**PROYECTO PRÁCTICO FASE II**

**GENERADOR DE SCANNER**

**Objetivo**

En la Fase II del proyecto es pre-requisito haber entregado completamente la Fase I utilizando árboles de expresiones, ya que en esta fase, el entregable es la tabla de First, Last, Follow y también la tabla de transiciones del autómata Finito Determinista.

Estas tablas deben ser obtenidas de la sección “TOKENS” del archivo de entrada

**Explicación de la salida**

En la segunda fase del proyecto es necesario la lectura de un archivo de texto llamado: GRAMATICA.txt el cual contiene la definición de la gramática.



Al leerlo el programa generará una cadena con la expresión regular proveniente de la sección tokens, por ejemplo utilizando el archivo anterior tenemos:

DIGITO DIGITO \* | '"' CHARSET '"' | ''' CHARSET ''' | '=' | '<' '>' | '<' | '>' | '>' '=' | '<' '=' | '+' | '-' | 'O' 'R' | '\*' | 'A' 'N' 'D' | 'M' 'O' 'D' | 'D' 'I' 'V' | 'N' 'O' 'T' | '(' '\*' | '\*' ')' | ';' | '.' | '{' | '}' | '(' | ')' | '[' | ']' | '.' '.' | ':' | ',' | ':' '=' | LETRA ( LETRA | DIGITO ) \*

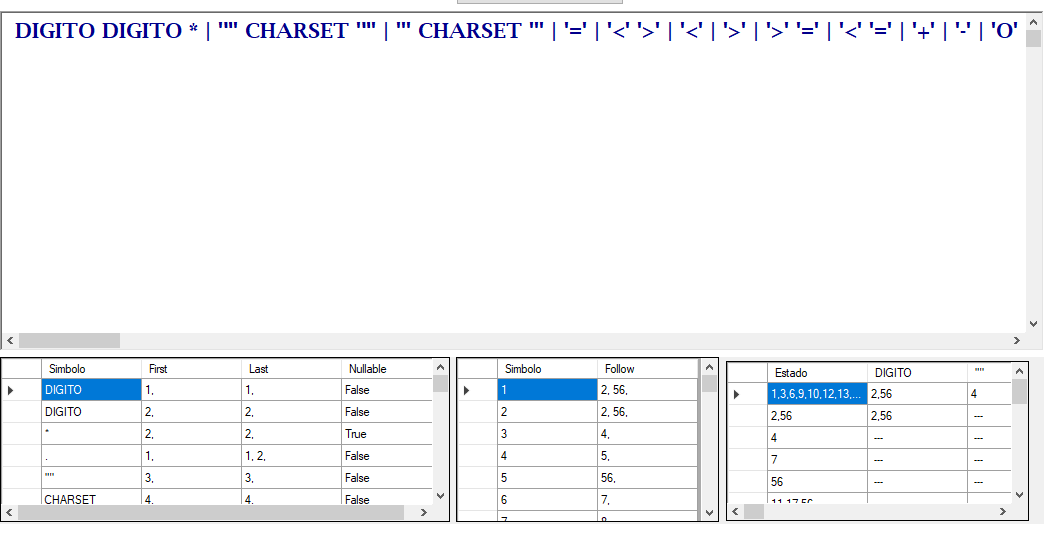
Con esa expresión regular, se genera el árbol de expresiones y se obtiene la tabla de First, Last, Follow y la tabla de transiciones del Autómata Finito Determinista, para el ejemplo anterior tendríamos:

|  |  |  |  |
| --- | --- | --- | --- |
| **SIMBOLO** | **FIRST** | **LAST** | **NULLABLE** |
| DIGITO | 1, | 1, | False |
| DIGITO | 2, | 2, | False |
| \* | 2, | 2, | True |
| . | 1, | 1, 2, | False |
| '"' | 3, | 3, | False |
| CHARSET | 4, | 4, | False |
| . | 3, | 4, | False |
| '"' | 5, | 5, | False |
| . | 3, | 5, | False |
| | | 1, 3, | 1, 2, 5, | False |
| ''' | 6, | 6, | False |
| CHARSET | 7, | 7, | False |
| . | 6, | 7, | False |
| ''' | 8, | 8, | False |
| . | 6, | 8, | False |
| | | 1, 3, 6, | 1, 2, 5, 8, | False |
| '=' | 9, | 9, | False |
| | | 1, 3, 6, 9, | 1, 2, 5, 8, 9, | False |
| '<' | 10, | 10, | False |
| '>' | 11, | 11, | False |
| . | 10, | 11, | False |
| | | 1, 3, 6, 9, 10, | 1, 2, 5, 8, 9, 11, | False |
| '<' | 12, | 12, | False |
| | | 1, 3, 6, 9, 10, 12, | 1, 2, 5, 8, 9, 11, 12, | False |
| '>' | 13, | 13, | False |
| | | 1, 3, 6, 9, 10, 12, 13, | 1, 2, 5, 8, 9, 11, 12, 13, | False |
| '>' | 14, | 14, | False |
| '=' | 15, | 15, | False |
| . | 14, | 15, | False |
| | | 1, 3, 6, 9, 10, 12, 13, 14, | 1, 2, 5, 8, 9, 11, 12, 13, 15, | False |
| '<' | 16, | 16, | False |
| '=' | 17, | 17, | False |
| . | 16, | 17, | False |
| | | 1, 3, 6, 9, 10, 12, 13, 14, 16, | 1, 2, 5, 8, 9, 11, 12, 13, 15, 17, | False |
| '+' | 18, | 18, | False |
| | | 1, 3, 6, 9, 10, 12, 13, 14, 16, 18, | 1, 2, 5, 8, 9, 11, 12, 13, 15, 17, 18, | False |
| '-' | 19, | 19, | False |
| | | 1, 3, 6, 9, 10, 12, 13, 14, 16, 18, 19, | 1, 2, 5, 8, 9, 11, 12, 13, 15, 17, 18, 19, | False |
| 'O' | 20, | 20, | False |
| 'R' | 21, | 21, | False |
| . | 20, | 21, | False |
| | | 1, 3, 6, 9, 10, 12, 13, 14, 16, 18, 19, 20, | 1, 2, 5, 8, 9, 11, 12, 13, 15, 17, 18, 19, 21, | False |
| '\*' | 22, | 22, | False |
| | | 1, 3, 6, 9, 10, 12, 13, 14, 16, 18, 19, 20, 22, | 1, 2, 5, 8, 9, 11, 12, 13, 15, 17, 18, 19, 21, 22, | False |
| 'A' | 23, | 23, | False |
| 'N' | 24, | 24, | False |
| . | 23, | 24, | False |
| 'D' | 25, | 25, | False |
| . | 23, | 25, | False |
| | | 1, 3, 6, 9, 10, 12, 13, 14, 16, 18, 19, 20, 22, 23, | 1, 2, 5, 8, 9, 11, 12, 13, 15, 17, 18, 19, 21, 22, 25, | False |
| 'M' | 26, | 26, | False |
| 'O' | 27, | 27, | False |
| . | 26, | 27, | False |
| 'D' | 28, | 28, | False |
| . | 26, | 28, | False |
| | | 1, 3, 6, 9, 10, 12, 13, 14, 16, 18, 19, 20, 22, 23, 26, | 1, 2, 5, 8, 9, 11, 12, 13, 15, 17, 18, 19, 21, 22, 25, 28, | False |
| 'D' | 29, | 29, | False |
| 'I' | 30, | 30, | False |
| . | 29, | 30, | False |
| 'V' | 31, | 31, | False |
| . | 29, | 31, | False |
| | | 1, 3, 6, 9, 10, 12, 13, 14, 16, 18, 19, 20, 22, 23, 26, 29, | 1, 2, 5, 8, 9, 11, 12, 13, 15, 17, 18, 19, 21, 22, 25, 28, 31, | False |
| 'N' | 32, | 32, | False |
| 'O' | 33, | 33, | False |
| . | 32, | 33, | False |
| 'T' | 34, | 34, | False |
| . | 32, | 34, | False |
| | | 1, 3, 6, 9, 10, 12, 13, 14, 16, 18, 19, 20, 22, 23, 26, 29, 32, | 1, 2, 5, 8, 9, 11, 12, 13, 15, 17, 18, 19, 21, 22, 25, 28, 31, 34, | False |
| '(' | 35, | 35, | False |
| '\*' | 36, | 36, | False |
| . | 35, | 36, | False |
| | | 1, 3, 6, 9, 10, 12, 13, 14, 16, 18, 19, 20, 22, 23, 26, 29, 32, 35, | 1, 2, 5, 8, 9, 11, 12, 13, 15, 17, 18, 19, 21, 22, 25, 28, 31, 34, 36, | False |
| '\*' | 37, | 37, | False |
| ')' | 38, | 38, | False |
| . | 37, | 38, | False |
| | | 1, 3, 6, 9, 10, 12, 13, 14, 16, 18, 19, 20, 22, 23, 26, 29, 32, 35, 37, | 1, 2, 5, 8, 9, 11, 12, 13, 15, 17, 18, 19, 21, 22, 25, 28, 31, 34, 36, 38, | False |
| ';' | 39, | 39, | False |
| | | 1, 3, 6, 9, 10, 12, 13, 14, 16, 18, 19, 20, 22, 23, 26, 29, 32, 35, 37, 39, | 1, 2, 5, 8, 9, 11, 12, 13, 15, 17, 18, 19, 21, 22, 25, 28, 31, 34, 36, 38, 39, | False |
| '.' | 40, | 40, | False |
| | | 1, 3, 6, 9, 10, 12, 13, 14, 16, 18, 19, 20, 22, 23, 26, 29, 32, 35, 37, 39, 40, | 1, 2, 5, 8, 9, 11, 12, 13, 15, 17, 18, 19, 21, 22, 25, 28, 31, 34, 36, 38, 39, 40, | False |
| '{' | 41, | 41, | False |
| | | 1, 3, 6, 9, 10, 12, 13, 14, 16, 18, 19, 20, 22, 23, 26, 29, 32, 35, 37, 39, 40, 41, | 1, 2, 5, 8, 9, 11, 12, 13, 15, 17, 18, 19, 21, 22, 25, 28, 31, 34, 36, 38, 39, 40, 41, | False |
| '}' | 42, | 42, | False |
| | | 1, 3, 6, 9, 10, 12, 13, 14, 16, 18, 19, 20, 22, 23, 26, 29, 32, 35, 37, 39, 40, 41, 42, | 1, 2, 5, 8, 9, 11, 12, 13, 15, 17, 18, 19, 21, 22, 25, 28, 31, 34, 36, 38, 39, 40, 41, 42, | False |
| '(' | 43, | 43, | False |
| | | 1, 3, 6, 9, 10, 12, 13, 14, 16, 18, 19, 20, 22, 23, 26, 29, 32, 35, 37, 39, 40, 41, 42, 43, | 1, 2, 5, 8, 9, 11, 12, 13, 15, 17, 18, 19, 21, 22, 25, 28, 31, 34, 36, 38, 39, 40, 41, 42, 43, | False |
| ')' | 44, | 44, | False |
| | | 1, 3, 6, 9, 10, 12, 13, 14, 16, 18, 19, 20, 22, 23, 26, 29, 32, 35, 37, 39, 40, 41, 42, 43, 44, | 1, 2, 5, 8, 9, 11, 12, 13, 15, 17, 18, 19, 21, 22, 25, 28, 31, 34, 36, 38, 39, 40, 41, 42, 43, 44, | False |
| '[' | 45, | 45, | False |
| | | 1, 3, 6, 9, 10, 12, 13, 14, 16, 18, 19, 20, 22, 23, 26, 29, 32, 35, 37, 39, 40, 41, 42, 43, 44, 45, | 1, 2, 5, 8, 9, 11, 12, 13, 15, 17, 18, 19, 21, 22, 25, 28, 31, 34, 36, 38, 39, 40, 41, 42, 43, 44, 45, | False |
| ']' | 46, | 46, | False |
| | | 1, 3, 6, 9, 10, 12, 13, 14, 16, 18, 19, 20, 22, 23, 26, 29, 32, 35, 37, 39, 40, 41, 42, 43, 44, 45, 46, | 1, 2, 5, 8, 9, 11, 12, 13, 15, 17, 18, 19, 21, 22, 25, 28, 31, 34, 36, 38, 39, 40, 41, 42, 43, 44, 45, 46, | False |
| '.' | 47, | 47, | False |
| '.' | 48, | 48, | False |
| . | 47, | 48, | False |
| | | 1, 3, 6, 9, 10, 12, 13, 14, 16, 18, 19, 20, 22, 23, 26, 29, 32, 35, 37, 39, 40, 41, 42, 43, 44, 45, 46, 47, | 1, 2, 5, 8, 9, 11, 12, 13, 15, 17, 18, 19, 21, 22, 25, 28, 31, 34, 36, 38, 39, 40, 41, 42, 43, 44, 45, 46, 48, | False |
| ':' | 49, | 49, | False |
| | | 1, 3, 6, 9, 10, 12, 13, 14, 16, 18, 19, 20, 22, 23, 26, 29, 32, 35, 37, 39, 40, 41, 42, 43, 44, 45, 46, 47, 49, | 1, 2, 5, 8, 9, 11, 12, 13, 15, 17, 18, 19, 21, 22, 25, 28, 31, 34, 36, 38, 39, 40, 41, 42, 43, 44, 45, 46, 48, 49, | False |
| ',' | 50, | 50, | False |
| | | 1, 3, 6, 9, 10, 12, 13, 14, 16, 18, 19, 20, 22, 23, 26, 29, 32, 35, 37, 39, 40, 41, 42, 43, 44, 45, 46, 47, 49, 50, | 1, 2, 5, 8, 9, 11, 12, 13, 15, 17, 18, 19, 21, 22, 25, 28, 31, 34, 36, 38, 39, 40, 41, 42, 43, 44, 45, 46, 48, 49, 50, | False |
| ':' | 51, | 51, | False |
| '=' | 52, | 52, | False |
| . | 51, | 52, | False |
| | | 1, 3, 6, 9, 10, 12, 13, 14, 16, 18, 19, 20, 22, 23, 26, 29, 32, 35, 37, 39, 40, 41, 42, 43, 44, 45, 46, 47, 49, 50, 51, | 1, 2, 5, 8, 9, 11, 12, 13, 15, 17, 18, 19, 21, 22, 25, 28, 31, 34, 36, 38, 39, 40, 41, 42, 43, 44, 45, 46, 48, 49, 50, 52, | False |
| LETRA | 53, | 53, | False |
| LETRA | 54, | 54, | False |
| DIGITO | 55, | 55, | False |
| | | 54, 55, | 54, 55, | False |
| \* | 54, 55, | 54, 55, | True |
| . | 53, | 53, 54, 55, | False |
| | | 1, 3, 6, 9, 10, 12, 13, 14, 16, 18, 19, 20, 22, 23, 26, 29, 32, 35, 37, 39, 40, 41, 42, 43, 44, 45, 46, 47, 49, 50, 51, 53, | 1, 2, 5, 8, 9, 11, 12, 13, 15, 17, 18, 19, 21, 22, 25, 28, 31, 34, 36, 38, 39, 40, 41, 42, 43, 44, 45, 46, 48, 49, 50, 52, 53, 54, 55, | False |
| # | 56, | 56, | False |
| . | 1, 3, 6, 9, 10, 12, 13, 14, 16, 18, 19, 20, 22, 23, 26, 29, 32, 35, 37, 39, 40, 41, 42, 43, 44, 45, 46, 47, 49, 50, 51, 53, | 56, | False |

**Tabla de First, Last y Nullable**

|  |  |
| --- | --- |
| **SÍMBOLO** | **FOLLOW** |
| 1 | 2, 56, |
| 2 | 2, 56, |
| 3 | 4, |
| 4 | 5, |
| 5 | 56, |
| 6 | 7, |
| 7 | 8, |
| 8 | 56, |
| 9 | 56, |
| 10 | 11, |
| 11 | 56, |
| 12 | 56, |
| 13 | 56, |
| 14 | 15, |
| 15 | 56, |
| 16 | 17, |
| 17 | 56, |
| 18 | 56, |
| 19 | 56, |
| 20 | 21, |
| 21 | 56, |
| 22 | 56, |
| 23 | 24, |
| 24 | 25, |
| 25 | 56, |
| 26 | 27, |
| 27 | 28, |
| 28 | 56, |
| 29 | 30, |
| 30 | 31, |
| 31 | 56, |
| 32 | 33, |
| 33 | 34, |
| 34 | 56, |
| 35 | 36, |
| 36 | 56, |
| 37 | 38, |
| 38 | 56, |
| 39 | 56, |
| 40 | 56, |
| 41 | 56, |
| 42 | 56, |
| 43 | 56, |
| 44 | 56, |
| 45 | 56, |
| 46 | 56, |
| 47 | 48, |
| 48 | 56, |
| 49 | 56, |
| 50 | 56, |
| 51 | 52, |
| 52 | 56, |
| 53 | 54, 55, 56, |
| 54 | 54, 55, 56, |
| 55 | 54, 55, 56, |
| 56 |  |

**Tabla de Follow**



**SET DE PRUEBAS**

En esta sección se muestra un set de pruebas que se deben incluir y sus respectivos resultados:

|  |  |
| --- | --- |
| **Archivo de pruebas** | **Resultado esperado** |
|  | Formato correcto / Tabla de Firsta, Last, Follow de acuerdo a la expresión regular ingresada, generación de la tabla de transiciones. |
|  | Formato correcto. / Tabla de Firsta, Last, Follow de acuerdo a la expresión regular ingresada, generación de la tabla de transiciones. |

**Entregables Fase II**

El programa debe codificarse en el lenguaje que dicidió en la fase I, muy importante el repositorio en github en donde se pueda dar seguimiento a los “commit” realizados.

Entregables:

* Programa fuente, debidamente documentado internamente
* Programa ejecutable
* Set de casos de prueba utilizados con sus respectivos archivos y resultados esperados.
* Documentación de análisis que contenga las expresiones regulares y árboles de expresiones creados para parte del archivo.
* Tablas de First, Last, Follow y Tabla de transiciones
* Extra: Mostrar el árbol de expresiones.

FECHA DE ENTREGA:

Calificación presencial e individual el día: **sábado 6 de abril de 2024** en horario de clase, el código fuente a evaluar será el que esté subido en la sección de entrega del portal.

**Aspectos a evaluar**

* Validación de errores.
* Adecuada aplicación de los conocimientos vistos en clase
* Calidad de la documentación: ortografía, orden, limpieza y que esté completa.
* Calidad de la solución propuesta: que solucione el problema (que haga lo que requiere el enunciado) en forma eficaz.
* Funcionalidad del programa: debe cumplir a cabalidad con todos los requerimientos.
* Evidencia de la creación del programa y dominio de los conceptos utilizados.
* Creatividad.

**Baremo de calificación**

|  |  |  |  |
| --- | --- | --- | --- |
| ITEM | Resultado | Ponderación | Observaciones |
| Solución entregada en el portal | Sí / No | -- | Si no está en el portal no se califican los demás ítems y se obtiene 0 puntos |
| Solución compila | Sí / No | -- | Si la solución no compila se obtiene 0 puntos |
| Lectura de archivos correcta | Sí / No | 10 puntos | Lee archivos de forma correcta con filtros y no se produce ninguna excepción durante la lectura de archivos. |
| Creación de tablas correcta | Sí / No | 15 puntos por cada archivo de prueba, total: 75 puntos | Genera los resultados esperados de cada archivo de prueba (El catedrático pondrá los archivos de prueba) |
| Creación correcta de tabla de transiciones | Sí / No | 10 puntos | No utiliza librerías de evaluación de expresiones regulares ni herramientas de análisis lexicográfico |
| Documentación técnica con expresiones regulares y árboles de expresión | Sí / No | 5 puntos | Presenta documento que muestra razonamiento utilizado. |
| Muestra el árbol | Sí / No | 5 puntos extra | De forma creativa muestra el árbol sintáctico |