

Universidad Rafael Landívar

Facultad de Ingeniería

Lenguajes Formales y Autómatas

Sección 01

Ingeniero Moises Alonso



Proyecto Fase 2

Generador de Scanner

Pablo Muralles 1113818

Guatemala 13 de abril del 2020

Objetivo General

Dentro del análisis de los lenguajes Formales es importante conocer las fases del proceso de compilación, la finalidad del proyecto es que el estudiante comprenda la función de los analizadores léxico y sintáctico de un compilador a través de la generación de un programa que sea capaz de reconocer un lenguaje y finalmente evaluar si las palabras utilizadas están bien formadas de acuerdo con una gramática. El proyecto consta de 3 fases, las cuales son dependientes, es decir, que para poder terminar la fase III, las anteriores deben estar completas, de lo contrario no se podrá entregar el funcionamiento completo.

Objetivos Específicos

- 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
- Realizar el análisis sintáctico de la gramática ingresada.
- Aplicar los conocimientos aprendidos en la clase hasta el momento para realizar la esta fase.
- Generar de manera adecuada las tablas de First, Last, Follows y Estados.

Análisis

- Entradas: Archivos de texto donde contenga la gramática para generar las tablas.
- Salida: Tablas de first, last, follows y estados.
- Procesos:
 1. Generar la expresión regular a partir de la definición de los tokens.
 2. Crear el árbol de expresiones de la nueva expresión regular.
 3. Calcular los first, last y follows
 4. Calcular los estados a partir del first de la raíz del árbol de expresiones y los follows.

Pseudocódigo

A continuación, encontrara el pseudocódigo de los métodos utilizados para encontrar las tablas de first, last, follows y estados.

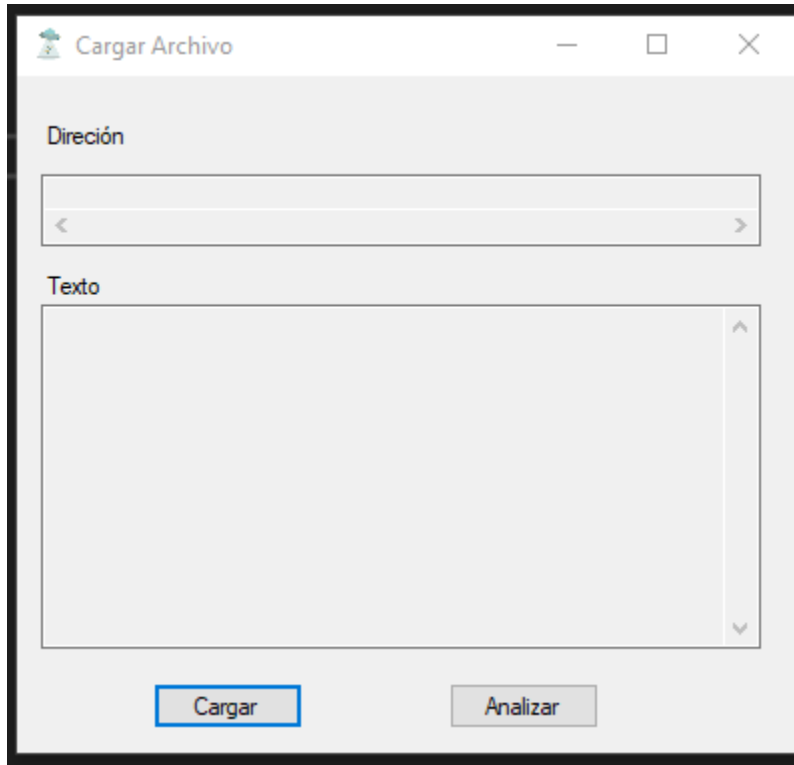
<p>Algoritmo para calcular las tablas de first, last y lo follows Nombre del método: RecorridoPostorden</p>
<p>Entradas: Nodo del árbol de expresiones “raíz”. Diccionario<int,list<int>> Follows . Lista <nodo> ReccorridoFirstLast Int contadorterminales</p>
<p>Salida: First y last para cada nodo del árbol que las tenga. Diccionario Con los Follws para cada símbolo.</p>
<p>Proceso:</p> <ol style="list-style-type: none">1. Si la raíz es diferente a nula<ol style="list-style-type: none">a. Llamar de nuevo al método y enviarle como parámetro el hijo izquierdo de la raíz.b. Llamar de nuevo al método y enviarle como parámetro el hijo derecho de la raíz.c. Agregar la raíz a la lista RecorrdioFirstLastd. Si la raíz es hoja<ol style="list-style-type: none">1. Se le agrega a la propiedad del nodo Numero el contadorterminales2. Se le agrega a la propiedad del nodo first el contadorterminales3. Se le agrega a la propiedad del nodo last el contadorterminales4. Se agrega al diccionario la propiedad del nodo numero y el valor inicializando la lista int5. Se le suma 1 al contadorterminalese. Sino la propiedad data de la raíz es igual a “*”<ol style="list-style-type: none">1. La propiedad de la raíz nutable se pone true2. La propiedad de la raíz first es igual al first de su hijo izquierdo3. La propiedad de la raíz last es igual al last de su hijo izquierdo<ol style="list-style-type: none">a. Mientras existan en el hijo izquierdo flast<ol style="list-style-type: none">1. Mientras existan first en el hijo izquierdo<ol style="list-style-type: none">a. Buscar en el diccionario de follows la lave del last y agregarle al valor del diccionario el firstf. Sino la propiedad data de la raíz es igual a “+”<ol style="list-style-type: none">1. La propiedad de la raíz first es igual al first de su hijo izquierdo2. La propiedad de la raíz last es igual al last de su hijo izquierdo<ol style="list-style-type: none">a. Mientras existan en el hijo izquierdo flast<ol style="list-style-type: none">1. Mientras existan first en el hijo izquierdo<ol style="list-style-type: none">a. Buscar en el diccionario de follows la lave del last y agregarle al valor del diccionario el first

- g. Sino la propiedad data de la raíz es igual a “?”
 - 1. La propiedad de la raíz nutable se pone true
 - 2. La propiedad de la raíz first es igual al first del hijo izquierdo de la raíz
 - 3. La propiedad de la raíz last es igual al last del hijo izquierdo de la raíz
- h. Sino la propiedad data de la raíz es igual a “|”
 - 1. Si el hijo izquierdo tiene la propiedad nutable igual a true o el hijo derecho tiene la propiedad nutable igual a true
 - a. La propiedad de la raíz nutable va a ser igual a true
 - 2. Mientras existan first en el hijo izquierdo
 - a. Agregar los first del hijo izquierdo al first de la raíz
 - 3. Mientras existan firsts en el hijo derecho
 - a. Agregar a la propiedad de la raíz firsts los first del hijo derecho
 - 4. Mientras existan last en el hijo izquierdo
 - b. Agregar los last del hijo izquierdo al last de la raíz
 - 5. Mientras existan last en el hijo derecho
 - b. Agregar a la propiedad de la raíz last los last del hijo derecho
- i. Sino la propiedad data de la raíz es igual a “.”
 - 1. Si el hijo de la raíz es nutable y el hijo derecho es nutable la propiedad de la raíz nutable es igual a true
 - 2. Si el hijo izquierdo es nutable
 - a. Mientras existan first en el hijo izquierdo
 - 1. Agregarlos a la propiedad first de la raíz
 - b. Mientras existan first en el hijo derecho
 - 1. Agregarlos a la propiedad de la raíz first
 - 3. Sino
 - a. Mientras existan first en el hijo izquierdo
 - 1. agregarlos a la propiedad de la raíz first
 - 4. Si el hijo derecho es nutable
 - a. Mientras existan last en el hijo izquierdo
 - 1. Agregarlos a la propiedad last de la raíz
 - b. Mientras existan last en el hijo derecho
 - 1. Agregarlos a la propiedad de la raíz last
 - 5. Sino
 - a. Mientras el hijo derecho tenga last
 - 1. agregarlos a la propiedad de la raíz last
 - 6. Mientras el hijo izquierdo tenga last
 - a. Mientras el hijo derecho tenga first
 - 1. Si en el valor del diccionario con la clave del last no está contenido el first
 - a. agregar el first en el diccionario con la clave del last.

<p style="text-align: center;">Algoritmo para calcular los Estados Nombre del Método: CrearEstados</p>
<p>Entradas:</p> <ol style="list-style-type: none"> 1. Nodo "Árbol" 2. Diccionario <List<int>, Dictionary<string, List<int>>> "TablaEstados" 3. Cola <List<int>> "EstadosAprobar" 4. Lista <List<int>> "EstadosHistorial" 5. Lista <t<int>> "Concordancia" 6. Lista <<int>> "follows" 7. Lista <string> "terminales" 8. List<int> "estadoinicial" 9. Diccionario<int, List<int>> "Follows"
<p>Salida: Diccionario <List<int>, Dictionary<string, List<int>>> "TablaEstados"</p>
<p>Proceso:</p> <ol style="list-style-type: none"> 1. Agregar al estadohistorial el estado inicial 2. Mientras existan terminales <ol style="list-style-type: none"> a. Mientras Existan números en el estado inicial <ol style="list-style-type: none"> 1. Si el símbolo de terminales es igual al símbolo del estado inicial <ol style="list-style-type: none"> a. agregarlo a la lista concordancia b. Mientras existan números en la listaconcardancia <ol style="list-style-type: none"> 1. buscar en el diccionario follows la llave con el numero de la Listaconcordancia y agregar los follows a la lista follows c. Se crea un diccionario <string,list<int>> dicmodificar y se agrega al de valor el símbolo y de valor la listafollows d. Si la lista follows no existe en estadosaprobar y estadoshistorial y la listafollows no está vacía <ol style="list-style-type: none"> 1. se agrega la lista follows a estadosaprobar 3. Agregar al diccionario tablaestados con la llave el estado inicial y de valor el dicccmodificar 4. Mientras estadosaprobar no este vacía <ol style="list-style-type: none"> a. Mientras existan terminales <ol style="list-style-type: none"> 1. Mientras Existan números en el estado inicial <ol style="list-style-type: none"> a. Si el símbolo de terminales es igual al símbolo del estado inicial <ol style="list-style-type: none"> 1. agregarlo a la lista concordancia 2. Mientras existan números en la listaconcardancia <ol style="list-style-type: none"> 1. buscar en el diccionario follows la llave con el numero de la Listaconcordancia y agregar los follows a la lista follows 3. Se crea un diccionario <string,list<int>> dicmodificar y se agrega al de valor el símbolo y de valor la listafollows 4. Si la lista follows no existe en estadosaprobar y estadoshistorial y la listafollows no está vacía <ol style="list-style-type: none"> 1. se agrega la lista follows a estadosaprobar b. Agregar al diccionario tablaestados con la llave e l estado inicial y de valor el dicccmodificar

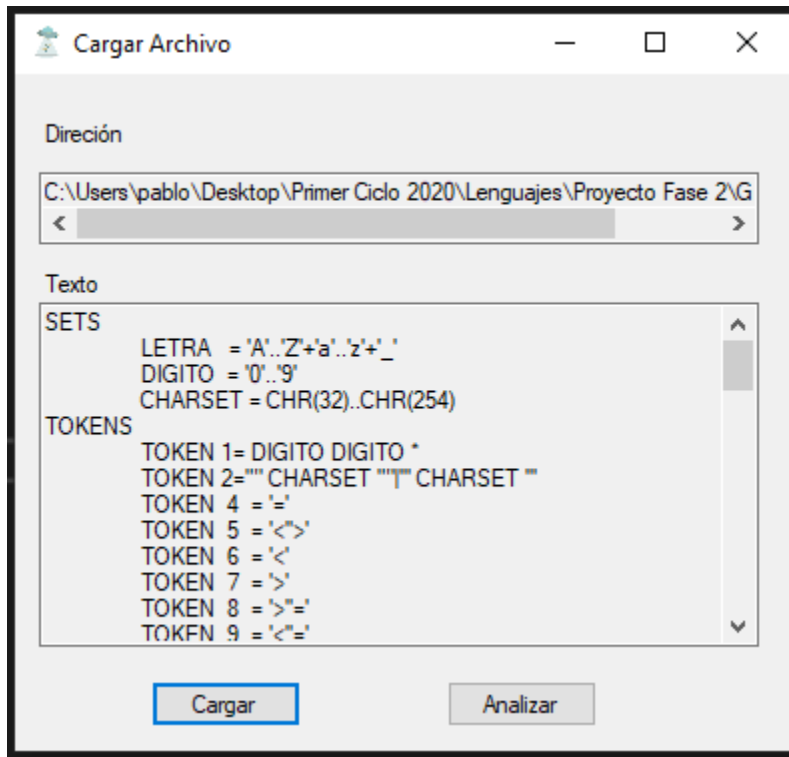
Manual de usuario

1. Para cargar el archivo de la gramática debe presionar cargar si el archivo es un txt lo aceptara sino mostrara un mensaje de error.

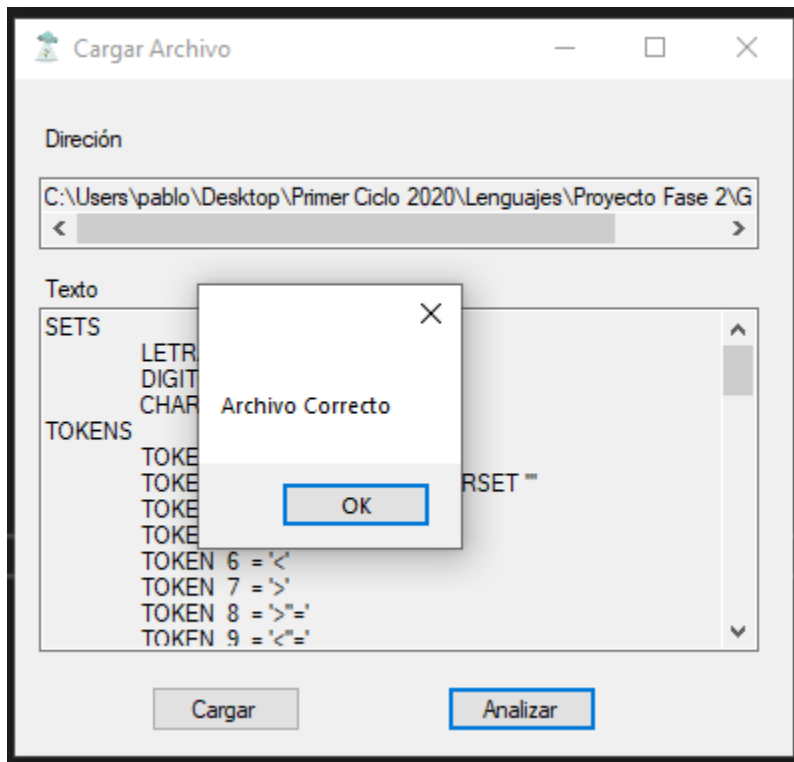


The image shows a software window titled "Cargar Archivo" (Load File). It has a standard Windows-style title bar with minimize, maximize, and close buttons. The window is divided into two main sections. The top section is labeled "Dirección" (Address) and contains a text input field with left and right arrow buttons. The bottom section is labeled "Texto" (Text) and contains a large, empty text area with a vertical scrollbar on the right side. At the bottom of the window, there are two buttons: "Cargar" (Load) and "Analizar" (Analyze). The "Cargar" button is highlighted with a blue border.

2. Ya cargado el archivo se podrá ver la dirección del archivo con el texto que tiene adentro que será la gramática para analizar. Para poder analizar la gramática se debe de presionar analizar.



3. Al presionar analizar el programa analizara la gramática si esta correcta mostrara un mensaje como el ejemplo siguiente sino es correcta mostrara un mensaje de en qué línea existe un error y que se esperaba que viniera.



4. Si la gramática es correcta mostrará otra ventana donde se podrá observar en la parte superior la expresión regular formada a partir de los tokens después se encontrará con el árbol formado por esa expresión regular y para finalizar en la parte de abajo con las tablas de first, last, follow y estados. Y para finalizar en la esquina inferior derecha se encontrará un botón que dice atrás para poder regresar a la ventana de carga para poder analizar otro archivo.

Tablas

Expresión Regular: DIGITO·DIGITO*|'|CHARSET·'|'|CHARSET·'|'|'<'>'|'|'<'>'|'|'>'>'|'|'<'>'|'|'+'|'|'-'|'|'0'-'R'|'|'A'-'N'-'D'|'|'M'-'O'-'D'|'|'D'-'T'-'V'|'|'N'-'O'-'T'|'|'('·'|'|'*)|'|'|'

Árbol:

First y Last:

Símbolo	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

Follows:

Símbolo	Follow
1	2,56.
2	2,56.
3	4.
4	5.
5	56.
6	7.
7	8.
8	56.

Estados:

Estados	DIGITO	'	CHARSET	'
1,3,6,9,10,12,13...	2,56	4	---	7
2,56	2,56	---	---	---
4	---	---	5	---
7	---	---	8	---
56	---	---	---	---
11,56,17	---	---	---	---
56,15	---	---	---	---
71	---	---	---	---

Atras