Teoría de Autómatas y Lenguajes Formales

Ejercicios de Autómatas a Pila

Autores:

Araceli Sanchis de Miguel
Agapito Ledezma Espino
Jose A. Iglesias Martínez
Beatriz García Jiménez
Juan Manuel Alonso Weber

* Algunos ejercicios están basados en enunciados de los siguientes libros:

- Enrique Alfonseca Cubero, Manuel Alfonseca Cubero, Roberto Moriyón Salomón. *Teoría de autómatas y lenguajes formales.* McGraw-Hill (2007).
- Manuel Alfonseca, Justo Sancho, Miguel Martínez Orga. *Teoría de lenguajes, gramáticas y autómatas.* Publicaciones R.A.E.C. (1997).
- Pedro Isasi, Paloma Martínez y Daniel Borrajo. *Lenguajes, Gramáticas y Autómatas. Un enfoque práctico.* Addison-Wesley (1997).

1. **Diseñar un Autómata a Pila para los siguientes lenguajes:**
   a. $\mathbf{L = \{ a^n \bullet b^n \mid n \geq 0\}}$
   b. $\mathbf{L = \{ a^n \bullet b^{2n} \mid n > 0\}}$
   c. $\mathbf{L = \{ a^{2n} \bullet b^n \mid n \geq 0\}}$
   d. $\mathbf{L = \{ a^{2n} \bullet b^n \mid n > 0\}}$

2. **Diseñar un Autómata a Pila para el lenguaje:** $\mathbf{L = \{ a^{n+m} \bullet b^{m+t} \bullet a^t \bullet b^n \mid n, t > 0, m \geq 0\}}$

3. **Obténgase el APV correspondiente a la gramática**

   G_{FNG}=({a,b,c,d}, {S,A,B}, S, P), con las siguientes reglas de producción:

   S ::= a S B | b A | b | d
   A ::= b A | b
   B ::= c

4. **Obtener formalmente el APf equivalente para cada el APv indicado a continuación:**

   APv_a=({1,2}, {A,B,B’,C}, {q}, A, q, f, {Φ}), donde f viene dada por:

   f(q,2,A) = (q, BC)
   f(q,1,A) = (q,B)
   f(q,λ,A) = (q, λ)
   f(q,1,B) = {(q,B’), (q,C), (q, λ)}
   f(q,2,B’) = {(q,B’), (q,C)}
   f(q,2,C) = (q, λ)

5. **Obtener formalmente el APv equivalente para el APf indicado:**

   APf_b=({a,b}, {A,B}, {q1,q2,q3,q4}, A, q1, f, {q4}), donde f viene dada por:

   f(q1,a,A) = {(q2,BA), (q4,A)}
   f(q1,λ,A) = {(q4, λ)}
   f(q2,a,B) = {(q2,BB)}
   f(q2,b,B) = {(q3, λ)}
   f(q3,λ,A) = {(q4,A)}
   f(q3,b,B) = {(q3, λ)}

6. Marque las afirmaciones verdaderas

a. Al obtener una G2 a partir de un APf, ésta se encontrará en FNG
b. Es posible que una G3 pueda ser transformada a APv
c. Dado un AP no determinista, existen algoritmos para transformarlo en AP determinista.
d. En un AP determinista, dado un estado, un símbolo leido y un símbolo en la cima de la pila, transitaremos al mismo estado, pudiendo apilar dos conjuntos diferentes de símbolos.

7. Marque las afirmaciones verdaderas

a. Dado un movimiento en un AP, es posible determinar el par (imagen, antiimagen) de la función de transición correspondiente.
b. Los autómatas de pila por vaciado no pueden transformarse en autómatas de pila por estados finales
c. Los autómatas de pila por estados finales reconocen una palabra cuando la pila está vacía y no queda nada por leer en la entrada.
d. Los autómatas de pila por estados finales no son nunca deterministas.

8. Marque las afirmaciones verdaderas

a. (p,a,A;p,Z) indica que se apila un solo símbolo Z
b. (p,a,A;p,Z) indica que se desapila A
c. (p,a,A;p,A) indica que la pila queda igual tras la transición.
d. (p,a,λ;p,λ) indica que la pila queda igual tras la transición.

9. Marque las afirmaciones verdaderas

a. $f(q, \lambda, A) = \{(q, \lambda)\}$es una transición independiente de la entrada.
b. La descripción instantánea (q, λ,, λ) en un autómata de pila que reconoce por vaciado indica que hemos llegado al final de la palabra con la pila vacía.
c. El alfabeto de pila y el alfabeto de entrada de un autómata de pila son conjuntos disjuntos.
d. La transición $f(q,a, A)=\{ (q2, z1),(q1, z1) \}$ nos indica que el autómata de pila es no determinista.

10. Describa las funciones de transición que dan lugar a los siguientes movimientos:

(p,1001, A) |- (p, 001,1A) |- (p, 01, 01A) |- (q, 1, 1A) |- (q, λ, A) |- (q, λ, λ)

SOLUCIONES

1. Diseñar un Autómata a Pila para los siguientes lenguajes:

a. L = { $a^n \bullet b^n$ | n ≥ 0}

b. L = { $a^n \bullet b^{2n}$ | n > 0}

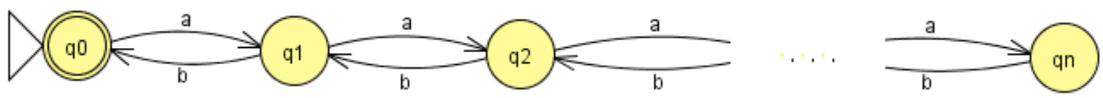
c. L = { $a^{2n} \bullet b^n$ | n ≥ 0}

d. L = { $a^{2n} \bullet b^n$ | n > 0}

Solución:

a. L = { $a^n \bullet b^n$ | n ≥ 0}

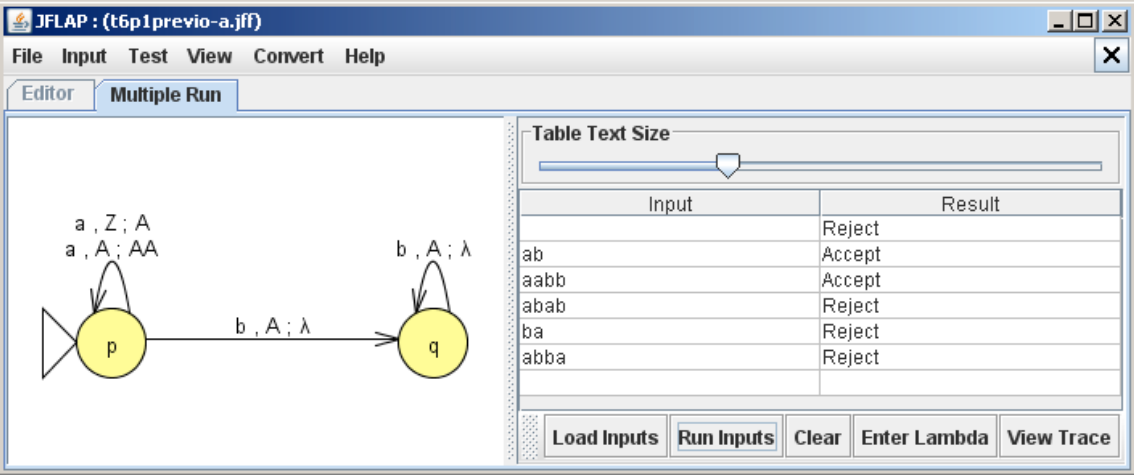
Este tipo de lenguaje no se puede reconocer con un AFD dado que es un dispositivo que carece de una memoria auxiliar para contabilizar los símbolos leídos. La única posibilidad de llevar la cuenta es disponer de tantos estados como valores de n pueda haber, cosa poco práctica:



El autómata de pila emplea una pila en la que pueda ir anotando lo que necesite recordar. Para el problema planteado se propone almacenar en la pila una A por cada *a* leída en la cinta de entrada. A la hora de leer las *b* de la cinta de entrada bastará con eliminar una A por cada *b* leída. La cadena de entrada se reconoce sí una vez terminada de leer la pila se queda vacía.

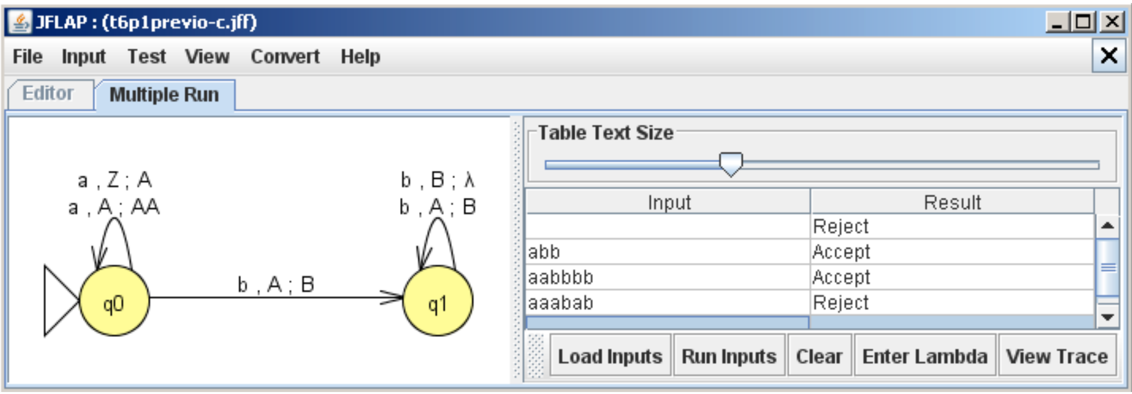
Emplearemos dos estados (p y q) para diferenciar la secuencia de *a* de la de *b*.

f(p, a, Z) = (p, A)	al iniciarse el APv, con la primera *a* leída
f(p, a, A) = (p, AA)	para sucesivas *a* en la cinta de entrada, va apilando otras tantas A.
f(p, b, A) = (q, λ)	para la primera *b* en la cinta de entrada, cambia de estado.
f(q, b, A) = (q, λ)	para las demás *b* en la cinta de entrada, sigue desapilando A.
f(p, λ, Z) = (q, λ)	para reconocer una cinta vacía (palabra vacía). Esta transición sólo lleva al reconocimiento de la palabra en la cinta, cuando esta haya sido leída por completo.



b. L = { $a^n \bullet b^{2n}$ | n > 0}

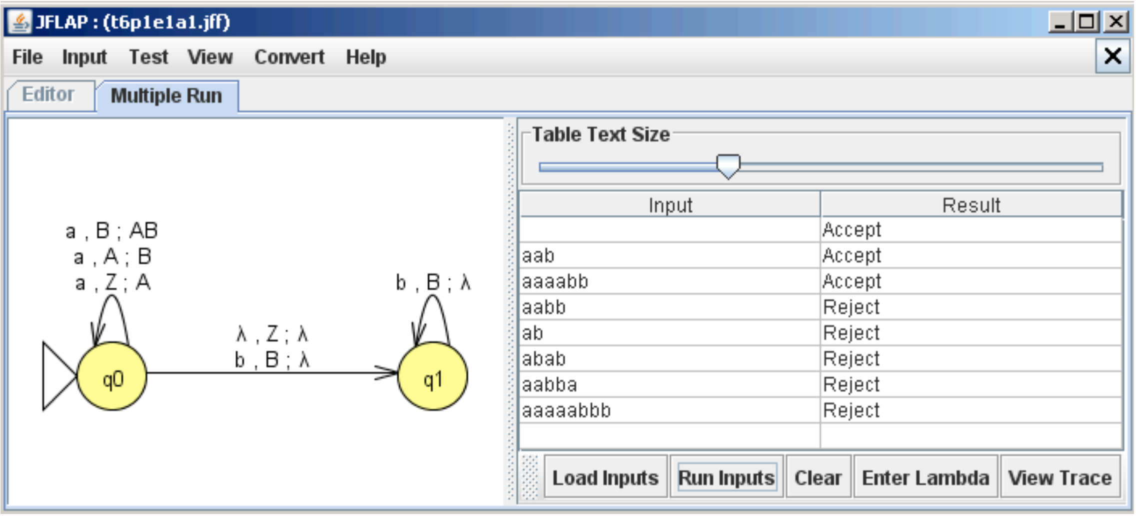
Solución de diseño:
Diferenciamos la cadena en dos partes para tratar la secuencia de *a* y la de *b* por separado (un estado para cada secuencia). Por cada *a* leída, ponemos una A en la pila. Por cada A en la pila tenemos que leer dos *b*. Esto lo solventamos en dos tiempos, sustituyendo una A por una B con la primera *b* leída, y eliminando la B con la segunda *b* leída.



Ejemplo de funcionamiento:
(p, aabbbb, Z) |- (p, abbbb, A) |- (p, bbbb, AA) |- (q, bbb, BA) |- (q, bb, A) |- (q, b, B) |- (q, λ, λ)

c. $L = \{ a^{2n} \bullet b^n \mid n \geq 0\}$

Solución de diseño:
Basándonos en el ejercicio anterior, cada *a* impar la memorizamos con una A, y cada *a* par sustituirá la A de la cima de la pila por una B.

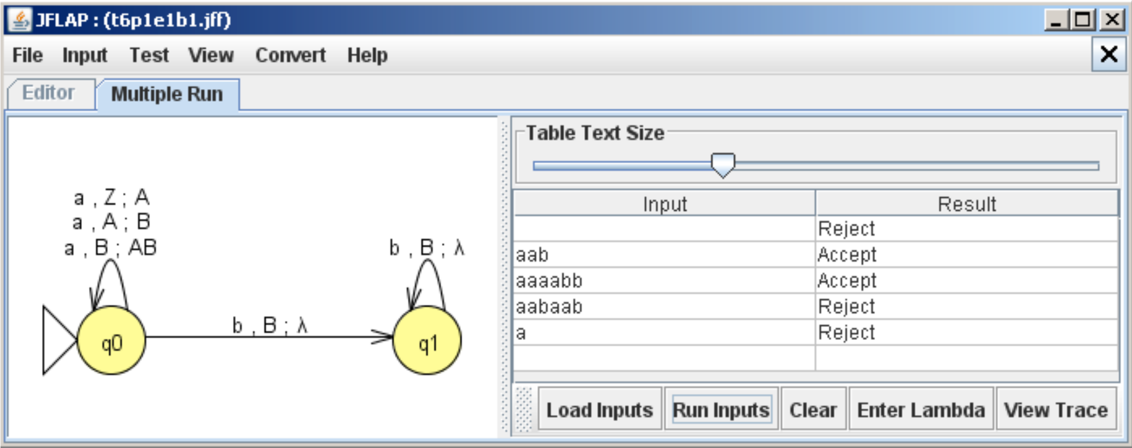


Ejemplo de funcionamiento:

(p, aaaabb, Z) ├ (p, aaabb, A) ├ (p, aabb, B) ├ (p, abb, AB) ├ (p, bb, BB) ├ (q, b, B) ├ (q, λ, λ)

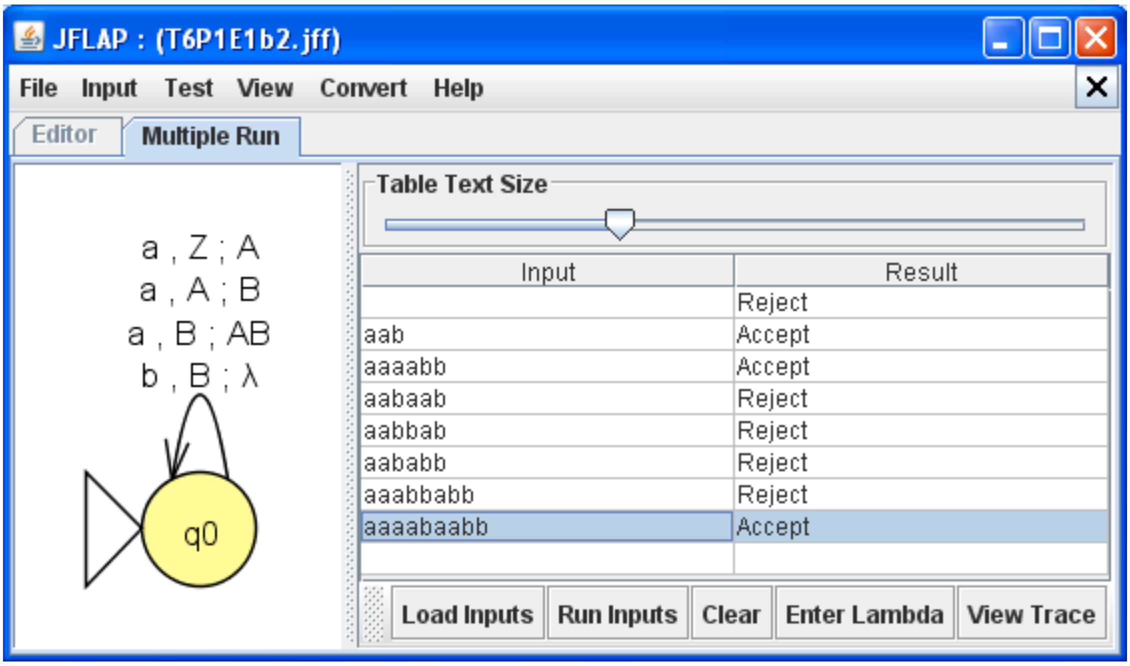
d. L = { $a^{2n}•b^{n}$ | n > 0}

Sobre la solución anterior se elimina la transición que elimina Z de la pila, con lo cuál no se puede reconocer lambda.



En algunos casos podemos condensar el APv en otro de un solo estado, pero conviene asegurarse de que no reconoce cadenas ajenas al lenguaje, por ejemplo aaaabaabb, que sí reconoce cuando no pertenece al lenguaje.

Este APv es incorrecto.



2. Diseñar un Autómata a Pila para el siguiente lenguaje: (1e)

$L = \{ a^{n+m} \bullet b^{m+t} \bullet a^{t} \bullet b^{n} \mid n, t > 0, m \geq 0\}$

Solución:

Expandimos la notación: $L = \{a^n(a^m b^m)(b^t a^t)b^n \mid n, t > 0, m \geq 0\}$

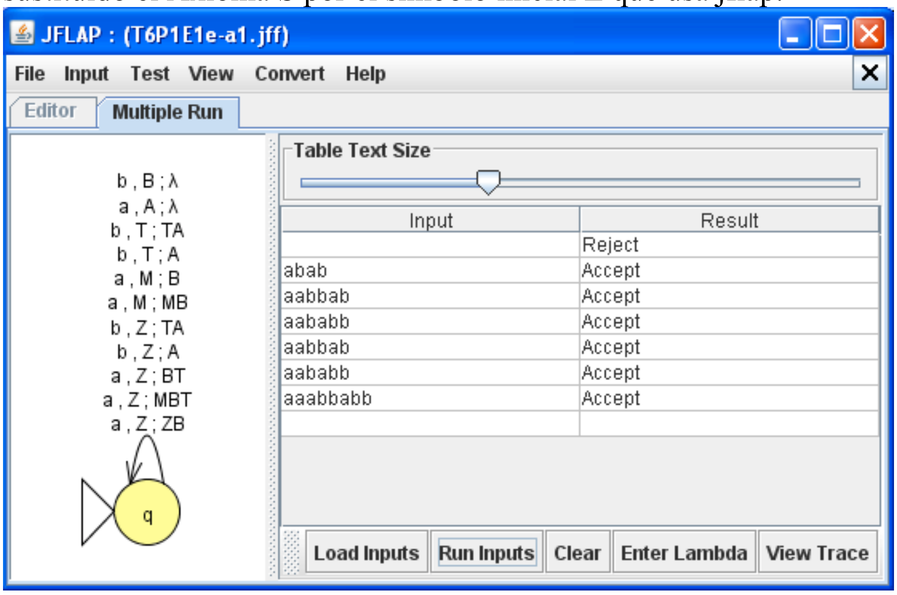
En este caso tenemos el problema que consiste en que para determinar el valor de los índices n, m y t la única posibilidad parece ser contar la última secuencia de *a* y *b*. Eso determina t y n y de ahí podemos determinar el valor de m. Pero para cuando hemos leído dicha secuencia de *a* ya no podemos retroceder en la cinta de entrada.

A) Pasando la gramática en FNG a Apv

La solución más inmediata parece ser partir de una gramática

Gramática:	Limpiamos lambda:	Paso a FNG:
S → aSb | MT	S → aSb | MT | T	S → aSB | aMBT | aBT | bA | bTA
M → λ | aMb | ab	M → aMb | ab	M → aMB | aB
T → ba| bTa	T → ba| bTa	T → bA| bTA
		A → a
		B → b

Siguiendo el algoritmo de las transparencias, se obtiene el APv de la figura. Se ha sustituido el Axioma S por el símbolo inicial Z que usa jflap.

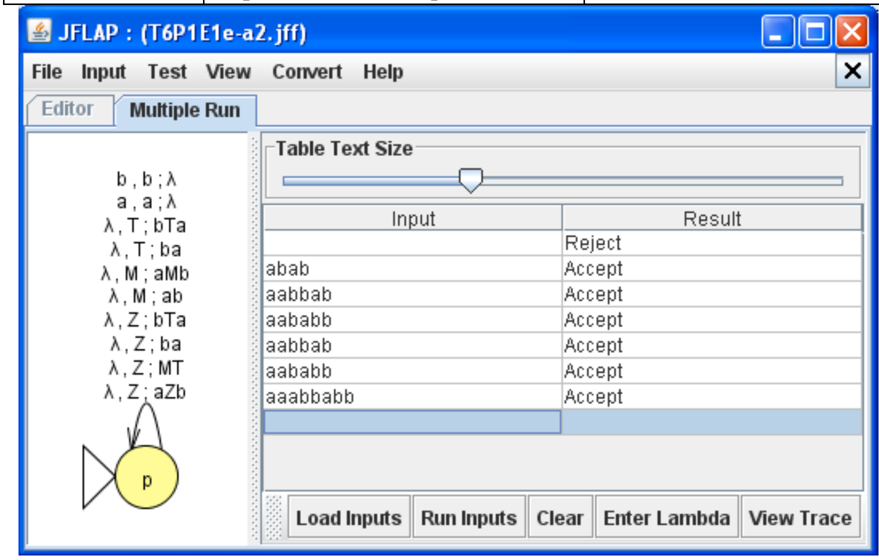


(q0, abab, Z) ├ (q0, abab, aZb) ├ (q0, bab, Zb) ├ (q0, bab, Tb) ├ (q0, bab, bab) ├ (q0, ab, ab) ├ (q0, b, b) ├ (q0, λ, λ)

B) Pasando G2 a APv

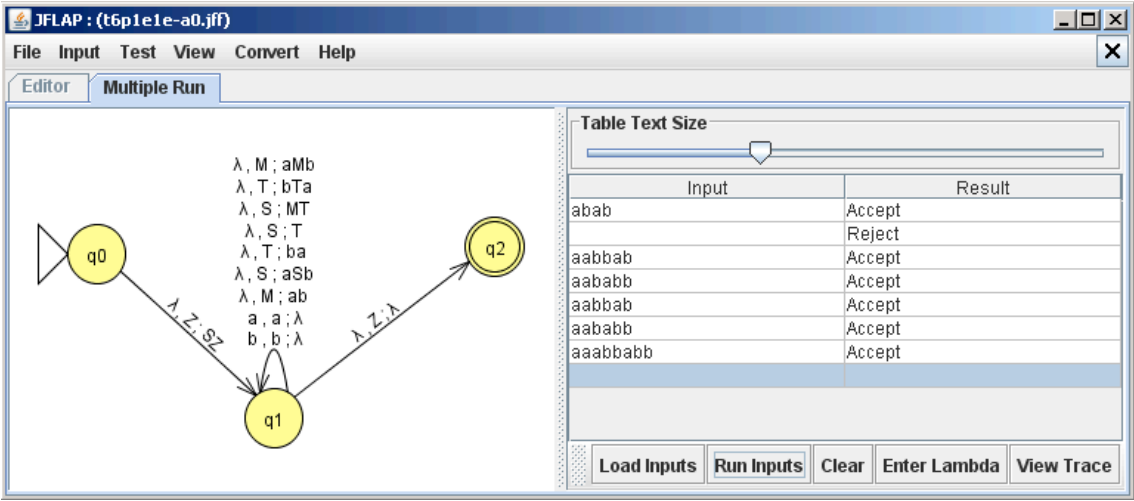
Se puede generalizar el algoritmo para gramáticas que no están en FNG, añadiendo una transición por cada terminal.

S → aSb	(q, aSb) ∈ f(q, λ, S)	Por cada terminal se incluye: f(q, a, a) = (q, λ) f(q, b, b) = (q, λ)
S → MT	(q, MT) ∈ f(q, λ, S)	
S → ba	(q, ba) ∈ f(q, λ, S)	
S → bTa	(q, bTa) ∈ f(q, λ, S)	
M → aMb	(q, aMb) ∈ f(q, λ, M)	
M → ab	(q, ab) ∈ f(q, λ, M)	
T → ba	(q, ba) ∈ f(q, λ, T)	
T → bTa	(q, bTa) ∈ f(q, λ, T)	



También aquí sustituimos el Axioma S por Z para que el APv pueda funcionar en jflap.

C) Solución de jflap, partiendo de la G2 limpia:
Jflap es capaz de generar un APv aplicando una variante del algoritmo anterior. Partiendo de la gramática sin lambda (en principio deberíamos eliminar también la redenominación S -> T, cosa que obviamos ahora para reducir transiciones), se añaden un estado inicial y otro final. El primer estado sirve para poner en la pila el Axioma sobre el el símbolo inicial del APv (Z), y el último para eliminar dicho símbolo cuando no queda nada más en la pila. Esta configuración la emplea jflap porque así puede operar **indistintamente** como APv o APf.



Podemos observar que los APv obtenidos son capaces de procesar las gramáticas con las que se generan. Al tratarse de APs no deterministas lo que hacen es probar a sustituir en la pila un no terminal (empezando por el axioma) por la parte derecha de alguna de las producciones. Cuando en la cima de la pila hay un terminal se emplean las transiciones de los terminales para eliminarlas. A base de tentativas intenta buscar un camino hasta vaciar la pila. En la práctica este tipo de proceso no determinista hay que implementarlo empleando la técnica de *backtracking*.

La ventaja del primer algoritmo, que parte de la FNG, es que puede resultar mucho más eficiente de operar el APv:
- Se combina en una sóla transición las operaciones de poner en la pila la parte derecha de cada producción y de eliminar el terminal que queda en la cima.
- El proceso para elegir la transición adecuada está más acotado, ya que el símbolo que se esté leyendo en la cinta determina una selección más reducida de transiciones aplicables.

D) Solución de diseño:

Otra solución, en la que obtenemos el APv "por diseño" se basa precisamente en el potencial que ofrece un dispositivo no determinista. Al comienzo del problema se mencionaba la dificultad de determinar los valores de n, m y t antes de leer por completo la cinta de entrada. Otra posibilidad consiste en determinar dichos valores por tanteo. Justo aquí es dónde resulta útil el no determinismo.

Al analizar la cadena de entrada que empieza por *a*, sabemos que la primera pertenecerá obligatoriamente al bloque determinado por n, pero a partir de la segunda podría pertenecer también al bloque determinado por m. La forma de resolución consiste en suponer que una *a* puede ser tanto del bloque n como del m. Es decir:

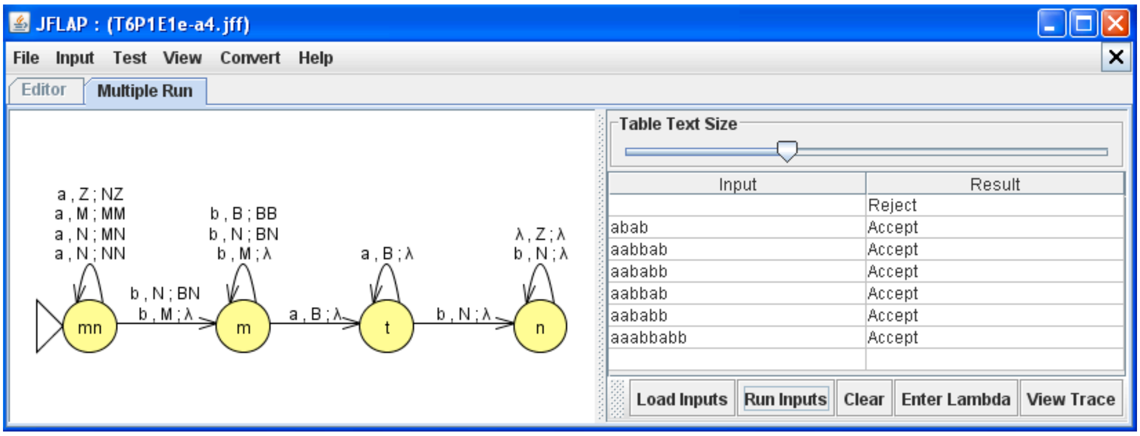
si en la entrada hay una *a*, el APv se bifurca suponiendo en un caso que pertenece al bloque n, y en el otro que pertenece al bloque m. Emplearemos los símbolos de pila N y M para memorizar ambos casos.

F(p, a, X) = {(p, NX), (p, MX)}

¿Qué símbolo X nos podemos encontrar en la pila?

- Z, símbolo inicial del APv ➔ la primera *a* leída sólo puede ser del bloque n ➔
  f(p, a, Z) = (p, NZ)
- N, símbolo que indica una *a* leída que pertenece al bloque n ➔
  f(p, a, N) = {(p, NN), (p, MN)}
- M, símbolo que indica una a leída que pertenece al bloque m ➔ mientras sigan llegando *aes* las consideraremos del bloque m ➔ f(p, a, M) = (p, MM)

Este sería el punto clave del diseño del autómata. El resto considera que al llegar la primera secuencia de *b*, vamos quitando las M que haya en la pila (estado m) y por cada *b* restante ponemos una B (del grupo t) en la pila. En el estado t se elimina una B de la pila por cada *a* leída, y en el último estado (n) se eliminan las N por cada *a* leída en la cinta de entrada.



3. Obténgase el APV correspondiente a la gramática:

G_{FNG}=({a,b,c,d}, {S,A,B}, S, P), con las siguientes reglas de producción:

S ::= a S B | b A | b | d
A ::= b A | b
B ::= c

Solución:

G = ({a,b,c,d}, {S,A,B}, {q}, S, q, f, {})

```
S → aSB | bA | b | d
A → bA | b
B → c
```

G = ({a,b,c,d}, {S,A,B}, {q}, S, q, f, {})

```
S → aSB        ➔     f(q,a,S) = (q, SB)
S → bA         ➔     f(q,b,S) = (q, A)
S → b          ➔     f(q,b,S) = (q, λ)
S → d          ➔     f(q,d,S) = (q, λ)
A → bA         ➔     f(q,b,A) = (q, A)
A → b          ➔     f(q,b,A) = (q, λ)
B → c          ➔     f(q,c,B) = (q, λ)
```

4. Obtener formalmente el APf equivalente para cada el APv indicado a continuación:

APv_a=({1,2}, {A,B,B’,C}, {q}, A, q, f, {Φ}), donde f viene dada por:

f(q,2,A) = (q, BC)
f(q,1,A) = (q,B)
f(q,λ,A) = (q, λ)
f(q,1,B) = {(q,B’), (q,C), (q, λ)}
f(q,2,B’) = {(q,B’), (q,C)}
f(q,2,C) = (q, λ)

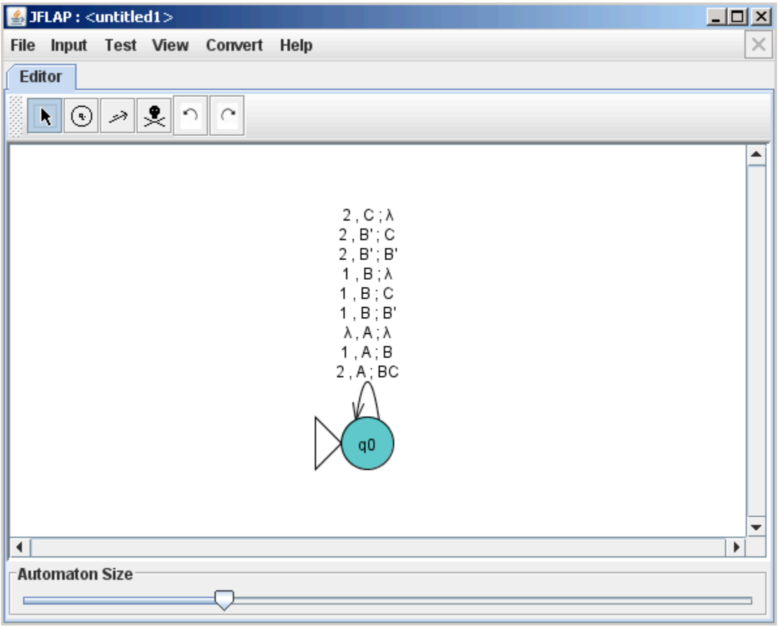
Solución:

APf_a=({1,2}, {A,B,B’,C,Z}, {q,p,r}, Z, p, f’, {r}), donde f viene dada por:

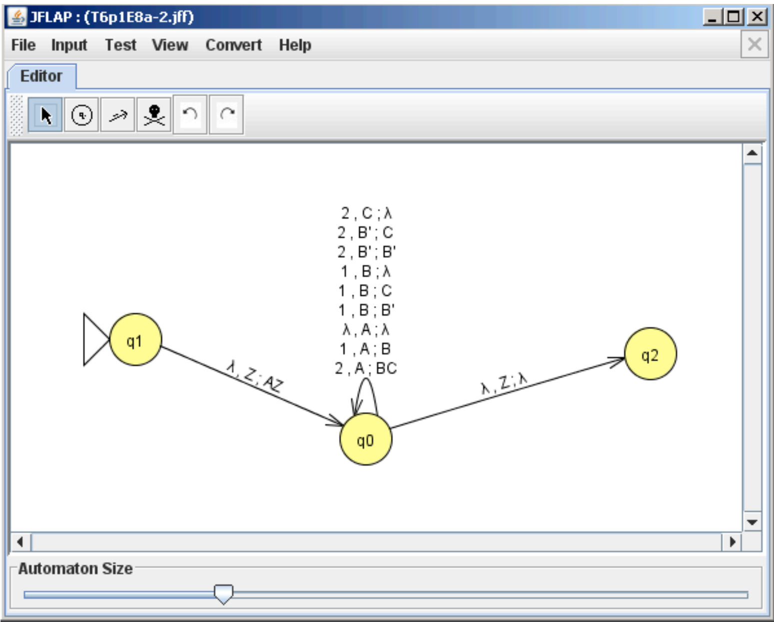
- Se añaden 2 nuevos estados, p y r, inicial y final. Se añade nuevo símbolo inicial de pila, Z.
- Se añade una transición lambda, del estado inicial nuevo al antiguo, extrayendo el nuevo símbolo inicial de pila (Z), e insertando el antiguo y el nuevo (AZ).
  f(p,λ,Z) = (q,AZ)
- Se copian todas las transiciones del APv:
  f(q,2,A) = (q, BC)
  f(q,1,A) = (q,B)
  f(q,λ,A) = (q, λ)
  f(q,1,B) = {(q,B’), (q,C), (q, λ)}
  f(q,2,B’) = {(q,B’), (q,C)}
  f(q,2,C) = (q, λ)
- Al llegar al símbolo Z se ha vaciado la pila, es decir, se ha reconocido la palabra en el APv, por lo que se debe llegar la estado final en el APf, para lo que se añade una transición lambda desde cada estado del APv al estado final del APf:
  f(q, λ, Z) = (r, λ)

```
f(q,2,A)   = (q, BC)
f(q,1,A)   = (q, B)
f(q,λ,A)   = (q, λ)
f(q,1,B)   = {(q, B’), (q, C), (q, λ)}
f(q,2,B’)  = {(q, B’), (q, C)}
f(q,2,C)   = (q, λ)
```

Para este problema es posible obtener otra solución mínima (de un solo estado):



Otro Apv posible emplea un símbolo de pila inicial (Z) que se pone al inicio, y que debe ser eliminado para que la palabra sea reconocida:



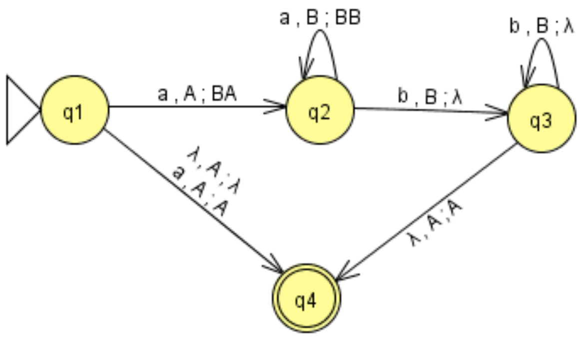
5. Obtener formalmente el APv equivalente para el AP indicado:

APf_b=({a,b}, {A,B}, {q1,q2,q3,q4}, A, q1, f, {q4}), donde f viene dada por:

f(q1,a,A) = {(q2,BA), (q4,A)}
f(q1,λ,A) = {(q4, λ)}
f(q2,a,B) = {(q2,BB)}
f(q2,b,B) = {(q3, λ)}
f(q3,λ,A) = {(q4,A)}
f(q3,b,B) = {(q3, λ)}

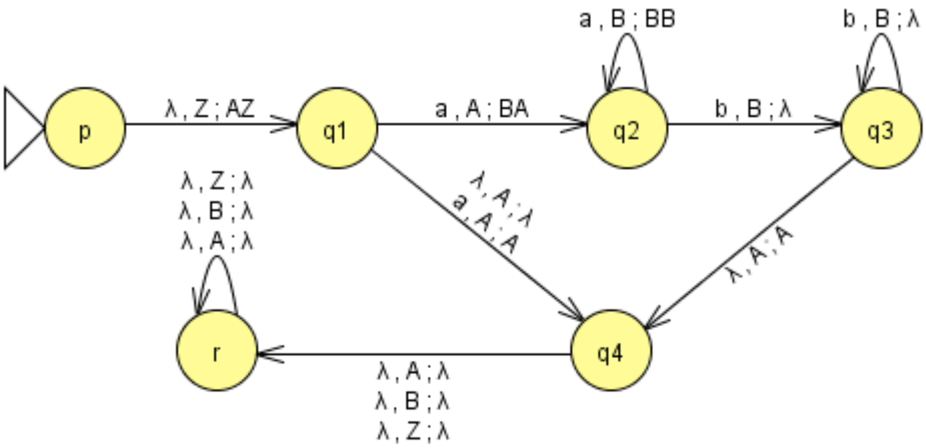
Solución:

APf origen:



APv equivalente final: APf_b=({a,b}, {A,B,Z}, {q1,q2,q3,q4,p,r}, Z, p, f', {Φ}), donde se han añadido de f a f' las siguientes transiciones:

- Para unir estados iniciales: f(p,λ,Z)=(q1,AZ)
- Para vaciar la pila:
  - De q4 a r: f(q4,λ,A)=(r, λ) f(q4,λ,B)=(r, λ) f(q4,λ,Z)=(r, λ)
  - De r a r: f(r,λ,A)=(r, λ) f(r,λ,B)=(r, λ) f(r,λ,Z)=(r, λ)



6. Marque las afirmaciones verdaderas

a. Al obtener una G2 a partir de un APf, ésta se encontrará en FNG

b. Es posible que una G3 pueda ser transformada a APv

c. Dado un AP no determinista, existen algoritmos para transformarlo en AP determinista.

d. En un AP determinista, dado un estado, un símbolo leido y un símbolo en la cima de la pila, transitaremos al mismo estado, pudiendo apilar dos conjuntos diferentes de símbolos.

7. Marque las afirmaciones verdaderas

a. Dado un movimiento en un AP, es posible determinar el par (imagen, antiimagen) de la función de transición correspondiente.

b. Los autómatas de pila por vaciado no pueden transformarse en autómatas de pila por estados finales

c. Los autómatas de pila por estados finales reconocen una palabra cuando la pila está vacía y no queda nada por leer en la entrada.

d. Los autómatas de pila por estados finales no son nunca deterministas.

8. Marque las afirmaciones verdaderas

a. (p,a,A;p,Z) indica que se apila un solo símbolo Z

b. (p,a,A;p,Z) indica que se desapila A

c. (p,a,A;p,A) indica que la pila queda igual tras la transición.

d. (p,a,λ;p,λ) indica que la pila queda igual tras la transición.

9. Marque las afirmaciones verdaderas

a. f(q, λ,A) = {(q, λ)}es una transición independiente de la entrada.

b. La descripción instantánea (q, λ,, λ) en un autómata de pila que reconoce por vaciado indica que hemos llegado al final de la palabra con la pila vacía.

c. El alfabeto de pila y el alfabeto de entrada de un autómata de pila son conjuntos disjuntos.

d. La transición f(q,a, A)={ (q2, z1),(q1, z1) } nos indica que el autómata de pila es no determinista.

10. Describa las funciones de transición que dan lugar a los siguientes movimientos:

(p,1001, A) |- (p, 001,1A) |- (p, 01, 01A) |- (q, 1, 1A) |- (q, λ, A) |- (q, λ, λ)

Solución:

(p,1001, A) |- (p, 001,1A)	f(p,1,A) = (p, 1A)
(p, 001,1A) |- (p, 01, 01A)	f(p,0,1) = (p, 01)
(p, 01, 01A) |- (q, 1, 1A)	f(p,0,0) = (q, λ)
(q, 1, 1A) |- (q, λ, A)	f(q,1,1) = (q, λ)
(q, λ, A) |- (q, λ, λ)	f(q,λ,A) = (q,λ)