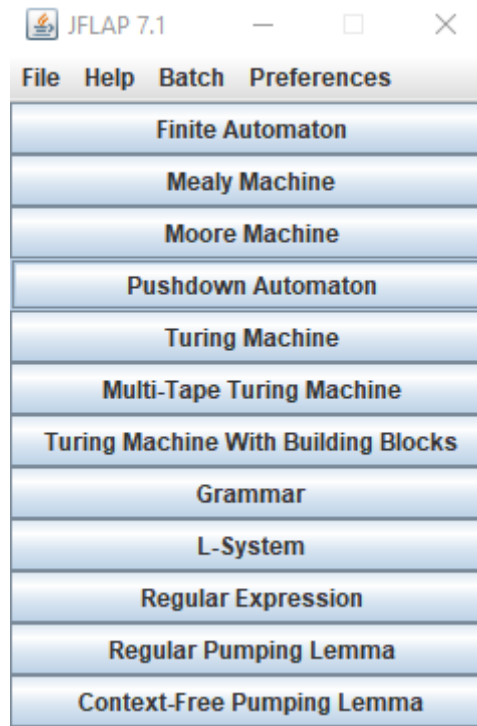


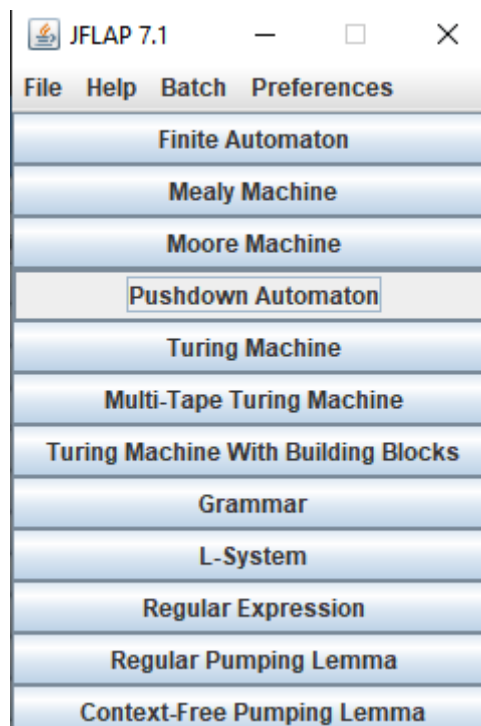
## Problema 5. Implementar y ejecutar un autómata con pila para un lenguaje libre del contexto.

Pasos para crear el autómata con pila en **jflap**:

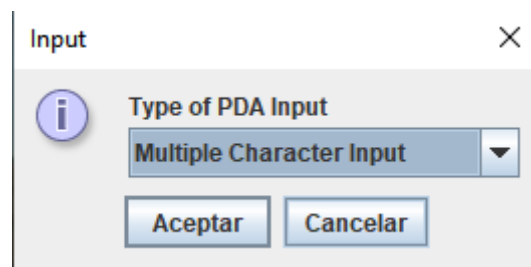
1) Abrimos **jflap**



2) Elegimos *Pushdown Automaton*

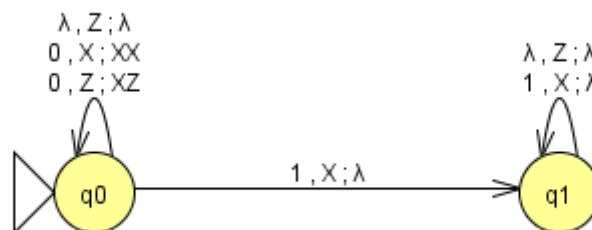


- 3) Seleccionamos la opción *Multiple Character Input* porque vamos a meter más de un símbolo en la pila a la vez.



El lenguaje que queremos reconocer son cadenas formadas por una secuencia de 0 seguida de otra secuencia de 1, ambas con el mismo número de caracteres.

El autómata que tendríamos que crear en **jflap** sería el siguiente:

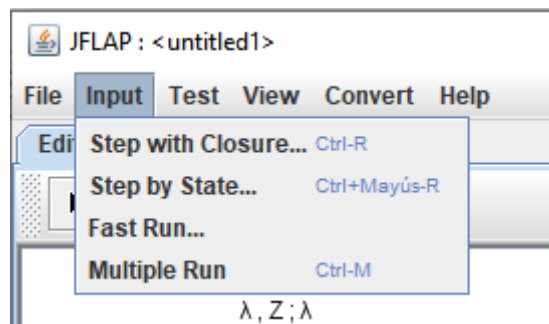


Como sabemos, al ser un autómata finito tiene un número de estados finito: q0 y q1. Además, sabemos que es un autómata no determinístico, que por lo general no pueden ser pasados a determinísticos y esto lo hace ineficiente. La interpretación de cada línea es:

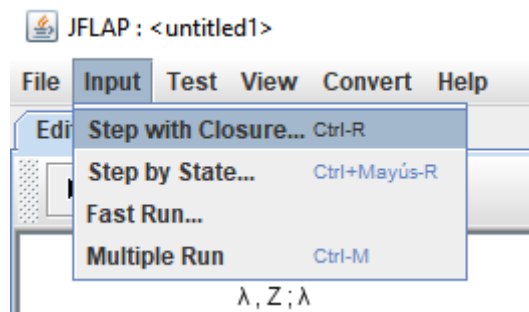
- $\lambda, Z, \lambda$ : Si está en  $q0$  sin leer nada de la cinta de entrada, sacando Z de la pila, que no meta nada en la pila. Es una transición nula porque no lee nada de entrada y no mete nada en la pila. Esta transición, a priori, podría no tener sentido en  $q0$ , pero está puesta para que si metemos la cadena vacía (que por defecto ya tendrá Z en pila como símbolo de pila vacía) nos diga que es válida también, pues sacará la Z y no nos quedará nada dentro de la pila.
- $0, Z, XZ$ : Leer 0 de la cinta de entrada, sacar Z de la pila y meter XZ, quedando esto último en el tope.
- $0, X, XX$ : Leer 0 de la cinta de entrada, sacar X de la pila y meter XX, quedando esto último en el tope.

Vamos a insertar cadenas para ver si nuestro autómata con pila las acepta o rechaza correctamente. Para ello:

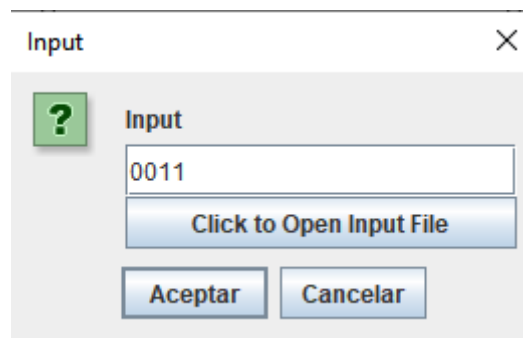
- 1) Seleccionamos la opción *input*



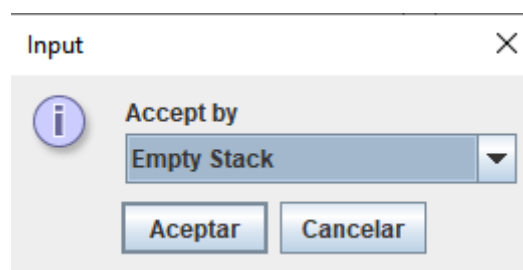
- 2) *Step by closure*

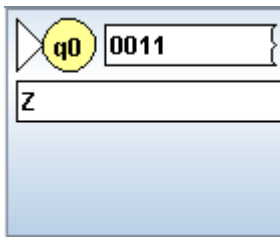


- 3) Insertamos la cadena que queremos comprobar



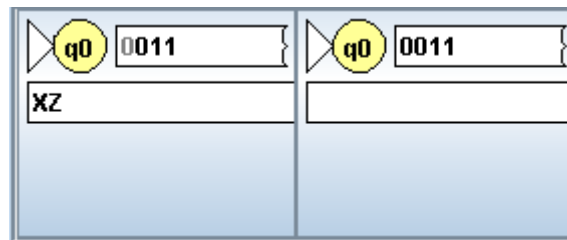
- 4) Aceptar
- 5) *Accept by Empty Stack* (debido a que en este autómata no hay estados finales)





Estado inicial de la pila: tiene un símbolo de tope de pila: Z (no está vacía). Como la pila tiene una estructura LIFO, el primero símbolo que se mete es el último que sale y, además, estas salen por el tope de la pila, que es la parte más a la izquierda.

El funcionamiento consiste en sacar un símbolo de la pila comprobar si tiene transición para ese carácter y, de tenerla, la realiza. Le damos a *Start* y vemos que se abren dos caminos:

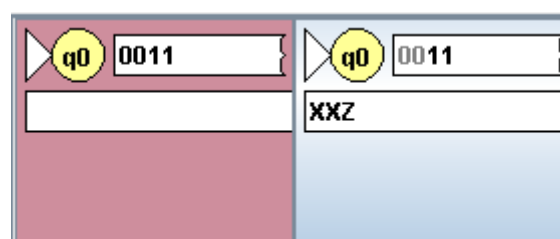


Nos interesa el camino de la izquierda que es aquel en el que ha leído un 0 de entrada, ha sacado la Z de la pila y ha metido en su lugar XZ. Como el objetivo es comprobar que tiene el mismo número de 0 que de 1, lo que hará es que por cada 0 que lea meterá una X en la pila. De esta forma, vemos que en *q0* va a leer la parte de los 0. En *q1* tenemos las siguientes líneas:

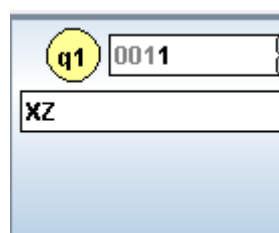
- $\lambda, Z, \lambda$ : Si no lee nada por pantalla saca una Z de la pila y no mete nada.
- 1, X,  $\lambda$ : Si lee un 1 saca una X de la pila y no mete nada.

De esto podemos concluir que en el lado de *q1* comprobaremos la correcta cantidad de 1 que tiene la cadena, pues el objetivo es sacar tantas X como se haya metido en *q0* (se mete una X en pila por cada 0 leído).

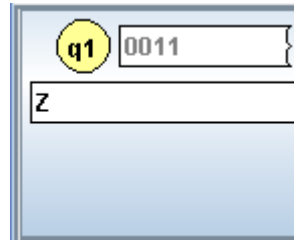
Le damos a *Step* para continuar y obtenemos lo siguiente:



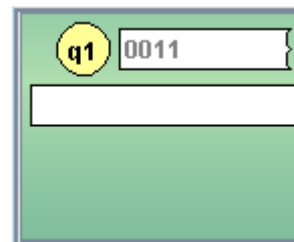
Pues al leer otro 0, ha sacado una X de la pila, pero ha metido dos. De nuevo a *Step* y ya leerá un 1, lo que hará que pase a *q1* y saque una X (que se ha metido al leer un 0) de la pila.



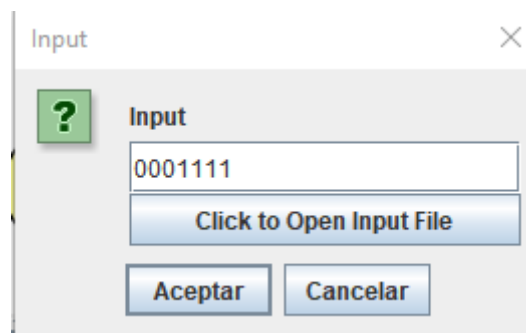
Vuelve a leer un 1 y volverá a sacar otra X:



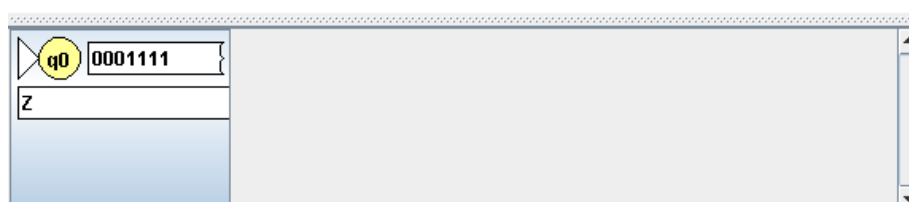
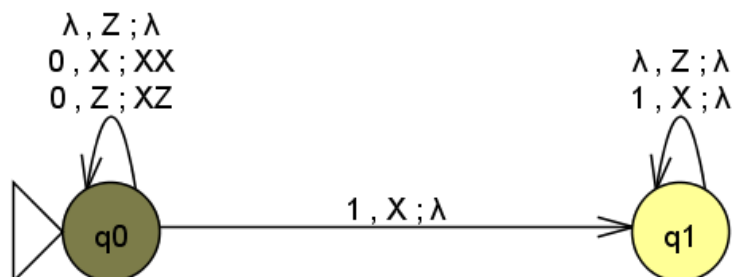
De terminar el proceso aquí, rechazaría cadena porque aún quedan cosas en la pila (Z), pero en  $q1$  tenemos la transición  $\lambda, Z, \lambda$  que sacará la Z y no meterá nada. La cadena verde indicará la aceptación de la cadena.



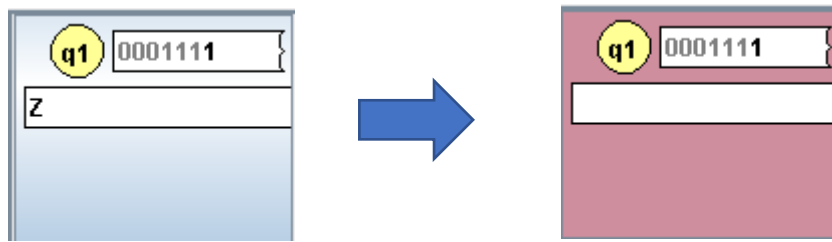
Ahora probaremos con otro ejemplo que sepamos que no es correcto para ver que la comprobación es correcta. Le damos la cadena [0001111]:



Inicialmente se verá así:



Y permitirá dar varios pasos sin errores, pues hasta el último carácter la cadena es correcta (ya que los 6 primeros dígitos son 0001111). Pero al llegar a este momento, esto es lo que ocurre:



Pues al leer un 1, intenta sacar una X de pila que no existe. Esto es una parada abrupta por error debido a que no tengo transición para el caso en el que leo un 1 sin que haya X, ignorando el 1 e interpretando que no lee nada. De esta forma, se activa la transición en la que no lee nada y saca una Z de la pila, quedando esta vacía y dando el mismo error de antes al no poder leer el 1. Tras no saber qué hacer, termina rechazando.