Universidad Nacional de Asunción
Facultad Politécnica

Ingeniería en Informática
Diseño de Compiladores

Generador de Autómatas
Compi-Win

San Lorenzo-Paraguay
Octubre-2013

Información de Contacto

Autores:

- ✓ Univ. Cristian Aceval
  E-mail: cristian.aceval@gmail.com.

- ✓ Univ. Víctor Franco
  E-mail: victorfranco90@gmail.com.

Tutor:

- ✓ Ing. Sergio Aranda.
  Email: sergio.aranda@consultora.com.py

Contenido

I. Abstract

El trabajo consiste en realizar un Analizador Léxico en base a una entrada que se encuentra definido por un Alfabeto y por una expresión regular. A partir del lenguaje de entrada, se busca construir las principales representaciones utilizadas para el Análisis Léxico, es decir, a partir de la entrada se generaran los Autómatas Finitos No Determinísticos, Autómatas Finitos Determinísticos, y además el Autómata Finito Determinístico con Minimización.

II. Descripción General

El sistema Compi-Win está realizado en el lenguaje java, con una interfaz de escritorio realizado con librerías swing. Además contamos con un software qué genera los grafos llamado GraphVitz que es necesario para la máquina que utilizará nuestro sistema.

El proyecto Compi-Win se encuentra divido en 4 paqueterías:

- ✓ Windows: contiene toda la interfaz GUI necesarias para la interacción del Software con el usuario.
- ✓ Algoritmos: en ella encontramos los algoritmos de Thompson, validación, subconjuntos, etc., que serán necesarios para la creación de los autómatas.
- ✓ Estructuras: referentes a la estructura de datos o modelos utilizados en el sistema.
- ✓ Análisis: en ella encontramos las validaciones del Alfabeto, Analizador Sintáctico, Analizador Léxico, Token y la Expresión Regular Correspondiente.

III. Interfaces Principales

El software cuenta con una ventana principal en la cual permite la selección de dos operaciones:

La primera opción permite al usuario ingresar de manera directa tanto el alfabeto como la expresión regular a ser utilizada.

La segunda opción ya cuenta con un alfabeto definido, el usuario del sistema deberá ingresar la expresión regular a ser evaluada.

A continuación presentamos las vistas mencionadas anteriormente:

Figura 1. Ventana Principal del Sistema Compi-Win.

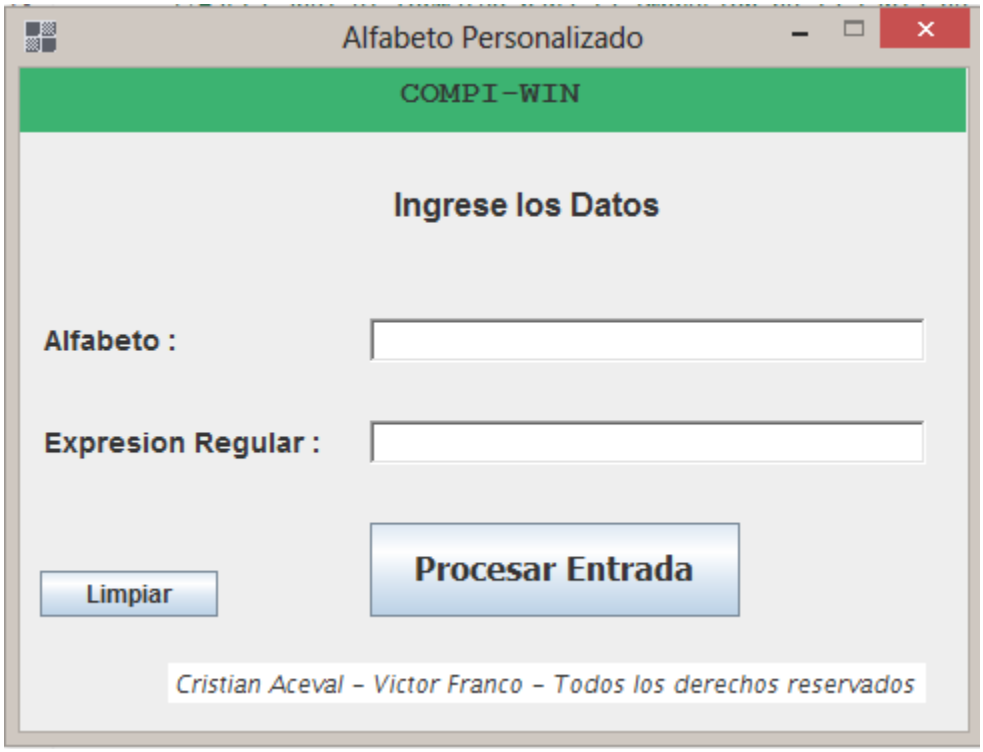


Figura 2. Ventana del Alfabeto Personalizado.

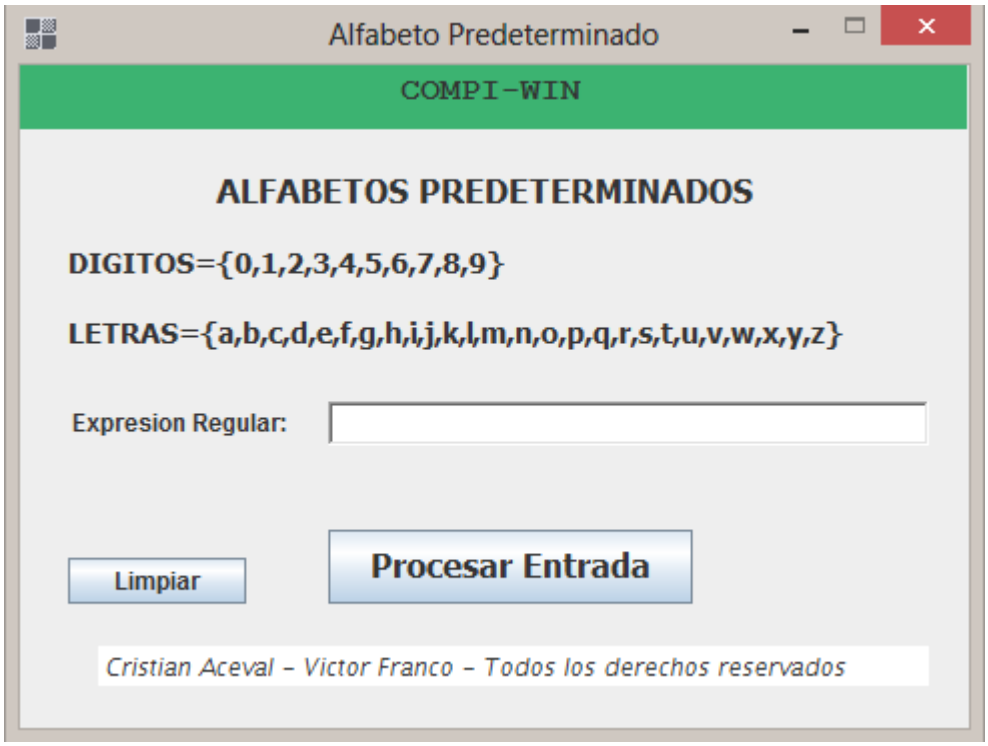


Figura 3. Ventana del Alfabeto Predeterminado.

La figura 2(dos) presenta una ventana que permite el ingreso tanto del Alfabeto a ser utilizado, así también se debe de agregar la expresión regular para que sea procesada.

La figura 3(tres) presenta una ventana que indica el alfabeto predeterminado que se encuentra de manera estática en el sistema sea este: ([a-z], [0-9]).

Al procesar la entrada ya se generan los autómatas, en caso de que el usuario haya ingresado valores inconsistentes el sistema emitirá un mensaje indicando dicho acontencimiento.

Una vez procesado los datos el sistema emitirá una ventana indicando todos los pasos realizados para llegar a la solución el cual se ven en la siguiente figura:

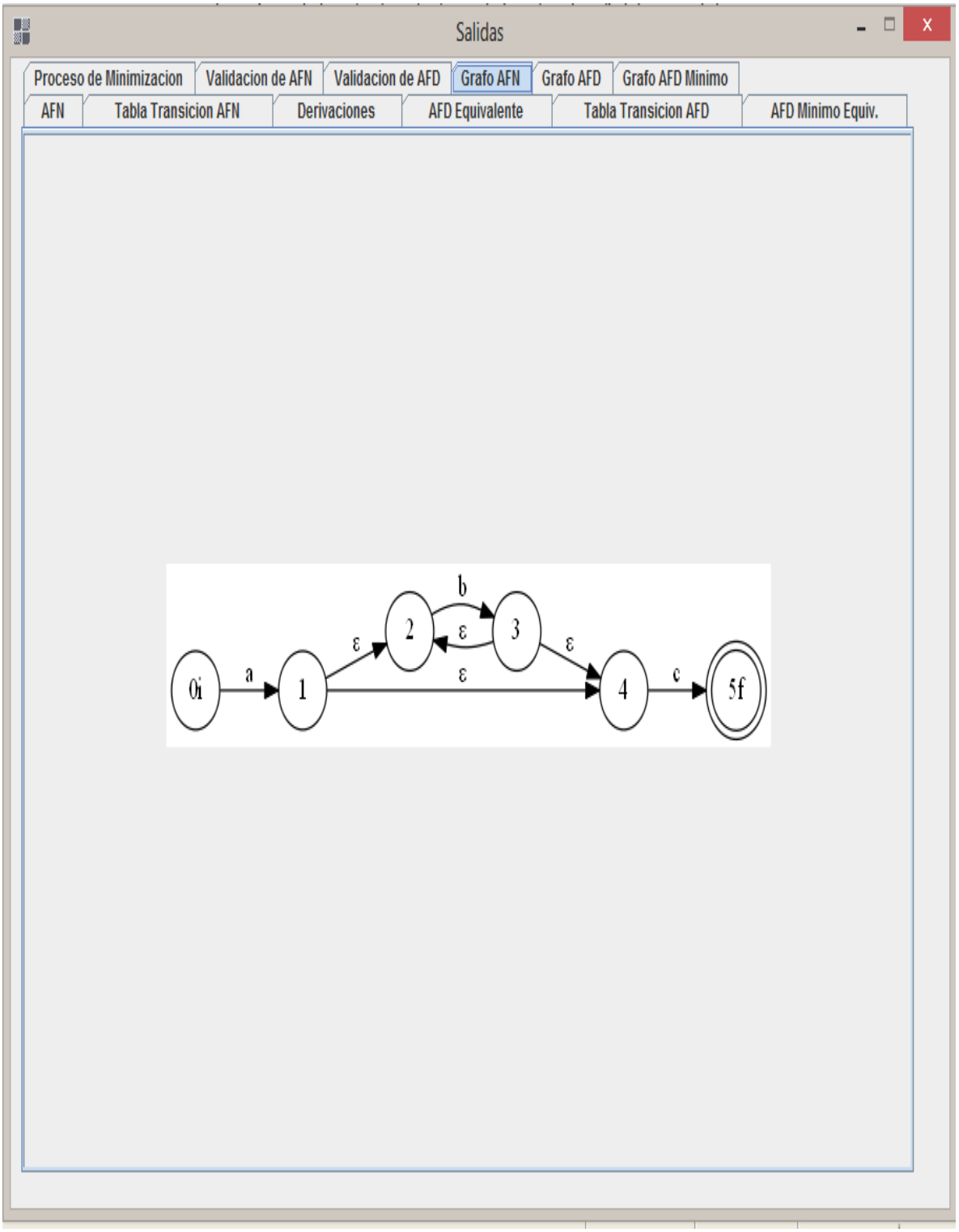


Figura 4. Ventana de Salidas del Sistema.

Esta ventana muestra todo lo referente al autómata que está representando. Entre sus pestañas se permite ver:

1. La tabla de transiciones del autómata para los diversos casos.
2. Dos opciones que pueden ser utilizadas para la validación de alguna cadena de entrada, sean estos AFD y AFN.
3. Descripción de los procesos realizados para obtener el correspondiente autómata
4. Los gráficos en forma de grafos dirigidos correspondiente al autómata a través de su tabla de transiciones por cada símbolo del alfabeto en cuestión, además como información importante destacamos que el estado inicial de un autómata tiene un identificador que consiste en la letra "i", de inicial. Cada estado final de un autómata tiene un identificador que consiste en la letra "f", de final

IV. Estructuras

Aquí se encuentran implementadas las estructuras de datos que representan a los Autómatas, tal como AFN, AFD y AFD mínimo, en función a sus estados y transiciones.

V. Análisis

Las expresiones regulares que fueron tomadas en cuenta para este proyecto soportan los siguientes operadores, además de los símbolos de un alfabeto dado:

- ✓ Unión (|)
- ✓ Cerradura positiva (+)
- ✓ Opción (?)
- ✓ Agrupaciones (a través de paréntesis)
- ✓ Concatenación (sin carácter representativo)
- ✓ Cerradura de Kleene (*)

Para validar las expresiones regulares de entrada y convertirlas a un AFN, se implementó un Traductor Dirigido por la Sintaxis, cuya gramática es la siguiente:

```
Terminales = {“|”, “)”, “(“, “*”, “+”, “?”, é, “símbolo del alfabeto”}
No Terminales = {ExprecionRegular, Concatenar, Grupo, Elemento, Operador, SimpleElemento}
Simbolo inicial = ExprecionRegular
Producciones =
            ExprecionRegular -> ExprecionRegular "|" Concatenar
            ExprecionRegular -> Concatenar
            Concatenar -> Concatenar Grupo
            Concatenar -> Grupo
            Grupo   -> Elemento    Operador
            Grupo   -> Elemento
            Elemento    -> "(" ExprecionRegular ")"
            Elemento    -> SimpleElemento
            Operador -> "*"
            Operador -> "+"
            Operador -> "?"
            Operador -> é
            SimpleElemento  -> simbolos_del_alfabeto
```

Luego de eliminar la recursividad por la izquierda y factorizar, la gramática resultante es la siguiente:

```
            ExprecionRegular -> Concatenar R1
            R1      -> "|" Concatenar R1 | €
            Concatenar   -> Grupo R2
            R2      -> Grupo R2 | €
            Grupo   -> Elemento    Operador
            Grupo   -> Elemento
            Elemento    -> "(" ExprecionRegular ")"
            Elemento    -> SimpleElemento
            Operador -> "*"
            Operador -> "+"
            Operador -> "?"
```

Operador -> €

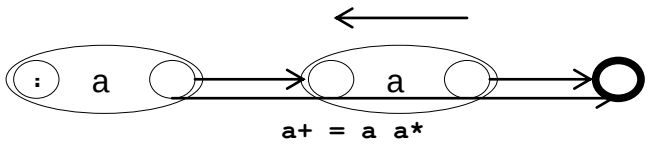
SimpleElemento -> simbolos_del_alfabeto

VI. Algoritmos

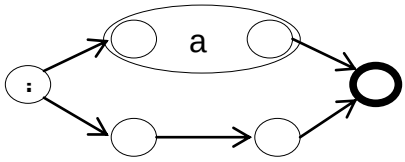
En este módulo se agregan los algoritmos para generar las distintos Autómatas. Los algoritmos son: Thompson, Subconjunto y Minimización.

Estos algoritmos se basan en los algoritmos implementados en el libro de Compiladores de "Aho".

Para los operadores de cerradura positiva (+) y de opción (?), en las expresiones regulares, las construcciones de Thompson utilizadas corresponden a las obtenidas de las siguientes relaciones:



a+ = a a*



- a? = a | ε

Para la Minimización se realiza primero la eliminación de nodos inalcanzables, luego se aplica el algoritmo de minimización y finalmente, se eliminan los nodos identidad.

VII. Consideraciones Especiales

Principales características implementadas:

- Se imprime una descripción de los procesos realizados para obtener cada autómata.
- En la tabla de transición del AFD se muestran los estados del AFN agrupados en cada uno de los estados del AFD.
- En la tabla de transición del AFD mínimo se muestran las etiquetas de los estado de tal manera que pueda indicarse cuales estados fueron agrupados.
- El código se encuentra documentado para su posterior entendimiento.
- Visualmente se pueden apreciar los grafos Generados por el Sistema permitiendo así una mejor apreciación del grafo.
- Se ha creado un repositorio en github.com de manera a contemplar los cambios realizados a lo largo de la elaboración del proyecto.(

https://github.com/cristoka7/compiladores).

El cual permitirá acceder a los datos desde cualquier parte del mundo.