

PROYECTO: GENERADOR DE SCANNER

Fase 1.

Fecha de entrega: MARZO 15

SCANNER (ANALIZADOR DE LÉXICO)

- Objetivos
 - La primera fase tiene como objetivo aplicar los conceptos asociados a las expresiones regulares y autómatas finitos determinísticos
 - Las expresiones regulares y los autómatas como reconocedores de lenguajes
 - El scanner (analizador de léxico) como fase inicial del proceso de compilación de un lenguaje de programación

DEFINICIÓN

- El scanner recibirá un archivo de entrada con las definiciones de las expresiones regulares y los tokens que reconoce
- Las secciones son:
 - Sets (conjuntos)
 - Tokens (caracteres)
 - Actions (acciones asociadas a los tokens)
 - Error (token de error)

SETS

- Acá vamos a definir los conjuntos de caracteres que puede reconocer el lenguaje, y que serán usados en la siguiente sección de tokens

- La estructura es:

SETS

letra = 'A'..'Z'+'a'..'z'+'_'

digito = '0'..'9'

charset = chr(32)..chr(254)

TOKENS

- Caracteres y expresiones regulares que pueden ser reconocidas por el lenguaje del scanner

- La estructura es:

Tokens

Token 1 = digito digito *

toKen 2="" CHARSET ""|"" CHARSET ""

token 3 = '='

token 4 = '<>'

token 5 = '>''='

token 6 = 'A''N''D'

toKEN 7 = 'O''R'

TOKEN 40 = '(' '*'

token 3 = letra (LETRA|digito)* { Reservadas() }

ACTIONS

- Acá se pueden definir acciones asociadas a los tokens para acciones particulares sobre los mismos

- La estructura es:

Actions

Reservadas()

{

18='PROGRAM'

19='INCLUDE'

20='CONST'

}

ERROR

- En esta sección, simplemente colocaremos el token de error cuando el símbolo no corresponda al lenguaje
- La estructura es:
Error = 54

ESTRUCTURA COMPLETA

```
SETS
    LETRA    = 'A'..'Z'+'a'..'z'+'_'
    DIGITO   = '0'..'9'
    CHARSET  = CHR(32)..CHR(254)

TOKENS
    TOKEN 1= DIGITO DIGITO *
    TOKEN 2=''' CHARSET '''|''' CHARSET '''
    TOKEN 4 = '='
    TOKEN 5 = '<'>'
    TOKEN 6 = '<'
    TOKEN 7 = '>'
    TOKEN 8 = '>' '='
    TOKEN 9 = '<' '='
    TOKEN 10 = '+'
    TOKEN 11 = '-'
    TOKEN 12 = 'O' 'R'
    TOKEN 13 = '*'
    TOKEN 14 = 'A' 'N' 'D'
    TOKEN 15 = 'M' 'O' 'D'
    TOKEN 16 = 'D' 'I' 'V'
    TOKEN 3= LETRA ( LETRA | DIGITO )* { RESERVADAS() }

ACTIONS
    RESERVADAS()
    {
        18 = 'PROGRAM'
        19 = 'INCLUDE'
        20 = 'CONST'
        21 = 'TYPE'
        22 = 'VAR'
    }

ERROR = 54
```


FASE 1 (ENTREGA 15 MARZO)

- La primera fase consiste en la lectura del archivo de entrada con las especificaciones y la construcción del DFA completo.
- Los pasos son:
 - Procesar archivo de entrada
 - Detectando errores y desplegándolos en pantalla
 - Mostrando la línea y columna procesada
 - Crear el árbol de expresiones
 - Calcular las funciones de First, Last y Follow
 - Crear el DFA completo de todos los tokens
 - Mostrar el DFA
 - Tabla /DFA gráfico

NORMAS

- El proyecto es individual
- La copia se sanciona: proyecto total (10 pts) con nota de 0 pts.
- Por ser la fase 1 de proyecto, la persona que no termine esta fase, puede dejar con 0 pts (10 pts de esta fase) y continuar para lograr entregar la fase final.
- No se aceptan entregas parciales de esta fase.