | /

<FACTOR> → <factor_expression> | <factor_aux>
<factor_expression> → ( <EXPRESSION> )
<factor_aux> → <factor_operations> <factor_cte>
<factor_cte> → id | <CTE>
<factor_operations> → <factor_operations_plus_minus> | ε
<factor_operations_plus_minus> → + | -

<FUNCS> → void id ( <funcs_args> ) [ <funcs_vars> <BODY> ] ;
<funcs_vars> → <VARS> | ε
<funcs_args> → <args_aux> | ε
<args_aux> → id : <TYPE> , <args_aux> | id : <TYPE>

<F_CALL> → id ( <f_call_expression> ) ;
<f_call_expression> → <f_call_expression_aux> | ε
<f_call_expression_aux> → <EXPRESSION> , <f_call_expression_aux>
                        | <EXPRESSION>
```