

Teoría de Lenguajes Autómatas y Compiladores

Compilador YACC

Matias Domingues - Legajo 50278

Objetivo

Generar una gramática e implementar un analizador léxico en lex con el objetivo de poder escribir un programa en este lenguaje que permita calcular la factorización en números primos de un número entero.

Descripcion de la Gramática

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

V_n = {VARTYPE VARNAME VARIABLE SPACE RETTYPE VARDEF MOREVARS SEVDEF
BOPERATORS ELEMENT OPERATIONS MOREELEMS SEVVAR FUNCTION COMPARATOR
COMPARATION VARIABLEOP DIGIT FUNCTION CONDITION CYCLE FORCYCLE }

V_t ={0,1,2,3,4,5,6,7,8,9,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, VOID, (,), ++, --, >, < >=, <=, ==, !=, =, +, -, / *, %, INT, CHAR,
STRING, BOOLEAN, MAIN, WHILE, FOR, IF}

S = Símbolo Inicial

P = Producciones

Desarrollo:

Se realizó el compilador en lex, utilizando una gramática parecida a una versión simplificada de C. Uno de los problemas con lo que me encontré, fue verificar que los corchetes estaban balanceados, para lo cual tuve que implementar funciones extras en C.

Futuras extensiones:

Se podrían agregar bloques para simular objetos dentro del código, también se podría intentar simplificar el lenguaje para sacar los paréntesis y solo usar indentaciones para marcar la apertura y el cierre de los mismos.