

Gramática proyecto 1 OLC

Tokens y expresiones regulares

Generales

MULTILINE_COMMENT := <! U* !>

SET_DECLARATION := CONJ

ID := [Identificador]

ONELINE_COMMENT := // U* \n

LBRACE := {

RBRACE := }

COLON := :

ARROW := ->

SEMICOLON := ;

SCOPE_BREAK := %%

WHITESPACE := \t | \n | ' '

Símbolos de la notación de conjuntos

TILDE := ~

COMMA := ,

DIGIT := [0-9]

LOWERCASE := [a-z]

UPPERCASE := [A-Z]

ASCII := [Caracteres ASCII del 32 al 125 omitiendo Letras y dígitos]

Símbolos de la notación de ER's

AND := .

OR := |

KLEENE := *

PLUS := +

QUESTION := ?

LETTER := [a-zA-Z]

DIGIT := [0-9]

ESCAPED_LINEBREAK := “ \n ”

ESCAPED_SINGLE_QUOTE := “ \' ”

ESCAPED_DOUBLE_QUOTE := “ \” ”

Gramática

PROGRAM ::= <LBRACE> SCOPES <RBRACE>

SCOPES ::= DECL_SCOPE <SCOPE_BREAK> <SCOPE_BREAK> STMT_SCOPE

DECL_SCOPE ::= DECL DECLS

DECL ::= SET_DECL | REGEX_DECL

DECLS ::= DECL DECLS | ε

SET_DECL ::= <SET_DECLARATION> <COLON> <ID> <ARROW> SET <SEMICOLON>

SET ::= COMPR_SET | SET_ELEMENTS

COMPR_SET ::= SET_ELEMENT <TILDE> SET_ELEMENT

SET_ELEMENT ::= <ASCII> | <UPPERCASE> | <LOWERCASE> | <DIGIT>

EXTEND_SET ::= SET_ELEMENT SET_ELEMENTS

SET_ELEMENTS ::= <COMMA> SET_ELEMENT | ε

REGEX_DECL ::= <ID> <ARROW> REGEX_EXPR

REGEX_EXPR ::= REGEX_TERM REGEX_EXPRS <SEMICOLON>

REGEX_EXPRS ::= REGEX_TERM REGEX_EXPRS | ε // <- !!!

REGEX_TERM ::= <AND> REGEX_TERM REGEX_TERM
| <OR> REGEX_TERM REGEX_TERM
| <KLEENE> REGEX_TERM
| <PLUS> REGEX_TERM
| <QUESTION> REGEX_TERM
| REGEX_TERMINAL

REGEX_TERMINAL ::= <LETTER>
| <DIGIT>

| <ESCAPED>

| <STRING >

| SET_REFERENCE

SET_REFERENCE ::= <LBRACE> <ID> <RBRACE>

STMT_SCOPE ::= EVAL_STMT EVAL_STMTS

EVAL_STMT ::= <ID> <COLON> <STRING> <SEMICOLON>

EVAL_STMTS ::= EVAL_STMT | ε