

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
CENTRO UNIVERSITARIO DE OCCIDENTE
DIVISIÓN DE CIENCIAS DE LA INGENIERÍA
ORGANIZACIÓN DE LENGUAJES Y COMPILADORES 1
ING. MOISES GRANADOS
AUX. BRYAN MONZON
PRACTICA 2

MANUAL TÉCNICO
ANALIZADOR WEB
VERSIÓN 1.0

POR:

201730347 HECTOR ADOLFO SON CHICHÉ.

Quetzaltenango, abril de 2021.

ANÁLISIS DE LA GRAMÁTICA PARA ANALIZADOR LÉXICO.

"#".* // comentario simple línea
[/][*][^*]*[*]+([^[/*][^*]*[*]+)*[/] // comentario multiple líneas

lexema

"\$_"[A-Za-z]+([A-Za-z]*[0-9]*"_"*)+
"%_"([A-Za-z]+[0-9]*"_"*)+
[!\"&i\._,]
[0-9]+

tipo token

'IDLEX'
'IDSYNTAX'
'SIMBOL'
'ENTERO'

//CARACTERES DEFINIDOS

"Wison"
"Terminal"
"Syntax"
"No_Terminal"
"Initial_Sim"
"Lex"

'WISON'
'TERMINAL'
'SYNTAX'
'NOTERMINAL'
'INITIALSIM'
'LEX'

Token para palabras
reservadas
"RESERVADO"

"i"
"?"
"/"
"*"
"+"
":"
"{"
"}"
"<"
"-"
";"
"["
"]"
" "
"("
")"
"="
"|"

'INPREG'
'OUTPREG'
'DIAGONAL'
'CERRADURA'
'MAS'
'DOSP'
'LLAVEIN'
'LLAVEOUT'
'MEQ'
'GUION'
'PUNTOCOMA'
'CORIZQ'
'CORDER'
'COMILLASIMPLE'
'PARIN'
'PAROUT'
'IGUAL'
'OR'

Tokens de datos
de los simbolos a
utilizar:
"SIMBOLOS"

'IDENTIFICADOR';

Entonces mi expresión regular quedaría así:
SIMBOLOS | IDLEX | IDSYNNTAX | SIMBOL | ENTERO |
IDENTIFICADOR | RESERVADO.

esto quiere decir que puede ingresar cualquier lexema para
simbolos o un lexema para id lexico o un lexema para id sintactico
o un simbol (carácter) o un numero entero o un identificador
(texto plano) o una palabra reservado.

ANÁLISIS DE LA GRAMÁTICA PARA ANALIZADOR SINTÁCTICO.

$G = (N, T, P, S)$ donde:

N = simbolos no terminales

{ expresion, expreon_lex, exp_lexsig, exp_sintax, exp_sintaxsig,
exp_sigsintx, produccion_sintax, produccion, exp_text, sigconcat,
clausula_ex, exp_concat, tex }

T = simbolos terminales

{ WISON, TERMINAL, SYNTAX, NOTERMINAL, INITIALSIM, LEX,
INPREG, OUTPREG, DIAGONAL, CERRADURA, MAS, DOSP, LLAVEIN,
LLAVEOUT, MEQ, GUION, PUNTOCOMA, CORIZQ, CORDER,
COMILLASIMPLE, PARIN, PAROUT, IGUAL, OR
IDENTIFICADOR, IDLEX, IDSYNNTAX, SIMBOL, ENTERO, WISON,
TERMINAL, SYNTAX, NOTERMINAL, INITIALSIM, LEX }

//estado inicial

$S = \{ \text{expresion} \}$

P = Producciones

P = {

**producciones para iniciar con la palabra wison **

expresion → WISON INPREG LEX LLAVEIN DOSP expreon_lex
| error

**producción para empezar a recibir los terminales que ingresa el usuario para su analizador **

expreon_lex → TERMINAL IDLEX MEQ GUION exp_text exp_lexsig

produccion para recibir varios terminales o ir a la parte del analizador sintactico

exp_lexsig → TERMINAL IDLEX MEQ GUION exp_text exp_lexsig
| DOSP LLAVEOUT exp_sintax

**producción para iniciar el analizador sintactico creado **

exp_sintax → SYNTAX LLAVEIN LLAVEIN DOSP exp_sintaxsig

**produccion para ingresar los no terminales del analizador **

exp_sintaxsig → NOTERMINAL IDSYNNTAX PUNTOCOMA exp_sigsintx

**producciones para crear varios no terminales y tambien indica el simbolo no terminal del analizador creado **

exp_sigsintx → NOTERMINAL IDSYNNTAX PUNTOCOMA exp_sigsintx
| INITIALSIM IDSYNNTAX PUNTOCOMA produccion_sintax

producciones paara agregar las producciones del analizador creado

produccion_sintax → IDSYNNTAX MEQ IGUAL produccion
produccion_sintax
| DOSP LLAVEOUT LLAVEOUT OUTPREG WISON

**produccion para crear todas las producciones del analizador que se va crear por el usuario **

produccion → IDLEX produccion
| IDSYNNTAX produccion
| OR produccion
| PUNTOCOMA

indica el valor que tendra cada simbolo terminal

```
exp_text -> COMILLASIMPLE tex PUNTOCOMA
| exp_concat clausula_ex PUNTOCOMA
| PARIN exp_concat clausula_ex PAROUT sigconcat
| PARIN IDLEX PAROUT sigconcat
```

**para concatenar varios valores para un simbolo terminal **

```
sigconcat -> PARIN exp_concat clausula_ex PAROUT sigconcat
| PARIN IDLEX PAROUT sigconcat
| PUNTOCOMA
```

**indica la clausula que tendra un simbolo terminal **

```
clausula_ex -> CERRADURA
| MAS
| OUTPREG
| /*lambda*/
```

**es para los valores que tendra una simbolo no terminal para definir una secuencia de caracteres **

```
exp_concat -> CORIZQ ENTERO GUION ENTERO CORDER
| CORIZQ IDENTIFICADOR GUION IDENTIFICADOR CORDER
```

** ipara tener como valor un carácter o una palabra especial de un simbolo terminal **

```
tex -> IDENTIFICADOR tex
| SIMBOL tex
| ENTERO COMILLASIMPLE
| INPREG COMILLASIMPLE
| OUTPREG COMILLASIMPLE
| DIAGONAL COMILLASIMPLE
| CERRADURA COMILLASIMPLE
| DOSP COMILLASIMPLE
| LLAVEIN COMILLASIMPLE
| LLAVEOUT COMILLASIMPLE
| MEQ COMILLASIMPLE
| GUION COMILLASIMPLE
| PUNTOCOMA COMILLASIMPLE
| CORIZQ COMILLASIMPLE
| CORDER COMILLASIMPLE
| COMILLASIMPLE COMILLASIMPLE
| PARIN COMILLASIMPLE
| PAROUT COMILLASIMPLE
| IGUAL COMILLASIMPLE
| OR COMILLASIMPLE
| MAS COMILLASIMPLE
| COMILLASIMPLE
```