Tp 2

1. Diseñar categorías léxicas

<Palabras reservada> -> una de **int short**

Identificadores:

<identificador> -> <letrasMin>| <letrasMin> <Digito>

<Digito> -> uno de **1 2 3 4 5 6 7 8 9**

<letrasMin> -> <letrasMin>< letras > |<letrasMin>

<letras> -> uno de **a e i o u**

Constantes:

<Constantes> -> <Digito><Constantes>|<Digito>

<Digito> -> uno de **0 1 2 3 4 5 6 7 8 9**

Operadores :

<Operadores> -> uno de **+ \* - =**

Caracteres de puntuación :

<CaracteresDePuntuacion>-> una de **( ) { } ;**

1. Formalizar cada Lenguaje Regular con Expresiones Regulares

Palabras Reservadas : int + short

Identificadores : [aeiou]+[1\_9] | [aeiou]+

Constantes : [0\_9]+

Operadores : [+\*-=]

Caracteres de Puntuación: [(){};]

1. Definir Autómatas Finitos para cada lenguaje

Palabras Reservadas: {{S-,P+},{int,short},S,{ P+},{S->int P| short P}}.

Identificadores: {{ S-,T+,P+},{a,e,i,o,u,0,1,2,3,4,5,6,7,8,9},S,{ T+,P+},{ S ->{a,e,i,o,u} P ,P-{a,e,i,o,u}P, P -> {1\_9} T}

Constantes: {{ S-,P+},{0,1,2,3,4,5,6,7,8,9},S,{P+},{S -> {0\_9} P , P -> {0\_9} P }}

Operadores: {{ S-,P+},{+,\*,-,=},S ,{P+},{S -> + P, S -> - P, S -> \* P , S -> = P }}

Caracteres de Puntuación: {{ S-,P+},{(,),[,],;},S ,{P+},{S -> ( P, S -> ) P, S -> [ P , S -> ] P , S -> ; P }}

1. Hacer un programa en C para determinar si un lexema pertenece a la Categoría

Léxica correspondiente

Gramatica Sintactica

Ejeplo Micro:

<programa> -> **inicio** <listaSentencias> **fin**

<listaSentencias> -> <sentencia> {<sentencia>}

<sentencia> -> <identificador> **:=** <expresion>**;**|

**leer(**<listaIdentificadores>**);**|

**escribir(**<listaExpresiones>**);**

**<**listaIdentificadores> ->

  <identificador> {, <identificador>}

<listaExpresiones> ->

  <expresion> {, <expresion>}

<expresion> -> <primaria> {<operadorAditivo> <primaria>}

<primaria> -> <identificador> | <constante> |**(** <expresion> **)**

Cada Token debe tener su nombre



ADF para implementar escanner

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TT | LETRA | DIGITO | + | - | \* | = | ( | ) | ; | fdt | sp | otro |
| 0 | 1 | 3 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 0 | 13 |
| 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 2+ | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 |
| 3 | 4 | 3 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 4+ | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 |
| 5+ | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 |
| 6+ | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 |
| 7+ | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 |
| 8+ | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 |
| 9+ | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 |
| 10+ | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 |
| 11+ | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 |
| 12+ | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 |
| 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 |