

# Teoría de Autómatas y Lenguajes Formales

## Ejercicios de Lenguajes y Gramáticas

### 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. Crear una gramática que genere los siguientes lenguajes:

- a)  $\{ a, aa, aaa \}$
- b)  $\{ a, aa, aaa, aaaa, aaaaa, \dots \}$
- c)  $\{ \lambda, a, aa, aaa \}$
- d)  $\{ \lambda, a, aa, aaa, aaaa, aaaaa, \dots \}$

La notación empleada para representar cada uno de los lenguajes será:

- a)  $\{ a^n \mid n \in [1, 3] \}$
- b)  $\{ a^n \mid n > 0 \}$
- c)  $\{ a^n \mid n \in [0, 3] \}$
- d)  $\{ a^n \mid n \geq 0 \}$

2. Dadas las gramáticas  $G=(\Sigma_T, \Sigma_{NT}, S, P_i)$  donde:

$G_1$	$G_2$	$G_3$	$G_4$	$G_5$
$\Sigma_T = \{c\}$	$\Sigma_T = \{c,d\}$	$\Sigma_T = \{c\}$	$\Sigma_T = \{c,d\}$	$\Sigma_T = \{c,d\}$
$\Sigma_{NT} = \{S, A\}$	$\Sigma_{NT} = \{S, A\}$	$\Sigma_{NT} = \{S, A\}$	$\Sigma_{NT} = \{S, A, T\}$	$\Sigma_{NT} = \{S, A\}$
$P_1: S \rightarrow \lambda \mid A$ $A \rightarrow AA \mid c$	$P_2: S \rightarrow \lambda \mid A$ $A \rightarrow cAd \mid cd$	$P_3: S \rightarrow \lambda \mid A$ $A \rightarrow AcA \mid c$	$P_4: S \rightarrow cA$ $A \rightarrow d \mid cA \mid Td$ $T \rightarrow Td \mid d$	$P_5: S \rightarrow \lambda \mid A$ $A \rightarrow Ad \mid cA \mid c \mid d$

Determinar el lenguaje asociado a dichas gramáticas.

3. Determinar el tipo de las siguientes gramáticas en la jerarquía de Chomsky, justificándolo:

- a)  $G=(\{a,b\}, \{A,B,S\}, S, P)$ ,  
 $P=\{S::=aA, A::=bB, A::=aA, A::=a, B::=\lambda\}$
- b)  $G=(\{a,b,c\}, \{A,B,C,S\}, S, P)$ ,  
 $P=\{S::=aAb, S::=Ba, S::=\lambda, aAbC::=aAbB, aAbC::=aabC, BCc::=AaCc, BCc::=BaAbc, C::=Ca, C::=a\}$
- c)  $G=(\{casa, jardín, gato\}, \{S, CASERON, BOSQUE, TIGRE\}, S, P)$ ,  
 $P=\{S::=TIGRE\ jardín, S::=BOSQUE\ CASERON, BOSQUE::=\lambda, jardín\ CASERON\ TIGRE\ casa::=jardín\ BOSQUE\ TIGRE\ casa, gato\ CASERON\ BOSQUE::=gato\ BOSQUE\ casa\ TIGRE\ BOSQUE, BOSQUE::=TIGRE\ casa, BOSQUE::=jardín\}$
- d)  $G=(\{x,y\}, \{C,A,B,S\}, S, P)$ ,  
 $P=\{S::=Cx, S::=Cy, S::=By, S::=Ax, S::=x, S::=y, A::=Ax, A::=Cx, A::=x, B::=By, B::=yA, C::=xA\}$
- e)  $G=(\{a,b,c\}, \{S,B\}, S, P)$ ,  
 $P=\{S::=abc, S::=aBSc, Ba::=aB, Bb::=bb\}$

4. Dada la gramática G, se pide:

$G = (\{a, b, c\}, \{S, A, B\}, S, P)$ ,  $P = \{S ::= \lambda, S ::= aAc, A ::= aA, A ::= Ac, A ::= B, B ::= b, B ::= Bb\}$

- Especificar el tipo de G en la jerarquía de Chomsky, razonadamente.
- Determinar el lenguaje L generado por la gramática G.
- Construir 2 árboles de derivación para una misma palabra perteneciente a  $L(G)$ .
- Comprobar si las siguientes formas sentenciales son válidas en G, y en caso afirmativo establecer una cadena de derivaciones que permite llegar a cada una de ellas.
  - aaAcc
  - ac
  - ababBcc
  - abbccc

5. Obtener una gramática de tipo 0 para el lenguaje  $L = \{a^n b^n c^n / n \geq 1\}$ .

6. Realizar las transformaciones necesarias del proceso de limpieza de gramáticas, para obtener una gramática limpia G' equivalente a la G dada.

$G = (\{a, b, c, d\}, \{X, Y, Z, O, P, Q, A\}, Z, P)$ ,  
 $P = \{Z ::= Z, Q ::= OP, X ::= aa, Z ::= aX, Y ::= aa, Z ::= Ya, O ::= b, Z ::= aaa, P ::= QO, Q ::= d, P ::= c, O ::= PQ\}$

7. Dada la gramática G LI, obtener una G' LD equivalente.

$G = (\{0, 1\}, \{A, S\}, S, P)$   
 $P = \{S ::= 1 \mid A1; A ::= S0\}$

8. Dada la gramática G:

$G = (\{e, f, g, z, a, b, d\}, \{Y, X, E, A, D, I, G\}, A, P)$ ,  
 $P = \{ \begin{array}{l} A ::= a \\ E ::= b \\ A ::= azb \\ A ::= aX \\ E ::= E \\ G ::= g \\ X ::= XE \\ D ::= eI \\ X ::= z \\ Y ::= b \\ I ::= fG \\ X ::= Xb \\ E ::= d \end{array} \}$

- Transformar a FNC, explicando cada paso realizado.
- Determinar si las palabras 'abz' y 'azdbb' pertenecen al lenguaje generado por G. En caso afirmativo, generar un árbol de derivación para dicha palabra. En caso negativo, justificar la no pertenencia.

9. Obtener una gramática en FNC equivalente a la siguiente:

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

$$P = \{ S ::= AaB \mid Cbb \mid B$$

$$A ::= Aa \mid cD$$

$$B ::= a \mid Ba \mid \lambda$$

$$C ::= Sa \mid a \mid abB$$

$$D ::= aaA$$

$$E ::= aa \}$$

10. Sea la gramática  $G$ , hallar su FNG equivalente.

$$G = (\{a, b\}, \{S\}, S, P), \text{ donde } P = \{S ::= aSb \mid SS \mid \lambda\}$$

## SOLUCIONES

### 1. Crear una gramática que genere los siguientes lenguajes: (T4p1E1)

- a)  $\{ a, aa, aaa \}$
- b)  $\{ a, aa, aaa, aaaa, aaaaa, \dots \}$
- c)  $\{ \lambda, a, aa, aaa \}$
- d)  $\{ \lambda, a, aa, aaa, aaaa, aaaaa, \dots \}$

La notación empleada para representar cada uno de los lenguajes será:

- a)  $\{ a^n \mid n \in [1, 3] \}$
- b)  $\{ a^n \mid n > 0 \}$
- c)  $\{ a^n \mid n \in [0, 3] \}$
- d)  $\{ a^n \mid n \geq 0 \}$

Solución:

- a) La gramática sólo debe generar tres palabras y se puede admitir que todas ellas están formadas por el símbolo “a”. Por tanto, una posible solución es:

$G = (\{a\}, \{S\}, S, P)$  donde:

$P = \{S ::= a \mid aa \mid aaa\}$

- b) En este caso, las palabras del lenguaje están formadas por una o varias *aes*. Por tanto, una posible solución es:

$G = (\{a\}, \{S, A\}, S, P)$  donde:

$P = \{S ::= A$

$A ::= a \mid aA\}$

o simplemente,

$G = (\{a\}, \{S\}, S, P)$  donde:

$P = \{S ::= a \mid aS\}$

- c) Este caso es similar al planteado en el primer apartado pero, a diferencia de éste, la palabra vacía debe pertenecer al lenguaje. Por tanto, una posible solución es:

$G = (\{a\}, \{S\}, S, P)$  donde:

$P = \{S ::= \lambda \mid a \mid aa \mid aaa\}$

- d) Este caso es similar al planteado en el segundo apartado pero, a diferencia de éste, la palabra vacía debe pertenecer al lenguaje. Por tanto, una posible solución es:

$G = (\{a\}, \{S, A\}, S, P)$  donde:

$P = \{S ::= \lambda \mid A$

$A ::= a \mid aA\}$

o simplemente,

$G = (\{a\}, \{S\}, S, P)$  donde:

$P = \{S ::= \lambda \mid aS\}$

**2. Dadas las gramáticas  $G=(\Sigma_T, \Sigma_{NT}, S, P_i)$  donde: (T4p1E2)**

$G_1$	$G_2$	$G_3$	$G_4$	$G_5$
$\Sigma_T = \{c\}$ $\Sigma_{NT} = \{S, A\}$	$\Sigma_T = \{c,d\}$ $\Sigma_{NT} = \{S, A\}$	$\Sigma_T = \{c\}$ $\Sigma_{NT} = \{S, A\}$	$\Sigma_T = \{c,d\}$ $\Sigma_{NT} = \{S, A, T\}$	$\Sigma_T = \{c,d\}$ $\Sigma_{NT} = \{S, A\}$
$P_1: S \rightarrow \lambda \mid A$ $A \rightarrow AA \mid c$	$P_2: S \rightarrow \lambda \mid A$ $A \rightarrow cAd \mid cd$	$P_3: S \rightarrow \lambda \mid A$ $A \rightarrow AcA \mid c$	$P_4: S \rightarrow cA$ $A \rightarrow d \mid cA \mid Td$ $T \rightarrow Td \mid d$	$P_5: S \rightarrow \lambda \mid A$ $A \rightarrow Ad \mid cA \mid c \mid d$

**Determinar el lenguaje asociado a dichas gramáticas.**

Solución:

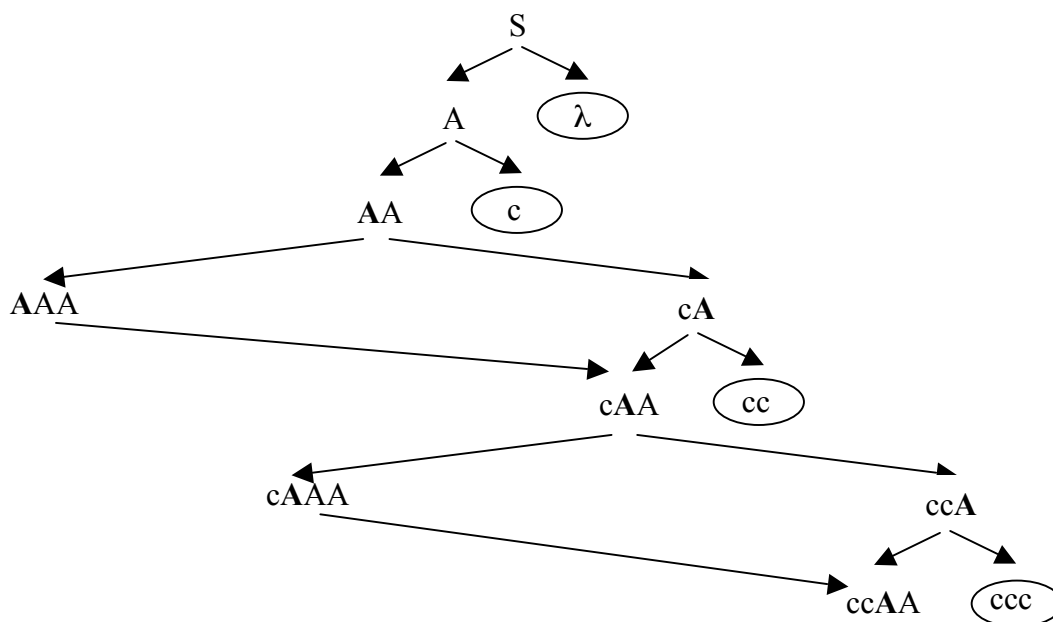
El lenguaje asociado a una gramática se corresponde con el conjunto de todas las sentencias de la gramática:

$$L(G) = \{x / S^* \rightarrow x \text{ AND } x \in \Sigma^*\}$$

Una forma de encontrar las sentencias (o palabras) del lenguaje asociado a una gramática es apoyarse en una estructura en forma de árbol en la que la raíz es el axioma y las hojas las palabras del lenguaje. Para construir este árbol, hay que tener presente que sus nodos, en general, se corresponden con palabras temporales que derivan en otras palabras temporales. El paso de una a otra se consigue aplicando producciones.

En gramáticas de tipo 2 ó 3 esta transformación se puede simplificar asumiendo que las nuevas palabras temporales se obtienen a partir de su progenitor derivando el símbolo NT situado más a la izquierda.

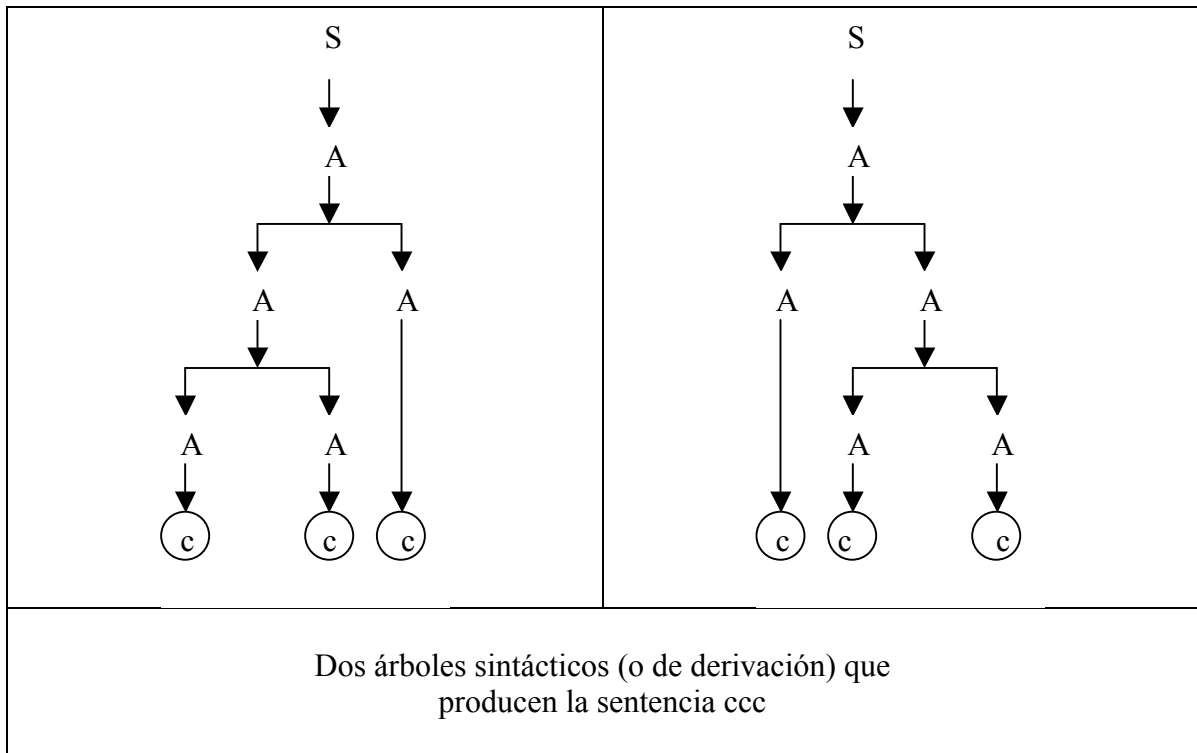
- a)  $S ::= \lambda \mid A$   
 $A ::= AA \mid c$



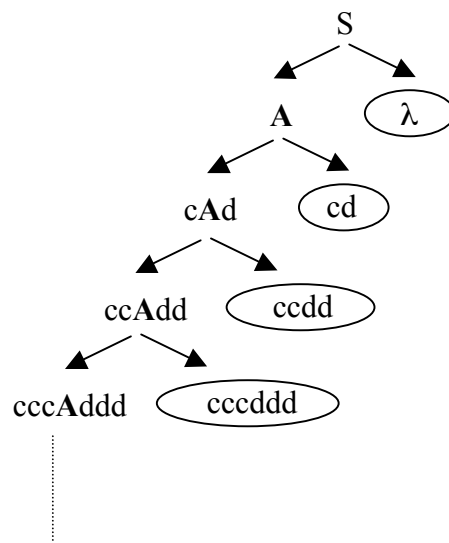
$$L(G_1) = \{ \lambda, c, cc, ccc, \dots \} = \{ \lambda, c^n \} \text{ con } n=1, 2, 3, \dots$$

Determinadas palabras se pueden encontrar por más de un camino:

**La Gramática es ambigua.**



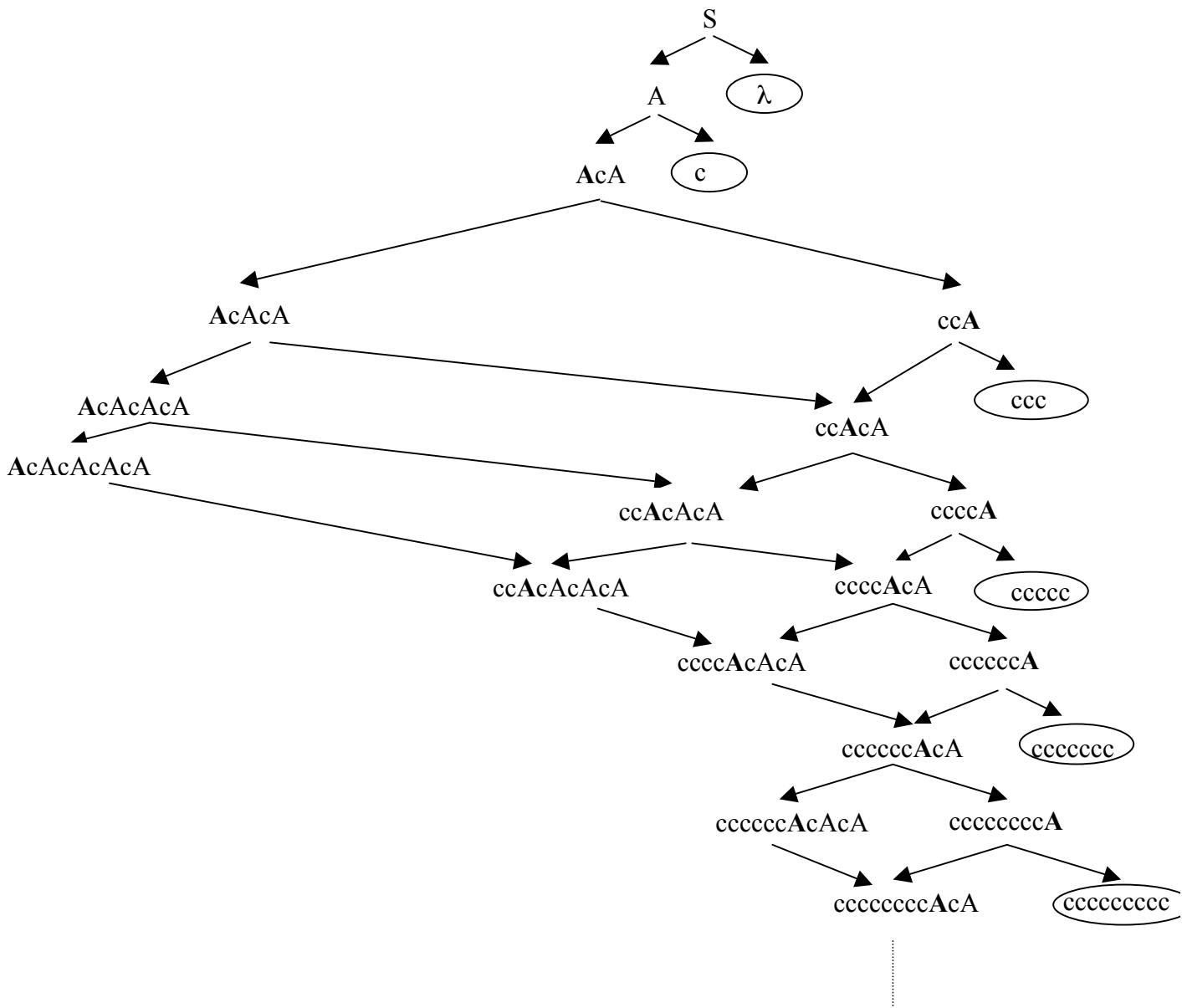
- b)  $S ::= \lambda \mid A$   
 $A ::= cAd \mid cd$



$$L(G_2) = \{ \lambda, cd, ccdd, cccddd, \dots \} = \{ \lambda, c^n d^n \} \text{ con } n=1, 2, 3, \dots$$

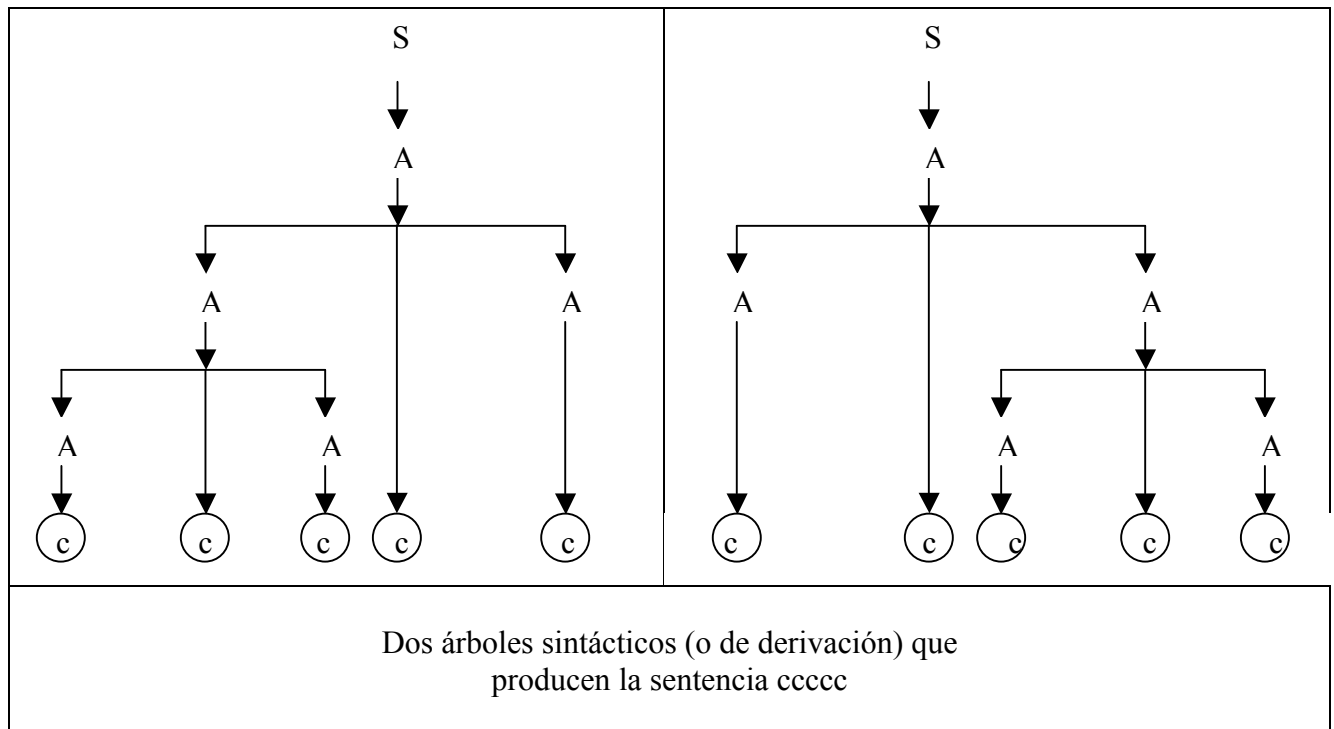


- c)  $S ::= \lambda \mid A$   
 $A ::= AcA \mid c$



$$L(G_3) = \{ \lambda, c, ccc, ccccc, ccccccc, \dots \} = \{ \lambda, c^n \mid n=1, 3, 5, \dots \} = \\ = \{ \lambda, c^{2n+1} \mid n=0, 1, 2, 3, \dots \}$$

Determinadas palabras se pueden encontrar por más de un camino:  
**La Gramática es ambigua.**



- d)  $S ::= cA$   
 $A ::= d \mid cA \mid Td$   
 $T ::= Td \mid d$

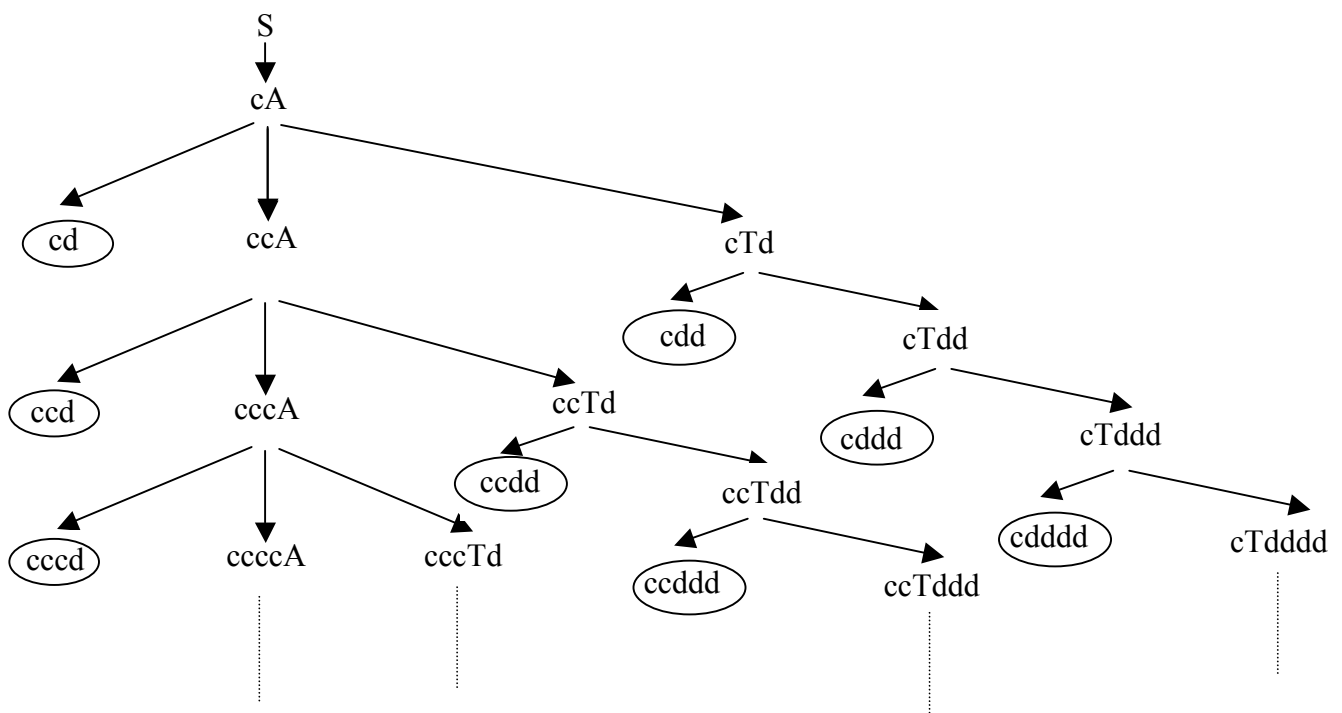
En ocasiones, analizar la semántica de las producciones que definen la gramática puede resultar más factible que intentar encontrar las palabras del lenguaje asociado a una palabra y, a partir de ellas, obtener su definición genérica. Así, por ejemplo, analizando las producciones que definen esta gramática es fácil obtener las siguientes conclusiones:

- $S \rightarrow cA$ : “la palabra empieza por *c*”  
 $A \rightarrow \mid cA$ : “añade ene *ces*”  
 $A \rightarrow d \mid Td$ : “la palabra termina en *d*”  
 $T \rightarrow Td \mid d$ : “añade eme *des*”

A partir de este análisis es relativamente fácil determinar que las palabras del lenguaje asociado a esta gramática se ajustan a:

$$L(G_d) = \{c^n d^m \mid n, m \geq 1\}$$

Construyendo el árbol, se encuentra que:



$$L(G_4) = \{cd, ccd, cccd, \dots, cdd, ccdd, \dots, cccdd, cccdd, cdddd, \dots\} = \{c^n d^m\} \text{ con } n, m \geq 1$$

$$\begin{aligned} \text{e) } S &::= \lambda \mid A \\ A &::= cd \mid Ad \mid cA \end{aligned}$$

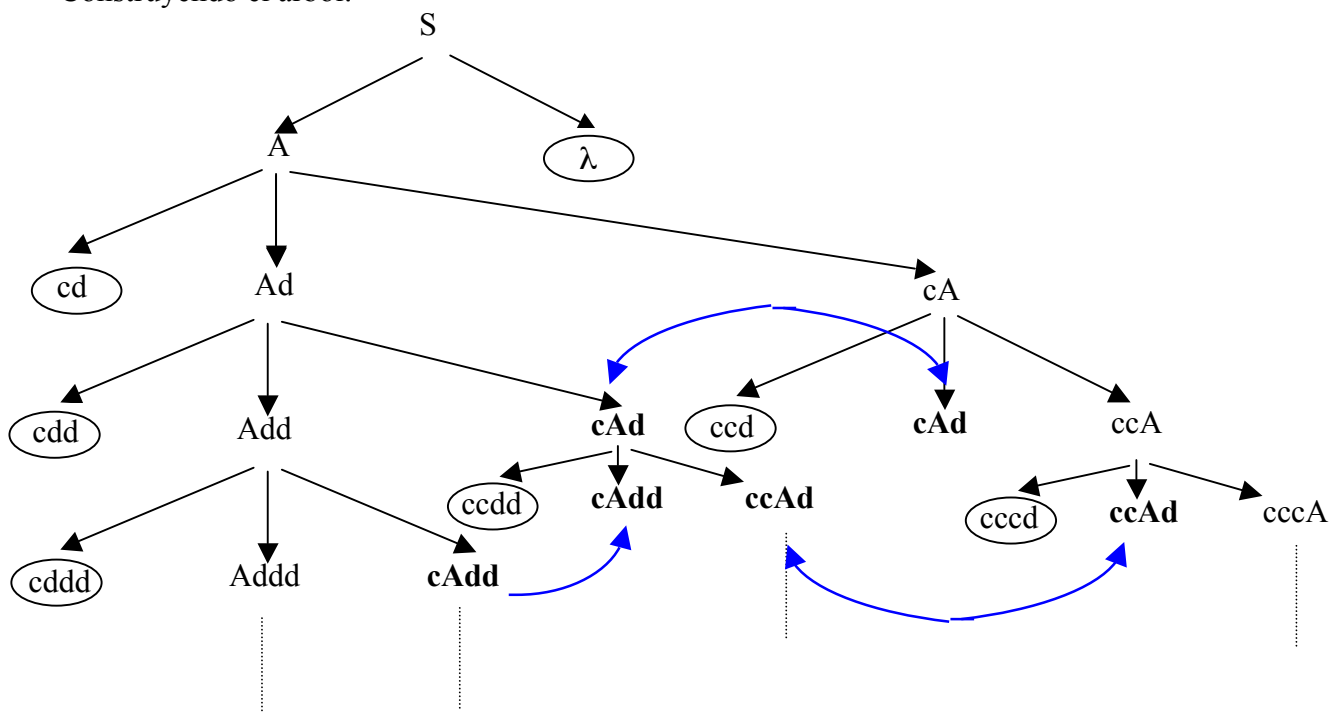
Analizando las producciones se puede concluir:

$S ::= \lambda$  "El lenguaje contiene la palabra vacía"  
 $S ::= A ; A ::= cd$  "cd es una palabra del lenguaje"  
 $S ::= A ; A ::= Ad$  "Las palabras terminan en una cadena de *des*"  
 $S ::= A ; A ::= cA$  "Las palabras comienzan por una cadena de *ces*"

A partir de este análisis se puede determinar que las palabras del lenguaje asociado a esta gramática se ajustan a:

$$L(G_e) = \{\lambda, c^n d^m \mid n, m \geq 1\}$$

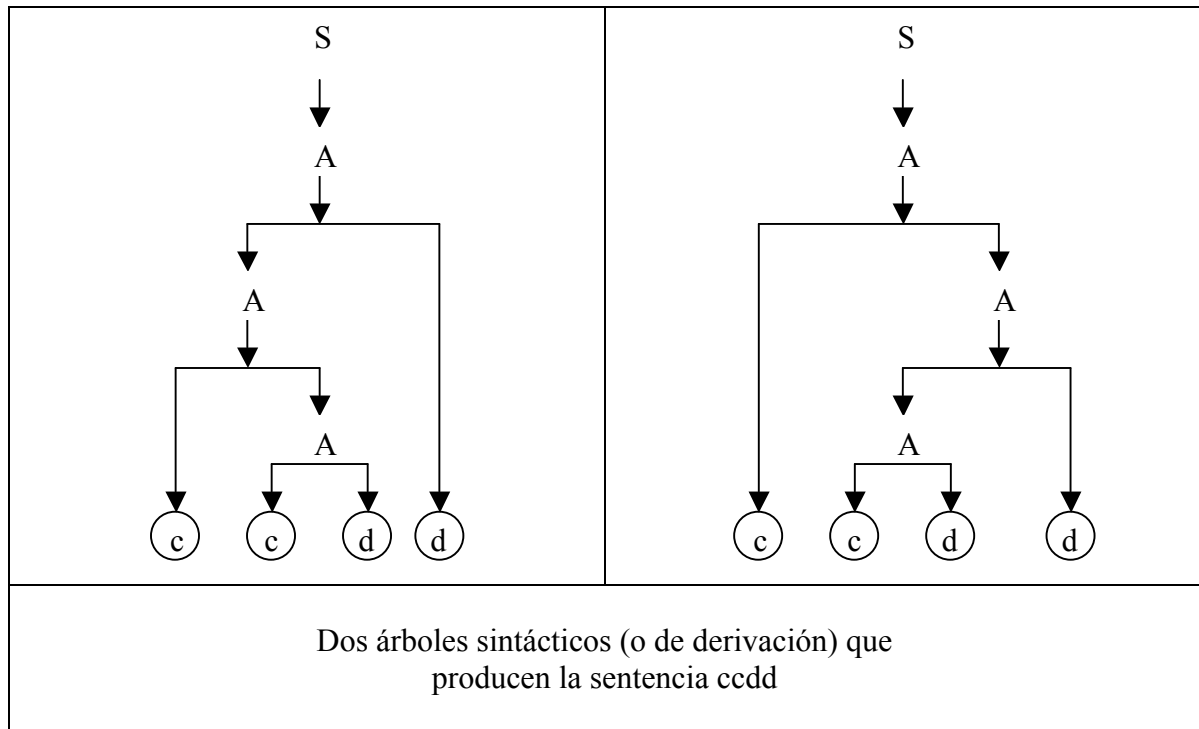
Construyendo el árbol:



$$L(G_5) = \{\lambda, cd, cdd, cddd, \dots, ccd, ccdd, cccdd, \dots\} = \{\lambda, c^n d^m\} = \text{con } n, m \geq 1$$

Determinadas palabras se pueden encontrar por más de un camino:

**La Gramática es ambigua.**



**3. Determinar el tipo de las siguientes gramáticas en la jerarquía de Chomsky, justificándolo: (T4p1E4)**

- a)  $G = (\{a, b\}, \{A, B, S\}, S, P)$ ,  
 $P = \{S ::= aA, A ::= bB, A ::= aA, A ::= a, B ::= \lambda\}$
- b)  $G = (\{a, b, c\}, \{A, B, C, S\}, S, P)$ ,  
 $P = \{S ::= aAb, S ::= Ba, S ::= \lambda, aAbC ::= aAbB, aAbC ::= aabC, BCc ::= AaCc, BCc ::= BaAbc, C ::= Ca, C ::= a\}$
- c)  $G = (\{casa, jardin, gato\}, \{S, CASERON, BOSQUE, TIGRE\}, S, P)$ ,  
 $P = \{S ::= TIGRE\ jardin, S ::= BOSQUE\ CASERON, BOSQUE ::= \lambda, jardin\ CASERON\ TIGRE\ casa ::= jardin\ BOSQUE\ TIGRE\ casa, gato\ CASERON\ BOSQUE ::= gato\ BOSQUE\ casa\ TIGRE\ BOSQUE, BOSQUE ::= TIGRE\ casa, BOSQUE ::= jardin\}$
- d)  $G = (\{x, y\}, \{C, A, B, S\}, S, P)$ ,  
 $P = \{S ::= Cx, S ::= Cy, S ::= By, S ::= Ax, S ::= x, S ::= y, A ::= Ax, A ::= Cx, A ::= x, B ::= By, B ::= yA, C ::= xA\}$
- e)  $G = (\{a, b, c\}, \{S, B\}, S, P)$ ,  
 $P = \{S ::= abc, S ::= aBSc, Ba ::= aB, Bb ::= bb\}$

Solución:

- a) Tipo 0. Podría ser de tipo 3 pero tiene una regla compresora ( $B ::= \lambda$ ).
- b) Tipo 1. En las producciones 4, 5, 6 y 7 se mantiene el contexto; y el resto serían válidas en una gramática de tipo 2.
- c) Tipo 0. Por regla compresora ( $BOSQUE ::= \lambda$ ). El resto de reglas mantienen el contexto, siendo válidas en una gramática de tipo 1.
- d) Tipo 2. Porque hay reglas de producción tipo  $Nt ::= t\ Nt$  (G3LD) y tipo  $Nt ::= Nt\ t$  (G3LI) mezcladas en la misma gramática.
- e) Tipo 0. Porque la regla 3 ( $Ba ::= aB$ ) no mantiene el contexto.

**4. Dada la gramática G, se pide: (T4p1E5)**

$G = (\{a, b, c\}, \{S, A, B\}, S, P)$ ,  $P = \{S ::= \lambda, S ::= aAc, A ::= aA, A ::= Ac, A ::= B, B ::= b, B ::= Bb\}$

- e) Especificar el tipo de G en la jerarquía de Chomsky, razonadamente.
- f) Determinar el lenguaje L generado por la gramática G.
- g) Construir 2 árboles de derivación para una misma palabra perteneciente a L(G).
- h) Comprobar si las siguientes formas sentenciales son válidas en G, y en caso afirmativo establecer una cadena de derivaciones que permite llegar a cada una de ellas.
  - d.1.- aaAcc
  - d.2.-ac
  - d.3.-ababBcc
  - d.4.-abbccc

Solución:

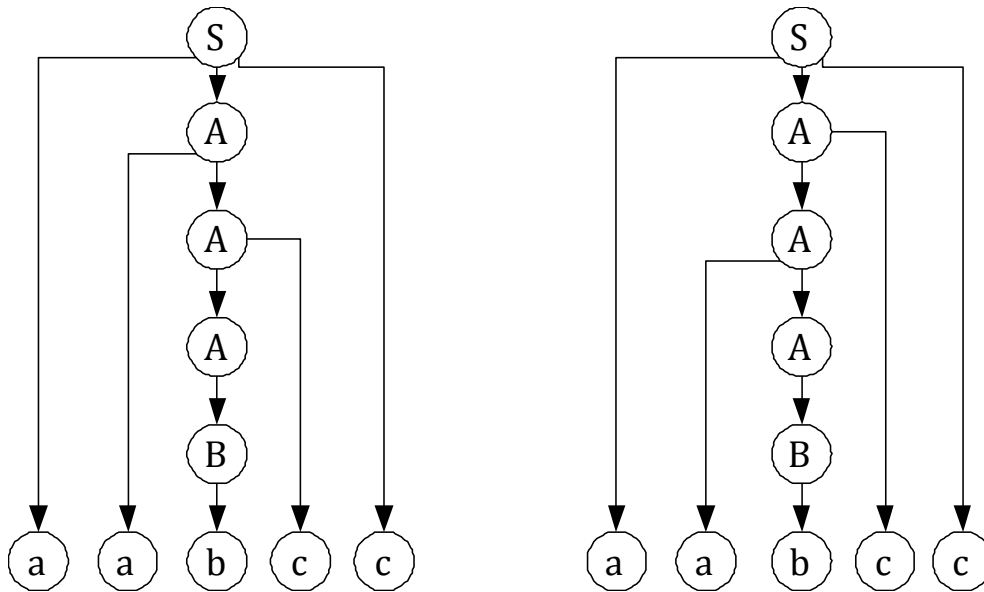
- a) G es de tipo 2 (de contexto libre).

Como todas las partes izquierdas están formadas por un sólo símbolo no terminal, puede ser G3 ó G2; pero en las partes derechas hay diversas combinaciones en las posiciones de terminal y no terminal ( $A ::= aA$  y  $A ::= Ac$ ), y además hay reglas con más de 2 símbolos ( $S ::= aAc$ ), por lo que es G2.

- b)  $L(G) = \{a^p b^q c^r, p=q=r=0 \text{ ó } p, q, r > 0\}$ .

Es decir, cadenas con  $p$  veces 'a',  $q$  veces 'b', y  $r$  veces 'c', manteniendo el orden (primero las a's, luego las b's y luego las c's). El número de veces que aparece cada símbolo es: o bien 0 para todos (obteniendo la palabra  $\lambda$ ), o bien al menos una aparición de cada símbolo, cada uno con un número de veces igual o independiente al resto.

c) Palabra=aabcc

Derivación de árbol izquierdo:  $S \rightarrow aAc \rightarrow aaAc \rightarrow aaAcc \rightarrow aaBcc \rightarrow aabcc$ Derivación de árbol derecho:  $S \rightarrow aAc \rightarrow aAcc \rightarrow aaAcc \rightarrow aaBcc \rightarrow aabcc$ 

d)

d.1.- **aaAcc**:  $S \rightarrow aAc \rightarrow aaAc \rightarrow aaAcc$ d.2.- **ac**: inválidad.3.- **ababBcc**: inválidad.4.- **abbccc**:  $S \rightarrow aAc \rightarrow aAcc \rightarrow aAccc \rightarrow aBccc \rightarrow aBbccc \rightarrow abbccc$



### 5. Obtener una gramática de tipo 0 para el lenguaje $L=\{a^n b^n c^n / n \geq 1\}$ (T4p1E10)

Solución:

Si  $L=\{a^n c^n\}$

las producciones de la gramática serían:

$S ::= ac \mid aSc$

¿Cómo introducir las “bes”?

Si se considera la producción:  $S ::= abc \mid abSc$

se obtendrían las siguientes palabras:

$L'=\{abc, ababcc, abababccc, ababababcccc, \dots\} = \{(ab)^n c^n / n \geq 1\}$

En este lenguaje, el número de veces que aparece cada carácter es el correcto pero no el orden en el que lo hacen. Para obtener el orden correcto se deberían utilizar reglas del tipo:

$ba \rightarrow ab$

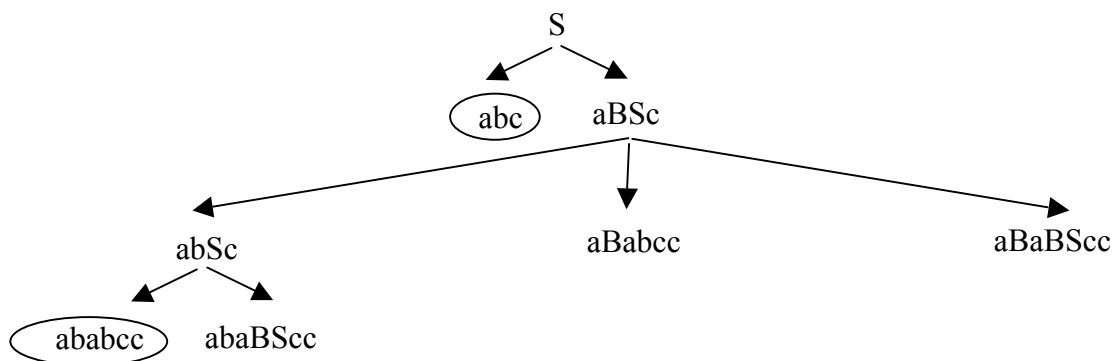
Ahora bien, puesto que esta regla no satisface las condiciones de una producción (al menos un símbolo NT en la parte izquierda) no puede utilizarse.

Se podría pensar que la solución es:

$S ::= abc \mid aBSc$

$Ba ::= aB$

$B ::= b$



¡¡Error!!

No obstante, y como se puede comprobar analizando el árbol de derivación, esta solución es errónea pues el orden de los caracteres sigue siendo incorrecto. Para conseguir el orden correcto es necesario garantizar que la sustitución  $B \rightarrow b$  se realiza única y exclusivamente una vez que las B están situadas en la posición correcta.

Puesto que la forma genérica de las cadenas obtenidas por las producciones:

$S ::= abc \mid aBSc$

$Ba ::= aB$

es:  $aBaBaB \dots abcccc \dots$

Es fácil ver que una de las B estará correctamente posicionada sí y sólo sí precede a la b:  $aBaBa \dots aBbcccc \dots$ . Si en este momento se realiza la sustitución, la forma genérica de las palabras será:

$aBaBa \dots abbcccc \dots$

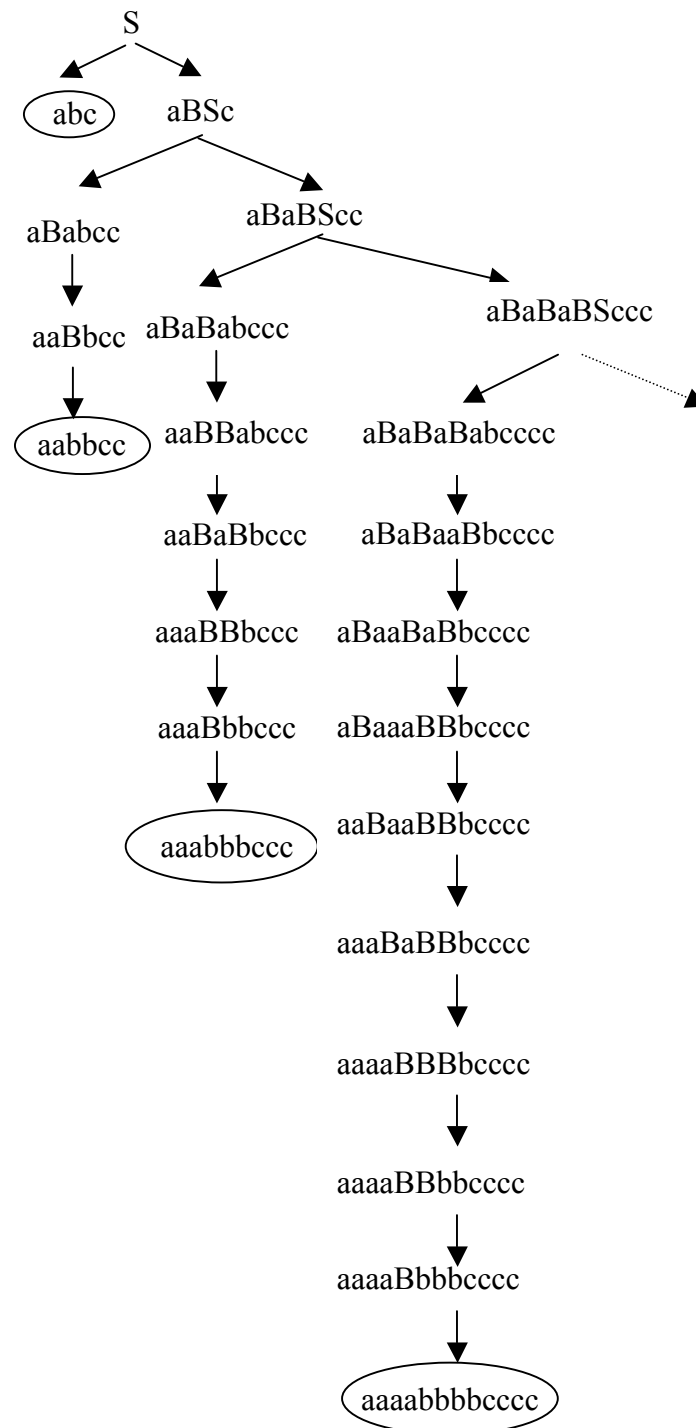
Las restantes B estarán correctamente posicionadas si preceden a la cadena de *bes* que se está generando, es decir, siempre que B preceda a b.

En base a esto, la solución del problema planteado es:

$S ::= abc \mid aBSc$

$Ba ::= aB$

$Bb ::= bb$



**6. Realizar las transformaciones necesarias del proceso de limpieza de gramáticas, para obtener una gramática limpia  $G'$  equivalente a la  $G$  dada. (T4p2E3)**

$G = (\{a,b,c,d\}, \{X,Y,Z,O,P,Q,A\}, Z, P),$

$P = \{ Z ::= Z, Q ::= OP, X ::= aa, Z ::= aX, Y ::= aa, Z ::= Ya, O ::= b, Z ::= aaa, P ::= QO, Q ::= d, P ::= c, O ::= PQ \}$

Solución:

$G' = (\{a\}, \{X,Y, Z\}, Z, P'), P' = \{ Z ::= aaa, Z ::= aX, Z ::= Ya, X ::= aa, Y ::= aa \}$

La gramática de entrada tiene una regla innecesaria asociada al axioma ( $Z ::= Z$ ).

Los símbolos inaccesibles son:

- No Terminales: O, P, Q
- Terminales: b, c, d (Estos símbolos forman parte de las producciones asociadas a los NT que se eliminan).

No hay reglas supérfluas.

**7. Dada la gramática G LI, obtener una G' LD equivalente. (T4p2E6)**

$$G = (\{0,1\}, \{A,S\}, S, P)$$

$$P = \{ S ::= 1 \mid A1; A ::= S0 \}$$

Solución:

Primero, como hay axioma inducido (axioma en la parte derecha) en la tercera regla, por lo que antes de aplicar el algoritmo de conversión G3LI en G3LD hay que eliminar el axioma inducido, obteniendo su gramática equivalente.

Se añade un nuevo NT  $\Rightarrow B \in \Sigma_N$ .

Se **añade** una regla  $B ::= x$ , para cada regla  $S ::= x$  (pero no se eliminan las reglas del axioma):

$$S ::= 1 \mid A1 \rightarrow B ::= 1 \mid A1$$

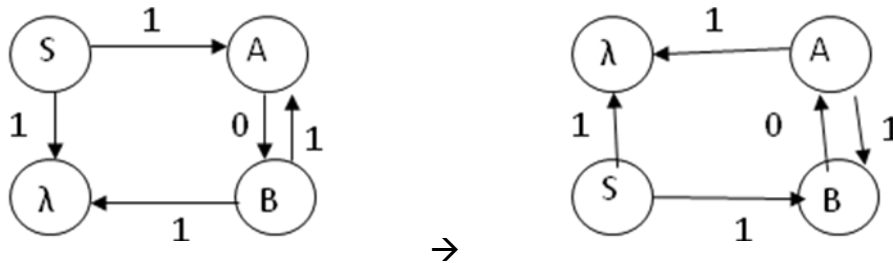
Se sustituye el axioma en la parte derecha (en todas las reglas que aparezca) por el nuevo símbolo NT añadido:

$$A ::= S0 \rightarrow A ::= B0$$

La gramática resultante equivalente sin axioma inducido es:

$$G' = (\{a,b\}, \{A,S,B\}, S, P'), P' = \{S ::= 1 \mid A1, A ::= B0, B ::= 1 \mid A1\}$$

Ahora se puede pasar a convertir en LD, construyendo el grafo correspondiente, e invirtiendo la orientación de las conexiones:



Reescribiendo las producciones leyendo el grafo de la derecha, la gramática G LD equivalente resultante es:

$$G LI = (\{a,b\}, \{A,S,B\}, S, P), P = \{S ::= 1 \mid 1B, B ::= 0A, A ::= 1 \mid 1B\}$$

**8. Dada la gramática G: (T4p2E8)**

$$G = (\{e, f, g, z, a, b, d\}, \{Y, X, E, A, D, I, G\}, A, P),$$

$$P = \{ \begin{array}{l} A ::= a \\ E ::= b \\ A ::= azb \\ A ::= aX \\ E ::= E \\ G ::= g \\ X ::= XE \\ D ::= eI \\ X ::= z \\ Y ::= b \\ I ::= fG \\ X ::= Xb \\ E ::= d \end{array} \}$$

- Transformar a FNC, explicando cada paso realizado.
- Determinar si las palabras 'abz' y 'azdbb' pertenecen al lenguaje generado por G. En caso afirmativo, generar un árbol de derivación para dicha palabra. En caso negativo, justificar la no pertenencia.

Solución:

**a) Transformación a FNC, limpiándola previamente:**Limpieza

- Reglas Innecesarias:  
 $E ::= E$
- Símbolos inaccesibles:
  - No aparecen en parte derecha: D, Y  
 $D ::= eI, Y ::= b$
  - Tras varios ciclos (indirectos): G, I  
 $G ::= g, I ::= fG$

Gramática limpia:

$$G' = (\{a, b, d, z\}, \{X, E, A\}, A, P'),$$

$$P' = \{ \begin{array}{l} A ::= a \\ A ::= aX \\ A ::= azb \\ E ::= b \\ E ::= d \\ X ::= Xb \\ X ::= XE \\ X ::= z \end{array} \}$$

Transformación a FNC (tabuladas en FNC)

- Tratamiento de  $A ::= aX$

$$A ::= BX$$

$$B ::= a$$

- Tratamiento de  $A ::= azb$

$$A ::= BC$$

$$C ::= zb$$

$$C ::= DF$$

$$D ::= z$$

$$F ::= b$$

- Tratamiento de  $X ::= Xb$

$$X ::= XF$$

## Gramática en FNC:

$$G' = (\{a,b,d,z\}, \{A, B, C, D, E, F, X\}, A, P''),$$

$$P'' = \{ A ::= a$$

$$A ::= BC$$

$$A ::= BX$$

$$B ::= a$$

$$C ::= DF$$

$$D ::= z$$

$$E ::= b$$

$$E ::= d$$

$$F ::= b$$

$$X ::= XE$$

$$X ::= XF$$

$$X ::= z$$

$$\}$$
**b) Palabras:**

- abz: No pertenece, porque no existe ningún árbol de derivación que la genere.
- azdbb: Sí pertenece:

$$A \rightarrow \mathbf{BX} \rightarrow \mathbf{aX} \rightarrow \mathbf{aXF} \rightarrow \mathbf{aXFF} \rightarrow \mathbf{aXEFF} \rightarrow \mathbf{azEFF} \rightarrow \mathbf{azdFF} \rightarrow \mathbf{azdbF} \rightarrow \mathbf{azdbb}$$

**9. Obtener una gramática en FNC equivalente a la siguiente: (T4p2E9)**

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

$$P = \{ \begin{array}{l} S ::= AaB \mid Cbb \mid B \\ A ::= Aa \mid cD \\ B ::= a \mid Ba \mid \lambda \\ C ::= Sa \mid a \mid abB \\ D ::= aaA \\ E ::= aa \end{array} \}$$

Solución:

a) Gramática Bien FormadaSímbolos superfluos o no generativos =  $\{A, D\}$ 

$$P = \{ \begin{array}{l} S ::= Cbb \mid B \\ B ::= a \mid Ba \mid \lambda \\ C ::= Sa \mid a \mid abB \\ E ::= aa \end{array} \}$$
Símbolos inaccesibles =  $\{E\}$ 

$$P = \{ \begin{array}{l} S ::= Cbb \mid B \\ B ::= a \mid Ba \mid \lambda \\ C ::= Sa \mid a \mid abB \end{array} \}$$
Eliminación de reglas no generativas:  $B ::= \lambda$ 

$$P = \{ \begin{array}{l} S ::= Cbb \mid B \mid \lambda \\ B ::= a \mid Ba \\ C ::= Sa \mid a \mid abB \mid ab \end{array} \}$$
Eliminación de reglas de red denominación:  $S ::= B$ 

$$P = \{ \begin{array}{l} S ::= Cbb \mid a \mid Ba \mid \lambda \\ B ::= a \mid Ba \\ C ::= Sa \mid a \mid abB \mid ab \end{array} \}$$

La gramática bien formada equivalente es la siguiente (obsérvese que “c” no pertenece al alfabeto de símbolos terminales de la gramática, puesto que no hay ninguna regla que lo genere):

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

$$P = \{ \begin{array}{l} S ::= Cbb \mid a \mid Ba \mid \lambda \\ B ::= a \mid Ba \\ C ::= Sa \mid a \mid abB \mid ab \end{array} \}$$
b) Transformación a FNC

La transformación de dicha gramática en una gramática en forma formal de Chomsky conlleva la substitución de las siguientes reglas:

$S ::= Cbb$   
 $S ::= CD$   
 $D ::= EE$   
 $E ::= b$   
 $S ::= Ba$   
 $S ::= BF$   
 $F ::= a$   
 $B ::= Ba$   
 $B ::= BF$   
 $C ::= Sa$   
 $C ::= SF$   
 $C ::= abB$   
 $C ::= FG$   
 $G ::= EB$   
 $C ::= ab$   
 $C ::= FE$

La gramática en forma normal de Chomsky resultante será, por tanto:

$G = (\{a, b\}, \{S, B, C, D, E, F, G\}, S, P)$   
 $P = \{ S ::= CD \mid a \mid BF \mid \lambda$   
 $B ::= a \mid BF$   
 $C ::= SF \mid a \mid FG \mid FE$   
 $D ::= EE$   
 $E ::= b$   
 $F ::= a$   
 $G ::= EB \}$



**10. Sea la gramática G, hallar su FNG equivalente. (T4p2E11)** **$G = (\{a,b\}, \{S\}, S, P)$ , donde  $P = \{S ::= aSb \mid SS \mid \lambda\}$** 

Solución:

1.- Hay que eliminar la recursividad a izquierdas: $S ::= aSb \mid SS \mid \lambda \rightarrow$  (se transforma en)  $\rightarrow S ::= aSb \mid aSbX \mid \lambda$  y  $X ::= SX \mid S$ 2.- Hay que ordenar los no terminales:  $\{X, S\}$  y clasificar las reglas según este orden: $S ::= aSb$  (G1, aunque hay que quitar el terminal que está detrás del no terminal) $S ::= aSbX$  (G1, aunque hay que quitar el terminal que está detrás del no terminal) $S ::= \lambda$  (G1) $X ::= SX$  (G2) $X ::= S$  (G2)3.- Transformamos las de grupo G2 en grupo G1: $X ::= SX \rightarrow S ::= aSb \rightarrow X ::= aSbX$  (G1, aunque hay que quitar el terminal que está detrás del no terminal) $\rightarrow S ::= aSbX \rightarrow X ::= aSbXX$  (G1, aunque hay que quitar el terminal que está detrás del no terminal) $X ::= S \rightarrow S ::= aSb \rightarrow X ::= aSb$  (G1, aunque hay que quitar el terminal que está detrás del no terminal) $\rightarrow S ::= aSbX \rightarrow X ::= aSbX$  (G1, aunque hay que quitar el terminal que está detrás del no terminal)4.- Introducimos un nuevo símbolo para quitar el b: $B ::= b$ 

y la Gramática en FNG queda:

 $G' = (\{a,b\}, \{S,X,B\}, S, P')$  $P' = \{$  $S ::= aSB \mid aSbB \mid \lambda$  $X ::= aSBX \mid aSBXX \mid aSB$  $B ::= b$  $\}$