



Universidad Nacional Autónoma de México
Licenciatura en Ciencias de la Computación, Facultad de Ciencias
Autómatas y Lenguajes Formales 2021-1

Noé S. Hernández
Alan E. Arteaga
Carlos Naranjo

Práctica 1

Fecha de entrega: Domingo 15 de noviembre.

1 Objetivo

El alumno aprenderá a utilizar el programa JFLAP para reforzar su conocimiento de los lenguajes regulares.

2 Introducción

JFLAP (Java Formal Languages and Automata Package) es un software que se describe como:

"un paquete de herramientas gráficas que puede ser usado como un apoyo en el aprendizaje de conceptos básicos de la teoría de Autómatas y Lenguajes Formales"

JFLAP nos permite la simulación de elementos tales como autómatas finitos deterministas (AFD), autómatas finitos no deterministas (AFN), expresiones regulares, gramáticas, etc.

JFLAP es desarrollado y mantenido por la Universidad de Duke desde 1993. Es importante señalar que su desarrollo es en el lenguaje de programación Java, y que este software corre como una aplicación Java.

Para descargar este software para su uso, puede recurrirse al enlace:

<http://www.jflap.org/jflaptmp/>

Una vez en el sitio, se debe descargar el archivo **JFLAP7.1.jar** que se muestra en la siguiente imagen (archivo señalado en el recuadro azul):

Back to JFLAP web page

JFLAP SOFTWARE

NOTE: These are .jar files. If your operating system saves them as .zip files, rename them to .jar files. Then you should be able to click on them to run them.

NOTE 2: If you have trouble with clicking on the .jar file, try [jarfix](#)

NOTE 3: Most people will just want the software, if you want the source, scroll down to the bottom. NOTE 4: JFLAP Beta Version 8 was never finished. Recommend you use JFLAP Version 7.1.

To get JFLAP SOFTWARE (jar file):

- JFLAP Software - will put newest version here as updates occur

This is a .jar file.

- Jul 27, 2018 - JFLAP Version 7.1
 - Note: JFLAP now uses Java 8.0.
 - [Changes made to JFLAP 7.0](#)
 - **JFLAP7.1.jar**
- Jan 24, 2015 - JFLAP Version 8.0 BETA
 - WARNING: This version is not complete and not as stable as Version 7. It is being released so people can use the new features. Strongly recommend you use the July 27, 2018 Version 7.1 instead if you are using JFLAP for a course.
 - [New items in JFLAP 8.0 BETA](#)

Para poder ejecutar JFLAP se debe de contar con una versión del JDK (Java Development Kit) mayor o igual a la versión 8.0 Existen varias implementaciones del lenguaje Java, las dos más populares y fáciles de descargar resultan ser: [Open JDK](#) y [Oracle Java](#).

Una vez con Java instalado y con las variables de entorno configuradas correctamente para llamarlo desde la terminal, únicamente es necesario ejecutar el paquete descargado con el comando -jar en java:

```
$ java -jar [Carpeta del paquete]/JFLAP7.1.jar
```

Como referencia adicional, pueden consultar más acerca de la documentación e información útil para el uso adecuado de JFLAP en la página oficial:

<http://www.jflap.org/tutorial/>

3 Desarrollo

3.1 Autómatas finitos

Para cada uno de los tres incisos en esta subsección, envíe un archivo de JFLAP con el autómata solicitado. Además, proporcione como un archivo de texto con 14 cadenas, las primeras 7 de aceptación y las segundas 7 de rechazo.

Compruebe con tales archivos el comportamiento del autómata diseñado con la función **Input/Multiple Run** de JFLAP.

1. (2.5 pts.) Construya un AFD que reconozca aquellas cadenas en el alfabeto $\{1, 2, 3\}$ tales que la suma de los dígitos en la cadena sea divisible por 4.

Por ejemplo, si M es el autómata que pide el ejercicio:

- $23 \notin L(M)$ ya que $2 + 3 = 5$
- $33123 \in L(M)$ ya que $3 + 3 + 1 + 2 + 3 = 12$
- $21 \notin L(M)$ ya que $2 + 1 = 3$
- $22 \in L(M)$ ya que $2 + 2 = 4$

2. (2 pts.) Las cadenas sobre $\Sigma = \{0, 1, 2\}^*$ tales que tienen al menos dos ceros, a lo más un 1 y ningún 2.

Por ejemplo, si M es el autómata que pide el ejercicio:

- $01 \notin L(M)$ pues tiene a lo más un 1, no tiene 2, pero tiene sólo un 0.
- $200000000 \notin L(M)$ ya que tiene al menos dos 0, menos que un 1 pero contiene un 2
- $001000 \in L(M)$
- $00100001 \notin L(M)$ pues repite un 1.

3.2 Expresiones regulares

Para las expresiones regulares de los siguientes incisos, realice los pasos que se solicitan de forma escrita en un editor de texto o en una hoja, y luego verifique que el resultado obtenido es correcto utilizando JFLAP

Es necesario guardar el resultado de verificar con JFLAP cada paso en un archivo por separado, además de adjuntar también la solución escrita (puede ser mediante un archivo PDF generado por L^AT_EX o cualquier otro software para generar documentos o también mediante un archivo escaneado).

- a) Convierta la expresión regular a un $AFN - \varepsilon$.
- b) Convierta el $AFN - \varepsilon$ a un AFN .

- c) Convierta el *AFN* a un *AFD*.
 - d) Minimice el autómata resultante si es posible hacerlo.
1. (2.5 pts.) $(ab)^*(ba + aba)$
 2. (2 pts.) $(011 + 10)^*00$
 3. (2.5 pts.) $(x + y)^*xy(x + y)^*$

Tip:

Como apoyo para realizar diagramas para Autómatas finitos, pueden usar la siguiente herramienta, la cual exporta también código L^AT_EX en caso de que decidan enviar archivos generados por este medio.

<http://madebyevan.com/fsm/>