

Tarea 2: Lenguajes libres de contexto y máquinas de Turing

CC3102 - Teoría de la Computación
Departamento de Ciencias de la Computación
UNIVERSIDAD DE CHILE

Gabriel De La Parra

14 de noviembre de 2015

Índice

1. Problema 1	3
2. Problema 2	6
3. Problema 3	7
4. Problema 4	8
5. Problema 5	9

1. Problema 1

Entregue los diagramas de estados de los autómatas de pila que reconocen los siguientes lenguajes. Adjunte también una descripción intuitiva y de alto nivel de cómo opera el autómata de pila.

(a). $L_1 = \{x_1\#x_2\#\dots\#x_k \mid k \geq 1, \text{ cada } x_i \in \{a, b\}^*, \text{ y existen } i, j \text{ tales que } x_i = x_j^R\}$

Para que un AP pueda reconocer el lenguaje propuesto, debe hacerse uso del no-determinismo. La idea central es que A_1 , el autómata que reconoce L_1 , pueda “elegir” un x_i para guardar en la pila, y un x_j para compararlo. Gracias al no-determinismo, se pueden considerar así todos los x_i .

Para guardar un x_i el autómata puede recorrer $\{a, b, \#\}^*$ elementos. Las transiciones de entrada hacia el estado q_{save} están precedidas por un $\#$, por lo que se puede empezar a guardar solo después de éste símbolo, salvo el caso inicial x_1 , donde no hay $\#$. Posteriormente, se puede avanzar, bajo el mismo principio, hasta encontrar otro grupo x_j y realizar la comparación.

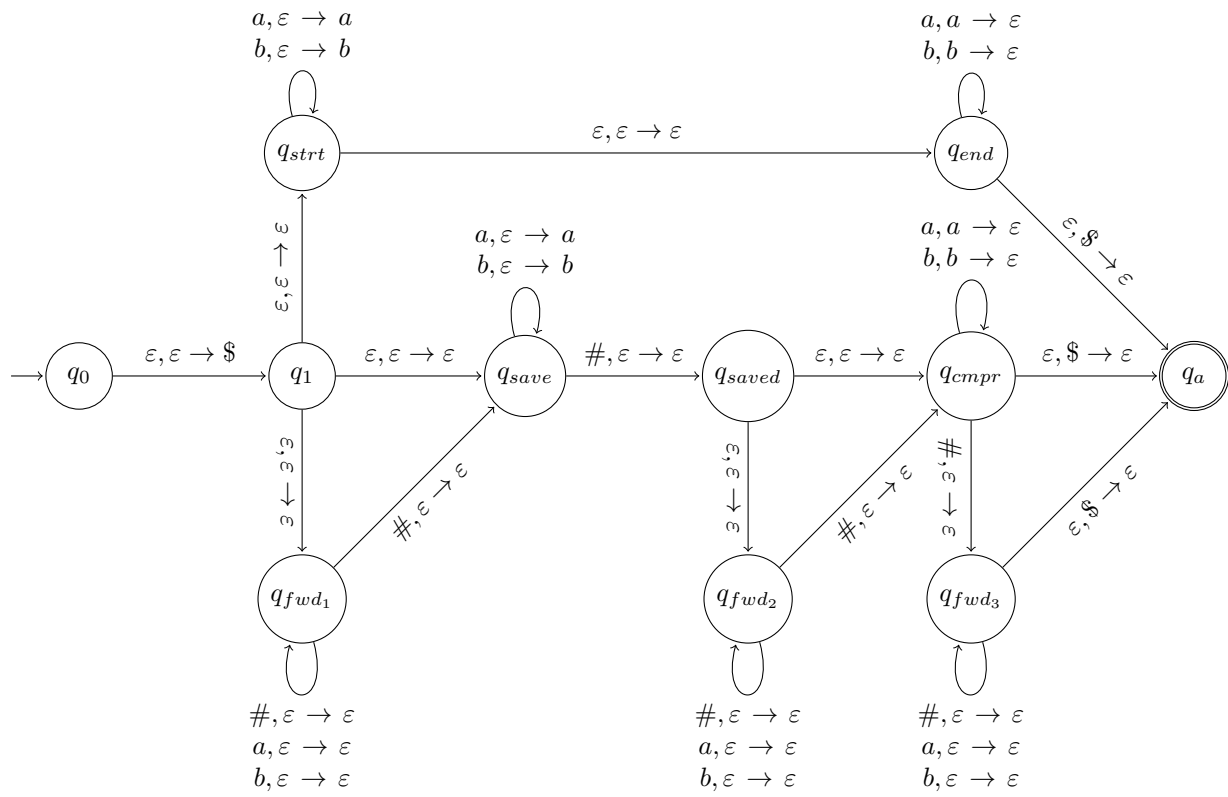
La comparación, que se realiza en q_{cmp} , es positiva cuando se saca todo lo que se metió a la pila de A_1 y queda solo el símbolo $\$,$ insertado al inicio, para denotar el final de la pila. Debe considerarse que, posterior a vaciar la pila, puede ser necesario avanzar hasta el final de la entrada, sobre otros x_k hasta llegar al final.

Se debe considerar adicionalmente el caso donde $k=1$ o $i=j$. En tal caso, no hay símbolo $\#$ y la palabra debe ser palíndromo. En tal caso, se agrega un estado q_{strt} , en el cual se guardan los símbolos iniciales de la palabra y posteriormente se debe sacar en q_{end} . Si la pila queda vacía al final de recorrer la entrada, se acepta. No se considera el caso de la palabra impar, sin embargo ya se ha visto anteriormente, que corresponde a no-determinísticamente saltarse el símbolo central.

Se presenta a continuación una descripción de alto nivel:

- Colocar un símbolo para denotar el inicio de la pila;
- Elegir una palabra x_i ;
- Guardar todos los símbolos de la palabra x_i en la pila;
- Elegir una palabra x_j ;
- Por cada símbolo de x_j sacar el mismo símbolo de la pila;
- Si se llega al final de la palabra y se llega al símbolo de inicio de la pila, acepta, de lo contrario rechaza;

El diagrama de estados para el autómata está dado por la siguiente figura:



(b). $L_2 = \{w \in \{a, b\}^* \mid \|w\|_a = 2 * \|w\|_b\}$, donde $\|w\|_x$ es el número de veces que aparece el símbolo $x \in \{a, b\}$ en w .

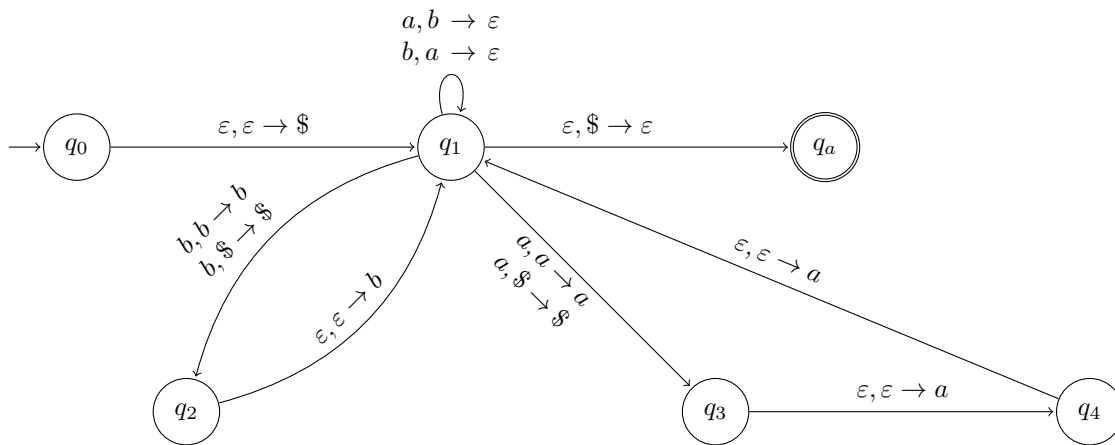
Las palabras dentro de L_2 son todas aquellas que tengan el doble de b 's que a 's. Para encontrar las palabras dentro del lenguaje se utiliza la misma idea del autómata que compara que la cantidad de a 's y b 's sean iguales. Este autómata utiliza la pila para guardar la referencia de comparación de a 's y b 's, es decir, si se han leído n a 's más que b 's, la pila tendrá n a 's. En el caso contrario, el autómata guardará n b 's. El autómata acepta si cuando se procesa toda la cadena, el símbolo de tope (\$) de la pila se procesa.

La adaptación a nuestro problema consiste en que por cada a que se lee, se deben agregar dos a 's, para compensar que sean el doble de b 's. Las condiciones de aceptación son las mismas.

Se presenta a continuación una descripción de alto nivel:

- Colocar un símbolo \$ para denotar el inicio de la pila;
- Si se lee una a ;
- Si al hacer pop, la pila tiene \$, colocar \$ y colocar $2xa$, continuar;
- Si al hacer pop, la pila tiene a , colocar $3xa$, una por la que se hizo pop, y dos más, continuar;
- Si al hacer pop, la pila tiene b , continuar;
- Si se lee una b ;
- Si al hacer pop, la pila tiene \$, colocar \$ y colocar b , continuar;
- Si al hacer pop, la pila tiene b , colocar $2xb$, una por la que se hizo pop, y una más, continuar;
- Si al hacer pop, la pila tiene a , continuar;
- Si se llega al final de la palabra y se llega al símbolo de inicio de la pila, acepta, de lo contrario rechaza;

El diagrama de estados para el autómata está dado por la siguiente figura:



2. Problema 2

Mostrar usando el lema de bombeo que los siguientes lenguajes no son libres de contexto.

(a). L_3 es el conjunto de las cadenas formadas por símbolos a , b y c con misma cantidad de a , b y c . Ejemplo: $abcabc$, $abbacaccab$, $bcaaacbc$, ...

La palabra $a^n b^n c^n$ pertenece al lenguaje. Se ha demostrado en cátedra, que esta palabra al ser bombeada, no pertenece a los lenguajes libres de contexto.

Sea $s = a^p b^p c^p$, donde el largo de la palabra, tal que $|s| = 3 \geq p$.

Se deben considerar las posibles particiones para $uv^i xy^i z$, que satisfagan las condiciones del bombeo:

- $\forall i \geq 0$ para $uv^i xy^i z \in L_3$
- $|vy| > 0$
- $|vxy| \leq p$

Las posibilidades se reducen a que vxy se compone de un solo símbolo o se compone de dos.

- Si v o y se componen de dos símbolos (ab o bc), al ser bombeado, genera estructuras del tipo $abab$ o bcb , las cuales rompen la estructura de la palabra. $\therefore \notin L_3$
- Si v o y se componen de un solo símbolo, basta bombear a 0 o 2 para que se haga una desigualdad entre los dos símbolos bombeados y el no bombeado. $\therefore \notin L_3$

(b). $L_4 = a^n b^n c^m : n \leq m \leq 2n$

La palabra vista anteriormente, $a^n b^n c^n$ también pertenece a este lenguaje. La demostración es la misma que en el caso anterior.

3. Problema 3

(b). Considere la siguiente gramática $G : S \rightarrow XY, X \rightarrow aXb|\varepsilon, Y \rightarrow bYa|\varepsilon$. Describa en forma concisa (en Castellano o con una fórmula corta) el lenguaje generado por la gramática. Convierta la gramática anterior G a la Forma Normal de Chomsky.

El lenguaje genera todas las palabras de la forma:

$$L_G = \{a^n b^{n+m} a^m \mid n, m \geq 0\}$$

La conversión a la forma normal de Chomsky implica que todas las reglas cumplan los siguientes criterios:

- El símbolo inicial puede derivar una sola vez.
- Las variables pueden ir a dos variables o un terminal, es decir $A \rightarrow BC$ o $A \rightarrow a$.

Inicialmente se agrega una nueva regla para la variable inicial S_0 . Si bien este paso se encuentra dentro del procedimiento, la regla S no está en la derivación de ninguna otra regla, por lo que S quedaría como una transición inalcanzable. Se elimina esta derivación adicional.

$$\begin{aligned} S &\rightarrow XY \\ X &\rightarrow aXb|\varepsilon \\ Y &\rightarrow bYa|\varepsilon \end{aligned}$$

El procedimiento de conversión implica reemplazar todas las reglas $\rightarrow \varepsilon$: $Y \rightarrow \varepsilon$ y $X \rightarrow \varepsilon$:

$$\begin{aligned} S &\rightarrow XY|X|Y \\ X &\rightarrow aXb|ab \\ Y &\rightarrow bYa|ba \end{aligned}$$

Luego es necesario reemplazar las reglas que son uno a uno en variable: $S \rightarrow X$, $S \rightarrow Y$:

$$\begin{aligned} S &\rightarrow XY|aXb|ab|bYa|ba \\ X &\rightarrow aXb|ab \\ Y &\rightarrow bYa|ba \end{aligned}$$

Se agregan reglas para convertir a la FNC:

$$\begin{aligned} A &\rightarrow a \\ B &\rightarrow b \end{aligned}$$

Al sustituir, queda:

$$\begin{aligned} S &\rightarrow XY|aXb|bYa|AB|BA \\ X &\rightarrow aXb|AB \\ Y &\rightarrow bYa|BA \\ A &\rightarrow a \\ B &\rightarrow b \end{aligned}$$

Adicionalmente se agregan dos reglas más:

$$\begin{aligned} U &\rightarrow AX \\ V &\rightarrow BY \end{aligned}$$

Finalmente, la gramática queda en la FNC definida por las siguientes reglas:

$$\begin{aligned} S &\rightarrow XY|UB|VA|AB|BA \\ X &\rightarrow UB|AB \\ Y &\rightarrow VA|BA \\ U &\rightarrow AX \\ V &\rightarrow BY \\ A &\rightarrow a \\ B &\rightarrow b \end{aligned}$$

4. Problema 4

Un autómata de cola (AC) se define de manera análoga a un autómata de pila, pero usando una cola. Recuerde que una cola es una estructura de datos con memoria ilimitada donde por un extremo se puede sólo insertar elementos y por el otro extremo sólo poner elementos.

- (a). Entregue una definición formal de un AC, incluyendo qué significa que un AC acepte una palabra.
- (b). Demuestre que un AC es equivalente en poder a una Máquina de Turing.

De manera general, un autómata de cola puede almacenar símbolos de tal forma que el último símbolo que se agrega, es el último que se saca (o alternatively, el primero que se introduce, es el primero que sale).

Esta capacidad, de poder sacar el primer elemento y colocarlo al final, se puede explotar de tal forma que al repetir varias veces, es posible rotar los elementos almacenados en la cola. Al poder rotar sobre sus elementos almacenados, es posible no solo moverse de izquierda a derecha, sino también simular el movimiento de derecha a izquierda, al rotar en $n - 1$ la cola de tamaño n , pudiendo agregar algún símbolo especial para denotar la el inicio y fin de la cola.

Si se considera que se podría almacenar el *input*, le daría también la capacidad de poder modificar su entrada. Adicionalmente, el rotar le permite agregar \sqcup a medida que agrega nuevos símbolos. Más aún, se puede extender a que se agregue un símbolo adicional para demarcar la posición del cabezal. Se podrían incluso implementar movimientos de desplazamiento (shift left/right). De todo lo anterior, la semejanza con una máquina de Turing.

La equivalencia en sentido contrario, es que una máquina de Turing puede mover el cabezal, permitiendo programar que introducir o remover elementos tenga que realizarse en cierto orden LILO o FIFO, como el de una cola. De esta forma, se podría tener la entrada en una parte de la cinta y en otra parte, separadas por algún símbolo especial, la cola. Lo mismo se vió anteriormente cuando se mencionaron las MT multicinta. En tal caso, se tendría una cinta para la entrada y una cinta para la cola.

La definición formal está dada por una 6-tupla $(Q, \Sigma, \Gamma, \delta, q_0, F)$, similar a la de un autómata de pila. La principal diferencia se encuentra en la forma en que se introducen y extraen los elementos.

- Q : Conjunto de estados;
- Σ : Alfabeto de entrada;
- Γ : Alfabeto de la cola;
- δ : Función de transición: En un estado determinado, lee un símbolo de la entrada, saca un símbolo de la cola, agrega un símbolo a la cola y se mueve al siguiente estado;
- q_0 : Estado inicial;
- $F \subseteq Q$: Conjunto de estados finales;

Un autómata de cola puede aceptar una palabra $w \in \Sigma$ si, existe una secuencia de transiciones, que permiten al autómata pasar del estado inicial q_0 a un estado final $\in F$, al leer la palabra completa y tener la cola vacía.

5. Problema 5

Entregue las MTs multicinta (descripción de alto nivel y diagramas de estados) que reconocen los siguientes lenguajes:

$$(a). L_3 = \{a^n b^m c^{\min\{n,m\}} \mid n, m \geq 0\}$$

La máquina multicinta, tiene las siguientes cintas:

I : Cinta de entrada

A : Cinta para símbolos a

B : Cinta para símbolos b

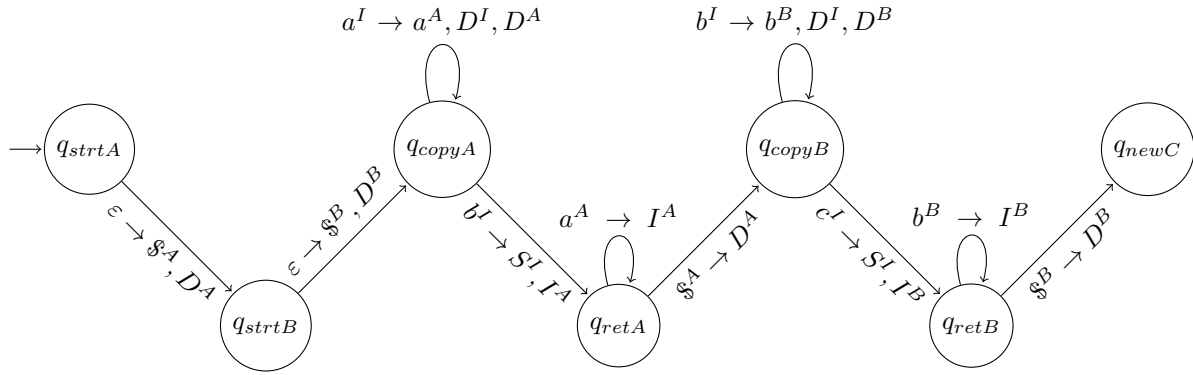
La máquina multicinta que reconoce el lenguaje, realiza de manera intuitiva lo siguiente:

Copia las a a A y b a B . Si en I lee una c , debe leer una a y una b , de lo contrario rechaza. Si lee un \sqcup en I debe leer un \sqcup en A y/o en B , de lo contrario rechaza.

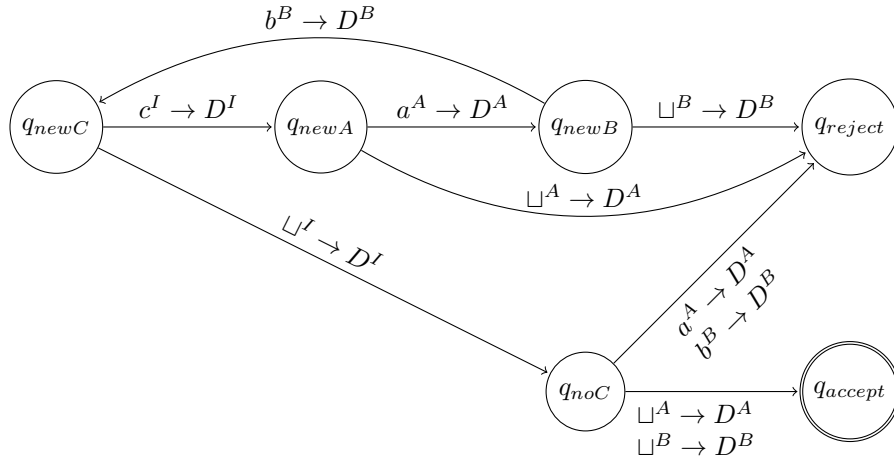
En una descripción de alto nivel, realiza lo siguiente:

- Copia, desde I hasta encontrar un b , todos los símbolos a a A y mueve el cabezal al principio de A , dado por un \$.
- Copia, desde I hasta encontrar un c , todos los símbolos b a B y mueve el cabezal al principio de B , dado por un \$.
- Desde la cinta I para cada c realiza la siguiente acción:
 - Si lee una c en I ,
 - Si lee un \sqcup en A , rechaza;
 - Si lee un \sqcup en B , rechaza;
 - Si lee una a en A , continúa.
 - Si lee una b en B , continúa.
 - Si lee un \sqcup en I
 - Si lee un \sqcup en A , acepta.
 - Si lee un \sqcup en B , acepta.
 - Si lee una a en A , rechaza.
 - Si lee una b en B , rechaza.

En el diagrama de estados, por presentación, se ha separado en 2 diagramas. Un primer diagrama para guardar los valores en las dos cintas A , B .



Se presenta la segunda parte, el diagrama para procesar c . La transición final desde q_{noC} puede llegar a un estado de aceptación y rechazo por el no-determinismo. Si existiera algún conflicto por estas transiciones, habría que modificar el diagrama para que revise en primera instancia los \sqcup y posteriormente los símbolos, como fue descrito en la descripción de alto nivel.



(b). $L_4 = \{x\#y\#z \mid x, y, z \in \{0, 1\}^*, |x| = |y|, z = x \oplus y\}$ donde \oplus es el XOR binario.

La máquina multicinta, tiene las siguientes cintas:

I : Cinta de entrada

X : Cinta para símbolos de x

Y : Cinta para símbolos de y

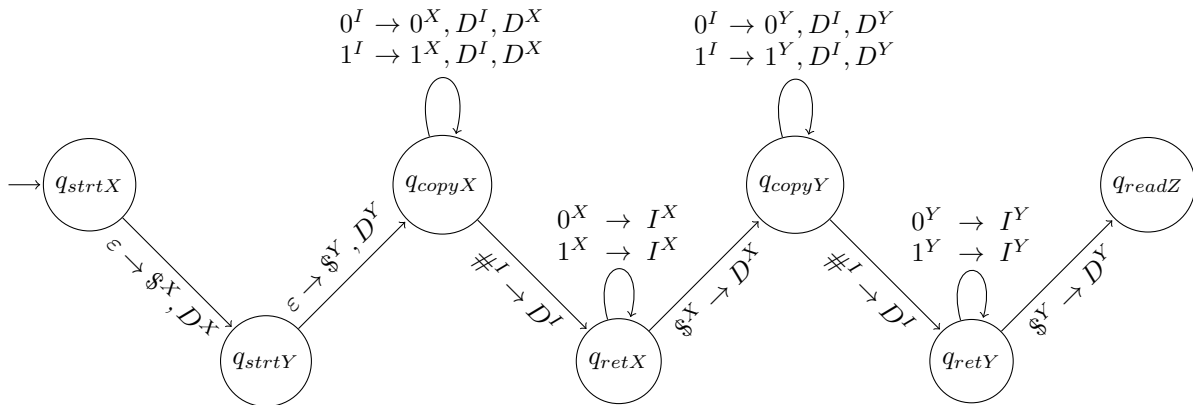
Intuitivamente, la máquina copia la entrada x a X , pasa el primer $\#$, copia y a Y , pasa el segundo $\#$ y finalmente empieza a procesar el contenido de z . Si z_i es 0, x_i e y_i deben ser distintos. Si z_i es 1, x_i e y_i deben ser iguales. Si en algún momento alguna de las cintas X e Y tienen un \sqcup , luego de haber leído un símbolo z_j , se rechaza. Si en cambio, al leer un z_j las otras dos cintas no son vacío, debe rechazar. La máquina acepta cuando se cumplen los criterios anteriores.

En una descripción de alto nivel, la máquina realiza la siguiente tarea:

- Copia, desde I hasta un $\#$, todos los símbolos de x a X y mueve el cabezal al principio de X .
- Copia, desde I hasta un $\#$, todos los símbolos de y a Y y mueve el cabezal al principio de Y .
- Desde la cinta I para cada símbolo en z realiza la siguiente acción:
 - Si lee un $z_i = 0$ en I ,
 - Si lee un \sqcup en X , rechaza.
 - Si lee un \sqcup en Y , rechaza.
 - Si lee un $x_i = 0$ en X ,
 - Si lee un $y_i = 0$ en Y , rechaza.
 - Si lee un $y_i = 1$ en Y , continúa.
 - Si lee un $x_i = 1$ en X ,
 - Si lee un 0 en Y , continúa.
 - Si lee un 1 en Y , rechaza.
 - Si lee un $z_i = 1$ en I ,
 - Si lee un \sqcup en X , rechaza.
 - Si lee un \sqcup en Y , rechaza.
 - Si lee un $x_i = 0$ en X ,
 - Si lee un $y_i = 0$ en Y , continúa.
 - Si lee un $y_i = 1$ en Y , rechaza.
 - Si lee un $x_i = 1$ en X ,
 - Si lee un 0 en Y , rechaza.
 - Si lee un 1 en Y , continúa.
 - Si lee un \sqcup en I ,
 - Si lee un $x_i = 0$ en X , rechaza.
 - Si lee un $x_i = 1$ en X , rechaza.

- Si lee un $y_i = 0$ en Y , rechaza.
- Si lee un $y_i = 1$ en Y , rechaza.
- Si lee un \sqcup en X , continúa.
- Si lee un \sqcup en Y , acepta.

El diagrama de estados, al igual que en (a) tiene una etapa de almacenamiento de los valores en las cintas X e Y .



La etapa de procesamiento para z , por presentación, ha sido modificada para excluir las transiciones de rechazo. Éstas se pueden inferir como todas las transiciones que no estén explícitamente graficadas.

