

Índice

Expresión Regular ----- pg.2

Gramática Regular ----- pg. 3

AFD ----- pg. 7

Métodos Utilizados ----- pg. 8

Expresión Regular

Expression: [`</>|diseño|variables|construccion|/|nombre|color|pared_asociada|ventana|pared|suelo|puerta|alto|ancho|inicio`]

Edit the regular expression above. ! is the empty string sub.

Regular Expression = (`Σ`, `E`)

Σ = {

Table Text Size

color|pared_asociada|ventana|pared|suelo|puerta|alto|ancho|inicio|fin|terreno|tipo|valor|[a-z|A-Z][0-9]cadena|doble|entero]

ub.

Regular Expression = (`Σ`, `E`)

Σ = {

Table Text Size

“ER:

`[</>|diseño|variables|construcción|/|nombre|color|pared_asociada|ventana|pared|suelo|puerta|alto|ancho|Inicio|fin|terreno|tipo|valor|[a-z|A-Z|[0-9]|cadena|doble|entero]`”

Gramática Regular

LHS	RHS
q0	→ <diseño>q1
q1	→ <variables>q2
	→ <construccion>q3
	→ </diseño>ε
q2	→ [nombre]q4
	→ [tipo]q4
	→ [valor]q4

Grammar = (V, T, P, S)

V = { D I N V }

T = { / 0 1 2 3 4 5 6 7 8 9 < > [] _ }

S = D

Table Text Size

LHS	RHS
	$\rightarrow </\text{diseño}>\epsilon$
q2	$\rightarrow [\text{nombre}]q4$
	$\rightarrow [\text{tipo}]q4$
	$\rightarrow [\text{valor}]q4$
	$\rightarrow </\text{variables}>q1$
q4	$\rightarrow [\text{ID_Nombre}]q2$
	$\rightarrow [\text{ID_Valor}]q2$
	$\rightarrow [\text{entero} \text{cadena} \text{doble}]...$
q3	$\rightarrow <\text{terreno}>q5$
	$\rightarrow <\text{pared}>q7$
	$\rightarrow <\text{ventana}>q9$
	$\rightarrow <\text{puerta}>q11$

Grammar = (V, T, P, S)

V = { D I N V }

T = { / 0 1 2 3 4 5 6 7 8 9 < > [] _ }

S = D

Table Text Size

LHS	RHS
	→ <puerta>q11
	→ <suelo>q13
	→ </construccion>q1
q5	→ [ancho largo]q6
	→ </terreno>q3
q6	→ [ID_Valor]q5
q7	→ [nombre color alto ini...
	→ </pared>q3
q8	→ [ID_Valor]q7
q9	→ [nombre tipo longitud...
	→ </ventana>q3
q10	→ [ID_Valor]q9

Grammar = (V, T, P, S)

V = { D I N V }

T = { / 0 1 2 3 4 5 6 7 8 9 < > [] _ }

S = D

Table Text Size

LHS	RHS
	→ </pared>q3
q8	→ [ID_Valor]q7
q9	→ [nombre tipo longitud...
	→ </ventana>q3
q10	→ [ID_Valor]q9
q11	→ [nombre alto ancho p...
	→ </puerta>q3
q12	→ [ID_Valor]q11
q13	→ [nombre color]q14
	→ </suelo>q3
q14	→ [ID_Valor]q13

Grammar = (V, T, P, S)

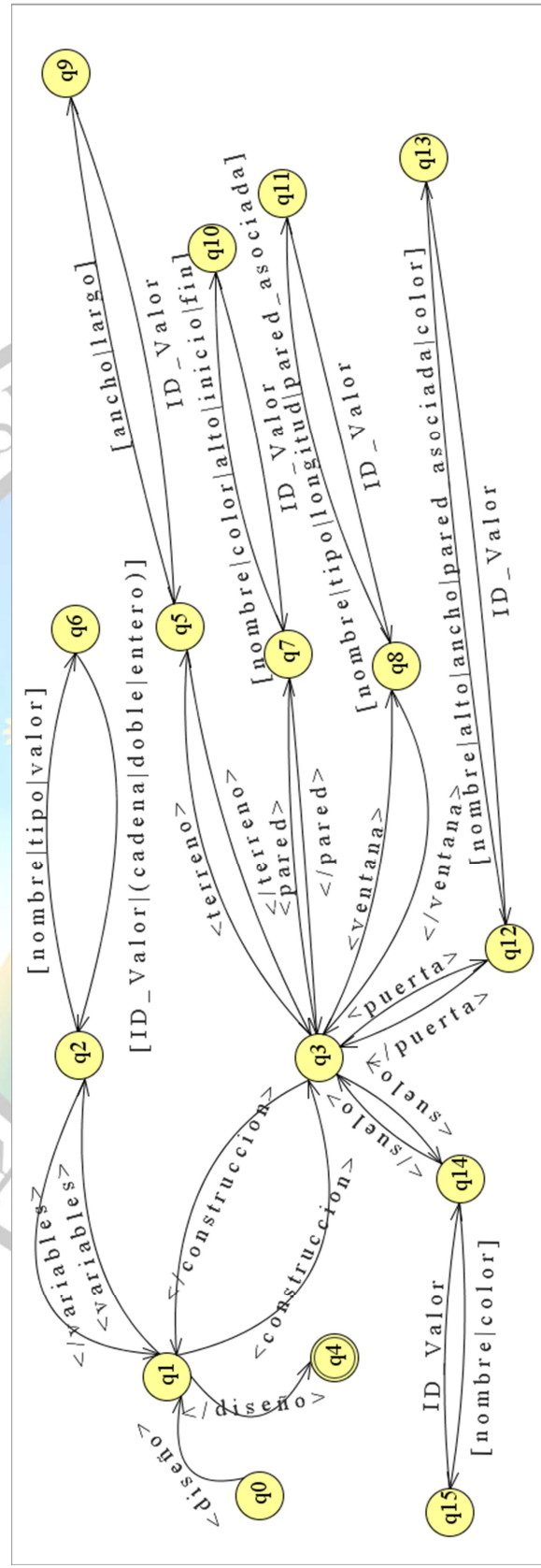
V = { D I N V }

T = { / 0 1 2 3 4 5 6 7 8 9 < > [] _ }

S = D

Table Text Size

Autómata Finito Determinista



Métodos utilizados

Analizador Léxico: Este método sirve para hacer un análisis completo de una cadena entrante, toma carácter por carácter, los agrupa y evalúa con respecto a la Expresión Regular utilizada (Expresión Regular dada en la pg. 2).

Analizador Sintáctico: Este método funciona para tomar todos los tokens obtenidos gracias al analizador léxico, y los compara con respecto al orden establecido en la gramática regular obtenida desde la expresión regular (Gramática Regular dada en la pg. 3).

AgregarAListaTokens: Este método funciona para que los caracteres agrupados en el analizador léxico, sean guardados en una lista de Tokens para su posterior análisis (guarda en base a si pertenece al lenguaje).

AgregarAListaSintactico: Este método funciona para que los tokens analizados de la lista de tokens, si viene en el orden correcto con respecto al lenguaje, los irá guardando.

AgregarAListaErroresLexicos: Este método funciona para que en un momento del analizador léxico encuentra un token al cual no pertenece el lenguaje, se almacene y siga con el análisis.

AgregarAListaErroresSintacticos: Este método funciona para que en el momento del análisis sintáctico, si encuentra un token al cual no está en el orden que pide el lenguaje, entonces será almacenado.

AgregarAListaVariables: Este método funciona para que cuando se detecte que hay una variable en el lenguaje, lo vaya almacenando, su nombre, su tipo y el valor que se le agregue.

AgregarVentana: Este método funciona para que cuando detecte que se pide los datos de una ventana para su posterior uso en la maqueta, guarde sus datos para su posterior uso.

AgregarPared: Este método funciona, para que cuando detecte que se esté dando datos de una pared a utilizar en la maqueta, vaya guardando todos sus datos para su posterior uso.

AgregarSuelo: Este método funciona para que cuando detecte que se estén dando datos de un suelo, se guarden sus datos para su posterior uso.

AgregarPuerta: Este método funciona para que cuando se detecte que se estén dando datos de una puerta, se guarden sus datos para su posterior uso.

AgregarTerreno: Este método funciona para que cuando se detecte que se estén dando los datos del terreno, se guarden para su posterior uso.

generarHTMLTokens: Este método funciona para imprimir la lista de todos los tokens recolectados y los datos almacenados de los elementos sintácticos reconocidos.

generarHTMLErrores: Este método funciona para imprimir la lista de todos los errores obtenidos a la hora del análisis, incluyendo errores de sintaxis.

retornarValorVariable: Este método sirve para, que si hay alguna variable que sea de tipo entero, se obtenga su valor numérico.

retornarValorDoble: Este método sirve para, que si hay alguna variable que sea de tipo doble, se obtenga su valor numérico.

OperacionSuma: Este método sirve para sumar 2 números.

OperacionResta: Este método sirve para restar 2 números.

OperacionMultiplicacion: Este método sirve para multiplicar 2 números.

OperaciónDivision: Este método sirve para dividir 2 números.

validacionDeVariable: Este método sirve para revisar si existe una variable introducida para su posterior uso.