MANUAL TÉCNICO: DRAW MATH

Nombre del sistema: Draw Mathque

Versión del sistema: 1.0

Tipo de Manual: Manual técnico

Fecha de elaboración: Octubre,2017

Área: Ciudad de Guatemala

PRESENTACIÓN

- Antecedentes

  El departamento de Matemáticas de la Facultad de Ingeniería requiere una aplicación que grafique funciones matemáticas a partir del análisis de un archivo de entrada que contendrá las instrucciones necesarias para generar la imagen solicitada.

- Objetivos

  -Graficar funciones matemáticas a partir de una entrada.

  -Obtener componentes léxicos del análisis léxico.

  -Analizar sintácticamente la entrada mediante una gramática de tipo 2.

- Introducción

  Manual dirigido a miembros de la facultad de ingeniería que quieran conocer la aplicación.

DESCRIPCIÓN DE ACTORES DEL SISTEMA

ACTOR DEL NEGOCIO	DESCRIPCIÓN
Usuario de la aplicación Draw Math	El usuario tendrá acceso a la aplicación en la cual a través de código podrá hacer funciones matemáticas.

ESPECIFICACIONES DEL SISTEMA

1. Requerimiento funcionales
   - Realizar graficas de funciones matemáticas a través de un código de entrada.
   - Obtener componentes léxicos mediante un autómata finito determinista.
   - Realizar análisis sintáctico de la entrada a partir de una gramática de tipo 2.
   - Galería de funciones donde se muestran todas las gráficas realizadas.

2. Requerimiento no funcionales
    - Tiempo prudente en que se tarda en realizar los procesos.
    - Tolerante a errores léxicos y sintácticos.
    - Uso intuitivo y dinámico para el usuario.

DESCRIPCIÓN DEL CÓDIGO

- ESTRUCTURA GENERAL

  Inicio Math

  Inicio declaracion constantes y variables
  //CONSTANTES
  Fin declaración constantes y variables

  Inicio declaración funciones
  //FUNCIONES
  Fin declaración funciones

  Inicio generación graficas
  //GRAFICAS
  Fin generación graficas

  Fin Math

- VARIABLES Y CONSTANTES

  Inicio declaracion constantes y variables

  Inicio constante
  Nombre = Nombre,
  Valor = valor,
  Tipo = tipo
  Fin constante

  //MÁS CONSTANTES

  Fin declaración constantes y variables

- FUNCIONES

Inicio declaración funciones

Inicio función
Nombre = nombre,
Valor = valor
Fin funcion

//MAS FUNCIONES

Fin declaración funciones

- GRAFICAS

Inicio generación graficas

Inicio grafica
Nombre = nombre,
X_positivo = valor,
X_negativo = valor,
Y_positivo = valor,
Y_negativo = valor,
Ancho = valor,
Largo = valor,
Ruta = "cadena",
Función = funcion
Fin grafica

//MAS GRAFICAS

Fin generación graficas

- COMENTARIOS

//COMENTARIO SIMPLE

/
COMENTARIO
COMPUESTO
**/*

Operaciones aritméticas

Suma	Suma(4,5) Suma(constante,10) Suma(constante,constante)
Resta	Resta(4,5) Resta(constante,10) Resta(constante,constante)
Multiplicación	Multiplicar(4,5) Multiplicar(constante,10) Multiplicar(constante,constante)
Dividir	Dividir(4,5) Dividir(constante,10) Dividir(constante,constante)
Potencia	Potencia(4,5) Potencia(constante,10) Potencia(constante,constante)
Raíz Cuadrada	RaizCuadrada(4) RaizCuadrada(constante)

Funciones trigonométricas

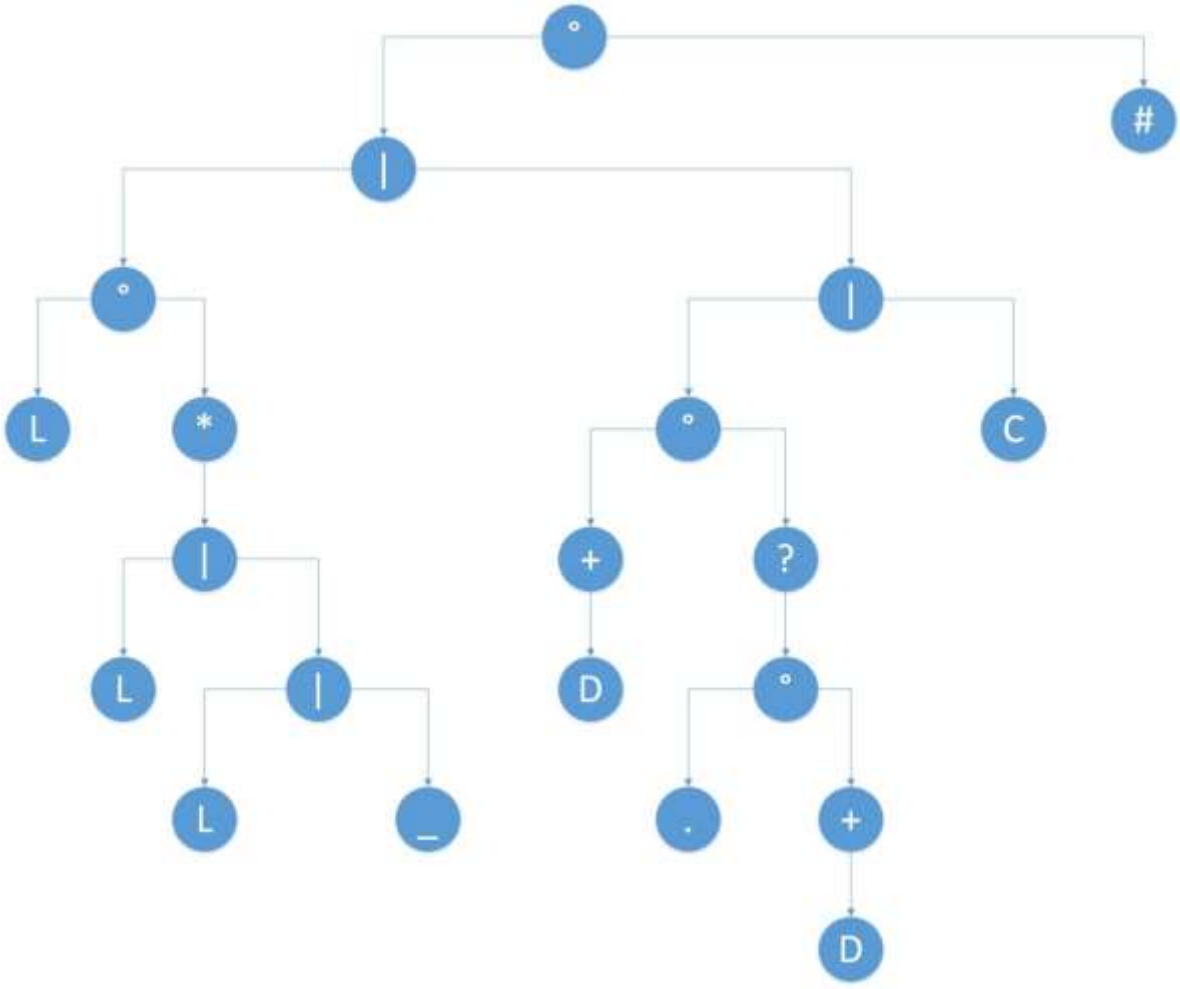
seno	Seno(constante) Seno(60) Seno(x)
coseno	coseno(constante) coseno(60) coseno(x)
tangente	tangente(constante) tangente(60) tangente(x)

DESCRIPCIÓN DETALLADA DEL ALGORITMO

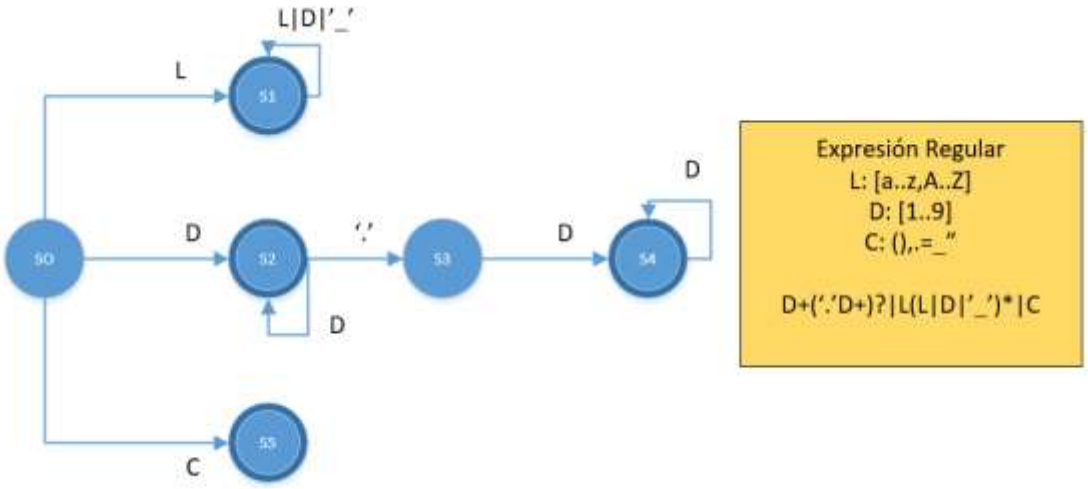
EXPRESIÓN REGULAR

D+(‘.’D+)?|L(L|D|’_’)*|C

ÁRBOL GENERADO A PARTIR DE LA EXPRESIÓN REGULAR



Autómata finito determinista (AFD)



GRAMATICA TIPO 2

ESTRUCTURA GENERAL

letra = (A..Z, a..z)

digito= (0..9)

identificador = letra(letra|digito|'_')*

numero = digito+('.'digito+)?

cadena = letra*

<INICIO> := <INICIO MATH> <CUERPO MATH> <FIN MATH>

<CUERPO> := <CONSTANTEVARIABLE> <FUNCIONES> <GRAFICAS>

<INICIO MATH> := "inicio math"

<FIN MATH> := "fin math"

<CONSTANTEVARIABLE> := <INICIO DECLARACION CONSTANTES Y VARIABLES> (<CONSTANTE>) <FIN DECLARACION VARIABLE Y CONSTANTES>

<FUNCIONES> := <INICIO DECLARACION FUNCIONES> (<FUNCION>|épsilon)<FIN DECLARACION FUNCIONES>

<GRAFICAS> := <INICIO GENERACION GRAFICAS> (<GRAFICA>|épsilon) <FIN GENERACION GRAFICAS>

MODULO 1: VARIABLES Y CONSTANTES

<CONSTANTE> := <INICIO CONSTANTE>

(<NOMBRE>’,’<TIPO>’,’<VALOR>|

<NOMBRE>’,’<VALOR>’,’<TIPO>|

<TIPO>’,’<NOMBRE>’,’<VALOR>|

<TIPO>’,’<VALOR>’,’<NOMBRE>|

<VALOR>’,’<NOMBRE>’,’<TIPO>|

<VALOR>’,’<TIPO>’,’<NOMBRE>)

<FIN CONSTANTE>(<CONSTANTE>| épsilon)

<INICIO DECLARACION CONSTANTES Y VARIABLES> := “inicio declaración constantes y variables”

<FIN DECLARACION VARIABLE Y CONSTANTES> := “fin declaración constantes y variables”

<INICIO CONSTANTE> := “inicio constante”

<FIN CONSTANTE> := “fin constante”

<NOMBRE> := “nombre””=” identificador

<TIPO> := (“Decimal”|”Entero”|”Cadena”)

<VALOR> := “valor=”(numero|cadena|<OPERACIÓNSIMPLE>|<OPERACIONCOMPUESTA>)

<OPERACIÓNCOMPUESTA> := <SUMA> | <RESTA> | <MULTIPLICACION> | <DIVIDIR> | <POTENCIA>

<OPERACIÓNSIMPLE> := <RAIZ CUADRADA> | <SENO> | <COSENO> | <TANGENTE>

<SUMA> := “Suma(“ (<OPERACIÓN>| identificador| numero) “,” (<OPERACIÓN>|identificador|numero) “)”

<RESTA> := “Resta(“ (<OPERACIÓN>| identificador| numero) “,” (<OPERACIÓN>|identificador|numero) “)”

<MULTIPLICACION> := “Multiplicacion (“ (<OPERACIÓN>| identificador| numero) “,” (<OPERACIÓN>|identificador|numero) “)”

<DIVIDIR> := “Dividir(“ (<OPERACIÓN>| identificador| numero) “,” (<OPERACIÓN>|identificador|numero) “)”

<POTENCIA> := “Potencia(“ (<OPERACIÓN>| identificador| numero) “,” (<OPERACIÓN>|identificador|numero) “)”

<RAIZ CUADRADA> := “raizCuadrada(“ (<OPERACIÓN>| identificador| numero) “)”

<SENO> := “Seno(“ (<OPERACIÓN>| identificador| numero) “)”

<COSENO> := “Coseno(“ (<OPERACIÓN>| identificador| numero) “)”

<TANGENTE> := “Tangente(“ (<OPERACIÓN>| identificador| numero) “)”

MODULO 2: FUNCIONES

<INICIO DECLARACION FUNCIONES>:= “inicio declaración funciones”

<FIN DECLARACION FUNCIONES>:= “fin funciones”

<FUNCION> := <INICIO FUNCION>

(<NOMBRE> ‘,’ <VALOR> |

<VALOR> ‘,’ <NOMBRE>)

<FIN FUNCION>

<INICIO FUNCION> := “inicio función>

<FIN FUNCION> := “fin función”

MÓDULO 3: GRÁFICAS

<GRAFICA> := <INICIO GRAFICA> <NOMBRE> “,” <X_POSITIVO> “,” <X_NEGATIVO> “,” <Y_POSITIVO> “,” <Y_NEGATIVO> “,” <ANCHO> “,” <LARGO> “,” <RUTA> “,” <FUNCIONG> “,” <FIN GRAFICA>

<INICIO GRAFICA> := “inicio grafica”

<FIN GRAFICA> := “fin grafica”

<X_POSITIVO> := “x_positivo = “ (numero | identificador)

<X NEGATIVO> := “x_negativo =” (numero | identificador)

<Y_POSITIVO> := “y_positivo =” (numero | identificador)

<Y_NEGATIVO> := “y_negativo =” (numero | identificador)

<ANCHO> := “ancho = “ (numero | identificador)

<LARGO> := “largo = “ (numero | identificador)

<RUTA> := “ruta = “ cadena

<FUNCIONG> := “función = “ identificador

GLOSARIO

TERMINO	DEFINICIÓN
Compilador	Programa que traduce un programa escrito en código fuente a un programa equivalente en código destino.
Token	Componente léxico con un significado colectivo.
Lexema	Secuencia de caracteres del programa fuente que concuerdan con el patrón de un componente léxico.
Reservadas	Lexema que tiene una función previamente definido por el programa por lo cual no puede usarse para otro objeto.
Análisis léxico	Fase del compilador el cual consiste en leer el código fuente carácter a carácter y agruparlos en componentes léxicos
Análisis sintáctico	Fase del compilador la cual consiste en examinar la estructura del flujo de token recibido del analizador léxico.
Autómata finito determinista (AFD)	Autómata finito en el cual para cada estado en que se encuentra, existe no más de una transición posible. Es una quíntupla formada por: Estados, alfabeto, estado inicial, función de transición y estados de aceptación
Método del árbol	Método para producir autómatas finitos más ordenados, eficientes y reducidos.
Tabla de símbolos	Estructura de datos que usa el proceso de traducción de un lenguaje de programación, por un compilador o un intérprete, donde cada símbolo en el código fuente de un programa está asociado con información tal como la ubicación, el tipo de datos y el ámbito de cada variable.
Autómata de pila	Autómata finito determinista con transiciones vacías y la capacidad de almacenar una cadena de “símbolos de pila”. Puede recordar una cantidad finita de información, sin embargo, solo se puede acceder a dicha información utilizando las formas de manipulación FIFO.