PRÁCTICA 3 - MODELOS DE COMPUTACIÓN

RAFAEL CALVO CÓRDOBA

ALBERTO LLAMAS GONZÁLEZ

Recuerda: La solución de un ejercicio incluye la respuesta que se solicita, los pasos seguidos para conseguir esa respuesta y una explicación concisa sobre cómo se ha obtenido la respuesta.

1.- Observando las siguientes gramáticas, indicar qué lenguaje generan y determinar cuáles de ellas son ambiguas. Justificar la respuesta.

- $S \rightarrow AbB$, $A \rightarrow aA \mid \varepsilon$, $B \rightarrow aB \mid bB \mid \varepsilon$
- $S \rightarrow abaS \mid babS \mid baS \mid \varepsilon$
- $S \rightarrow aSA \mid \varepsilon$, $A \rightarrow bA \mid \varepsilon$

Nota: Si la gramática no es ambigua explicarlo con lenguaje natural.

2.- Sea la gramática:

$S \rightarrow A \mid B$
$A \rightarrow aaA \mid \varepsilon$
$B \rightarrow aaaB \mid \varepsilon$

- Demostrar que es ambigua
- Dar una expresión regular para el lenguaje generado.
- Construir una gramática no ambigua que genere el mismo lenguaje

3.- Sea G = (V, Σ, P, {STMT}) la siguiente gramática:

$$\begin{aligned}
\langle \text{STMT} \rangle &\rightarrow \langle \text{ASSIGN} \rangle \mid \langle \text{IF-THEN} \rangle \mid \langle \text{IF-THEN-ELSE} \rangle \\
\langle \text{IF-THEN} \rangle &\rightarrow \texttt{if condition then } \langle \text{STMT} \rangle \\
\langle \text{IF-THEN-ELSE} \rangle &\rightarrow \texttt{if condition then } \langle \text{STMT} \rangle \texttt{ else } \langle \text{STMT} \rangle \\
\langle \text{ASSIGN} \rangle &\rightarrow \texttt{a:=1}
\end{aligned}$$

$$\Sigma = \{\texttt{if}, \texttt{condition}, \texttt{then}, \texttt{else}, \texttt{a:=1}\}.$$
$$V = \{\langle \text{STMT} \rangle, \langle \text{IF-THEN} \rangle, \langle \text{IF-THEN-ELSE} \rangle, \langle \text{ASSIGN} \rangle\}$$

G es una gramática intuitivamente similar a fragmentos de un código de un lenguaje de programación. Sin embargo, G es ambigua:

- Demuestre que G es ambigua
- De una nueva gramática no ambigua para el mismo lenguaje.

4.- Eliminar los símbolos y producciones inútiles de la siguiente gramática (siguiendo el algoritmo visto en clase).

S → aAb | cEB | CE
A → dBE | eeC
B → ff | D
C → gFB | ae
D → h
E → Fa
F → cEa

5.- Elimina las producciones nulas y unitarias de la siguiente gramática (siguiendo el algoritmo visto en clase).

S → Aa | Ba | B
A → Aa | ε
B → aA | BB | ε

OJO: $\varepsilon \in L(G)$

6.- Describe el lenguaje generado por la siguiente gramática G = ({S;A,B,C,D}; {a; b}; P; S), con

S → aAa | bAa | B
A → aAa | bAa | ε
B → bBa | aBa | CD
C → ba | aa
D → B

- Normaliza la gramática G en la Forma Normal de Chomsky (siguiendo todos los algoritmos vistos en clase)

7.- Dar dos autómatas con pila (uno por el criterio de pila vacía y otro por el criterio de estado final) que acepte las cadenas definidas sobre el alfabeto *A* del siguiente lenguaje $L = \{0^i1^j2^k3^m \mid i, j, k \geq 0, m = i + j + k\}$. Mostrar algún ejemplo de uso para aceptar (y rechazar) cadenas siguiendo la notación vista en clase (q, u, α).

8.- Construir un autómata con pila que acepte el lenguaje generado por la siguiente gramática siguiendo el algoritmo visto en clase. Mostrar un ejemplo de uso para aceptar una cadena de al menos cinco letras siguiendo la notación vista en clase.

S → aSb | bY | Ya
Y → bY | aY | ε

Práctica 3 - MC

① Observando las siguientes gramáticas, indicar qué lenguaje generan y determinar cuales de ellas son ambiguas.

- $S \to AbB$, $A \to aA \mid \varepsilon$, $B \to aB \mid bB \mid \varepsilon$

No es ambigua ya que no existe una palabra con dos árboles de derivación distintos

$L = \{ a^i b u \;,\; u \in \{a,b\}^*,\; i \geq 0 \}$

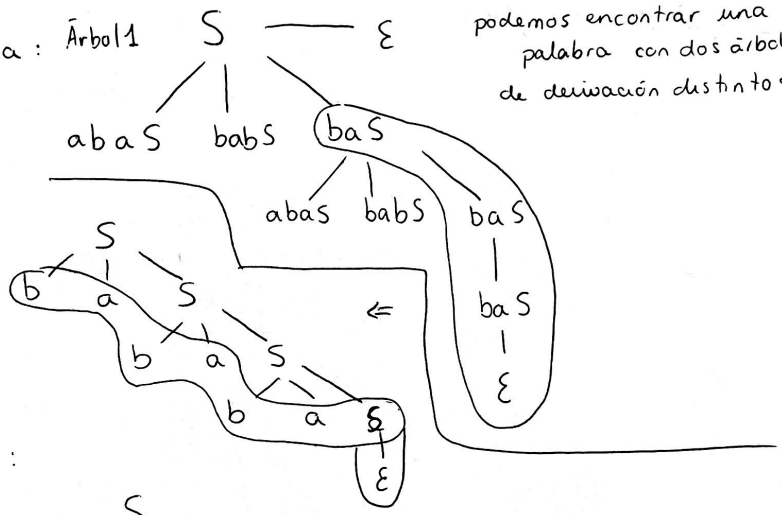
ER: $a^* b (a+b)^*$

- $S \to abaS \mid babS \mid baS \mid \varepsilon$

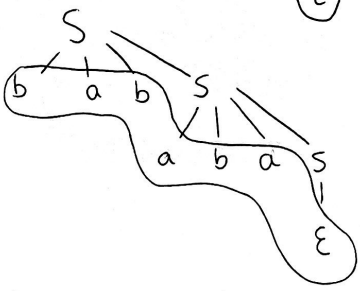
~~$abaS \mid babS \mid baS$, cualquiera~~

Es ambigua ya que podemos encontrar una palabra con dos árboles de derivación distintos.

bababa : Árbol 1



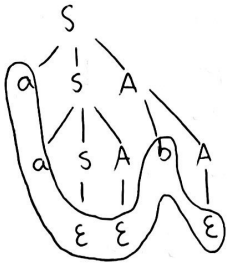
Árbol 2 :



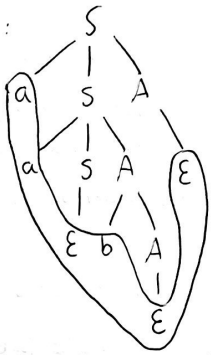
- $S \rightarrow aSA \mid \varepsilon$, $A \rightarrow bA \mid \varepsilon$

Es ambigua ya que podemos encontrar una palabra con dos árboles de derivación distintos

aab : Árbol 1



Árbol 2 :



$L = \{ a^i b^j , i \geq 0 , j \geq 0 \}$

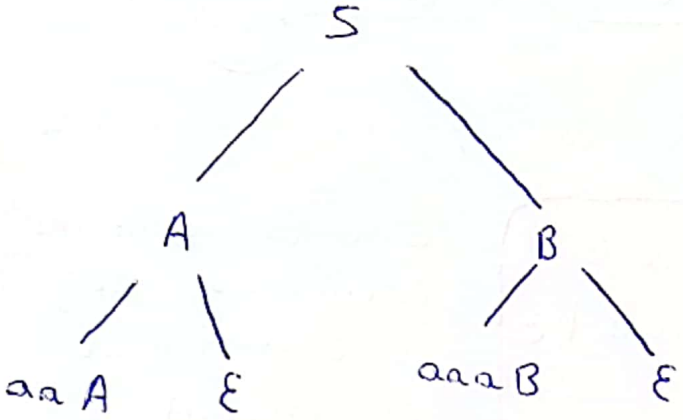
2. SEA LA GRAMÁTICA

$S \to A \mid B$

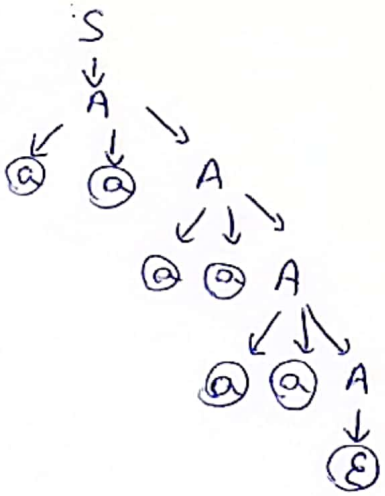
$A \to aaaA \mid \varepsilon$

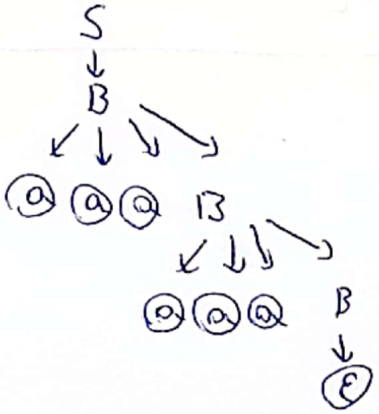
$B \to aaaB \mid \varepsilon$

- DEMOSTRAR QUE ES AMBIGUA



Tomamos una palabra, por ejemplo aaaaaa





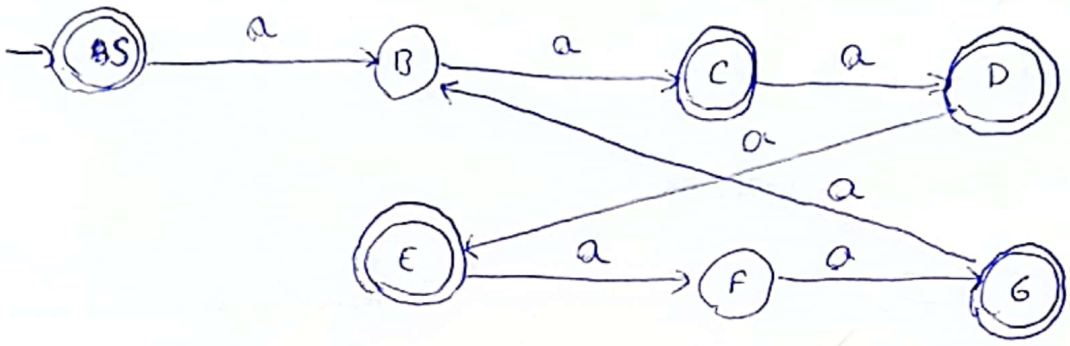
Dos árboles distintos dan lugar a la misma palabra por lo que la gramática es ambigua

- DAR UNA EXPRESIÓN REGULAR PARA EL LENGUAJE GENERADO

ER: $(aa)^* + (aaa)^*$

- CONSTRUIR UNA GRAMÁTICA NO AMBIGUA QUE GENERE EL MISMO LENGUAJE

GENERO UN AFD A PARTIR DE LA EXPRESIÓN REGULAR



A partir del autómata construyo la gramática

$S \rightarrow aB \mid \varepsilon$
$B \rightarrow aC$
$C \rightarrow aD \mid \varepsilon$
$D \rightarrow aE \mid \varepsilon$
$E \rightarrow aF \mid \varepsilon$
$F \rightarrow aG$
$G \rightarrow aB \mid \varepsilon$

COMO LA GRAMÁTICA GENERA UN AFD LA GRAMÁTICA NO ES AMBIGUA

③ Sea $G = (V, \Sigma, P, \{STMT\})$ la siguiente gramática:

$\{STMT\} \rightarrow \{ASSIGN\} \mid \{IF\text{-}THEN\} \mid \{IF\text{-}THEN\text{-}ELSE\}$

$\{IFTHEN\} \rightarrow$ if condition then $\{STMT\}$

$\{IFTHENELSE\} \rightarrow$ if condition then $\{STMT\}$ else $\{STMT\}$

$\{ASSIGN\} \rightarrow a := 1$

$\Sigma = \{$if, condition, then, else, $a := 1\}$

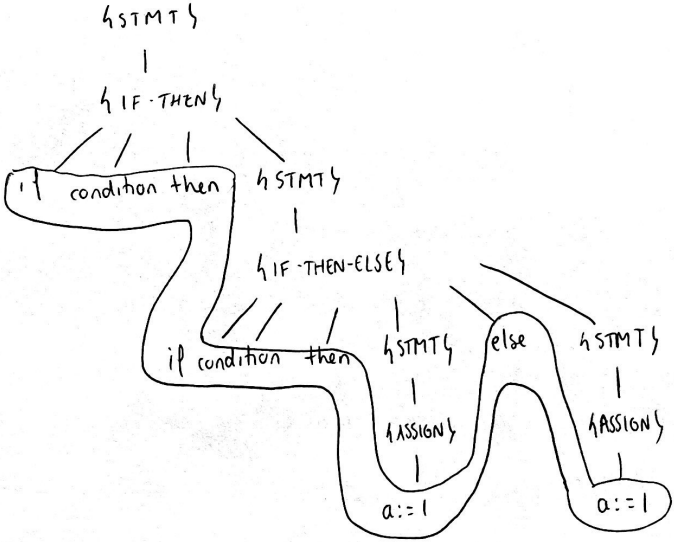
$V = \{\{STMT\}, \{IF\text{-}THEN\}, \{IFTHENELSE\}, \{ASSIGN\}\}$

G es una gramática intuitivamente similar a fragmentos de un código de un lenguaje de programación. Sin embargo, G es ambigua:

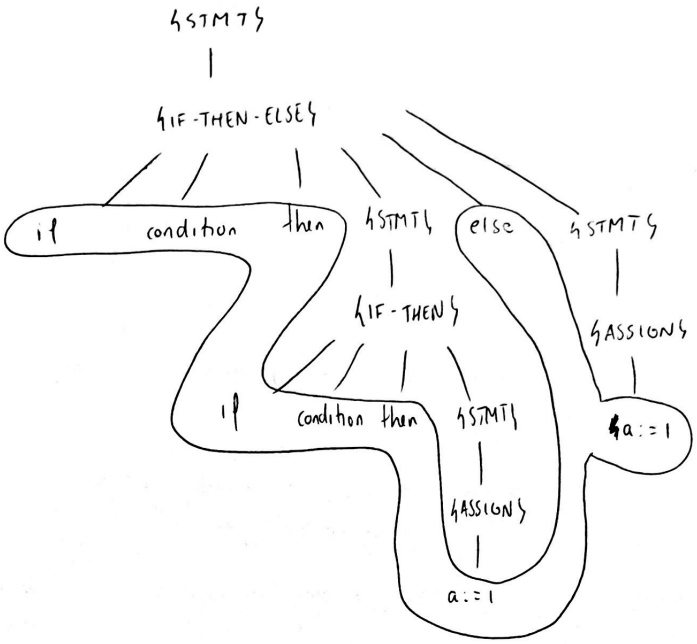
- Demuestre que G es ambigua. Es ambigua si una misma palabra tiene 2 árboles de derivación distintos.

Para la palabra if condition then if condition then $a := 1$ else $a := 1$:

Árbol 1:



Árbol 2:



- De una nueva gramática no ambigua para el mismo lenguaje.

$L_1 = \{a := 1\}$ $L_2 = \{$ if condition then u , $u \in A\}$

$L_3 = \{$if condition then u else v , $u, v \in A\}$ $L = L_1 \cup L_2 \cup L_3$

La gramática no ambigua que hemos sacado ha sido la siguiente:

⟨STMT⟩ → ⟨ASSIGN⟩ | ⟨IF-THEN⟩

⟨ASSIGN⟩ → a:=1

⟨IF-THEN⟩ → if condition then A

A → ⟨ASSIGN⟩ | ⟨IF THEN⟩ | A else A

4: ELIMINAR LOS SÍMBOLOS Y PRODUCCIONES INÚTILES DE LA SIGUIENTE GRAMÁTICA (siguiendo el algoritmo visto en clase).

$S \to aAb \mid \cancel{cB} \mid \cancel{CE}$

$A \to \cancel{dE} \mid eeC$

$B \to ff \mid D$

$C \to \cancel{gFB} \mid ae$

$D \to h$

$\cancel{E} \to fa$

$\cancel{F} \to cEa$

Buscamos los símbolos que alcanzan terminales y eliminamos los que no los alcanzan

$V_t = \{S, A, B, C, D\}$ $\quad V - V_t = \{E, F\}$

Buscamos y eliminamos los símbolos inalcanzables desde S

	J	V_s	T_s
—	S	$\emptyset$	$\emptyset$
Expando S	A	S	a, b
Expando A	C	S, A	a, b, e
Expando C	$\emptyset$ FIN	S, A, C	a, b, e

$V_t - V_s = \{B, D\}$ $\quad T - T_s = \{h, f\}$

$S \to aAb$

$A \to eeC$

~~$B \to ff \mid D$~~

$C \to ae$

~~$D \to h$~~

$S \to aAb$	FIN
$A \to eeC$	
$C \to ae$	

5: Elimina las producciones nulas y unitarias de la siguiente gramática (siguiendo el algoritmo visto en clase).

$S \to Aa \mid Ba \mid B$

$A \to Aa \mid \cancel{\varepsilon}$

$B \to aA \mid BB \mid \cancel{\varepsilon}$

Primero eliminamos las producciones nulas

$H = \{A, B\}$

Como S es anulable, la palabra vacía se puede generar mediante esta gramática

Ahora añadimos producciones

$S \to Aa \mid Ba \mid B \mid a$

$A \to Aa \mid a$

$B \to aA \mid BB \mid a \mid B$

Una vez eliminadas las producciones nulas, eliminamos las unitarias

$S \to Aa \mid Ba \mid \cancel{B} \mid a$

$A \to Aa \mid a$

$B \to aA \mid BB \mid a \mid \cancel{B}$

$H = \{(S, B), (B, B)\}$

Añadimos producciones

$S \to Aa \mid Ba \mid a \mid aA \mid BB$

$A \to Aa \mid a$

$B \to aA \mid BB \mid a$

FIN

⑥ Describe el lenguaje generado por la siguiente gramática

$G = (\{S, A, B, C, D\}, \{a, b\}, P, S)$ con

$S \to aAa \mid bAa \mid B$ $\quad$ $C \to ba \mid aa$

$A \to aAa \mid bAa \mid \varepsilon$ $\quad$ $D \to B$

$B \to bBa \mid aBa \mid CD$

- Normaliza la gramática G en la Forma Normal de Chomsky (siguiendo todos los algoritmos vistos en clase)

Normalizamos la gramática en la Forma Normal de Chomsky.

① Eliminamos los símbolos que no alcanzan terminales

$V_t = \{C, A, S\}$

$V - V_t = \{B, D\}$

$S \to aAa \mid bAa$

$A \to aAa \mid bAa \mid \varepsilon$

$C \to ba \mid aa$

② Eliminamos los símbolos inalcanzables desde S

	J (no explorado)	V_S (explorado)	T_S (símbolos terminales)
-	S	$\emptyset$	$\emptyset$
Expando S	A	S	a, b
Expando A	$\emptyset$	S, A	a, b, ε

$S \to aAa \mid bAa$

$A \to aAa \mid bAa \mid \varepsilon$

Luego $L = \{a^i b a^{(i+j)} : i, j \geq 0\}$

Pasamos ahora a FN CHOMSKY

1. Cogemos los símbolos terminales + variables | varios terminales y los convertimos en solo variables.

2. Más de 3 variable → 2 variables

$C_a \to a$ $C_b \to b$ [redacted] $D \to AC_a$ $E \to$ [redacted] C_bA]

$S \to C_aD \mid EC_a$

$A \to C_aD \mid EC_a \mid \varepsilon$]

Luego ya tendríamos G en la forma Normal de Chomsky

⑦ Dar dos autómatas con pila (uno por el criterio de pila vacía y otro por el criterio de estado final) que acepte cadenas sobre el alfabeto A del lenguaje $L = \{0^i 1^j 2^k 3^m \mid i,j,k \geq 0, m = i+j+k\}$. Mostrar algún ejemplo de uso.

a) Por el criterio de pila vacía.

Del lenguaje podemos sacar la siguiente gramática libre de contexto

$S \to 0\,S\,3 \mid A \mid \varepsilon$

$A \to 1\,A\,3 \mid B \mid \varepsilon$

$B \to 2\,B\,3 \mid \varepsilon$

Luego $Q = \{q\}$ $A = \{0,1,2,3\}$ $B = \{0,1,2,3,S,A,B\}$

$Z_0 = S$ $q_0 = q$

$\delta(q, \varepsilon, S) = \{(q, 0S3), (q, A), (q, \varepsilon)\}$

$\delta(q, \varepsilon, A) = \{(q, 1A3), (q, B), (q, \varepsilon)\}$

$\delta(q, \varepsilon, B) = \{(q, 2B3), (q, \varepsilon)\}$

$\delta(q,0,0) = \{(q,\varepsilon)\}$ $\delta(q,1,1) = \{(q,\varepsilon)\}$ $\delta(q,2,2) = \{(q,\varepsilon)\}$

$\delta(q,3,3) = \{(q,\varepsilon)\}$

Para aceptar la palabra 012333:

$(q, 012333, S) \vdash (q, 012333, 0S3)$ (Leo 0) $\vdash (q, 12333, S3) \vdash$

$\vdash (q, 012333, A)$ X (Seguiré ahora sólo con la posibilidad correcta)

$\vdash (q, 012333, \varepsilon)$ X

$\vdash (q, 12333, A3) \vdash (q, 12333, 1A33)$ (Leo 1) $\vdash (q, 2333, A33) \vdash$

$\vdash (q, 2333, B33) \vdash (q, 2333, 2B333)$ (Leo 2) $\vdash (q, 333, B333) \vdash$

$\vdash (q, 333, 333) \vdash (q, 33, 33) \vdash (q, 3, 3) \vdash (q, \varepsilon, \varepsilon)$

↑ Pila vacía ⇒ Se acepta

b) Por el criterio de estados finales

$L = \{ 0^i 1^j 2^k 3^m \mid i,j,k \geq 0, m = i+j+k \}$

La idea sería hacer que cada estado cuente una variable distinta y que el estado al que llegemos tras un 3 vacíe la pila.

$\delta(q_0, 0, z_0) = \{(q_1, Zz_0)\}$

$\delta(q_0, 1, z_0) = \{(q_2, Yz_0)\}$

$\delta(q_0, 2, z_0) = \{(q_3, Xz_0)\}$

$\delta(q_1, 0, z) = \{(q_1, Zz)\}$

$\delta(q_1, 1, z) = \{(q_2, Yz)\}$

$\delta(q_1, 2, z) = \{(q_3, Xz)\}$

$\delta(q_1, 3, z) = \{(q_4, \varepsilon)\}$

$\delta(q_2, 1, Y) = \{(q_2, YY)\}$

$\delta(q_2, 2, Y) = \{(q_3, XY)\}$

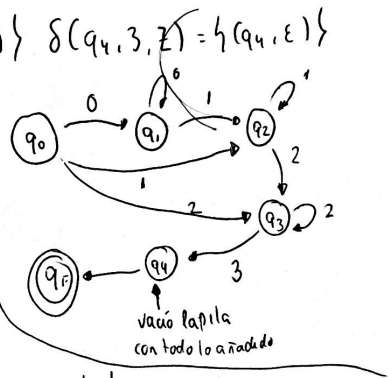
$\delta(q_2, 3, Y) = \{(q_4, \varepsilon)\}$

$\delta(q_3, 2, X) = \{(q_3, XX)\}$

$\delta(q_3, 3, X) = \{(q_4, \varepsilon)\}$

$\delta(q_4, 3, X) = \{(q_4, \varepsilon)\}$ $\delta(q_4, 3, Y) = \{(q_4, \varepsilon)\}$ $\delta(q_4, 3, Z) = \{(q_4, \varepsilon)\}$

$\delta(q_4, \varepsilon, z_0) = \{(q_F, \varepsilon)\}$



Para aceptar la palabra 0 1 2 3 3 3

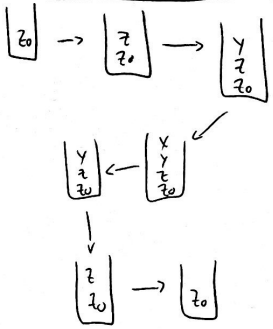
$(q_0, 012333, z_0) \vdash (q_1, 12333, Zz_0) \vdash$

$\vdash (q_2, 2333, YZz_0) \vdash (q_3, 333, XYZz_0) \vdash$

$\vdash (q_4, 33, YZz_0) \vdash (q_4, 3, Zz_0) \vdash$

$\vdash (q_4, \varepsilon, z_0) \vdash (q_F, \varepsilon, \varepsilon)$

↑ Acepta por estado final q_F



8 = CONSTRUIR UN AUTÓMATA CON PILA QUE ACEPTE EL LENGUAJE GENERADO POR LA SIGUIENTE GRAMÁTICA SIGUIENDO EL ALGORITMO VISTO EN CLASE. MOSTRAR UN EJEMPLO DE USO PARA ACEPTAR UNA CADENA DE AL MENOS CINCO LETRAS SIGUIENDO LA NOTACIÓN VISTA EN CLASE.

$S \to aSb \mid bY \mid Ya$

$Y \to bY \mid aY \mid \varepsilon$

$Q = \{q\}$ $\quad B = \{a, b, S, Y\}$

$A = \{a, b\}$ $\quad q_0 = q$ $\quad z_0 = S$

$\delta(q, \varepsilon, S) = \{(q, aSb), (q, bY), (q, Ya)\}$

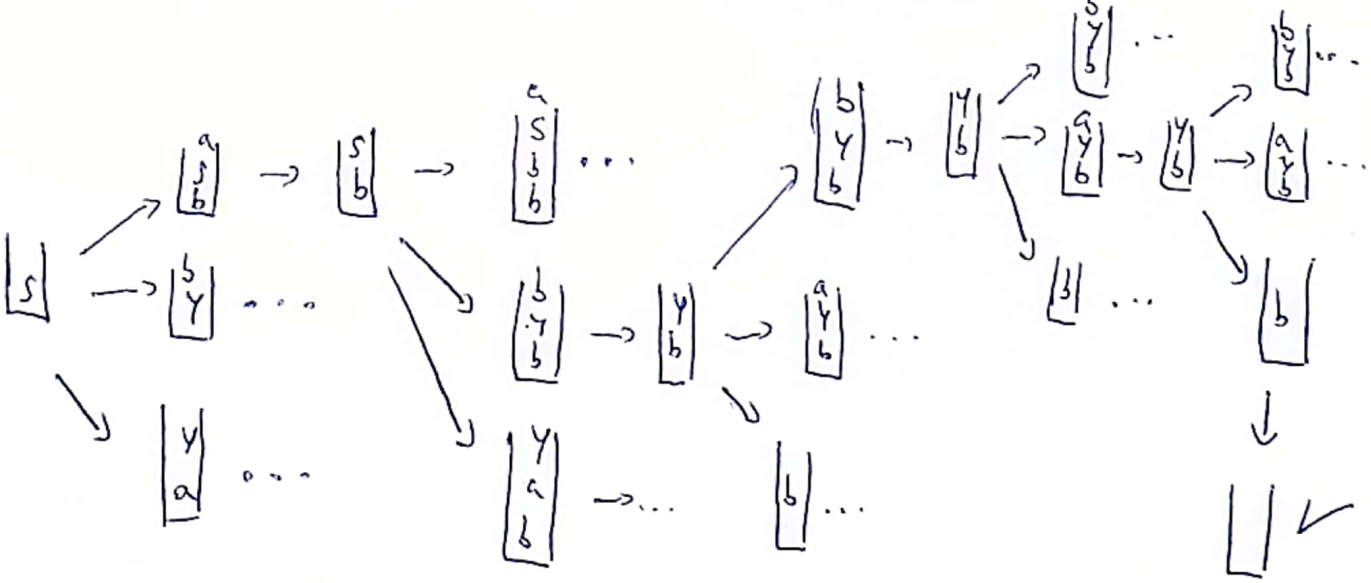
$\delta(q, S, Y) = \{(q, bY), (q, aY), (q, \varepsilon)\}$

$\delta(q, a, a) = \{(q, \varepsilon)\}$

$\delta(q, b, b) = \{(q, \varepsilon)\}$

VEMOS UN EJEMPLO DE ACEPTACIÓN

$aSb \to abYb \to abbYb \to \underline{abbab}$



Aceptada por el criterio de la pila vacía

aaabb

$(q, aaabb, S) \vdash (q, aaabb, aSb) \vdash (q, aabb, SSb) \vdash$

$\vdash (q, aabb, aSSb) \vdash (q, abb, SSb) \vdash (q, abb, aYSb) \vdash$

$\vdash (q, bb, YSb) \vdash (q, bb, Sb) \vdash (q, b, b) \vdash (q, \varepsilon)$

Se acepta