



## Práctica 1

Fecha de entrega: Domingo 19 de diciembre a las 23:59:59 hrs.

### 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"*

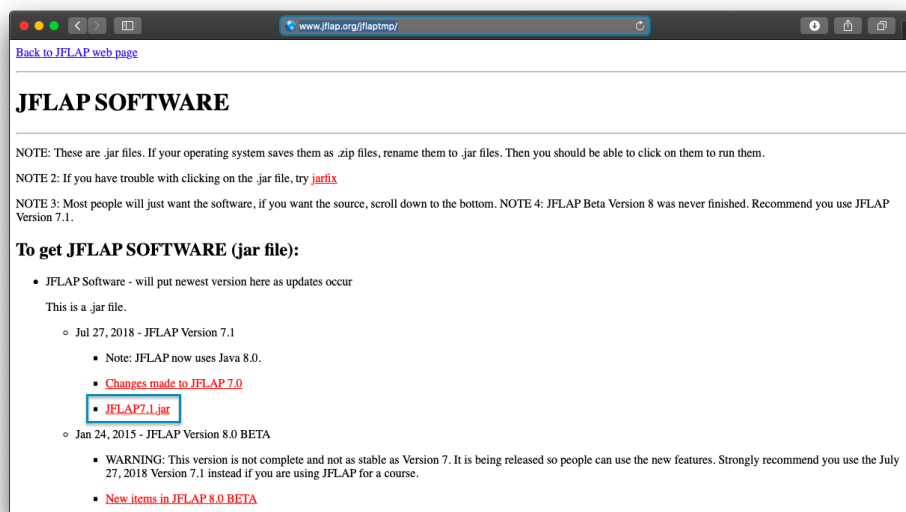
JFLAP nos permite la simulación de elementos tales como autómatas finitos deterministas (AFD), autómatas finitos no deterministas (AFN), autómatas finitos no deterministas con movimientos  $\varepsilon$  (AFN-  $\varepsilon$ ), expresiones regulares, gramáticas, etc.

JFLAP es desarrollado 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:

<https://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):



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 instalado Java 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:

<https://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 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  $\{0,1\}$  (binario) tales que las cadenas binarias representen un número decimal múltiplo de 3.

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

- $0 \in L(M)$  ya que 0 binario es 0 decimal, y 0 es múltiplo de 3.
- $101 \notin L(M)$  ya que 101 binario es 5 decimal, y 5 no es múltiplo de 3.
- $0110 \in L(M)$  ya que 0110 binario es 6 decimal, y 6 es múltiplo de 3 (nótese que 6 puede representarse también como 110 o con cualquier cantidad de ceros a la izquierda).

2. (2 pts.) Construya un AFD que reconozca aquellas cadenas en el alfabeto  $\{a,b\}$  tales que tengan un número par de  $b$  consecutivas.

Por ejemplo, si  $N$  es el autómata que pide el ejercicio:

- $aaa \in L(N)$  pues no tiene ninguna  $b$  consecutiva.
- $aaaaba \notin L(N)$  ya que tiene un número impar de  $b$  consecutivas (aunque sea una única  $b$ ).
- $aaaabbba \notin L(N)$  por el motivo anterior.
- $bb \in L(N)$ .
- $abbbbaabbbaabb \in L(N)$ .
- $bbabbb \notin L(N)$  pues aunque tiene al principio un número par de  $b$  consecutivas, no lo tiene al final.

### 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, tenga especial cuidado con la precedencia de operadores.

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<sup>A</sup>T<sub>E</sub>X 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.)  $(bb + aba)(ba)^*$
  2. (2 pts.)  $1011^* + 00$
  3. (2.5 pts.)  $(x + y)^*x + yy$

**Tip:**

Como apoyo para realizar diagramas para Autómatas finitos, pueden usar la siguiente herramienta, la cual exporta también código  $\text{\LaTeX}$  en caso de que decidan enviar archivos generados por este medio.

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