

# MANUAL JFLAP

## Autómatas Finitos – Parte 1

### (Java Formal Language and Automata Package)

#### 1. Introducción a JFLAP

JFLAP (Java Formal Language and Automata Package) es una herramienta para la enseñanza y la visualización interactiva de lenguajes formales. Permite crear y operar sobre autómatas (autómatas finitos, máquinas de Moore y Mealy, máquinas de Turing...), gramáticas, expresiones regulares y L-systems.

En esta práctica inicial sólo nos vamos a centrar en la parte enfocada a los autómatas finitos, y según avance el curso, iremos profundizando en los distintos apartados de la aplicación. Aunque algunas secciones no se verán en la asignatura.

#### 2. Iniciar JFLAP

La herramienta se puede descargar gratis de la url: [www.jflap.org](http://www.jflap.org) (rellenando un formulario previamente), o bien os la podéis bajar de aula global.

JFLAP está implementado en Java, por lo que es necesario tener la versión Java SE 1.6 o posterior instalada en vuestro ordenador. En el caso de las aulas informáticas de la clase de prácticas, ya estará instalado.

JFLAP se distribuye como un ejecutable *.jar*. Para ejecutar este tipo de ficheros en un sistema operativo Windows, basta con hacer doble clic en *JFLAP.jar*. Si descargáis la aplicación desde *jflap.org* deberéis cambiar la extensión del fichero (de *.zip* a *.jar*). En el caso de Linux, desde la consola, con el fichero *JFLAP.jar* en el directorio actual, tendréis que introducir el comando *java -jar JFLAP.jar* (este comando funciona de la misma forma en la consola de Windows). Una vez abierta la aplicación, aparecerá en pantalla una interfaz como la que se muestra en la Figura 1.



Figura 1: Ventana inicial de JFLAP

## 2. Autómatas Finitos

En este capítulo construiremos un autómata determinista en JFLAP, ilustrando algunos métodos de simulación, etc... En las secciones 2.1-2.4, mostraremos la definición estándar de AFD, y en la 2.5 veremos como JFLAP maneja una definición más general de un AFD con múltiples transiciones de caracteres.

### 2.1 Un autómata finito simple

Para empezar a construir un AF, pincha en el menú de inicio de JFLAP en el botón **Finite Automaton**. Aparecerá una ventana, con un menú, una pestaña que es el Editor, una barra de herramientas y un área en blanco que ocupará casi toda la ventana.

Como ejemplo, realizaremos el siguiente AF:

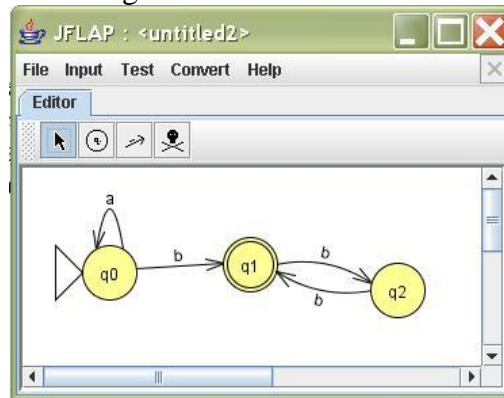




Figura 2. Ejemplo: AFD\_1

#### 2.1.1 Crear estados

Un autómata está formado por un conjunto de estados. Antes de crear uno, primero debes activar el botón , en la barra de herramientas. Mientras esté seleccionado, el botón estará sombreado.


La zona grande en blanco, llamada *canvas*, es donde construiremos el autómata. Ahora que la herramienta para crear estados está activada, pincha en el canvas para crear un estado. Aparecerá en la zona donde hayas pinchado. JFLAP los identificará en el orden en el que se creen como q0, q1, q2...

#### 2.1.2 Definir el estado inicial y el final

Todos los autómatas requieren un estado inicial y un conjunto de estados finales. Para marcar q0 como el estado inicial, selecciona el botón  y después, con el botón derecho del ratón pincha en el estado q0. Aparecerá un menú sobre el estado del autómata (el resto de las opciones del menú se explicarán en la sección 7.1.5), en el que escogeremos **Initial**. Ahora el estado q0 queda marcado con una flecha que indica que es el inicial. Igualmente, realiza el mismo procedimiento con q3 pero seleccionando **Final** en el menú. El estado q3 quedará con un doble círculo.

Para modificar los estados iniciales/finales, se realiza el mismo procedimiento, volviendo a escoger **Initial/Final** en el menú, para eliminar la marca de *check* que habíamos puesto antes.


### 2.1.3 Creando transiciones


En el ejemplo, hay 3 transiciones. Para crearlas, pincha en el botón . Una vez seleccionado, pincha en el estado  $q_0$ , y sin soltar el botón, arrastra el cursor del ratón hacia el estado  $q_1$  y después, suéltalo. Un campo de texto aparecerá entre los dos estados. Escribe una 'b' y dale a *intro*. Una nueva transición b del estado  $q_0$  a  $q_1$  apareció. De la misma forma, crea el resto de transiciones para que el autómata quede igual que el de la Figura 21. Para transiciones  $\lambda$ , deja el campo de texto en blanco.

Para los *loops*, simplemente pincha sobre el estado en el que quieras hacer el *loop* y suelta el ratón. Aparecerá la ventana donde deberás escribir el texto.


Si quieres cancelar en algún momento una transición, o una edición, pulsa escape.

### 2.1.4 Borrando estados y transiciones

Si te confundes y creas estados o transiciones de más, borrarlos es muy sencillo. Selecciona el botón , y haz doble clic sobre el elemento que quieras eliminar. En el caso de las transiciones puedes pinchar en la misma flecha o en la etiqueta. En los estados, cuando se borra uno, se eliminarán automáticamente todas las transiciones que entren o salgan del estado.

Para no eliminar algo por error, selecciona el botón  de nuevo, y así ya no estará activada la herramienta de borrado.

### 2.1.5 Herramienta de editor

En el punto 2.1.2 ya se ha explicado alguna función para el botón , pero tiene otras muchas funciones relacionadas a la modificación de los atributos de los estados existentes y de las transiciones:

- Marcar estados como inicial/final: esto ya lo hemos visto en la sección 2.1.2
- Mover estados y transiciones: pincha sobre el elemento que desees mover y arrástralo a la nueva localización. Si mueves una transición, moverás también los estados implicados.
- Editar transiciones existentes: haz clic sobre la transición que quieras modificar, y cambia lo que había antes en el campo de texto.
- Etiquetas: Si pinchas con el botón derecho del ratón sobre un estado, veras en el menú **Change Label**. Al seleccionar esa opción, aparecerá una nueva ventana donde podrás escribir la nueva etiqueta. Estas etiquetas ayudan a identificar el significado del estado.

En el ejemplo de la Figura 2, cuando estemos en  $q_2$ , será cuando llegue al autómata un número par de b's. En  $q_1$  tendremos un número impar. Por ello, podríamos etiquetar  $q_2$  con "*nº par de b's*" y  $q_1$  con "*nº impar de b's*".

Para borrar una etiqueta existente, haz clic en **Clear Label**, y para borrar todas las que haya, pincha en **Clear All Labels**.

Si pinchas con el botón derecho, en una zona en blanco, aparecerá un menú diferente con la opción **Display State Labels**. Esta opción está activada por defecto. Si la desactivas, las etiquetas serán invisibles. Las podrás ver, posicionando el cursor del ratón durante un par de segundos sobre un estado en el que hubieras añadido previamente una etiqueta.

- Disposición automática: pulsa el botón derecho en una zona en blanco. Observa que hay un elemento en el menú, que pone **Layout Graph**. Cuando está seleccionado, JFLAP aplicará un algoritmo de ordenación gráfico sobre el autómata. Esta opción es útil sobre todo, cuando hay un gran número de

estados y transiciones, ya que JFLAP los colocará para que el autómata se vea de la forma más sencilla posible, ahorrándonos el proceso, a veces tedioso, de ir moviendo cada elemento por separado.

Nota: Existen comandos para ejecutar algunas de estas opciones más rápido. Por ejemplo, manteniendo el cursor del ratón sobre el icono de la herramienta de creación de estados, aparecerá **(S)tate Creator**. El paréntesis que encierra a la **S** indica que esa es la tecla rápida para crear estados. Pulsando la letra correspondiente (en minúsculas), activaremos la opción deseada sin necesidad de usar el ratón.

## 2.2 Simulación de una entrada en el autómata

En esta sección estudiaremos tres de los métodos de JFLAP para simular el resultado de una entrada en un autómata: simulación por pasos, simulación rápida y simulación múltiple. El cuarto tipo “stepping by state” se verá brevemente en 7.3.

### 2.2.1 Simulación por pasos (Stepping simulation or Stepping with closure)

La simulación por pasos te permite ver cada configuración generada por un autómata al procesar una entrada. La siguiente figura muestra un ejemplo de una simulación por pasos de la entrada *aabbbb* en el autómata de la Figura 21. Para seleccionar esta herramienta, ve a **Input > Stepping with closure**. Te aparecerá una ventana, donde tendrás que introducir la entrada del autómata.

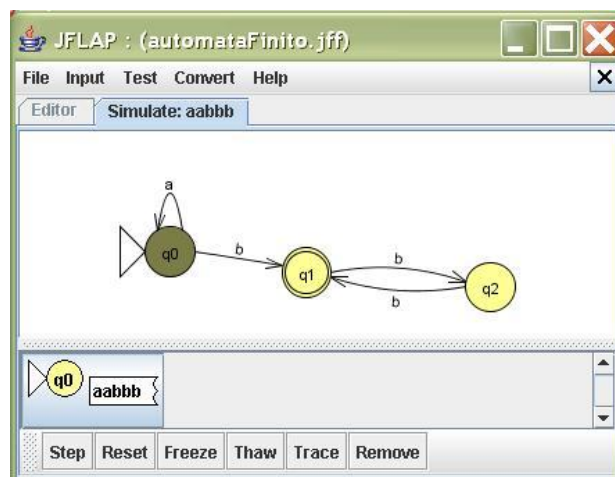


Figura 3: Inicio de la simulación por pasos con la entrada *aabbb*.

En la Figura 3 vemos como el estado activo del AF está sombreado. La zona por debajo del dibujo del AF muestra la configuración actual. Comenzamos en el estado inicial *q0*, y al pulsar **Step** las letras de la palabra de entrada se irán sombreado, a la vez que cambiará el estado actual, según sea la entrada recibida en cada caso. Si la palabra es aceptada, el dibujo con la configuración actual tendrá un fondo verde. Si no es aceptada por el AF, el fondo será rojo.

Algunas de las operaciones de la barra de herramientas que se encuentra en la parte baja de la ventana (Figura 3) solo funcionan cuando hay una configuración seleccionada. Para seleccionarla, pincha en ella. Puedes ver como al hacerlo, el color se oscurece un poco. Si pinchas ahora en **Remove**, eliminarás la configuración actual. Para empezar de nuevo con la simulación, pincha en **Reset**.

Para ver todos los pasos anteriores que se han dado para llegar a una configuración, selecciónala y después pincha en **Trace**. Se abrirá una nueva ventana con todo el proceso desde la configuración inicial, arriba del todo, a la actual.

### 2.2.2 Simulación rápida (Fast Simulation)

Esta simulación permite ver rápidamente si el autómata acepta o no una entrada. Si la acepta, aparecerá la lista con los pasos dados hasta llegar al final (similar a obtener la traza en el punto anterior). En caso contrario, aparecerá un mensaje para informar.

Selecciona **Input > Fast Run**. Escribe *aabbb* como entrada al autómata y el resultado que te dará JFLAP será como el de la Figura 4.

Observa los dos botones de la ventana. **I'm Done** cerrará la ventana. **Keep Looking** será útil para autómatas no deterministas y eso lo veremos en la sección 2.3.2.

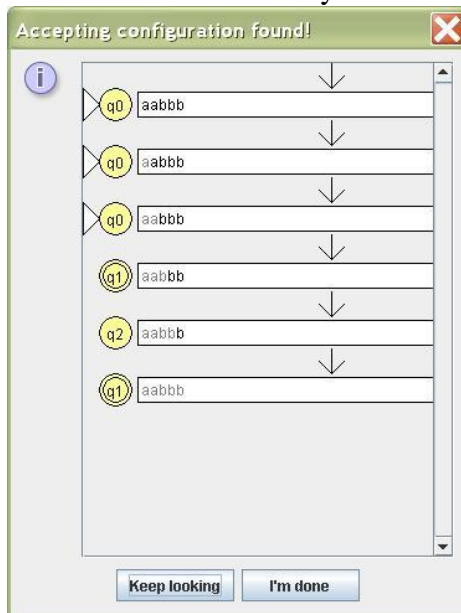


Figura 4: Resultado de una simulación rápida.

### 2.2.3 Simulación múltiple

Este método permite realizar funcionamientos múltiples de una vez, rápidamente. Selecciona **Input > Multiple Run**. Aparecerá una ventana similar a lo que se muestra en la Figura 5.

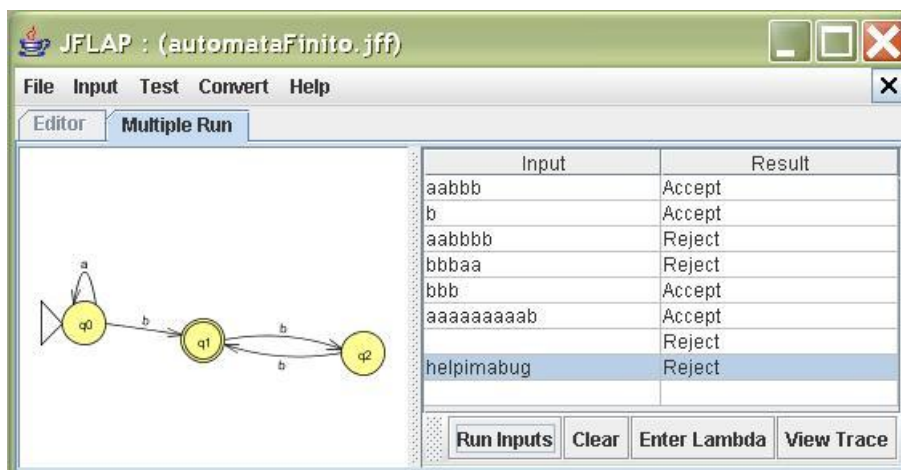


Figura 5:

*Ejemplo de simulación múltiple. En la fila 7 de inputs se introdujo lambda.*

El procedimiento es similar al del derivador múltiple por fuerza bruta que utilizamos en gramáticas. Escribe las entradas que quieras probar en la columna de inputs, como se muestra en el ejemplo, y pulsa en **Run Inputs**. La columna de la derecha mostrará los resultados obtenidos.

La lista de inputs que introduzcas, JFLAP los recordará la próxima vez que abras la simulación múltiple, mientras no reinicies el programa.