



Compilador MFJG

Martin Purita, Legajo 51187
Facundo Menzella, Legajo 51533
Julián Gutierrez Ferrara, Legajo 51141
Gustavo Del Guidice, Legajo 51239

Objetivo Desarrollar un compilador que sea capaz de factorizar un número primo. Este compilador también funcionaría para implementar otras funciones como obtener el número de Fibonacci.

Gramática Para explicar el funcionamiento de nuestro compilador hemos desarrollado una gramática de segundo orden. A continuación definimos dicha gramática:

Se define la siguiente gramática $G = (V_n, V_t, S, P)$ donde:

$V_n =$

$\{S, VAR_NAME, DIGIT, VAR_RIGHT_VALUE, VAR_LEFT_VALUE, INTEGER, CHARACTER, BOOLEAN_TYPE, BOOLEAN, CONSTAN$

$ENDFUNCTION, LESS_THAN, GREATER_THAN, EQUALS, NOT_EQUALS, IF, ENDIF, WHILE, ENDWHILE, ADD, LESS, DIVI$
 $VAR_DEFINITION\};$

$V_t = \{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, \{, \}, (,), <, >, :, =, _ \};$

S símbolo inicial;

P producciones;

Desarrollo Para realizar el análisis léxico hemos utilizado la herramienta LEX, la cual en la próxima versión se acoplará la herramienta YACC para realizar el análisis sintáctico.

Código fuente El código fuente se puede ver en el repositorio de GitHub: <https://github.com/MartuPuri/TLA/>