

Manual Tecnico

Analizador léxico y sintáctico

Descripción

Sistema desarrollado para el análisis léxico y sintáctico de un lenguaje formal. Consta de dos componentes principales: un **analizador léxico (lexer)**, encargado de identificar y clasificar los tokens presentes en la entrada, y un **analizador sintáctico predictivo LL(1)**, que valida la estructura gramatical de la cadena conforme a una gramática libre de contexto.

Requerimientos de software

Openjdk 21

Herramientas para desarrollo

Openjdk 21

JDK 21

JFlex

NetBeans

Java Swing

Maven

Instalación

Ejecutar el archivo analizador.jar desde la interfaz grafica, o desde la terminal usando 'java -jar rutaAlArchivo/analizador.jar'

Diseño

El diseño del parser consta de los siguientes componentes:

Gramática:

NT_INICIO -> NT_INSTRUCCION NT_INICIO'

NT_INICIO' -> NT_INSTRUCCION NT_INICIO'

NT_INICIO' -> ϵ

NT_INSTRUCCION -> NT_PRINT

NT_INSTRUCCION -> NT_REPEAT

NT_INSTRUCCION -> NT_CONDICIONAL

NT_INSTRUCCION -> NT_ASIGNACION

NT_PRINT -> PRINT NT_VALOR END

NT_VALOR -> LITERAL

NT_VALOR -> NUMERO_ENTERO

NT_VALOR -> IDENTIFICADOR

NT_VALOR -> DECIMAL

NT_REPEAT -> REPEAT NT_VALOR INIT NT_CONTENIDO_REPEAT END
NT_CONTENIDO_REPEAT -> NT_PRINT NT_CONTENIDO_REPEAT'
NT_CONTENIDO_REPEAT' -> NT_PRINT NT_CONTENIDO_REPEAT'
NT_CONTENIDO_REPEAT' -> ϵ

NT_CONDICIONAL -> IF NT_BOOLEANO THEN NT_CONTENIDO_COND END
NT_BOOLEANO -> TRUE
NT_BOOLEANO -> FALSE
NT_CONTENIDO_COND -> NT_PRINT
NT_CONTENIDO_COND -> ϵ

NT_ASIGNACION -> IDENTIFICADOR ASIGNACION NT_EXP END

NT_EXP -> NT_T NT_EXP'
NT_EXP -> SUMA NT_T NT_EXP'
NT_EXP -> RESTA NT_T NT_EXP'
NT_EXP' -> SUMA NT_T NT_EXP'
NT_EXP' -> RESTA NT_T NT_EXP'
NT_EXP' -> ϵ
NT_T -> NT_F NT_T'
NT_T' -> MULTIPLICACION NT_F NT_T'
NT_T' -> DIVISION NT_F NT_T'
NT_T' -> POTENCIA NT_F NT_T'
NT_T' -> ϵ
NT_F -> PARENTESIS_APERTURA NT_EXP PARENTESIS_CIERRE
NT_F -> NUMERO_ENTERO
NT_F -> IDENTIFICADOR

Primeros de la gramática:

First(NT_INICIO) = {PRINT, REPEAT, IF, IDENTIFICADOR}

First(NT_INICIO') = {PRINT, REPEAT, IF, IDENTIFICADOR}

First(NT_INSTRUCCION) = {PRINT, REPEAT, IF, IDENTIFICADOR}

First(NT_PRINT) = {PRINT}

First(NT_VALOR) = {LITERAL, NUMERO_ENTERO, IDENTIFICADOR}

First(NT_REPEAT) = {REPEAT}

First(NT_CONTENIDO_REPEAT) = {PRINT}

First(NT_CONTENIDO_REPEAT') = {PRINT}

First(NT_CONDICIONAL) = {IF}

$\text{First}(\text{NT_BOOLEANO}) = \{\text{TRUE}, \text{FALSE}\}$

$\text{First}(\text{NT_CONTENIDO_COND}) = \{\text{PRINT}\}$

$\text{First}(\text{NT_ASIGNACION}) = \{\text{IDENTIFICADOR}\}$

$\text{First}(\text{NT_EXP}) = \{\text{SUMA}, \text{RESTA}, \text{PARENTESIS_APERTURA}, \text{NUMERO_ENTERO}, \text{IDENTIFICADOR}\}$

$\text{First}(\text{NT_EXP}') = \{\text{SUMA}, \text{RESTA}\}$

$\text{First}(\text{NT_T}) = \{\text{PARENTESIS_APERTURA}, \text{NUMERO_ENTERO}, \text{IDENTIFICADOR}\}$

$\text{First}(\text{NT_T}') = \{\text{MULTIPLICACION}, \text{DIVISION}, \text{POTENCIA}\}$

$\text{First}(\text{NT_F}) = \{\text{PARENTESIS_APERTURA}, \text{NUMERO_ENTERO}, \text{IDENTIFICADOR}\}$

Siguientes de la gramática:

$\text{Follow}(\text{NT_INICIO}) = \{\text{EOF}\}$

$\text{Follow}(\text{NT_INICIO}') = \{\text{EOF}\}$

$\text{Follow}(\text{NT_INSTRUCCION}) = \{\text{PRINT}, \text{REPEAT}, \text{IF}, \text{IDENTIFICADOR}, \text{EOF}\}$

$\text{Follow}(\text{NT_PRINT}) = \{\text{PRINT}, \text{REPEAT}, \text{IF}, \text{IDENTIFICADOR}, \text{END}, \text{EOF}\}$

$\text{Follow}(\text{NT_VALOR}) = \{\text{END}, \text{INIT}\}$

$\text{Follow}(\text{NT_REPEAT}) = \{\text{PRINT}, \text{REPEAT}, \text{IF}, \text{IDENTIFICADOR}, \text{EOF}\}$

$\text{Follow}(\text{NT_CONTENIDO_REPEAT}) = \{\text{END}\}$

$\text{Follow}(\text{NT_CONTENIDO_REPEAT}') = \{\text{END}\}$

$\text{Follow}(\text{NT_CONDICIONAL}) = \{\text{PRINT}, \text{REPEAT}, \text{IF}, \text{IDENTIFICADOR}, \text{EOF}\}$

$\text{Follow}(\text{NT_BOOLEANO}) = \{\text{THEN}\}$

$\text{Follow}(\text{NT_CONTENIDO_COND}) = \{\text{END}\}$

$\text{Follow}(\text{NT_ASIGNACION}) = \{\text{PRINT}, \text{REPEAT}, \text{IF}, \text{IDENTIFICADOR}, \text{EOF}\}$

$\text{Follow}(\text{NT_EXP}) = \{\text{END}, \text{PARENTESIS_CIERRE}\}$

$\text{Follow}(\text{NT_EXP}') = \{\text{END}, \text{PARENTESIS_CIERRE}\}$

$\text{Follow}(\text{NT_T}) = \{\text{END}, \text{SUMA}, \text{RESTA}, \text{PARENTESIS_CIERRE}\}$

$\text{Follow}(\text{NT_T}') = \{\text{END}, \text{SUMA}, \text{RESTA}, \text{PARENTESIS_CIERRE}\}$

Follow(NT_F) = {END, SUMA, RESTA, MULTIPLICACION, DIVISION, POTENCIA, PARENTESIS_CIERRE}

Tabla de análisis:

	EOF	PRINT	END	LITERAL	NUMERO_ENTERO	IDENTIFICADOR	REPEAT	INIT	IF	THEN
NT_INICIO		NT_INICIO -> NT_INSTRUCCION NT_INICIO'				NT_INICIO -> NT_INSTRUCCION NT_INICIO'	NT_INICIO -> NT_INSTRUCCION NT_INICIO'		NT_INICIO -> NT_INSTRUCCION NT_INICIO'	
NT_INICIO'	NT_INICIO' -> ϵ	NT_INICIO' -> NT_INSTRUCCION NT_INICIO'				NT_INICIO' -> NT_INSTRUCCION NT_INICIO'	NT_INICIO' -> NT_INSTRUCCION NT_INICIO'		NT_INICIO' -> NT_INSTRUCCION NT_INICIO'	
NT_INSTRUCCION		NT_INSTRUCCION -> NT_PRINT				NT_INSTRUCCION -> NT_ASSIGNACION	NT_INSTRUCCION -> NT_REPEAT		NT_INSTRUCCION -> NT_CONDICIONAL	
NT_PRINT		NT_PRINT -> PRINT NT_VALOR END								
NT_VALOR				NT_VALOR -> LITERAL	NT_VALOR -> NUMERO_ENTERO	NT_VALOR -> IDENTIFICADOR				
NT_REPEAT							NT_REPEAT -> REPEAT NT_VALOR INIT NT_CONTENIDO_REPEAT END			
NT_CONTENIDO_REPEAT		NT_CONTENIDO_REPEAT -> NT_PRINT NT_CONTENIDO_REPEAT'								
NT_CONTENIDO_REPEAT'		NT_CONTENIDO_REPEAT' -> NT_PRINT NT_CONTENIDO_REPEAT'	NT_CONTENIDO_REPEAT' -> ϵ							
NT_CONDICIONAL									NT_CONDICIONAL -> IF NT_BOOLEANO THEN NT_CONTENIDO_COND END	
NT_BOOLEANO										
NT_CONTENIDO_COND		NT_CONTENIDO_COND -> NT_PRINT	NT_CONTENIDO_COND -> ϵ							
NT_ASSIGNACION						NT_ASSIGNACION -> IDENTIFICADOR ASIGNACION NT_EXP END				
NT_EXP					NT_EXP -> NT_T NT_EXP'	NT_EXP -> NT_T NT_EXP'				
NT_EXP'			NT_EXP' -> ϵ							
NT_T					NT_T -> NT_F NT_T'	NT_T -> NT_F NT_T'				
NT_T'			NT_T' -> ϵ							
NT_F					NT_F -> NUMERO_ENTERO	NT_F -> IDENTIFICADOR				

TRUE	FALSE	ASIGNACION	SUMA	RESTA	MULTIPLICACION	DIVISION	POTENCIA	PARENTESIS_APERTURA	PARENTESIS_CIERRE
NT_BOOLEANO -> TRUE	NT_BOOLEANO -> FALSE								
			NT_EXP -> SUMA NT_T NT_EXP'	NT_EXP -> RESTA NT_T NT_EXP'				NT_EXP -> NT_T NT_EXP'	
			NT_EXP' -> SUMA NT_T NT_EXP'	NT_EXP' -> RESTA NT_T NT_EXP'					NT_EXP' -> ε
								NT_T -> NT_F NT_T'	
			NT_T' - > ε	NT_T' - > ε	NT_T' -> MULTIPLICACION NT_F NT_T'	NT_T' -> DIVISION NT_F NT_T'	NT_T' -> POTENCIA NT_F NT_T'		NT_T' -> ε
								NT_F -> PARENTESIS_APERTURA NT_EXP PARENTESIS_CIERRE	

Desarrollo:

El algoritmo usado para aceptar la gramática es el siguiente:

INICIO Evaluar

REPETIR

P = cima de la pila de símbolos

E = tipo del token actual en la lista de tokens

SI P es un terminal o es "EOF" ENTONCES

SI P es igual a E ENTONCES

desapilar símbolo de pila

avanzar apuntador

SINO

registrar error

SINO

SI existe una producción para (P, E) en la tabla de análisis ENTONCES

desapilar símbolo y nodo

produccion = obtener producción de la tabla para (P, E)

invertir produccion

PARA CADA símbolo p en produccion

SI p no es "ε" ENTONCES

apilar p en la pila de símbolos

SINO

registrar error

HASTA que la pila de símbolos esté vacía

RETORNAR árbol sintáctico

FIN