

Grupo Miércoles 17:00-19:00 semanas A

– Práctica 2 –

Autor: Javier Martínez Virtó

NIP:930853

Autor: Víctor Marteles Martínez

NIP:928927

Ejercicio 1:

1. Resumen

Este ejercicio se puede dividir en 3 problemas, para los cuales hemos empleado estos estados:

```
%x COMENTBARRAS  
%x COMENTASTERISK  
%x UPS  
%x PARAM_1  
%x PARAM_2
```

Comentarios con barras

Si hay “//” se entra en el estado (condición de arranque) “COMENTBARRAS”.

Si se detecta una cadena de letras y números que sea igual a “ups” en alguna parte del comentario empleamos una variable llamada “ApareceUps”, la cual se pone a 1.

Al llegar a un salto de línea (“\n”), se imprime un “NO ACTUALIZADO” en ese lugar y se va a “INITIAL”.

Comentarios con asteriscos

Si hay “/*” se entra en el estado “COMENTASTERISK”.

Si se detecta “ups” en alguna parte del comentario “ApareceUps” se pone a 1.

Al llegar a “*/”, se imprime un “NO ACTUALIZADO” antes de “*/” y se va a “INITIAL”.

Llamadas ups(...)

Si se encuentra un “ups” que no se encuentre dentro de un comentario, se pasará al estado “UPS”.

Cuando se encuentre el primer “(“ se entrará en “PARAM_1” (solo valdrá si hay cero o más espacios entre “ups” y “(“, no valdrá con tabuladores (en tal caso, se va a “INITIAL”)).

Se captura el primer parámetro y lo guarda en p1. Consume todas las comas que haya antes del primer parámetro antes de entrar en “PARAM_1”.

(si era solo “ups()”, se salta a “INITIAL”).

Si aparece una “,”, pasa a “PARAM_2”, consumiendo todas las comas. (si aparece “)” imprime el primer parámetro y cierra la llamada (caso “ups(p1)” sin segundo argumento)).

Se captura el segundo parámetro y lo guarda en p2. Cuando llega a ")", intercambia los parámetros y salta al estado "INITIAL".

Si aparecen más comas las consume, y si después de las comas aparecen más parámetros, se imprimen los parámetros capturados en el orden original con una coma entre ellos y se salta al "INITIAL".

En el estado "INITIAL", cualquier otro texto se imprime sin modificar nada.

2. Pruebas

ENTRADA	SALIDA
// holas sou ups(a1,a2) //:ups: //pups() //ups /*hola soy ups asterisco */ /*hola soy ups asterisco con un \n */ /*hola soy upps asterisco erroneo con un \n */ ups (a1,a2) hola int ups() upsi(a1,a2) ups(a1 a23456() ups (a1,a2) ups(a1,,a2) ups(,a1,,a2,,a3,,a4) ups:(a1,a2) ups(,a1) ups(a1,) ups(,a2) ups(ups(a1,a2),a2)	// holas sou ups(a1,a2)NO ACTUALIZADO //:ups: NO ACTUALIZADO //pups() //upsNO ACTUALIZADO /*hola soy ups asterisco NO ACTUALIZADO*/ /*hola soy upps asterisco erroneo con un \n NO ACTUALIZADO*/ /*hola soy ups asterisco con un \n */ ups (a2,a1) hola int ups() upsi(a1,a2) ups(a1 a23456() ups (a1,a2) ups(a2,a1) ups(a1,a2,a3,,a4) ups:(a1,a2) ups(,a1) ups(a1,) ups(,a2) ups(ups(a1,a2),a2)

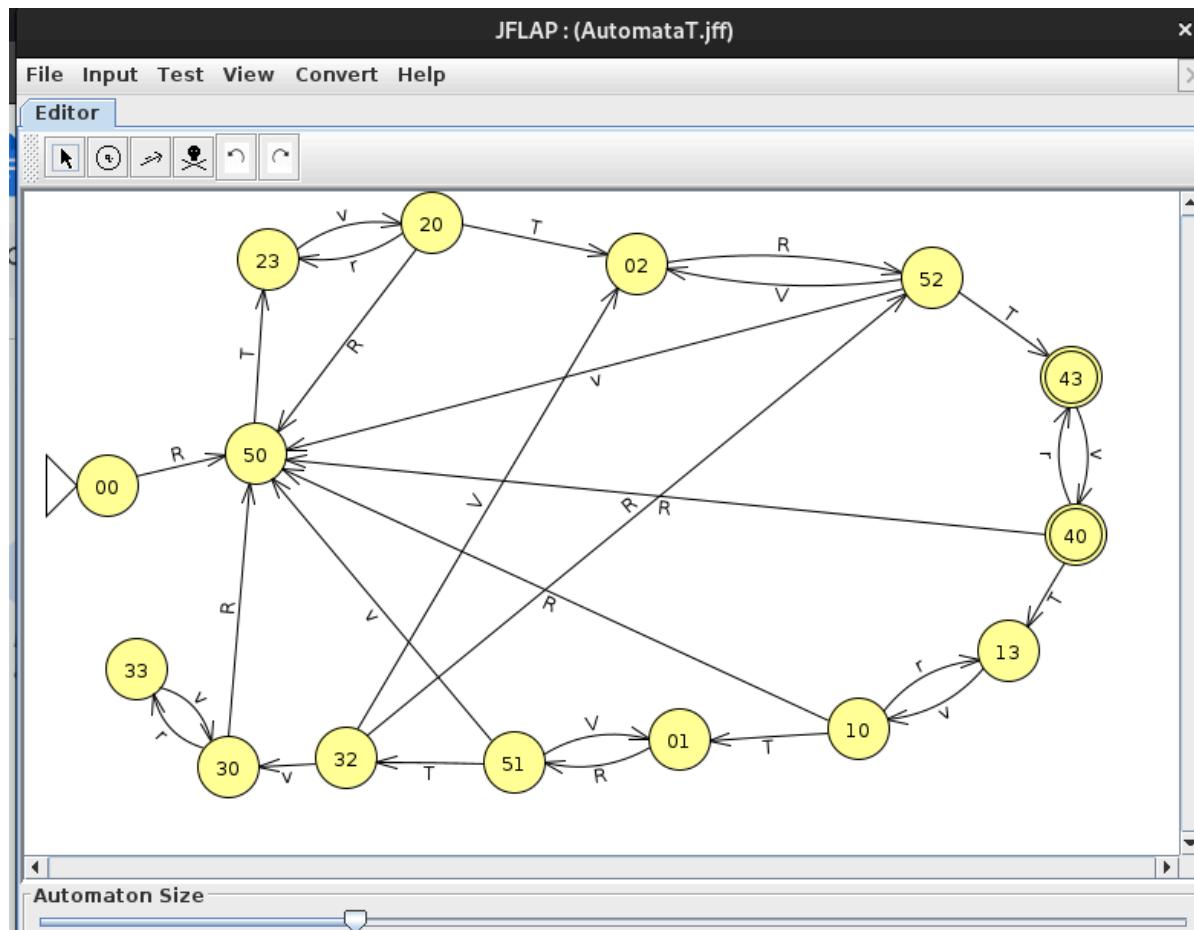
Ejercicio 2:

1. Resumen.

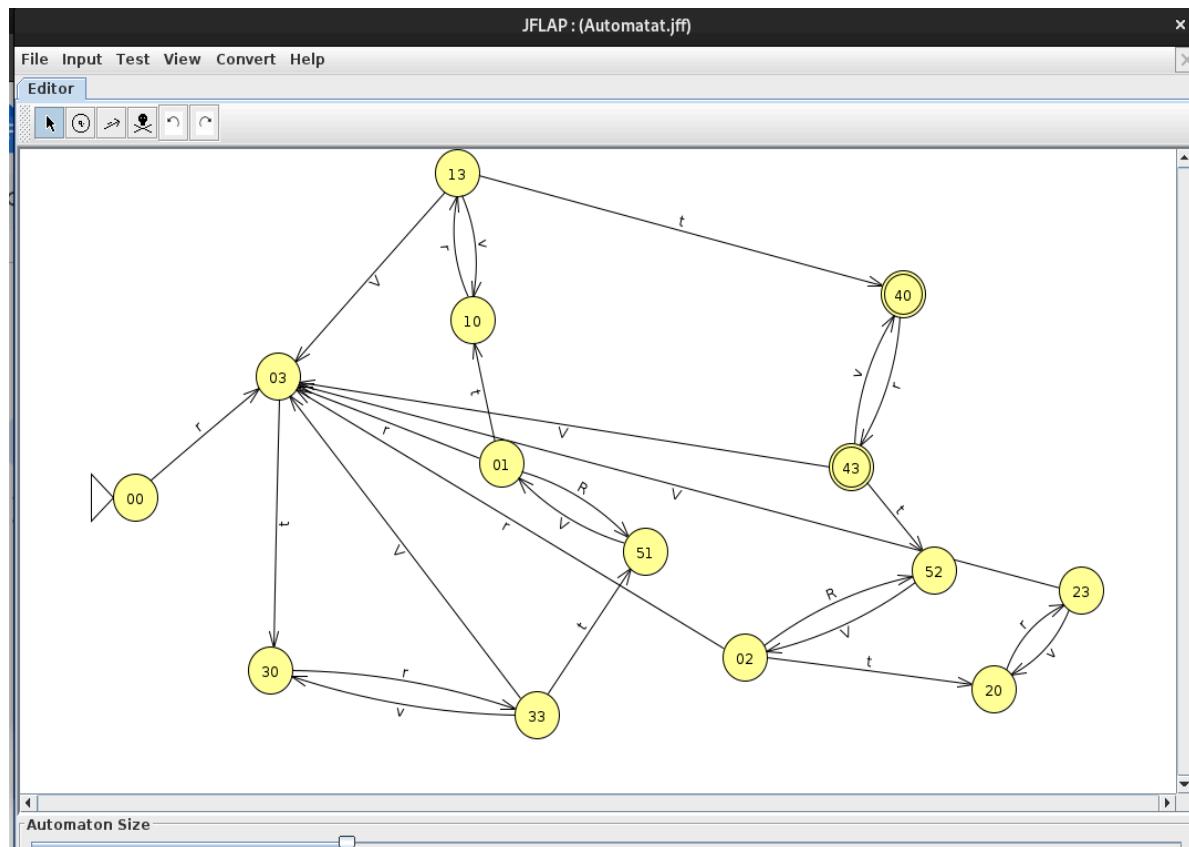
Hemos realizado los autómatas respectivos a los casos “T” y “t”, para posteriormente minimizarlos en el programa JFLAP, y con el mismo programa transformar el autómata en expresión regular. Después, hemos adaptado la ER a flex para que pueda procesarla.

En los autómatas, el número izquierdo representa el valor del barril grande de tamaño 5 y el derecho representa el barril pequeño de tamaño 3.

Autómata “T”



Autómata "t"



Todos los estados que no aparecen o son pozo (como el 53) o llegan inevitablemente al pozo (como el 50 en el autómata "t" o el 03 en el autómata "T".

Expresiones obtenidas al minimizar y adaptarlas a Flex:

T_SOLUCION:

$RTv((r|RT)v)^*TR((V|vTv((r|RT)v)^*T)R)^*T|RTrv((r|RT)v)^*TR((V|vTv((r|RT)v)^*T)R)^*Tv((r|((R|T v(rv)^*R)Tv((r|RT)v)^*T|Tv(rv)^*TR(VR)^*vTv((r|RT)v)^*T)R|Tv(rv)^*TR(VR)^*T(R|(V|v(rv)^*RTv((r|RT)v)^*T)R))((V|vTv((r|RT)v)^*T)R)^*(r|(((R|Tv(rv)^*R)Tv((r|RT)v)^*T|Tv(rv)^*TR(VR)^*vTv((r|RT)v)^*T)R|Tv(rv)^*TR(VR)^*T(R|(V|v(rv)^*RTv((r|RT)v)^*T)R))((V|vTv((r|RT)v)^*T)R)^*T?$

t_SOLUCION:

$rtr((v|Vt)r)^*tV((R|rtr((v|Vt)r)^*t)V)^*tr((v|Vtr((v|Vt)r)^*tV((R|rtr((v|Vt)r)^*t)V)^*t)r)^*t|rtr((v|Vt)r)^*tV((R|rtr((v|Vt)r)^*t)V)^*tr((v|Vtr((v|Vt)r)^*tV((R|rtr((v|Vt)r)^*t)V)^*t)r)^*tr((v|Vtr((v|Vt)r)^*tV((R|rtr((v|Vt)r)^*t)V)^*t)r)^*tr((v|Vtr((v|Vt)r)^*tV((R|rtr((v|Vt)r)^*t)V)^*t)r)^*tr((v|Vtr((v|Vt)r)^*tV((R|rtr((v|Vt)r)^*t)V)^*t)r)^*tr((v|Vtr((v|Vt)r)^*tV((R|rtr((v|Vt)r)^*t)V)^*t)r)?$

Al ser tan largas, las declaramos antes del primer %% para posteriormente llamarlas con {"definición"} para que flex pueda pre procesarlas y no de error al compilar.

2. Pruebas

Entrada:	Salida
RTvrvTRT	++RTvrvTRT
RTvTRTr	**RTvTRTr
rtrtVtrt	--rtrtVtrt
rtrvrVtrtVRVtrtr	--rtrvrVtrtVRVtrtr
RTtRT	**RTtRT
RTvTRTv	++RTvTRTv
RTvTRTvR	**RTvTRTvR
vRTvTRT	**vRTvTRT
RTvrvTRVRTvTvRTvrvTRT	++RTvrvTRVRTvTvRTvrvTRT
RTvTRT	++RTvTRT
rtrVtrtr	**rtrVtrtr
RTv	**RTv
rtrV	**rtrV
R	**R
r	**r
RTvTRTv	++RTvTRTv
RTvRTvRT	**RTvRTvRT
rtrVt	**rtrVt
vRT	**vRT
RTvv	**RTvv
RTvTRTV	**RTvTRTV
rtrVtrtrV	**rtrVtrtrV
RR	**RR
rr	**rr
RTr	**RTr
rtR	**rtR
rR	**rR
Rr	**Rr
T	**T
t	**t
vT	**vT
Vt	**Vt
RTRT	**RTRT
rtrVr t	rtrVr t
RTvrt	**RTvrt
rtVRT	**rtVRT
RTvTRTr	**RTvTRTr
RRTvTRT	**RRTvTRT
RTtRT	**RTtRT
RTvTRTvR	**RTvTRTvR
vRTvTRT	**vRTvTRT
RTTRRT	**RTTRRT
rtrtr	**rtrtr
RVrT	**RVrT

rtVtR rTRtrVrt RTvrvTRVRTvTvRTvrvTRT RTvrvTRT rtrtVtrt rtrvrVtrtVRVtrtr r R t v r V V R r T t	**rtVtR **rTRtrVrt ++RTvrvTRVRTvTvRTvrvTRT ++RTvrvTRT --rtrtVtrt --rtrvrVtrtVRVtrtr r R t v r V V R r T t
--	--