Práctica 8:
Autómatas de Pilas

Cátedra: Lenguajes Formales y Computabilidad
Aulas Virtuales - Pandemia Covid19

Docentes en Práctica
- Alejandro Hernandez
- Natalia Colussi
- Valeria Perez Moguetta

Para Arrancar Primero!

- Te proponemos una **actividad** para que practiques con unos autómatas de pila (AP) que ya hemos diseñado y sobre los cuales te proponemos lo siguiente:

  1. Revísalos de forma completa.
  2. Observes con atención todos los comentarios que te dejamos y cada detalle que remarcamos en los mismos.
  3. Te mostramos en cada uno cómo ejecutarlos con la pila que aparece abajo del diagrama de transición a la hora de verificar si reconoce una palabra, es decir, si es aceptada por el lenguaje que describe el autómata.
  4. Te dejamos de tarea que ejecutes de igual manera las transiciones necesarias y los movimientos en la pila para las cadenas que mencionamos como ejemplo.

- **Subí tu solución en la jamboard** de esta sección y el jueves trabajaremos con esto al ppio de la clase.

Actividad 1:

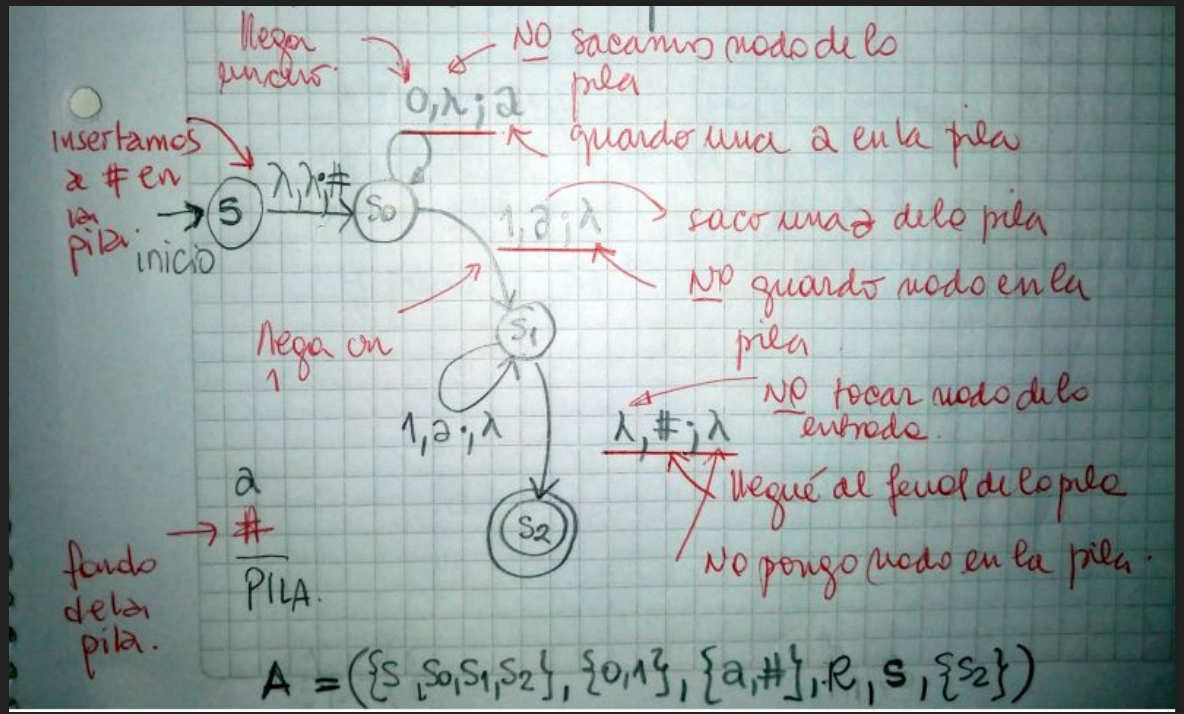
Dado el siguiente lenguaje

$L_1 = \{ 0^n 1^n \cdot n \geq 1 \}$

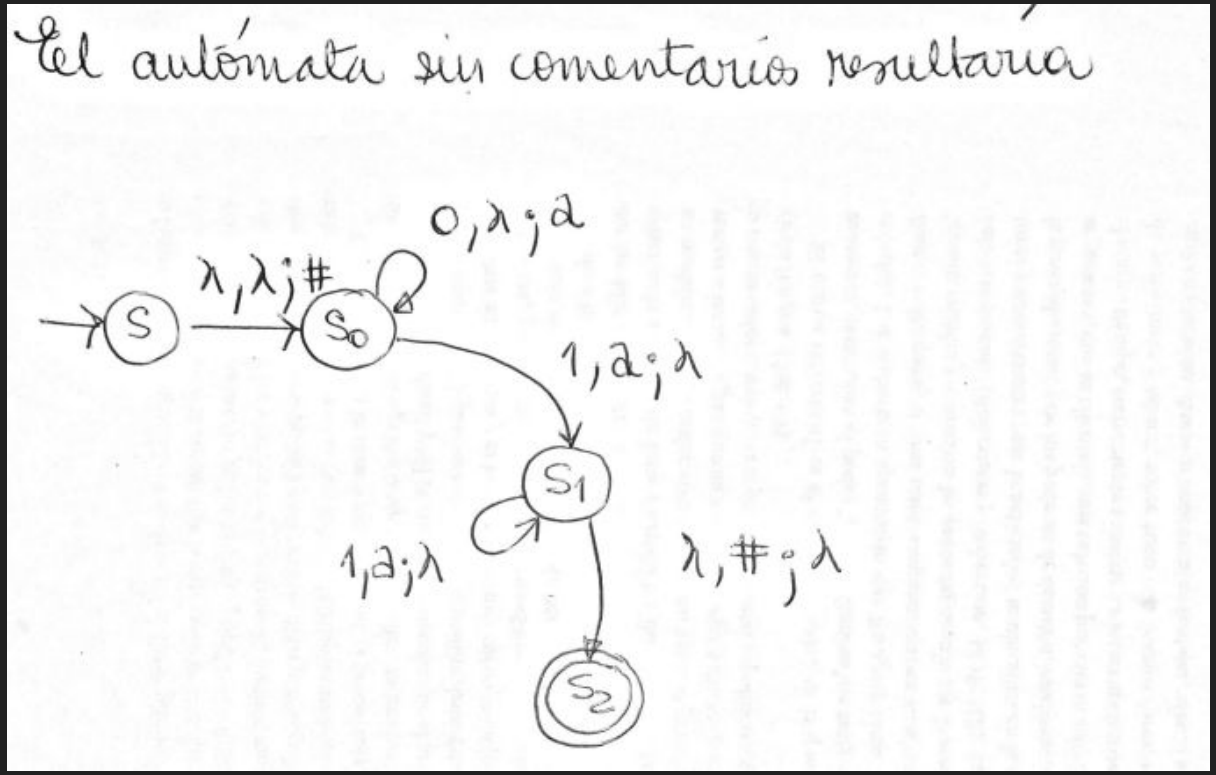
y el autómata de pila A que lo reconoce.

Lenguaje con el que vamos a trabajar.
L2, libre de contexto.

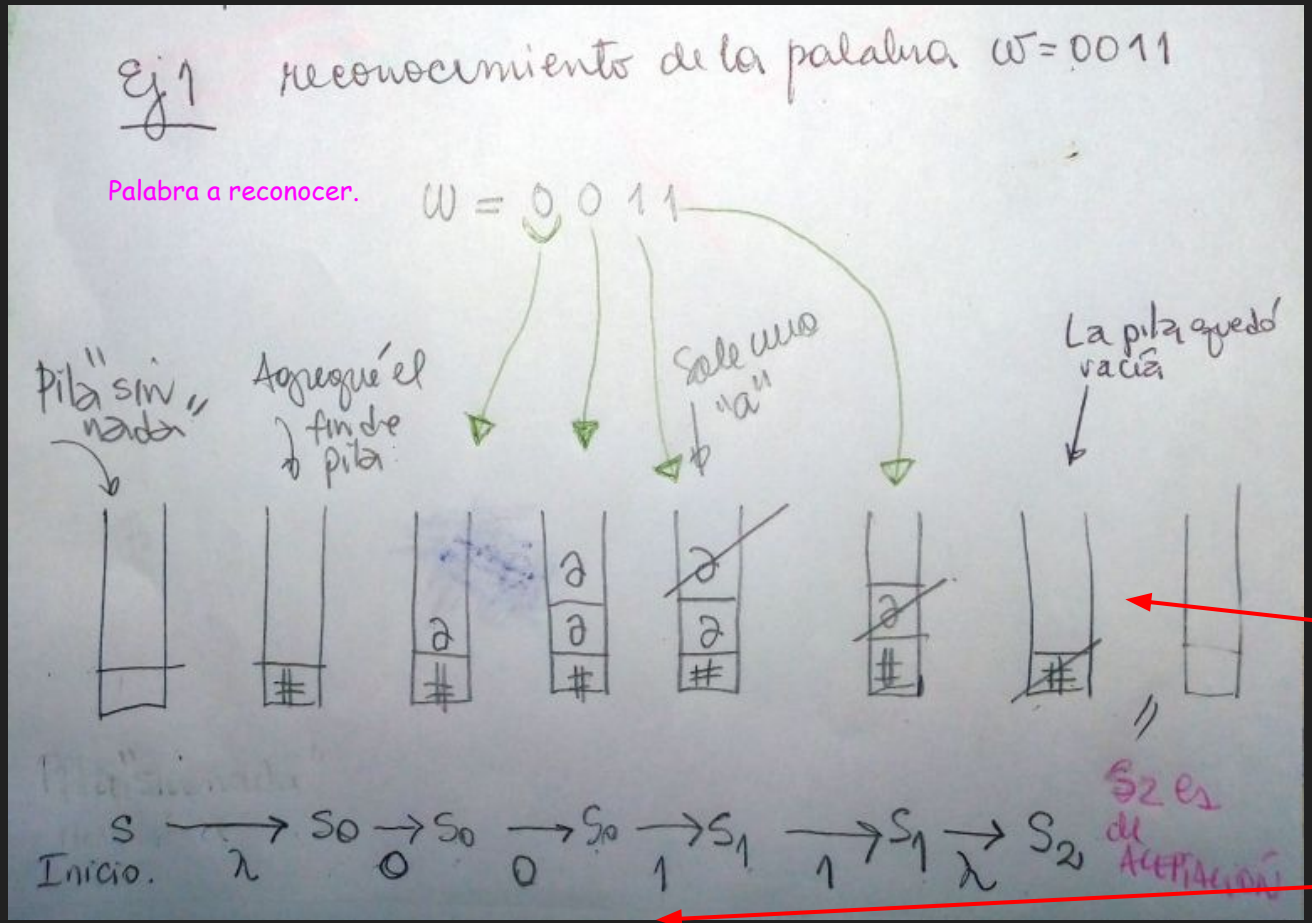
Autómata de pila explicado paso a paso.
Sólo con fines didácticos.

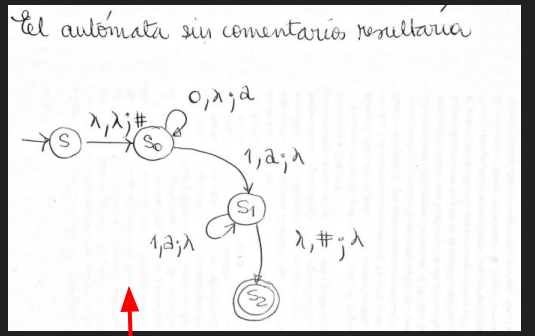


Autómata como una 6 u-pla



Autómata de base sobre el cual ejecutaremos las transiciones y los movimientos de la pila para comprender cómo acepta las palabras y cómo funciona realmente.





Para ejecutar las transiciones del autómta y los movimientos de la pila te pedimos que siempre tengas a mano el diagrama del autómata.

Estados con los movimientos de la pila según la lectura de la aada dato de entrada de la palabra.

Vamos marcando los estados que visitamos y lo que consumimos de la entrada. Marcamos el inicio y el estado final que es de aceptación.

Te pedimos que en base a las transiciones del ejemplo 1 realices las transiciones para los ejemplos 2 y 3, recorriendo el autómata y operando sobre la pila.

Dibuja entonces las pilas y los estados que transicionas para las siguientes palabras

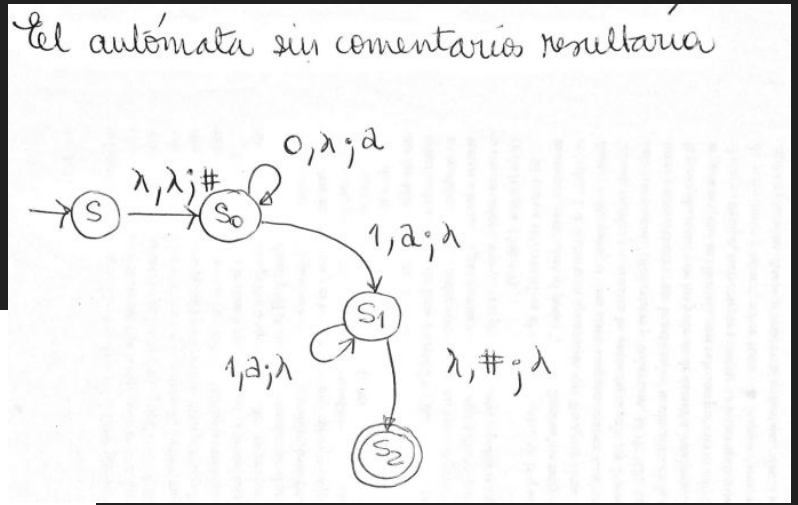
Ejemplo 2, $w = 000111$

Ejemplo 3, $w = 01$

Subí tus soluciones en la Jamboard que te dejamos en comunidades. Dejá los movimientos en la pila y las transiciones con los estados y las entradas. Tal cual te lo mostramos en el ejemplo.

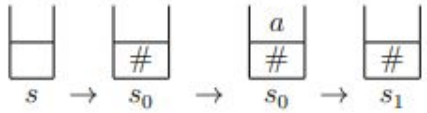
Actividad 2:

En este caso te pedimos que elijas una o dos cadenas que no sean reconocidas por el autómatas y procedas como en el ejemplo para demostrar que no es posible aceptarla.



Ejemplo 2 "NO" reconocimiento de la palabra $\omega = 011$.

$\omega =$ 0 1 1 ♯



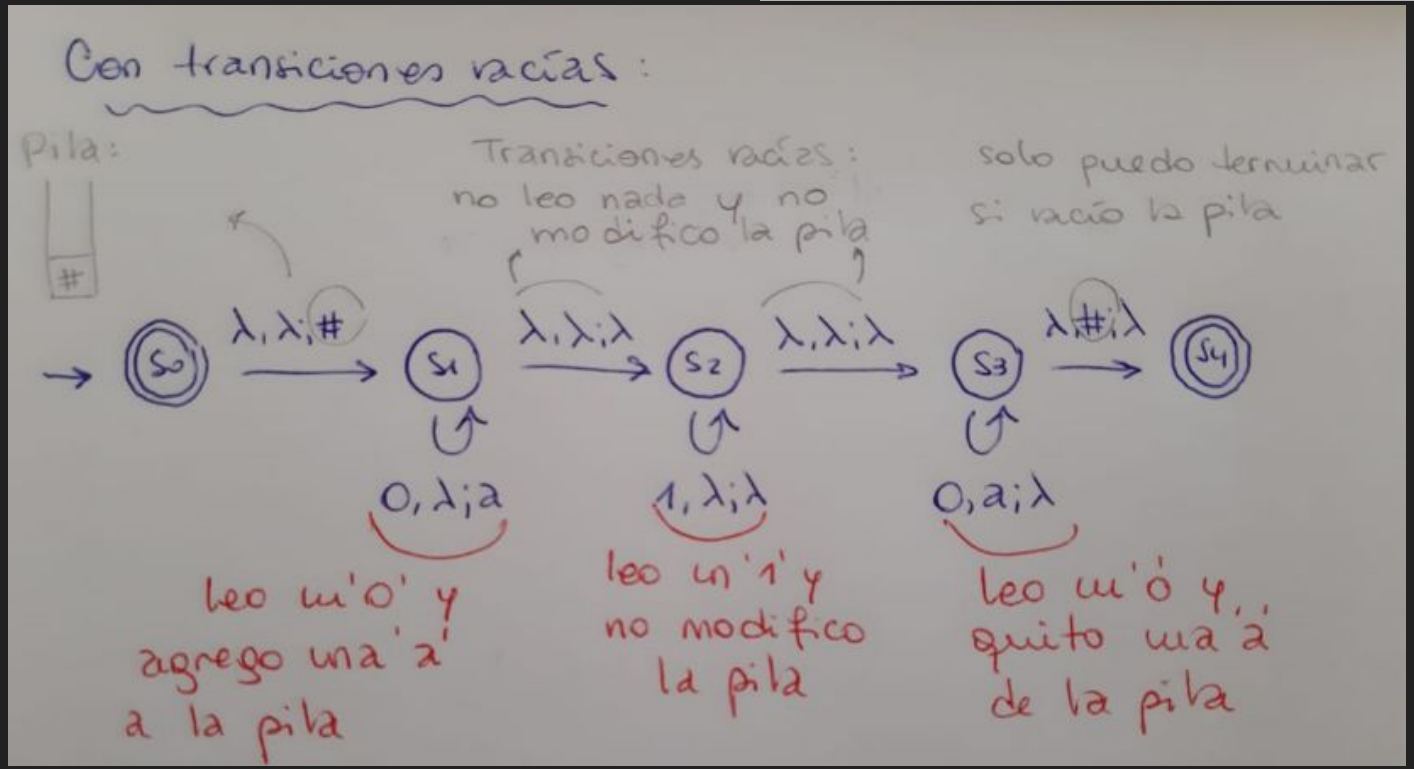
Vale la pena notar que no es posible transicionar cuando llega el último 1 pues para poder llevar a cabo dicha transición es necesario que haya una "a" en la pila, cosa que evidentemete no sucede.

Subí tus soluciones en la Jamboard que te dejamos en comunidades. Dejá los movimientos en la pila y las transiciones con los estados y las entradas. Tal cual te lo mostramos en el ejemplo.

Actividad 3:

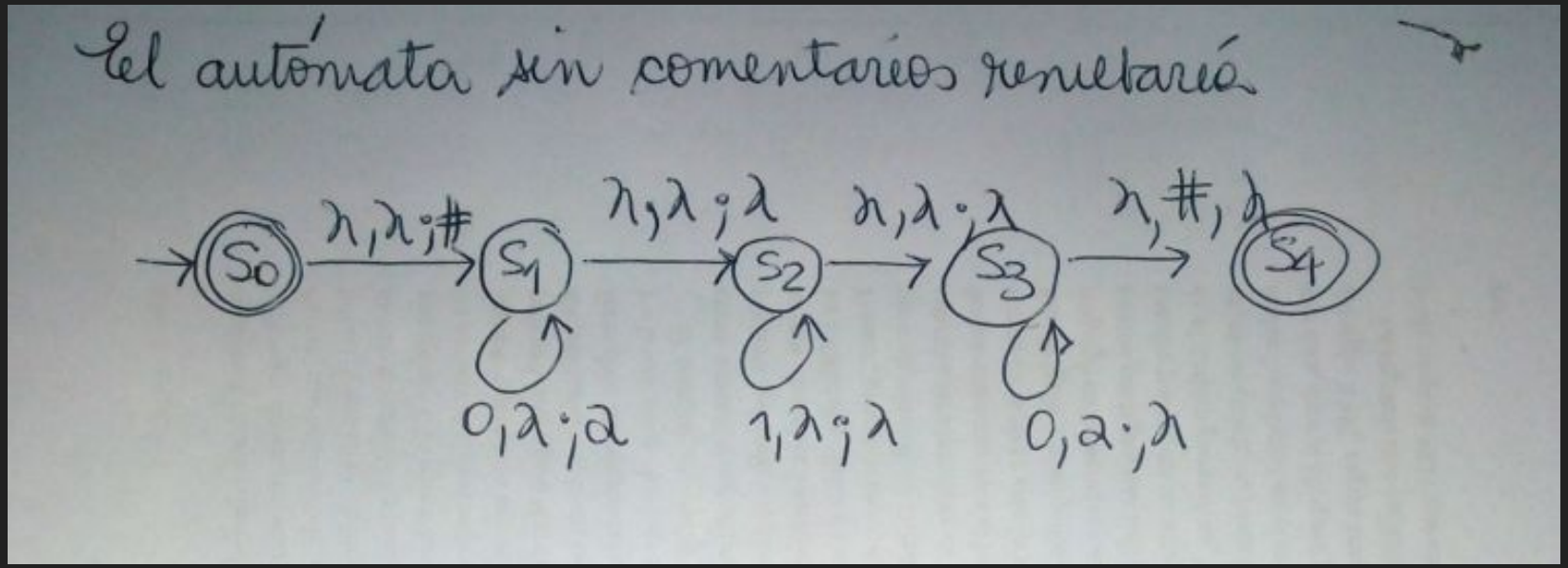
Dado el siguiente lenguaje,

$$L = \{ 0^n 1^m 0^n / n, m \in \mathbb{N}_0 \}$$



y el autómata de pila que lo reconoce.

Autómata de base sobre el cual ejecutaremos las transiciones y los movimientos de la pila para comprender cómo acepta las palabras y cómo funciona realmente.

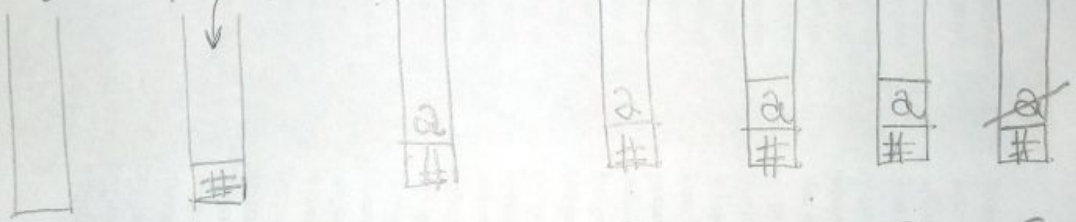


Ej Reconocimiento de la palabra.

$w = 0\ 1\ 0$

Pila sin nada

Agrego el fin de pila

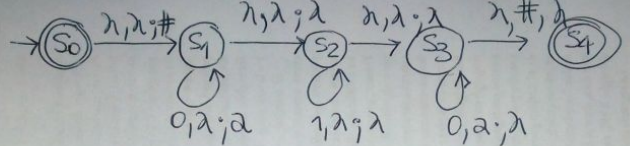


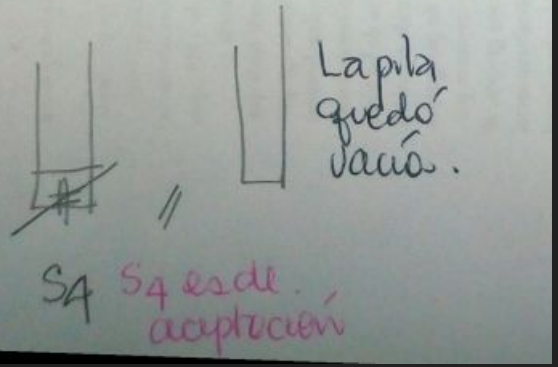
$s_0 \xrightarrow{\lambda} s_1 \xrightarrow{0} s_1 \xrightarrow{\lambda} s_2 \xrightarrow{1} s_2 \xrightarrow{\lambda} s_3 \xrightarrow{0} s_3 \xrightarrow{\lambda}$

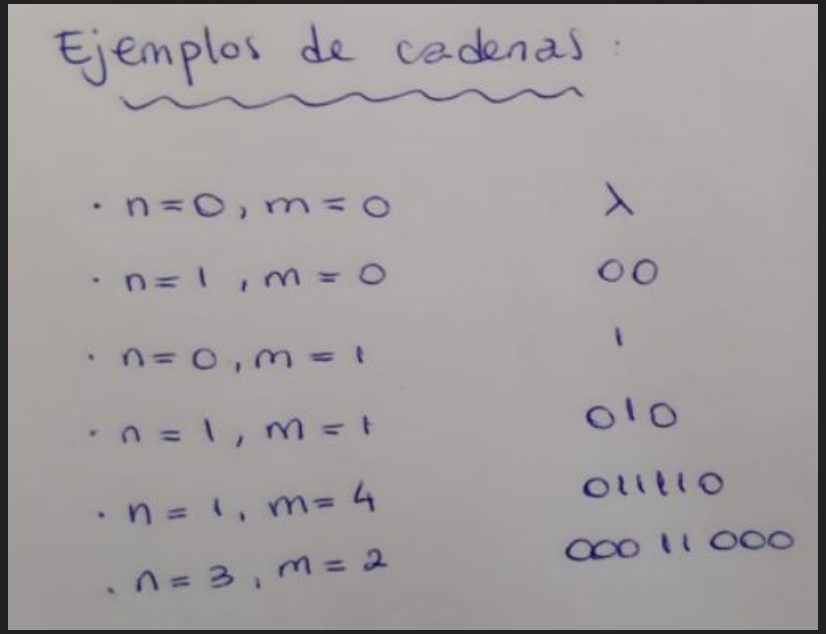
observar que las pilas quedan iguales.

Transiciones vacías.

El autómata sin comentarios reconocerá





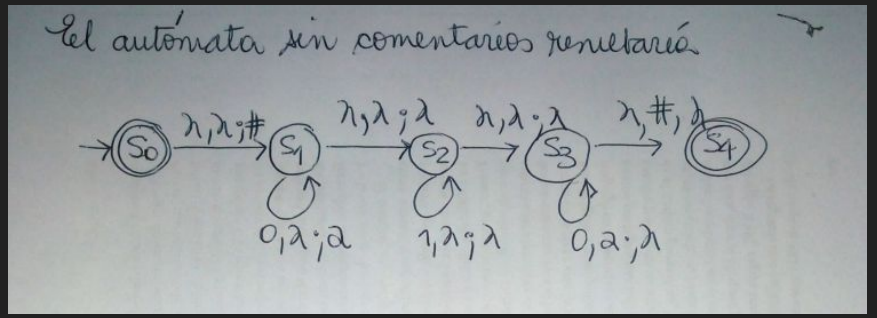


Te pedimos que en base a las transiciones del ejemplo 1, realices las transiciones para los restantes ejemplos que te proponemos, recorriendo el autómata y operando sobre la pila en cada caso. Dibujá las pilas y escribí las transiciones para cada ejemplo.

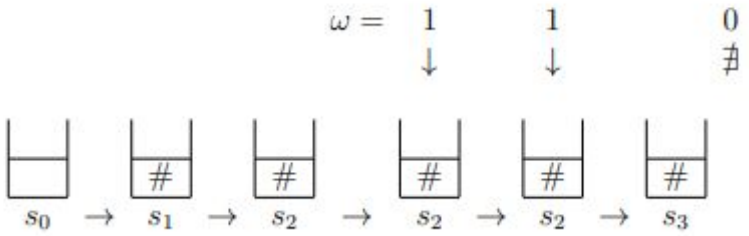
Subí tus soluciones en la Jamboard que te dejamos en comunidades. Dejá los movimientos en la pila y las transiciones con los estados y las entradas. Tal cual te lo mostramos en el ejemplo.

Actividad 4:

En este caso te pedimos que elijas una o dos cadenas que no sean reconocidas por el autómatas y procedas como en el ejemplo para demostrar que no es posible aceptarla.



"NO" reconocimiento de la palabra $\omega = 110$.

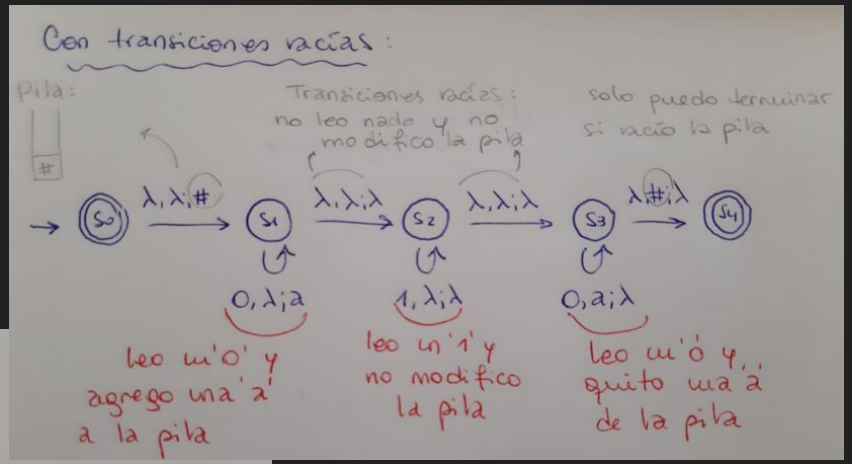


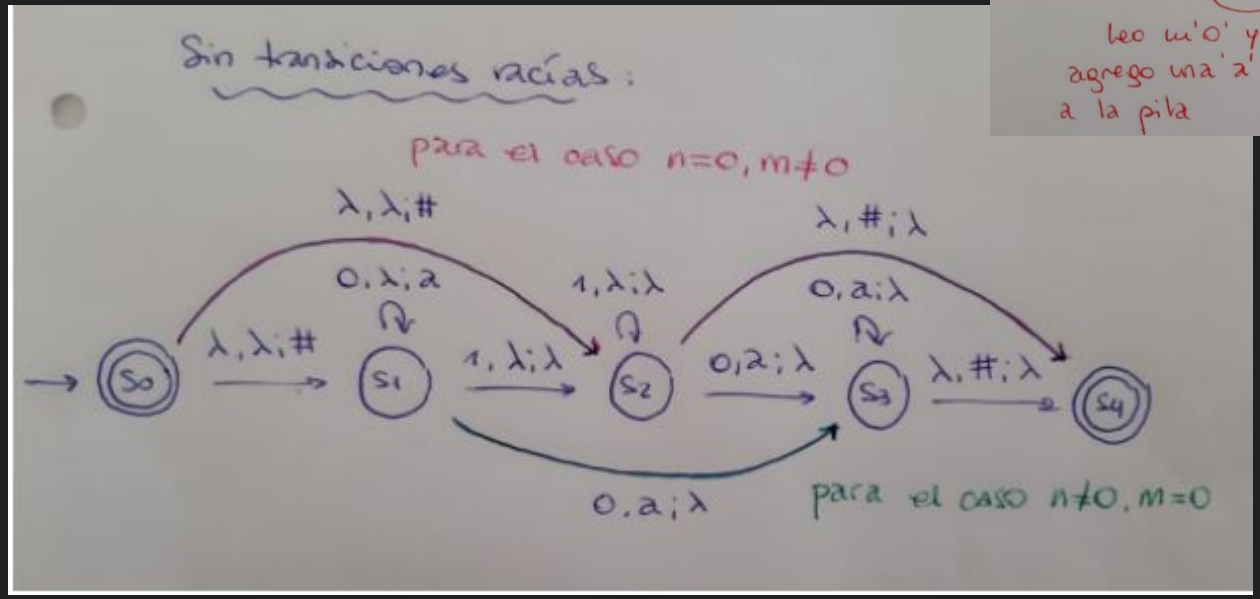
Vale la pena notar que no es posible transicionar cuando llega el último 0 pues para poder llevar a cabo dicha transición es necesario que haya una "a" en la pila, cosa que evidentemete no sucede.

Subí tus soluciones en la Jamboard que te dejamos en comunidades. Dejá los movimientos en la pila y las transiciones con los estados y las entradas. Tal cual te lo mostramos en el ejemplo.

Observaciones:

Este último autómata tiene transiciones que llamamos vacías. Podemos construir uno equivalente sin ellas. Te lo mostramos a continuación





Podes practicar la actividades 3 y 4 con este autómata sin transiciones vacías y comparar ambas soluciones. Discutiremos la solución en clase.

Ahora seguimos con los ejercicios de la **práctica 8**

Ejercicio 1:

1. Dé un autómata de pila que reconozca cada uno de los siguientes lenguajes y que, al aceptar una cadena, su pila quede vacía:

c) $\{a^n b^{2n} c^m \mid m, n \in \mathbb{N}_0\}$

Lo primero que hacemos es describir el lenguaje, antes que comenzar con el diagrama del autómata. Vemos: ...

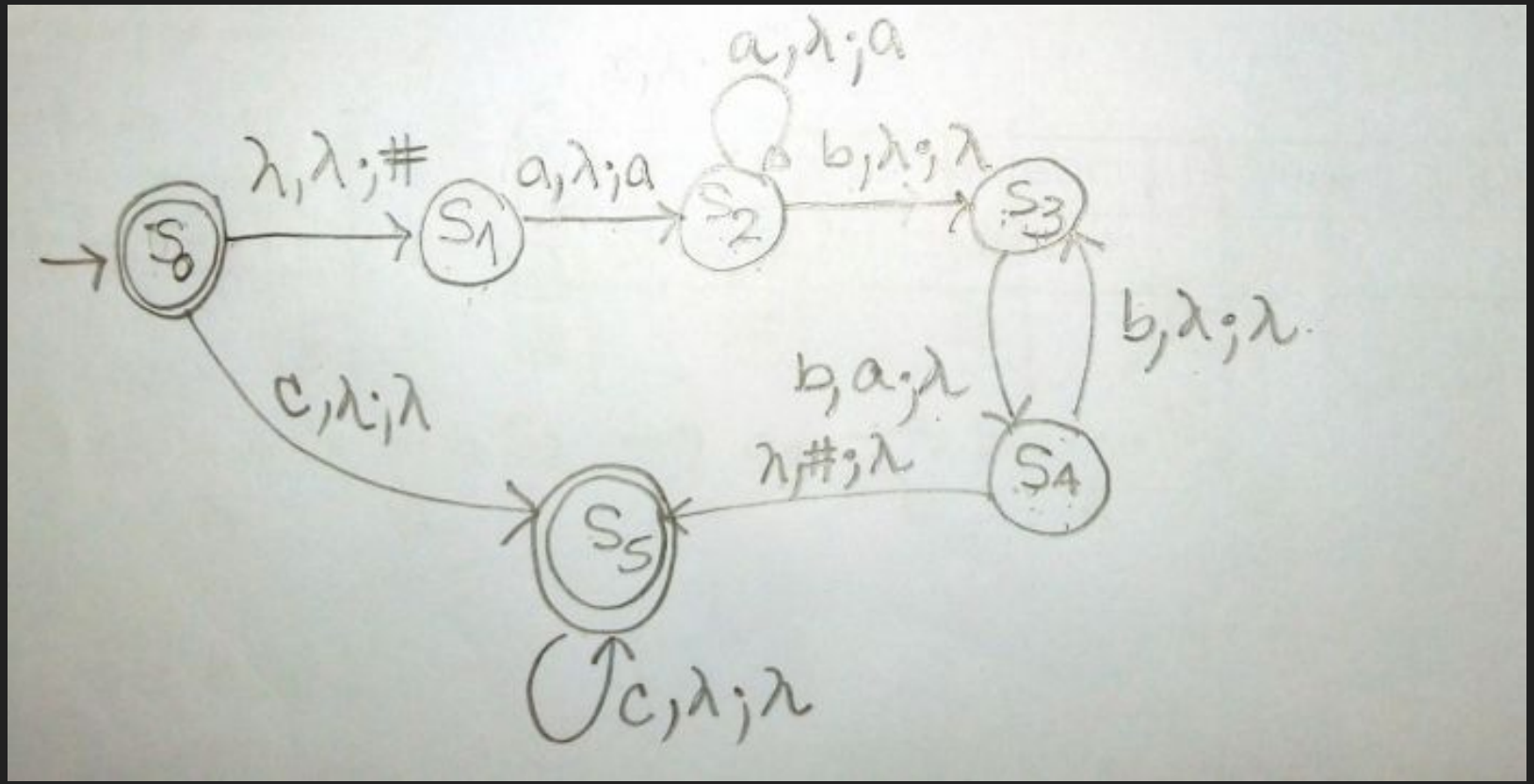
Vamos algumas cadeias.

$n=0$, $m=0$ $\quad a^0 b^{2\cdot 0} c^0 = \lambda$

$n=0$, $m>0$ $\quad a^0 b^{2\cdot 0} c^m = c^m = \{c, cc, ccc, \cdots\}$

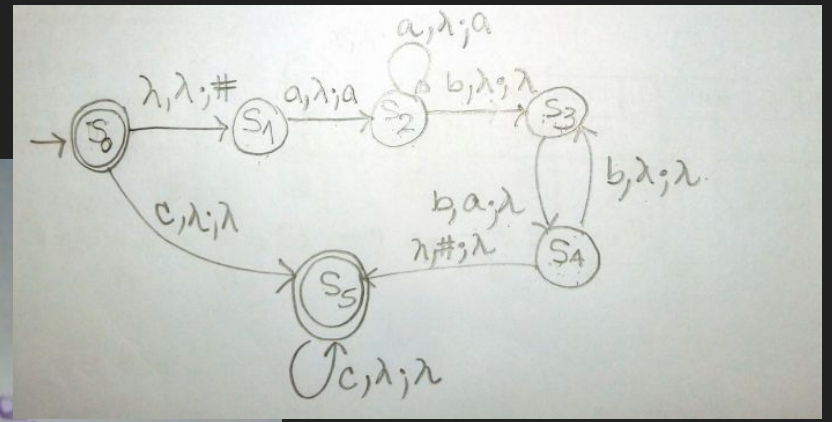
$n>0$, $m=0$ $\quad a^n b^{2n} c^0 = a^n b^{2n} = \{abb, aabbbb, \cdots\}$

$n>0$, $m>0$ $\quad a^n b^{2n} c^m = \{abbc,$ $aabbbbcc \cdots\}$.

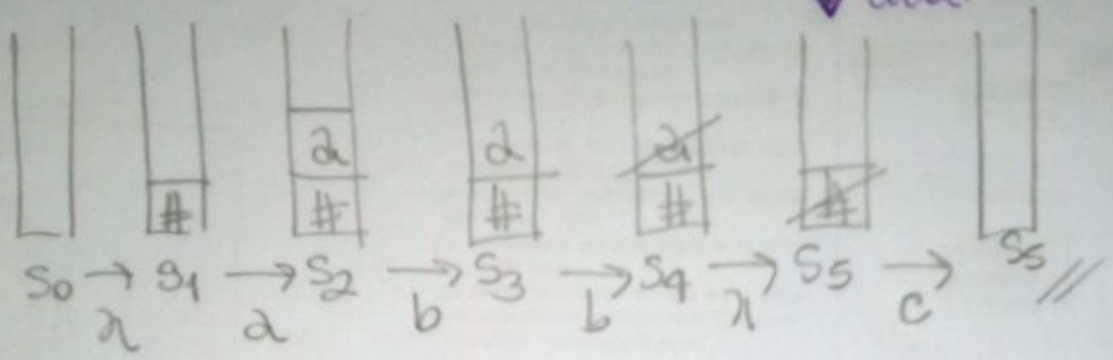

a, λ; a
λ, λ; #
a, λ; a
b, λ; λ
S0
S1
S2
S3
b, λ; λ
c, λ; λ
b, a; λ
λ, #; λ
S4
S5
c, λ; λ

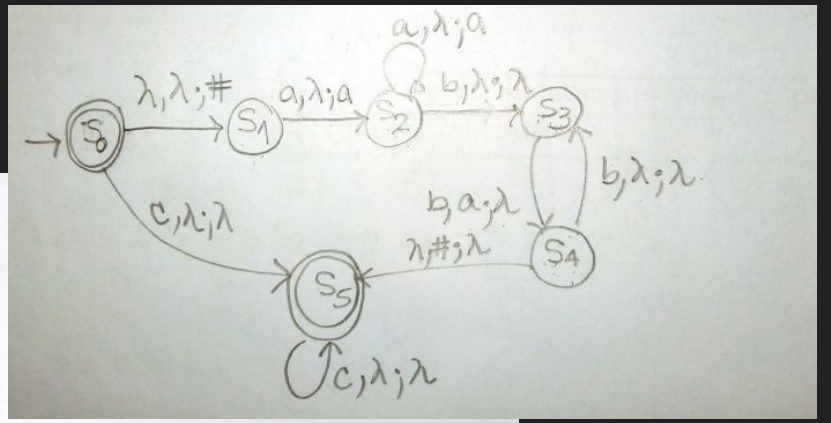
Ej: de palabras aceptadas

$w = abbc$.

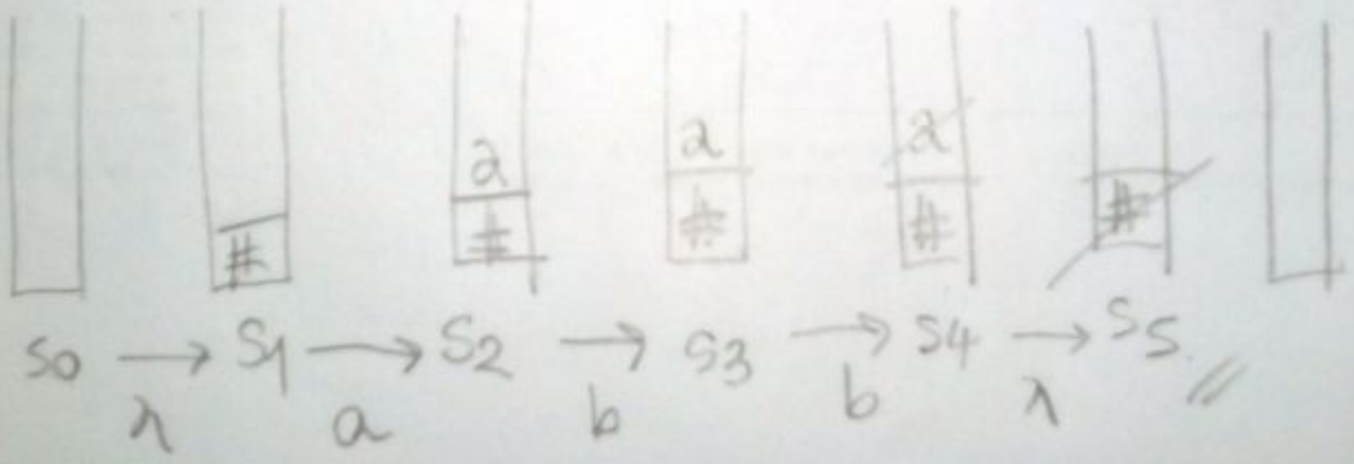


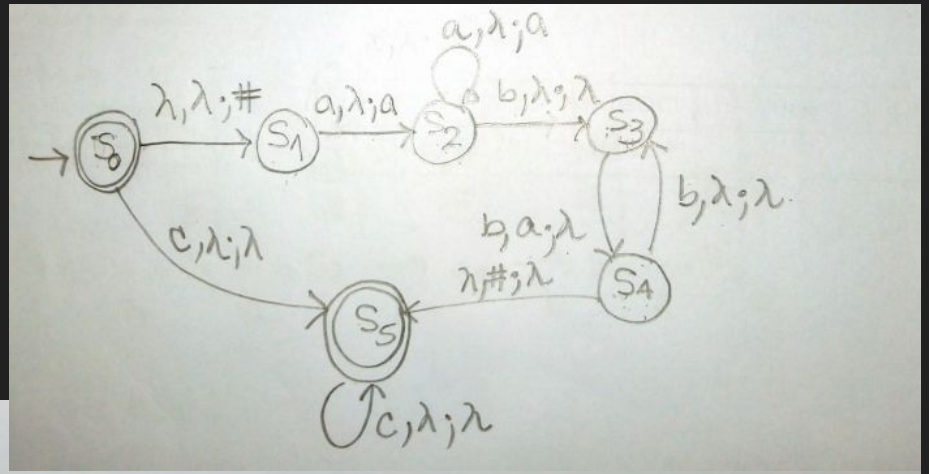
ya la pila lo rechazó aca.



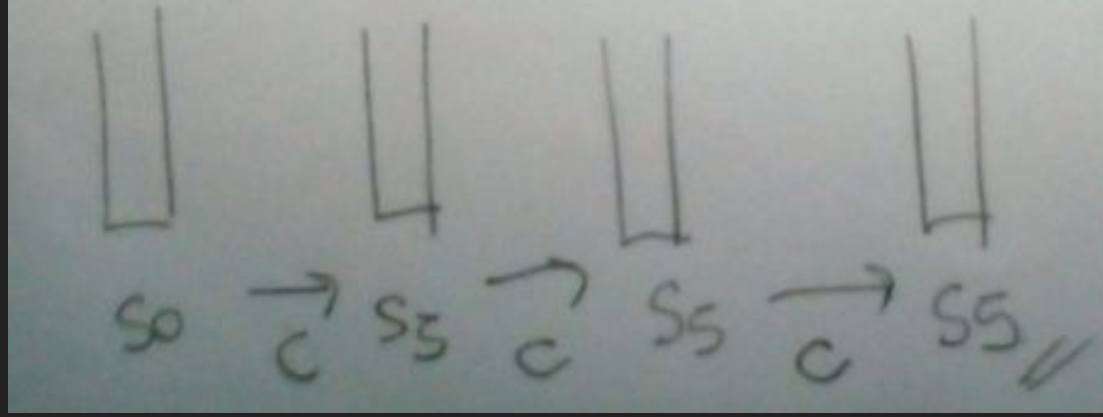


$w = abb$





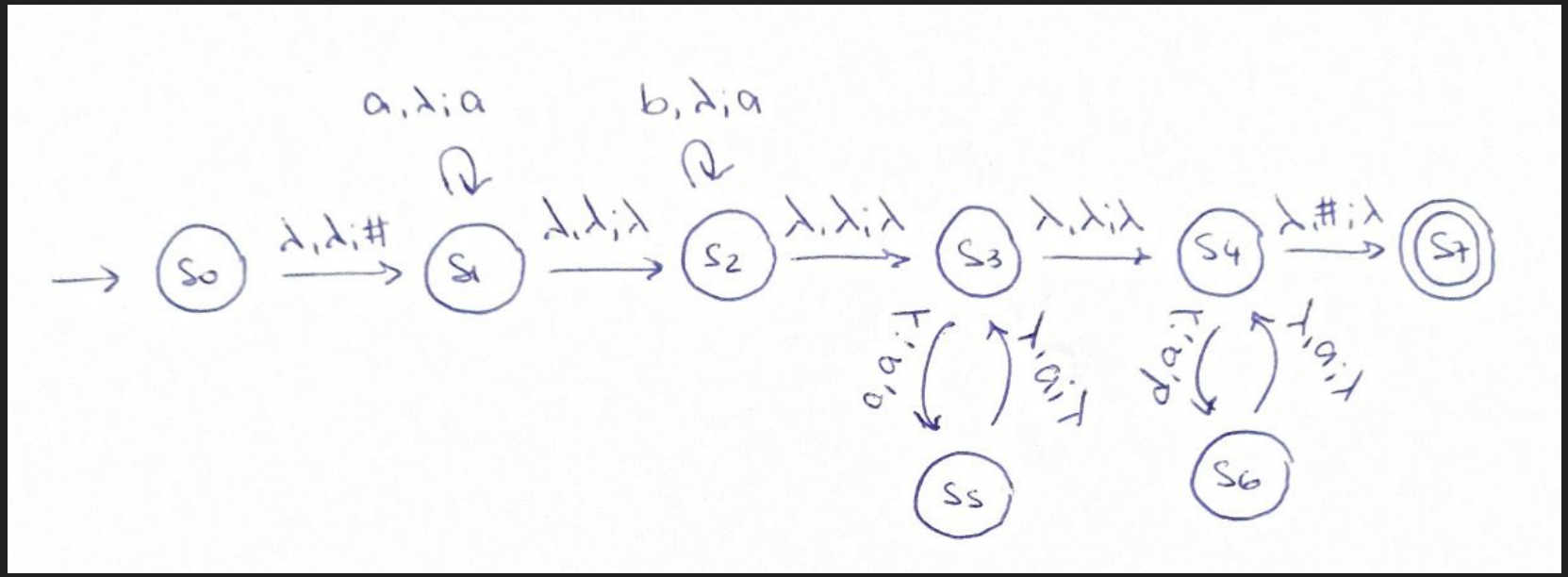
$w = ccc$.



Acá la pila siempre quedó vacía, no la usamos nunca.

Ejercicio 1 - item e)

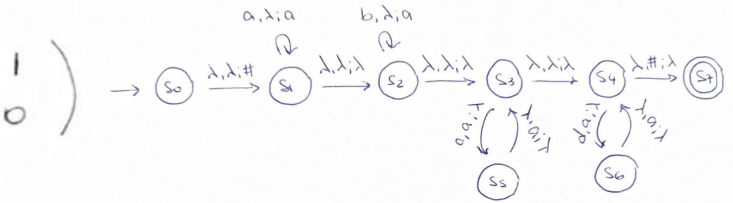
$$e)\ \{a^r b^s c^t d^u \mid r, s, t, u \in \mathbb{N}_0 \ \wedge\ r + s = 2\,(t + u)\}$$



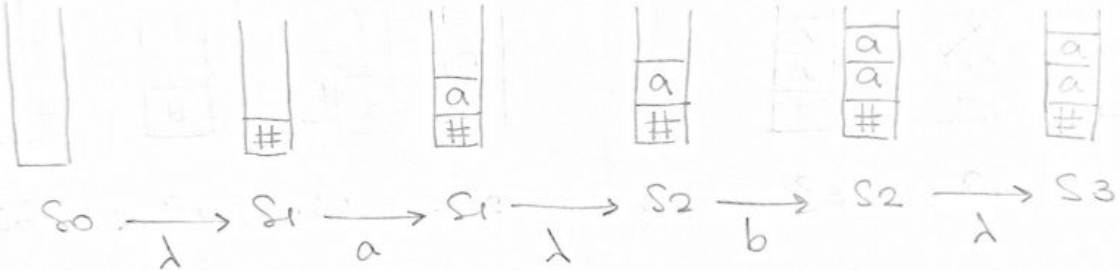
Ejemplo : Reconocimiento de la palabra

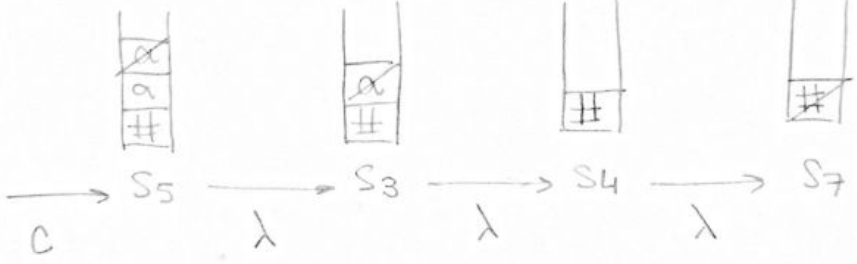
$\omega = abc$

$\begin{pmatrix} r=1 & t=1 \\ s=1 & u=0 \end{pmatrix}$



pila vacía



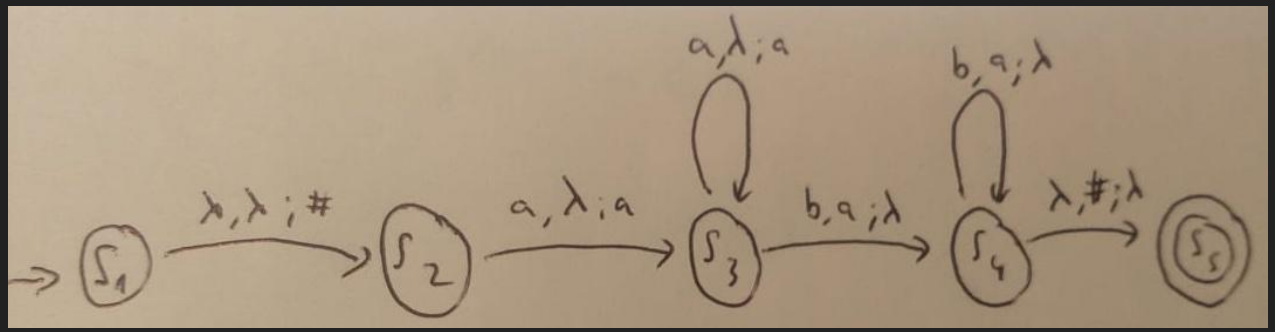


Ejercicio 3:

3. Muestre cómo pueden combinarse dos autómatas de pila M_1 y M_2 para formar un solo autómata que acepte el lenguaje $L(M_1) \cup L(M_2)$.

Podemos intentarlo primero con un ejemplo. Pensemos en el AP que acepta $\mathbf{a^n b^n}$. Y podemos por ejemplo intentar combinarlo con otro AP que acepte $\mathbf{b^n a^n}$.

Así, tenemos el siguiente AP:



Y tenemos también otro análogo. Para combinarlos, si empezamos con **a**, tenemos es una palabra del 1ro, y si empezamos con **b** es una palabra del 2do. Podremos hacer un AP que incluya todos los estados?

Ejercicio 3:

3. Muestre cómo pueden combinarse dos autómatas de pila M_1 y M_2 para formar un solo autómata que acepte el lenguaje $L(M_1) \cup L(M_2)$.

Podemos entonces pensar el caso general.

Tenemos que tener un nuevo estado inicial, y con una transición (λ,λ;λ) ir hacia el estado inicial de uno u otro AP.

Así, si tenemos:

M1 = (S1, A1, V1, T1, i1, F1) y M2 = (S2, A2, V2, T2, i2, F2)

Nos quedaría:

M1 U M2 = (S1 U S2 U {i3}, A1 U A2, V1 U V2,

T1 U T2 U {(i3,λ,λ;i1,λ), (i3,λ,λ;i2,λ)},

i3, F1 U F2)

Ejercicio 4:

4. Muestre cómo puede modificarse un autómata de pila M para que acepte el lenguaje $L(M)^*$.

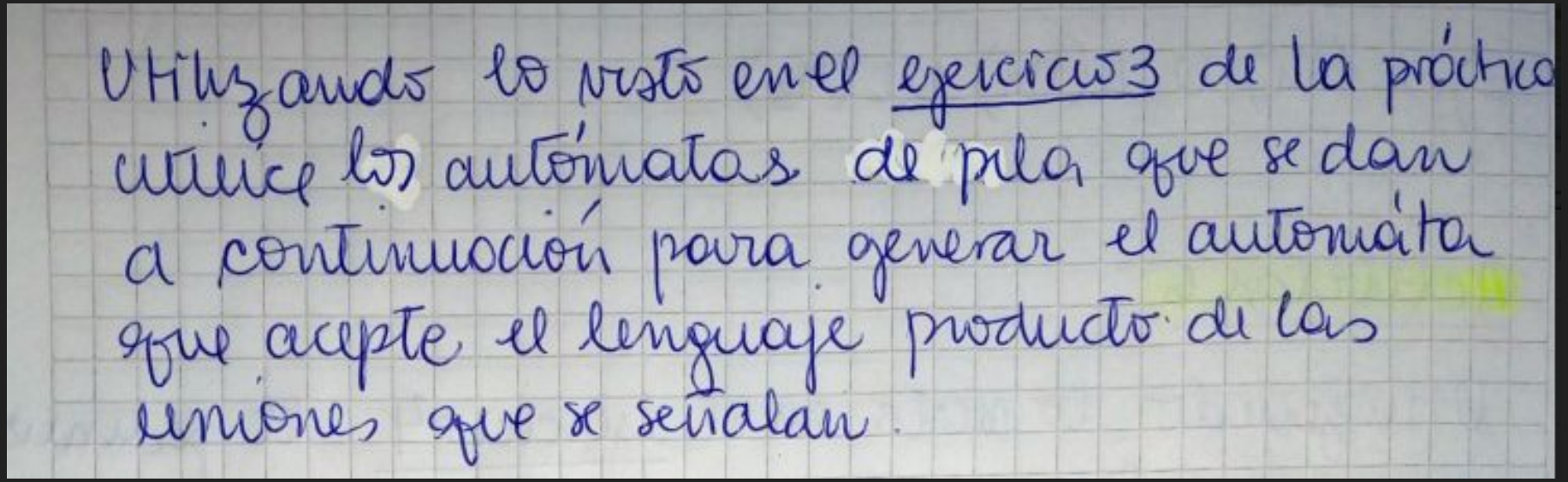
Tomando la idea del ejercicio anterior, pensemos que lo que queremos es construir un AP que acepte todas las cadenas que sean concatenación de 0 o más cadenas aceptadas en el AP original.

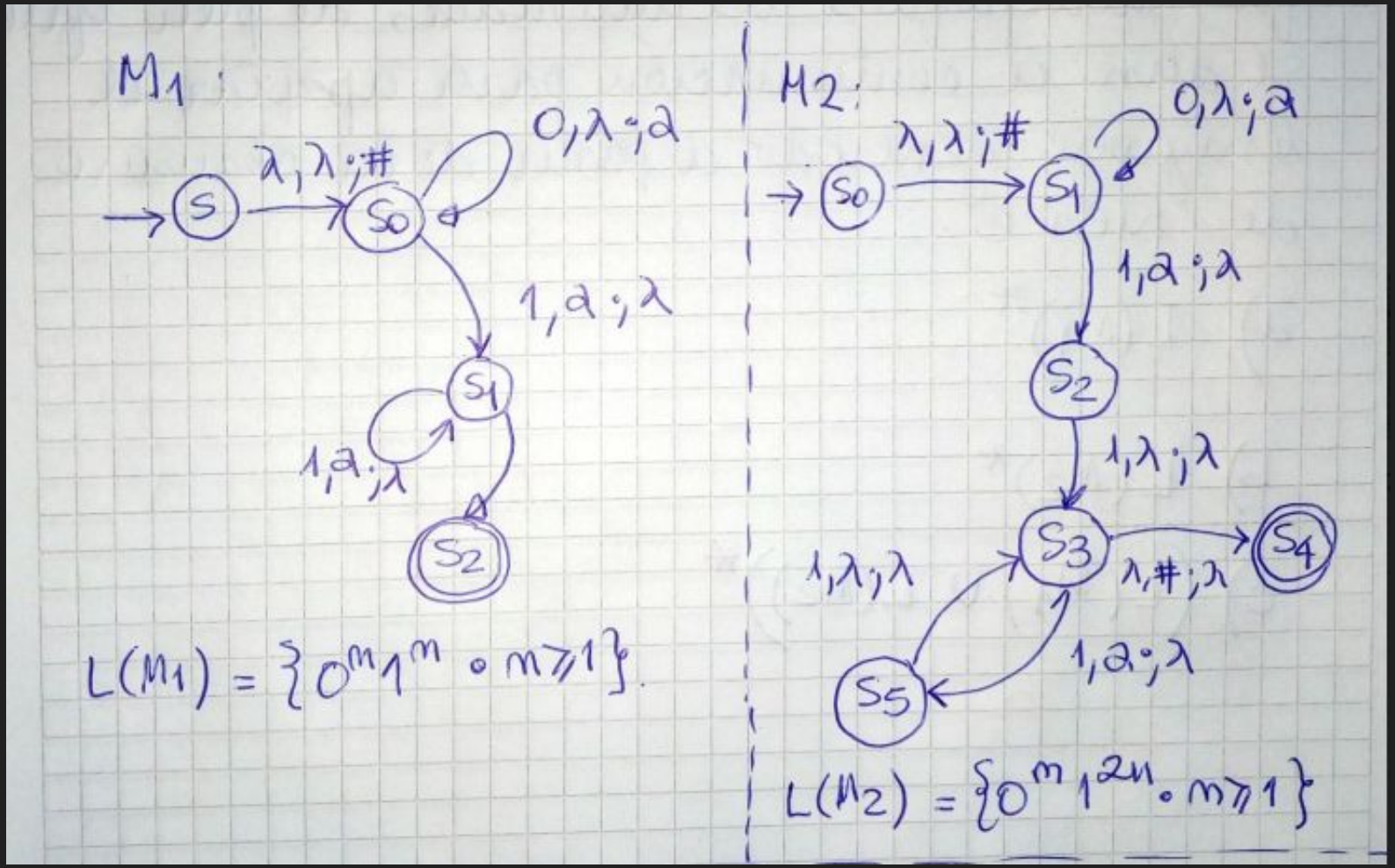
Por ende, tenemos que considerar 2 cuestiones:

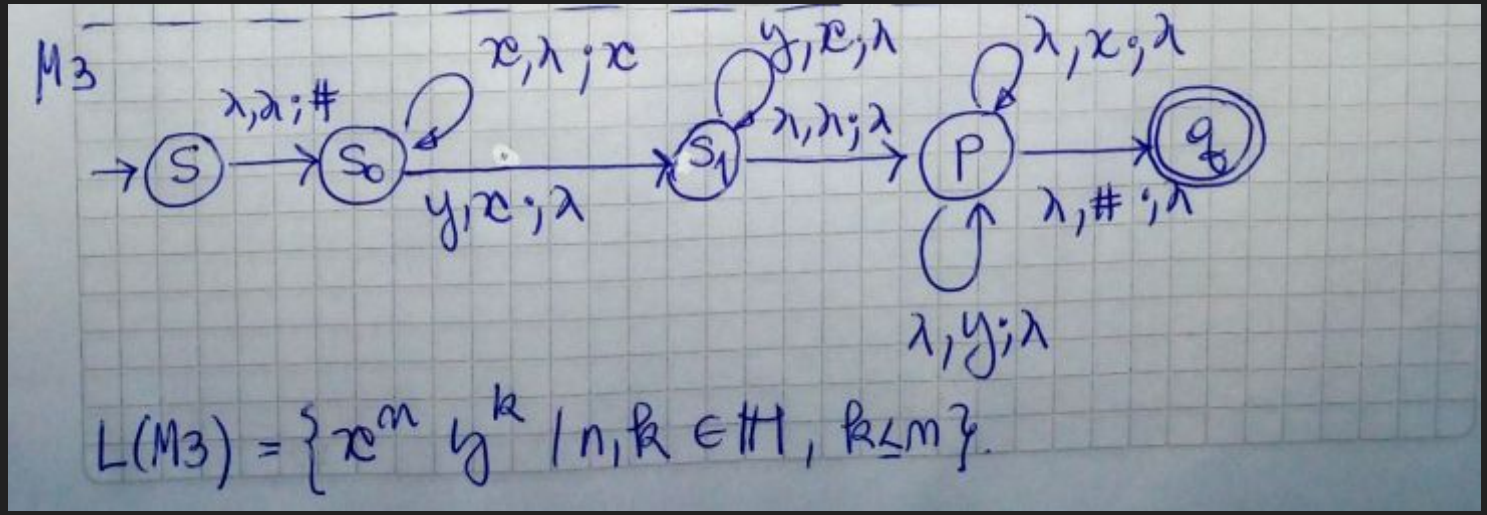
- Para aceptar cadenas que sean concatenación de 0 cadenas aceptadas, estaríamos aceptando la cadena vacía. Ergo, necesitamos que el nuevo estado inicial sea de aceptación.
- Para aceptar cadenas que sean concatenación de más de 1 cadena aceptada, necesitamos volver al estado inicial original después de haber llegado a un estado de aceptación.

Esto no cambia ni el alfabeto, ni los símbolos de pila. Simplemente agregamos ese nuevo estado inicial, que es de aceptación, y una serie de transiciones para lograr movernos por los estados considerando las 2 cuestiones recién mencionadas.

Actividad 5:


Utilizando lo visto en el ejercicio 3 de la práctica
utilice los autómatas de pila que se dan
a continuación para generar el autómata
que acepte el lenguaje producto de las
unions que se señalan.


M1:
s
s0
s1
s2
λ,λ;#
0,λ;a
1,a;λ
1,a;λ
L(M1) = {0^m 1^m ∘ m ≥ 1}.
M2:
s0
s1
s2
s3
s4
s5
λ,λ;#
0,λ;a
1,a;λ
1,λ;λ
1,λ;λ
λ,#;λ
1,a;λ
L(M2) = {0^m 1^2m ∘ m ≥ 1}



a) $L(M_1) \cup L(M_3)$

b) $L(M_1) \cup L(M_2)$

c) $L(M_1) \cup L(M_2) \cup L(M_3)$

Actividad 6:

Utilizando lo visto en el ejercicio 4 te pedimos que modifiques los autómatas de pila que se dan a continuación para aceptar el lenguaje generado a partir de la clausura de Kleene.

a) $L(M_1)^*$

b) $L(M_2)^*$

c) $(L(M_1) \cup L(M_2))^*$

Ejercicio 5:

5. Utilice el *pumping lemma* para lenguajes independientes de contexto[1] para probar que los siguientes lenguajes no pueden ser aceptados por ningún autómata de pila:

Pumping Lema para Independientes del Contexto:

[1]Si L es un lenguaje independiente de contexto que contiene un número infinito de cadenas, entonces debe existir en L una cadena de la forma $xuyvz$, donde x, u, y, v, z son subcadenas, al menos una de u y v es no vacía, y $xu^n yv^n z \in L$ para cada $n \geq 1$.

Recordemos que la demostración se basa en que la cantidad de no terminales es finita, y la parte derecha de cada regla de producción también debe ser finita. Por ende, si asumimos que tenemos una cadena lo suficientemente larga, vamos a estar repitiendo algún no terminal (que nos determina cómo subdividir la cadena en xuyvz). Con esto, si ese no terminal lo repetimos varias veces, vamos a estar repitiendo 2 partes de la cadena asumida (la parte u, y la parte v).

$a)\ L_1 = \{a^n b^n c^n \mid n \in \mathbb{N}_0\}$

Asumimos que puede ser aceptado por un AP. Concluímos que debe existir una gramática IC que lo genere. Ahora tomamos una cadena C_1 con la forma $a^n b^n c^n$ y lo suficientemente larga (si asumimos que m es cantidad de símbolos en la regla más larga, y t es la cantidad de símbolos no terminales, suficientemente larga es mayor que m^t, para asegurar que el árbol de derivación tiene profundidad mayor a t).

Por el Lema de Bombeo, la cadena puede ser subdividida en xuyvz. Sin pérdida de generalidad, asumimos que u es no vacío (si lo fuera, tomamos v que debe no serlo).

$a)\ L_1 = \{a^n b^n c^n \mid n \in \mathbb{N}_0\}$

Tenemos que analizar los siguientes casos:

- Caso 1: En u ocurren todos símbolos iguales (sean todos a, todos b, o todos c). Supongamos que son a. Entonces, al repetir u varias veces, estamos repitiendo a, y generamos una cadena C_2 con la forma $a^{n+k}b^nc^n$, donde k es mayor a 0. Esto no tiene la forma $a^nb^nc^n$.
- Caso 2: En u ocurren símbolos diferentes (sean a y b, o b y c, si fueran los 3 podríamos tomar una cadena más larga). Supongamos que son a y b. Entonces, al repetir u varias veces, estamos repitiendo a y b, y generamos una cadena C_3 con la forma $a^{n-k}(a^kb^i)^+b^{n-i}c^n$. Esto no tiene la forma $a^nb^nc^n$.

En ambos casos, vemos que la gramática generaría cadenas que no tienen la forma $a^nb^nc^n$, por ende concluimos que $a^nb^nc^n$ no puede ser aceptado por un AP.

$$b)\ L_2 = \{a^i b^j c^k \mid i, j, k \in \mathbb{N}_0 \ \wedge \ i < j < k\}$$

Supongamos que tenemos una gramática IC que general L_2, y que tenemos una cadena C_1 de la forma $a^ib^jc^k$ lo suficientemente larga, de tal manera que i es mayor que m^t (con m longitud de regla más larga, t cantidad de no terminales). Como i es mayor que m^t, algún no terminal (llamémosle N_1) se repite en la generación de las a's de la cadena C_1. Por el Lema de Bombeo, la cadena puede ser subdividida en xuyvz de tal manera que tanto u como y como v están todas compuestas solo por solamente a's, y asumimos que u es no vacío.

Si repetimos varias veces el no terminal N_1 vamos a estar generando una cadena con de la forma $a^{i+p}b^jc^k$. Con la suficiente cantidad de repeticiones, i+p > j, y la cadena no tiene la forma $a^ib^jc^k$, por ende concluimos que $a^ib^jc^k$ no es IC.