Universidad Nacional Autónoma de México

Facultad de Ciencias

Automatas y Lenguajes Formales

Tarea 02

Arriaga Santana Estela Monserrat
Castañon Maldonado Carlos Emilio
Fernández Blancas Melissa Lizbeth

1 **Demostrar que el algoritmo para encontrar producciones anulables, que se resume como:**

(a) Si $A \to \varepsilon$, A es anulable.

(b) Si $= 1, ..., k, C_i$ ya han sido encontrados y $A \to C_1C_2...C_k \in R$, entonces A es anulable.

encuentra un símbolo si y sólo si el símbolo es anulable.

Demostración:

- P.D. si el algoritmo encuentra un símbolo, entonces el símbolo es anulable.
  Haremos inducción sobre el número de pasos.
  - Paso base:
    Si el símbolo es encontrado en el primer paso, entonces $A \to \varepsilon$, por lo que $A \Rightarrow \varepsilon$. Por lo tanto, A es anulable.
  - Hipótesis de inducción: Sea $i = 1, ..., k$, supongamos que cada C_i es encontrada en menos de n pasos y que es anulable.
  - Paso inductivo:
    Si el símbolo es encontrado en n pasos, entonces $A \to C_1C_2...C_k$, por lo que $A \Rightarrow C_1C_2...C_k$ y por hipótesis de inducción sabemos que cada C_i se ha encontrado en menos de n pasos y que es anulable ($C_i \Rightarrow^* \varepsilon$). Por lo tanto, $C_1C_2...C_k \Rightarrow^* \varepsilon$ y $A \Rightarrow^* \varepsilon$. Por lo que A es anulable.

  Por lo tanto, si el algoritmo encuentra un símbolo, entonces el símbolo es anulable.

- P.D Si un símbolo es anulable, entonces el algoritmo lo encuentra.
  Haremos inducción sobre el número de producciones.
  - Paso base:
    Si la producción es en un paso $A \Rightarrow \varepsilon$, entonces $A \to \varepsilon \in R$ y es encontrado en el primer paso del algoritmo.
  - Hipótesis de inducción: Supongamos que para toda $i = 1, ..., k$, C_i es anulable y se ha encontrado en menos de n pasos.
  - Paso inductivo:
    Sea A un símbolo anulable en n pasos ($A \Rightarrow^* \varepsilon$) tal que $A \Rightarrow C_1C_2...C_k \Rightarrow^* \varepsilon$, con C_i anulable para toda i. Por hipótesis de inducción sabemos que toda C_i ha sido encontrada por el algoritmo, por lo que $A \to C_1C_2...C_k \in R$. Por lo tanto el algoritmo encuentra a A.

  Por lo tanto, si un símbolo es anulable, entonces es encontrado por el algoritmo.

Por lo tanto, el algoritmo encuentra un símbolo si y sólo si el símbolo es anulable.

2 **Demostrar que dada una gramática G_1 construida eliminando producciones ε a partir de G se tiene que si $A \Rightarrow_G^* w$ en G, entonces $A \Rightarrow_{G_1}^* w$ y $w \neq \varepsilon$ en G_1.**

Demostración:
PD. Dada una gramática G_1 construida eliminando producciones ε a partir de G se tiene que si $A \Rightarrow_G^* w$ en G, entonces $A \Rightarrow_{G_1}^* w$ y $w \neq \varepsilon$ en G_1.

Haremos inducción sobre el número de pasos, por consiguiente supondremos que $A \Rightarrow_G^* w$ y $w \neq \varepsilon$.

- Paso Base:
  En un paso, $A \rightarrow w$ y como $w \neq \varepsilon$, la producción se incluye en G_1.
  Por lo tanto, $A \Rightarrow_G^* w$.
- Hipótesis de Inducción:
  Supongamos que cada Y_i es derivada en menos de n pasos y $Y_i \Rightarrow_{G_1}^* w_i \neq \varepsilon$, para toda $i = 1, ..., k$.
- Paso Inductivo:
  En n pasos, tenemos que $A \Rightarrow G\ Y_1Y_2...Y_m \Rightarrow_G^* w$ y $w = w_1w_2...w_m$.
  Por H.I sabemos que $X_1X_2...X_k$ con $k \leq m$ han sido derivadas en menos de n pasos y cada X_i representa una Y_i sin producciones ε.
  Por la construcción de G_1, tenemos que $A \rightarrow X_1X_2...X_k$ por lo que $A \Rightarrow_{G_1}^* X_1X_2...X_k \Rightarrow_{G_1}^* w$ y $w = w_1w_2...w_m$
  Por lo tanto $A \Rightarrow^* w$.

3 **Demostrar que la gramática G_1 construida a partir de G por el procedimiento:**

(a) Encontrar pares unitarios (A, B) de G.

(b) Para todo par unitario (A, B) sustituir $A \rightarrow \alpha$, donde α es producción no unitaria y $B \rightarrow \alpha$ en G.

Genera una gramática equivalente, esto es: $L(G) = L(G_1)$.

Para demostrar que $L(G) = L(G_1)$, necesitamos probar que cualquier cadena generada por G, también puede ser generada por G_1 y viceversa.

- $L(G) \subseteq L(G_1)$
  Como G_1 está construido a partir de G, al reemplazar los pares unitarios por producciones no unitarias y como cada regla de producción $A \rightarrow_{G_1} w$ existe $A \Rightarrow_G^* w$ que puede obtenerse a través de una secuencia de cero o más producciones unitarias seguidas de una producción no unitaria en G, entonces cualquier cadena generada por G también es generada en G_1.
- $L(G_1) \subseteq L(G)$
  Sea w una cadena generada por G_1, en donde existe una derivación en G_1 de la forma $S \Rightarrow^* w$, donde S es el símbolo inicial de G_1. Probaremos por inducción que por cada derivación de $S \Rightarrow^* w$ en G_1, existe una derivación $S \Rightarrow^* w$ en G.

  Caso Base: Si la derivación tiene solo un paso, entonces está en la forma $S \Rightarrow w$, donde $w \in L(G_1)$. Como no hay pares unitarios en G_1, entonces este paso de derivación también es válido en G. Por lo que el caso base funciona.

  Hipótesis de Inducción: Suponemos que la derivación es posible para la longitud k.

  Paso Inductivo: Consideremos una derivación, $S \Rightarrow^* w$ en G_1 de longitud $k + 1$, esta puede ser escrita de la forma $S \Rightarrow A_1 \Rightarrow A_2 \Rightarrow ... \Rightarrow A_k \Rightarrow w$, donde cada A_i representa un símbolo no terminal.
  Si uno de los pasos $A_i \Rightarrow A_{i+1}$ es una producción unitaria, de acuerdo al paso (b) de la construcción de G_1, reemplazamos $A_i \Rightarrow A_{i+1}$ con $A_i \Rightarrow \alpha$. Como α es una producción no unitaria en G, este paso de derivación es válido en G. Entonces, tenemos que $S \Rightarrow A_1 \Rightarrow ... \Rightarrow A_{i-1} \Rightarrow \alpha \Rightarrow A_{i+1} \Rightarrow k \Rightarrow w$, lo que es una derivación en G

Como $L(G) \subseteq L(G_1)$ y $L(G_1) \subseteq L(G)$, queda demostrado que se genera una gramática equivalente.

4 **Considerar la siguiente gramática $G = (\{0,1\}, \Delta, S, R)$, donde las producciones son:**

$$\begin{aligned} S &\to X_1X_2X_301|1X_2X_301X_4 \\ X_1 &\to Z \\ Z &\to 0|\varepsilon \\ X_2 &\to 1X_2|1 \\ X_3 &\to 0X_2|X_1 \\ X_4 &\to \varepsilon \\ X_5 &\to 10 \end{aligned}$$

Convertir la gramática en forma normal simplificada (eliminar producciones ε, unitarias y símbolos inútiles).
Procedemos a seguir el algoritmo para obtener la forma normal simplificada.

- Eliminamos las producciones ε
  A lo que encontramos por consiguiente los símbolos anulables.
  - Z y X_4 son anulables ya que $Z \to \varepsilon$ y $X_4 \to \varepsilon$
  - X_1 es anulable porque tiene $X_1 \to Z$
  - X_3 es anulable porque tiene $X_3 \to X_1$
  - S es anulable porque tiene $S \to 1X_2X_301X_4$
- Ahora consideramos todos los casos en los que las variables anulables se puedan eliminar

$$\begin{aligned} S &\to X_1X_2X_301|X_2X_301|1X_2X_301|X_1X_201|X_201|1X_201 \\ X_1 &\to Z \\ Z &\to 0 \\ X_2 &\to 1X_2|1 \\ X_3 &\to 0X_2|X_1 \\ X_5 &\to 10 \end{aligned}$$

- Procedemos a encontrar las producciones unitarias
  - $(S,S),(X_1,X_1),(Z,Z),(X_2,X_2),(X_3,X_3)$ y (X_5,X_5)
  - (X_1,Z) es unitaria por la producción $X_1 \to Z$
  - (X_3,X_1) es unitaria por la producción $X_3 \to X_1$
  - (X_3,Z) es unitaria por la producción $X_1 \to Z$
- Procedemos a sustituir las producciones unitarias por las producciones de la variable

$$\begin{aligned} S &\to X_1X_2X_301|X_2X_301|1X_2X_301|X_1X_201|X_201|1X_201 \\ X_1 &\to 0 \\ X_2 &\to 1X_2|1 \\ X_3 &\to 0X_2|0 \\ X_5 &\to 10 \end{aligned}$$

- Eliminamos los símbolos inútiles , por lo que procedemos a encontrar los símbolos generadores.
  - 0 y 1 son generadores
  - X_1, X_2, X_3 son generadores por que $X_1 \to 0$, $X_2 \to 1$ y $X_3 \to 0$
  - S es generador por todas sus producciones
  - X_5 también es generador por la producción 10
- Eliminamos las variables no alcanzables

$$\begin{aligned} S &\to X_1X_2X_301|X_2X_301|1X_2X_301|X_1X_201|X_201|1X_201 \\ X_1 &\to 0 \\ X_2 &\to 1X_2|1 \\ X_3 &\to 0X_2|0 \end{aligned}$$

5 **De la gramática anterior, obtener su forma normal de Chomsky, aplicar el algoritmo de CYK a las siguientes cadenas:**

Haremos su forma normal de Chomsky.

Quitamos los símbolos terminales:

$$\begin{aligned}
S &\to X_1X_2X_3X_1B|X_2X_3X_1B|BX_2X_3X_1B|X_1X_2X_1B|X_2X_1B|BX_2X_1B \\
X_1 &\to 0 \\
B &\to 1 \\
X_2 &\to BX_2|1 \\
X_3 &\to X_1X_2|0
\end{aligned}$$

Hacemos $A \to X_1B$

$$\begin{aligned}
S &\to X_1X_2X_3A|X_2X_3A|BX_2X_3A|X_1X_2A|X_2A|BX_2A \\
X_1 &\to 0 \\
B &\to 1 \\
X_2 &\to BX_2|1 \\
X_3 &\to X_1X_2|0 \\
A &\to X_1B
\end{aligned}$$

Hacemos $C \to X_3A$, $D \to X_1X_2$ y $E \to BX_2$

$$\begin{aligned}
S &\to DC|X_2C|EC|DA|X_2A|EA \\
X_1 &\to 0 \\
B &\to 1 \\
X_2 &\to BX_2|1 \\
X_3 &\to X_1X_2|0 \\
A &\to X_1B \\
C &\to X_3A \\
D &\to X_1X_2 \\
E &\to BX_2
\end{aligned}$$

(a) 011001

S					
Ø	S				
Ø	Ø	S			
X_3, D	Ø	Ø	C		
A, D, X_3	E, X_2	Ø	Ø	A, D, X_3	
X_1, X_3	B, X_2	B, X_2	X_1, X_3	X_1, X_3	B, X_2
0	1	1	0	0	1

(b) 01001

S				
Ø	S			
Ø	Ø	C		
X_3, A, D	Ø	Ø	X_3, A, D	
X_1, X_3	B, X_2	X_1, X_3	X_1, X_3	B, X_2
0	1	0	0	1

(c) 01111101

S, C							
$\emptyset$	S						
X_3, D	$\emptyset$	S					
X_3, D	X_2, E	$\emptyset$	S				
X_3, D	X_2, E	X_2, E	$\emptyset$	S			
X_3, D	X_2, E	X_2, E	X_2, E	$\emptyset$	S		
X_3, A, D	X_2, E	X_2, E	X_2, E	X_2, E	$\emptyset$	X_3, A, D	
X_1, X_3	B, X_2	B, X_2	B, X_2	B, X_2	B, X_2	X_1, X_3	B, X_2
0	1	1	1	1	1	0	1

6 **Transformar la gramática de 4 en su forma normal de Greibach y realizar los árboles para las cadenas:**

Haremos la forma normal de Greibach. Notemos que no tenemos recursiones por la izquierda, por lo que sólo haremos la expansión de todo cuerpo de producciones hasta obtener un símbolo terminal.

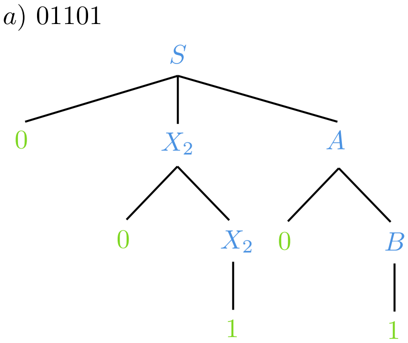
$$\begin{aligned}
S &\to 0X_2C|1X_2C|1C|0X_2A|1X_2A|1A \\
X_1 &\to 0 \\
X_2 &\to 1X_2|1 \\
X_3 &\to 0X_2|0 \\
B &\to 1 \\
A &\to 0B \\
C &\to 0X_2A|0A \\
D &\to 0X_2 \\
E &\to 1X_2
\end{aligned}$$

Eliminamos las producciones inalcanzables que se generaron, obteniendo lo siguiente:

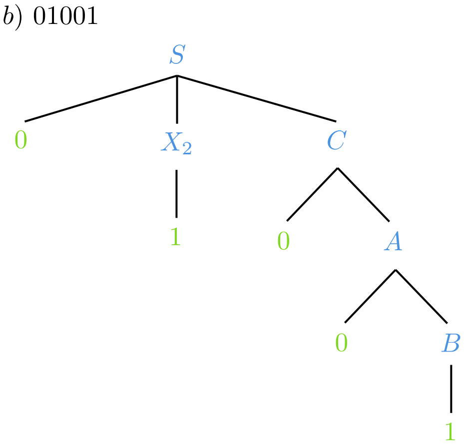
$$\begin{aligned}
S &\to 0X_2C|1X_2C|1C|0X_2A|1X_2A|1A \\
X_2 &\to 1X_2|1 \\
B &\to 1 \\
A &\to 0B \\
C &\to 0X_2A|0A
\end{aligned}$$

Ahora haremos los árboles sintácticos de las cadenas.

$a)$ 01101



$b)$ 01001



7 **Para el siguiente lenguaje con alfabeto $\Sigma = \{0, 1\}$:**

$$L = \{w \in \Sigma^* : w = xcx^R, x \in \Sigma^*, c \in \Sigma \cup \{\varepsilon\}\}$$

(a) Definir la gramática que lo genera.

Sea la gramática $G = (\Sigma, \Delta, S, R)$, donde $\Sigma = \{0, 1\}$, $\Delta = \{S\}$, S es el símbolo inicial y las reglas R están dadas por:

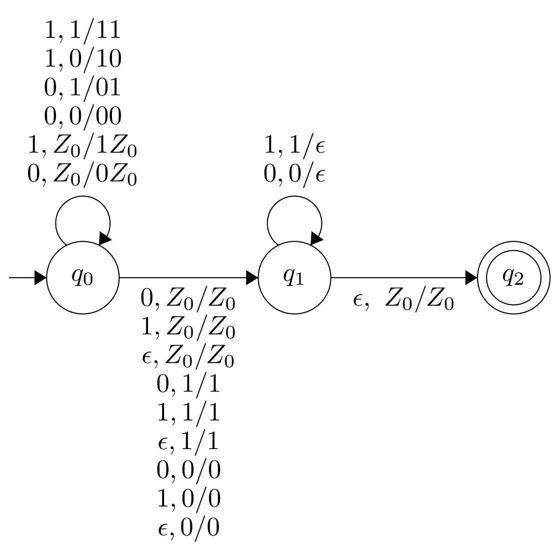
$$S \to 0S0|1S1|1|0|\varepsilon$$

(b) Dar un autómata de pila que acepte el lenguaje.

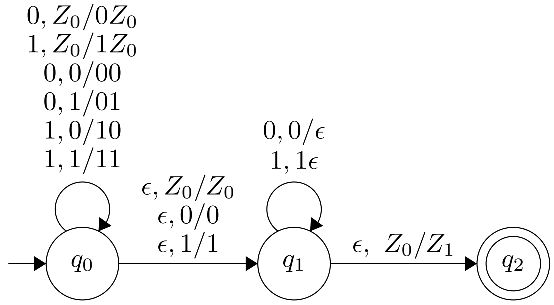
Sea $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ un autómata de pila donde $\{q_0, q_1, q_2\}$, $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, Z_0\}$, q_0 es el estado inicial, Z_0 es el símbolo de inicio de la pila, $F = \{q_2\}$ con función de transición δ dada por:

$$\begin{aligned}
\delta(q_0, 0, Z_0) &= (q_0, 0Z_0) \\
\delta(q_0, 1, Z_0) &= (q_0, 1Z_0) \\
\delta(q_0, 0, 0) &= (q_0, 00) \\
\delta(q_0, 0, 1) &= (q_0, 01) \\
\delta(q_0, 1, 0) &= (q_0, 10) \\
\delta(q_0, 1, 1) &= (q_0, 11) \\
\delta(q_0, \varepsilon, Z_0) &= (q_1, Z_0) \\
\delta(q_0, 0, Z_0) &= (q_1, Z_0) \\
\delta(q_0, 1, Z_0) &= (q_1, Z_0) \\
\delta(q_0, \varepsilon, 1) &= (q_1, 1) \\
\delta(q_0, 0, 1) &= (q_1, 1) \\
\delta(q_0, 1, 1) &= (q_1, 1) \\
\delta(q_0, \varepsilon, 0) &= (q_1, 0) \\
\delta(q_0, 0, 0) &= (q_1, 0) \\
\delta(q_0, 1, 0) &= (q_1, 0) \\
\delta(q_1, 0, 0) &= (q_1, \varepsilon) \\
\delta(q_1, 1, 1) &= (q_1, \varepsilon) \\
\delta(q_1, \varepsilon, Z_0) &= (q_2, Z_0)
\end{aligned}$$

El diagrama del autómata es el siguiente:



8 **Dado el siguiente autómata de pila:**



(a) Describirlo formalmente: estados, alfabeto, símbolos de pila, inicial e inicial de pila, finales.

Sea el autómata de pila $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ con $Q = \{q_0, a_1, a_2\}$, q_0 el estado inicial, $\Sigma = \{0, 1\}$, $\Gamma = \{Z_0, 0, 1\}$, con Z_0 inicial de pila y $F = \{q_2\}$.

(b) Para cada una de las siguientes cadenas, generar si es posible con $\vdash$ que lleve al estado final, explicitando cada una de las configuraciones, indicar cuando no es posible llegar a estado final:

I. 01

$$(q_0, 01, Z_0) \vdash (q_0, 1, 0Z_0)(q_0, \varepsilon, 10Z_0) \vdash (q_1, \varepsilon, 10Z_0)$$

Con esta cadena no podemos llegar al estado final y la pila está llena.

II. 00100

$(q_0, 00100, Z_0) \vdash (q_0, 0100, 0Z_0) \vdash (q_0, 100, 00Z_0) \vdash (q_0, 00, 100Z_0) \vdash (q_0, 0, 0100Z_0) \vdash (q_0, \varepsilon, 00100Z_0) \vdash (q_1, \varepsilon, 00100Z_0)$

Con esta cadena no podemos llegar al estado final y la pila está llena.

III. 010010

$(q_0, 010010, Z_0) \vdash (q_0, 10010, 0Z_0) \vdash (q_0, 0010, 10Z_0) \vdash (q_0, 010, 010Z_0) \vdash (q_1, 010, 010Z_0) \vdash (q_1, 10, 10Z_0) \vdash (q_1, 0, 0Z_0) \vdash (q_1, \varepsilon, Z_0) \vdash (q_2, \varepsilon, Z_0)$

Con esta cadena sí fue posible llegar al estado final.

IV. 101

$(q_0, 101, Z_0) \vdash (q_0, 01, 1Z_0) \vdash (q_0, 1, 01Z_0) \vdash (q_0, \varepsilon, 101Z_0) \vdash (q_1, \varepsilon, 101Z_0)$

Con esta cadena no podemos llegar al estado final y la pila está llena.

V. 1001

$(q_0, 1001, Z_0) \vdash (q_0, 001, 1Z_0) \vdash (q_0, 01, 01Z_0) \vdash (q_1, 01, 01Z_0) \vdash (q_1, 1, 1Z_0) \vdash (q_1, \epsilon, Z_0) \vdash (q_2, \varepsilon, Z_0)$

Con esta cadena sí fue posible llegar al estado final.

9 **Demostrar que si** $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ **es un autómata de pila y** $(q, xw, \alpha) \vdash^* (p, yw, \beta)$**,** $w \in \Sigma^*$**, entonces** $(q, x, \alpha) \vdash (p, y, \beta)$**.**

Haremos la demostración por inducción sobre el número de configuraciones.

- Paso base: Si se da en un paso, $(q, xw, \alpha) \vdash (p, yw, \beta)$, por lo que $x = ay$, con $a \in \Sigma$ tal que $(q, a, \alpha) = (p, \beta)$. Por lo tanto $(q, x, \alpha) \vdash (p, y, \beta)$.

- Hipótesis de inducción: Supongamos que $(q, xw, \alpha) \vdash^* (r, zw, \gamma)$ en n pasos, entonces $(q, x, \alpha) \vdash (r, z, \gamma)$.

- Paso inductivo: P.D que en $n + 1$ pasos se cumple que $(q, xw, \alpha) \vdash^* (p, yw, \beta)$.
  Tenemos que $(q, xw, \alpha) \vdash^* (p, yw, \beta) = (q, xw, \alpha) \vdash^* (r, zw, \gamma) \vdash (p, yw, \beta)$.
  Por hipótesis de inducción sabemos que $(q, xw, \alpha) \vdash^* (r, zw, \gamma)$ en n pasos y que $(q, x, \alpha) \vdash (r, z, \gamma)$.
  Como $(r, zw, \gamma) \vdash (p, yw, \beta)$ entonces hay una transición $\delta(r, a, \gamma) = (p, \beta)$, con $z = ay$, $a \in \Sigma$, por lo que $(r, z, \gamma) \vdash (p, y, \beta)$.
  Como $(q, x, \alpha) \vdash (r, z, \gamma)$ y $(r, z, \gamma) \vdash (p, y, \beta)$, entonces por transitividad$(q, x, \alpha) \vdash (p, y, \beta)$.

Por lo tanto, si $(q, xw, \alpha) \vdash^* (p, yw, \beta)$, $w \in \Sigma^*$, entonces $(q, x, \alpha) \vdash (p, y, \beta)$.

10 **Usando el lema de Ogden (lema de bombeo para gramáticas libres de contexto) demostrar que el siguiente lenguaje no es libre de contexto:**

$$L = \{a^n b^m c^n d^m : n, m \geq 1\}$$

Supongamos que L es un lenguaje libre de contexto, entonces por el lema de Ogden existe una constante n dependiente de L tal que si $z \in L$ con $|z| \geq n$, entonces z puede separarse en 5 subcadenas $z = uvwxy$ tal que $|vwx| \leq n$, $v \neq \varepsilon \neq x$ y para toda $i \geq 0$, $uv^i wx^i y \in L$.

Sea n la constante del lema y $z = a^n b^n c^n d^n \in L$, dividiéndose en $uvwxy$ con $|vwx| \leq n$, $v, x \neq \varepsilon$.

Por el lema de Ogden sabemos que si $i = 0$ y $z = uvwxy \in L$, entonces $uv^0 wx^0 xy \in L$.

Notemos que como $|vwx| \leq n$, entonces vwx sólo puede contener uno o dos símbolos distintos.Por lo tanto tenemos dos casos:

- Si vwx es una cadena que contiene sólo un símbolo, supongamos b, entonces $vwx = b^k$, con $k \leq n$. Entonces al hacer $uv^0 wx^0 y = a^n b^{n-(|v|+|x|)} c^n d^n$. Como $v \neq \varepsilon \neq x$, entonces $|b^{n-(|v|+|x|)}| \neq |d^n|$, por lo tanto, $uv^0 wx^0 y \notin L$.
- Si vwx tiene dos símbolos distintos, supongamos que son b y c, entonces $vwx = b^j c^k$ con $0 < j < n$, $0 < k < n$, $0 < j + k \leq n$, por lo que $uv^0 wx^0 y = a^n b^{n-j} c^{n-k} d^n$ y como $|a^n| \neq |c^{n-k}|$ y $|b^{n-j}| \neq |d^n|$, entonces $uv^0 wx^0 y \notin L$.

Como $z = uvwxy \in L$ y $uv^0 wx^0 y \notin L$, entonces L no es un lenguaje libre de contexto.