

Modelos de Computación: Relación 1

Daniel Monjas Miguélez

30 de octubre de 2020

Índice

1. Ejercicio 1	3
2. Ejercicio 2	3
3. Ejercicio 3	4
4. Ejercicio 4	4
5. Ejercicio 5	5
6. Ejercicio 6	5
7. Ejercicio 7	7
8. Ejercicio 8	7
9. Ejercicio 9	9
10.Ejercicio 10	11
11.Ejercicio 11	13
12.Ejercicio 12	13
13.Ejercicio 13	13
14.Ejercicio 14	13
15.Ejercicio 16	13
16.Ejercicio 17	14
17.Dificultad Media	15
18.Dificiles	17

1. Ejercicio 1

Describir el lenguaje generado por la siguiente gramática,

$$\begin{aligned}S &\rightarrow XYX \\ X &\rightarrow aX|bX|\epsilon \\ Y &\rightarrow bbb\end{aligned}$$

Defino la gramática generativa $G=(V,T,P,S)$ donde $T = \{a, b\}$, P es el conjunto de las reglas de producción del enunciado y S es el estado inicial. Si al estado inicial le aplicamos su correspondiente regla de producción tenemos que $S \rightarrow XYX$ y como Y solo tiene una regla de producción tenemos que $XYX \rightarrow XbbbX$. La regla de producción sobre X nos permite crear palabras formadas por a y por b , llegando a que $L(G) = \{ubbbv/u, v \in T^*\}$, es decir, el lenguaje generado por la gramática G es el lenguaje que tiene al menos tres b consecutivas. Esto demuestra la primera de las inclusiones.

Para ver la inversa seleccionamos una palabra con tres b consecutivas, la cual solo puede tener cualquiera de las siguientes formas $ebbbe, ebbbu, ubbbe, ubbbv$, donde $u, v \in (a, b)^*$ y vemos que la cual pertenece a nuestro lenguaje generado, lo cual es trivial ya que al ser $u, v \in T^* = (a, b)^*$ el cual incluye ϵ podemos construir cualquiera de las posibilidades anteriores.

2. Ejercicio 2

Describir el lenguaje generado por la siguiente gramática,

$$\begin{aligned}S &\rightarrow aX \\ X &\rightarrow aX|bX|\epsilon\end{aligned}$$

Defino la gramática generativa $G=(V,T,P,S)$ donde $T = \{a, b\}$, P es el conjunto de las reglas de producción del enunciado y S es el estado inicial. Si al estado inicial le aplicamos su correspondiente regla de producción llegamos a $S \rightarrow aX$, luego el lenguaje generado se va a ver obligado a empezar siempre por a . Luego al aplicarle a X su regla de producción podemos obtener cualquiera de las palabras del conjunto de palabras $(a, b)^*$, llegando así al siguiente lenguaje generado, $L(G) = \{au/u \in T^* = \{a, b\}^*\}$, que es claramente el lenguaje de las palabras comenzadas por a .

Para ver que cualquier palabra comenzada por a pertenece a nuestro lenguaje generado tomamos una palabra aleatoria empezada por a , que tendrá una de las formas siguientes $a\epsilon, au$ con $u \in \{a, b\}^*$. Como $\epsilon \in T^* = \{a, b\}^*$, vemos que claramente cualquier palabra empezada por a pertenece a nuestro lenguaje.

3. Ejercicio 3

Describir el lenguaje generado por la siguiente gramática,

$$\begin{aligned} S &\rightarrow XaXaX \\ X &\rightarrow aX|bX|\epsilon \end{aligned}$$

Defino la gramática generativa $G=(V,T,P,S)$ donde $T = \{a, b\}$, P es el conjunto de las reglas de producción del enunciado y S es el estado inicial. Si al estado inicial le aplicamos su correspondiente regla de producción llegamos a $S \rightarrow XaXaX$, y como con la regla de producción de X podemos producir cualquier palabra de T^* llegamos finalmente a que nuestro lenguaje generado por la gramática G es $L(G) = \{uavaw/u, v, w \in T^* = \{a, b\}^*\}$, que se trata del lenguaje cuyas palabras tienen al menos 2 a's.

Para ver que cualquier palabra con 2 a's o más pertenece a nuestro lenguaje generado tomamos una palabra con 2 a's o más. Si tiene solo 2 a's claramente pertenece a nuestra gramática, pues u, v y w pueden ser la palabra vacía o cualquier otra palabra. Si tiene más de dos a's solamente tenemos que dividir esa palabra, de forma que seleccionamos 2 as cualesquiera y todo lo que haya a la izquierda, derecha y entre ellas se considera una palabra u del conjunto de palabras $\{a, b\}^*$, siendo la palabra vacía cuando no haya nada, de forma que nuestra palabra se representa ahora de la forma $uavaw$ donde u, v y w son palabras de $\{a, b\}^* = T^*$, llegando así a que se trata de una palabra de nuestro lenguaje generado.

4. Ejercicio 4

Describir el lenguaje generado por la siguiente gramática,

$$\begin{aligned} S &\rightarrow SS|XaXaX|\epsilon \\ X &\rightarrow bX|\epsilon \end{aligned}$$

Defino la gramática generativa $G=(V,T,P,S)$ donde $T = \{a, b\}$, P es el conjunto de las reglas de producción del enunciado y S es el estado inicial. Si ahora defino una gramática auxiliar G_1 en la que la regla de producción de S la limitamos a $S \rightarrow XaXaX$, nos podemos fijar en que si $S \rightarrow XaXaX$ y luego aplicamos la regla de producción $XaXaX \rightarrow bX$ llegamos a la siguiente gramática generada, $L(G_1) = \{b^i a b^j a b^k / i, j, k \geq 0\}$. Si ahora defino que $L(G_1)^0 = \{\epsilon\}$, y $L(G_1)^{i+1} = L^i L$ y aplico la clausura de Kleene tengo que $L^* = \cup_{i \geq 0} L^i$, llego a que con las reglas de producción dadas $L(G)$ es la clausura de Kleene de $L(G_1)$, siendo el lenguaje que contiene un numero par de aes.

5. Ejercicio 5

Encontrar la gramática libre de contexto que genera el lenguaje sobre el alfabeto $\{a, b\}$ de las palabras que tiene más a que b (al menos una más).

Buscamos una gramática tal que $L(G) = \{u/u \in \{a, b\}^*, N_a(u) > N_b(u) \geq 0\}$ luego vamos a buscar una gramática $G=(V, T, P, S)$, de la cual sabemos que $T = \{a, b\}$ y S es el estado inicial. Ahora buscaremos las reglas de producción.

6. Ejercicio 6

Encontrar gramáticas de tipo 2 para los siguientes lenguajes sobre el alfabeto $\{a, b\}$. En cada caso determinar si los lenguajes generados son de tipo 3, estudiando si existe una gramática de tipo 3 que los genera.

1. Palabras en las que el número de b no es tres.
2. Palabras que tienen 2 o 3 b.
3. Palabras que no contienen la subcadena ab
4. Palabras que no contienen la subcadena baa

1.- Defino la gramática generativa $G=(V, T, P, S)$ donde $T = \{a, b\}$, P es el conjunto de las reglas de producción y S es el estado inicial. Y en el conjunto P se incluyen las siguientes reglas de producción, $S \rightarrow aS|bB_1\epsilon$, $B_1 \rightarrow aB_1|bB_2\epsilon$, $B_2 \rightarrow aB_2|bB_3\epsilon$, $B_3 \rightarrow aB_4|bB_4$ y $B_4 \rightarrow aB_4|bB_4\epsilon$. Aquí nuestro objetivo es que no haya un total de tres b en la palabra. Para ello establecemos las reglas anteriores, donde al aplicar la regla al estado inicial se pueden poner tantas a como se quieran, una b o la palabra vacía. Si llegamos a B_1 la palabra tendrá al menos una b, si llegamos a B_2 la palabra tendrá dos b. Si llegamos a B_3 tendrá tres b y se obligará a llegar a B_4 , lo que conllevará que la palabra tenga cuatro b. En B_4 lo que se permite es añadir cualquier palabra de $\{a, b\}^*$ a la palabra que teníamos previamente.

2.- En este caso se parece bastante al caso anterior. Defino la gramática generativa $G=(V, T, P, S)$ donde $T = \{a, b\}$, P es el conjunto de las reglas de producción y S es el estado inicial. Y en el conjunto P se incluyen las siguientes reglas de producción, $S \rightarrow aS|bB_1$, $B_1 \rightarrow aB_1|bB_2$, $B_2 \rightarrow aB_2|bB_3\epsilon$ y $B_3 \rightarrow aB_3\epsilon$. Aquí establecemos un enfoque parecido al anterior. En la primera regla de producción se obliga a llegar a B_1 , luego antes de aplicar B_1 tenemos una palabra de $\{a, b\}^*$, que tiene 1 b. B_1 obliga a llegar a B_2 , y antes de aplicar B_2 tenemos una palabra de $\{a, b\}^*$ con 2 b, luego B_2 nos

da la posibilidad de añadir otra b, varias a o terminar la palabra. En el caso de llegar a B_3 la palabra tiene un total de 3 b, y en el caso de aplicar B_3 se le pueden añadir todas las a que queramos o terminar la palabra.

3.- Defino la gramática generativa $G=(V,T,P,S)$ donde $V = \{S, X, Y, Z\}$, $T = \{a, b\}$, P es el conjunto de las reglas de producción y S es el estado inicial. Y en el conjunto P se incluyen las siguientes reglas de producción, $S \rightarrow bX|aY|\epsilon$, $Y \rightarrow aY|\epsilon$ y $X \rightarrow bX|aY|\epsilon$. Al no poder incluir la subcadena ab, tenemos que obligar que tras añadir una a no se puedan añadir b. Para ello usamos las reglas de producción anteriores. La regla sobre el estado inicial nos permite añadir tantas a como queramos, una b o terminar la palabra. Si añadimos una b llegamos a la regla de producción sobre X, lo cual nos permite añadir todas las a que queramos detrás de la b o terminar la palabra. En el caso de que sobre el estado inicial hayamos añadido todas las a que queramos estaremos en el regla de producción sobre Y que solo nos permite seguir añadiendo a o terminar la palabra.

4.- Defino la gramática generativa $G=(V,T,P,S)$ donde $V = \{S, X, Z\}$, $T = \{a, b\}$, P es el conjunto de las reglas de producción y S es el estado inicial. Y en el conjunto P se incluyen las siguientes reglas de producción,

$$\begin{aligned} S &\rightarrow bX|aS|\epsilon \\ X &\rightarrow bX|aZ|\epsilon \\ Z &\rightarrow bX|\epsilon \end{aligned}$$

Con este lenguaje generamos claramente $L(G)$, que es el lenguaje que no contiene la subcadena baa. Para ello vemos que se impide que detrás de una b venga más de una a. Si tenemos $S \rightarrow bX$ la regla de producción sobre X nos permite añadir todas las b que queramos, terminar la palabra o una a, pero si ponemos una a la regla de producción sobre Z nos obliga a añadir una b tras la a y volver de nuevo a X, luego de momento inmediatamente después de una b solo se permite una a. Supongamos que la b se optiene con la regla de producción $Y \rightarrow bX$, lo que nos manda de nuevo a la regla de producción sobre X, que ya hemos visto que esta solo permite una a tras las b.

Por otro lado tomamos una palabra que pertenezca al lenguaje L de palabras que no contienen la subcadena baa. Supongamos que esta palabra empieza por b, obligatoriamente tendrá la siguiente forma $b^i.(ab)^j.a^k, i > 0, j \geq 0, k = 0, 1$. Y por tanto al llegar a la regla de producción sobre X podemos añadir todas las b que queramos o ninguna, obteniendo el b^i , seguido de la concatenación de tantos (ab) como queramos, que se obtiene con las reglas de producción X y Z y una a al final que la de la regla de

producción sobre X. Para la el caso de que empiece por a se haría de forma análoga, se pone tantas a como se quieren con la regla sobre S, y se puede terminar la palabra o añadir una b, si se añade la b se llega al caso de antes, se pueden seguir añadiendo b, subcadenas (ab) o terminar la palabra.

7. Ejercicio 7

Encontrar una gramática libre del contexto que genera el lenguaje,

$$L = \{1u1 | u \in \{0,1\}^*\}$$

Defino la gramática generativa $G=(V,T,P,S)$ donde $T = \{0,1\}$, P es el conjunto de las reglas de producción y S es el estado inicial. Y en el conjunto P se incluyen las siguientes reglas de producción, $S \rightarrow 1X1$, la cual está permitida por tratarse de una gramática libre de contexto y finalmente la regla de producción $X \rightarrow 0X|1X|\epsilon$. Con lo que el lenguaje generado sería $L(G) = \{1u1 | u \in \{0,1\}^*\}$, siendo esto el lenguaje cuyas palabras tienen al menos 2 unos, uno al final y otro al principio. Para ello simplemente vemos que al aplicar la regla de producción sobre S ya forzamos a que la palabra final vaya a tener 2 unos, y por otro lado la regla de producción sobre X nos permite que entre esos 2 unos podemos meter cualquier palabra de $\{0,1\}^*$. Y si tomamos una palabra con dos unos, uno al final y otro al principio solamente tenemos que representar esa palabra como 1u1, de forma que los dos 1 nos los da la regla de producción sobre S y la palabra u nos la da la regla de producción sobre X incluso si u es la palabra vacía.

8. Ejercicio 8

Encontrar si es posible una gramática lineal por la derecha o na gramática libre del contexto que genere el lenguaje L supuesto que $L \subset \{a,b,c\}^*$ y verifica:

- $u \in L$ si y solamente si verifica que u no contiene dos símbolos b consecutivos.
- $u \in L$ si y solamente si verifica que u contiene dos símbolos b consecutivos.
- $u \in L$ si y solamente si verifica que contiene un número impar de símbolos c.
- $u \in L$ si y solamente si verifica que no contiene el mismo número de símbolos b que de símbolos c.

Para la primera condición es suficiente con indicar que tras una b obligatoriamente tiene que venir una a. Para ello defino la gramática generativa $G=(V,T,P,S)$ donde $T = \{a, b, c\}$, P es el conjunto de las reglas de producción y S es el estado inicial. Y en el conjunto P se incluyen las siguientes reglas de producción, $S \rightarrow aS|cS|bX|\epsilon$ y $X \rightarrow aS|cS|\epsilon$, donde la primera regla de producción nos permite situar tantas a como queramos seguidas de una b o la palabra vacía y la segunda regla de producción nos permite situar tantas a como queramos detras de una b o la palabra vacía. Luego se obliga a que inmediatamente después de una b se introduzca una a o una c o se termine la palabra obligando a que no haya dos b consecutivas.

Para la segunda condición solo tenemos que asegurarnos que si se añade una b se añadan obligatoriamente dos b, para ello defino la gramática generativa $G=(V,T,P,S)$ donde $T = \{a, b, c\}$, P es el conjunto de las reglas de producción y S es el estado inicial. Y en el conjunto P se incluyen las siguientes reglas de producción, $S \rightarrow aS|bS|cS|bbX$ y $X \rightarrow aX|bX|cX|\epsilon$. En la primera regla de producción permitimos que se añadan tantas a y b como se quieran en el orden que se quieran, pero la palabra no puede terminar ahí, para eso tenemos la segunda regla de producción a la que solo se llega una vez añadimos dos b consecutivas y la cual ya nos permite o terminar la palabra o seguir añadiendo a y b.

Para la tercera condición tenemos que asegurarnos que la palabra no se termine con un número par de c, para ello defino la gramática generativa $G=(V,T,P,S)$ donde $V = \{S, X\}$, $T = \{a, b, c\}$, P es el conjunto de las reglas de producción y S es el estado inicial. Y en el conjunto P se incluyen las siguientes reglas de producción, $S \rightarrow aS|bS|cX$ y $X \rightarrow aX|bX|cS|\epsilon$, la primera regla nos permite añadir tantas a como queramos y tantas b como queramos y una c, no nos permite terminar la palabra pues no se tendría un numero impar de c. La segunda nos permite o terminar la palabra pues en este caso se tendría un número impar de c y añadir tantas a y b como queramos, así como añadir una c más y volver al principio. Luego de esta manera antes de aplicar la regla de producción sobre S se tiene o 0 c o un número par de c y antes de aplicar X se tiene un número impar de c por lo que la regla de X nos permite terminar la palabra. Ahora tomamos una palabra del lenguaje L que es el lenguaje con un número impar de c. Esta palabra tendrá la siguiente forma $a^{n_1}.b^{m_1}.c \dots a^{n_n}.b^{m_n}.c$, donde $n_1, n_n, m_1, m_n \geq 0$, y el número de c es impar y mayor que 0. Si u es una palabra del lenguaje L , contamos el número de c de la palabra y obtendremos que este número es impar y podremos representar la palabra como $u.c.v.w \dots$, donde $u, v, w \in \{a, b\}^*$, vemos que tanto la regla de producción sobre S como la regla de producción sobre X nos permite generar cualquier palabra de $\{a, b\}^*$ y además por como están definidas nos aseguran que la palabra solo termine si el número de c es impar luego cualquier palabra del lenguaje L se puede generar con las reglas

de producción dadas en la gramática G .

Para la cuarta condición tenemos que asegurarnos que el número de b sea distinto siempre del número de c , para ello defino la gramática generativa $G=(V,T,P,S)$ donde $T = \{a, b, c\}$, P es el conjunto de las reglas de producción y S es el estado inicial. Y en el conjunto P se incluyen las siguientes reglas de producción,

$$\begin{aligned} S &\rightarrow XbX|YcY \\ X &\rightarrow aX|bX|XbXcX|XcXbX|\epsilon \\ Y &\rightarrow aY|cY|YbYcY|YcYbY|\epsilon \end{aligned}$$

La primera regla de producción nos permite poner tantas a como queramos y luego una b o una c y pasar a la regla de producción X o la regla de producción Y . La regla de producción X nos permite añadir todas las a que queramos, todas la b que queramos y nos asegura que no se añaden mas c que b así como terminar la palabra, luego al coger este camino habrá al menos un b más que c , la que se añade antes de entrar a la regla de producción de X . Por otro lado la regla de producción Y hace lo mismo que X para c , es decir, nos permite añadir todas las a que queramos, nos permite añadir todas las c que queramos y nos asegura que no se añaden mas b que c , así como terminar la palabra, luego si cogemos Y nos asegura que al menos hay una c mas que b . Luego $u \in L(G)$, entonces $u \in L$, donde L es el lenguaje donde las palabras tiene distinto numero de b que de c . Ahora tomamos una palabra cualquiera de este lenguaje y veamos que se puede generar con nuestra gramática. Sea $u \in L$ comprobamos si tiene más b que c o viceversa. Supongamos que hay mas b que c y el caso contrario sería análogo cambiando X por Y . Ahora lo único que tenemos que hacer es dividir la palabra de tal forma que nos quede con la forma siguiente, $v.b.w$ donde v y w son palabra en las que obligatoriamente $N_b(u) \geq N_c(v)$ y $N_b(w) \geq N_c(w)$. Vemos que además la forma $u.v.w$ se ajusta a la regla de producción $S \rightarrow XbX$, donde la regla de producción de X nos permite obtener cualquier palabra u y v asegurandonos que el número de b no supere el número de c .

9. Ejercicio 9

- Dado el alfabeto $A = \{a, b\}$ determinar si es posible encontrar una gramática libre de contexto que genera las palabras de longitud impar, y mayor igual que 3, tales que la primera letra coincida con la letra central de la palabra.
- Dado el alfabeto $A = \{a, b\}$ determinar si es posible encontrar una gramática libre de contexto que genere las palabras de

longitud par, y mayor o igual que 2, tales que las dos letras centrales coincidan.

Defino la gramática generativa $G=(V,T,P,S)$ donde $T = \{a,b\}$, $V = \{S,Y\}$, P es el conjunto de las reglas de producción y S es el estado inicial. Y en el conjunto P se incluyen las siguientes reglas de producción,

$$\begin{aligned} S &\rightarrow aXb|bYa|aXa|bYb \\ X &\rightarrow aXb|bXa|aXa|bXb|a \\ Y &\rightarrow aYb|bYa|aYa|bYb|b \end{aligned}$$

Esta gramática genera el lenguaje pedido. La regla de producción sobre S nos permite que la palabra empiece y termine por cualquiera de los símbolos terminales, pero en función de porque símbolo terminal empiecen se utiliza una u otra regla de producción, aunque ambas trabajan de forma similar. Supongamos que la palabra empieza por a , luego se aplicaría la regla de producción sobre X , la cual nos permite formar cualquier subcadena empezada por a a la derecha y terminada por a o b a la izquierda, además de ir creciendo en múltiplos de dos y forzar a que la palabra termine con un símbolo terminal a . Como crece de dos en dos al terminar la palabra tendrá longitud impar como se pide, y como la regla de producción obliga a que la palabra empiece por el mismo símbolo terminal que se tiene en medio obtenemos el lenguaje pedido.

Ahora tomamos una palabra del lenguaje y veamos que se puede obtener a partir de las reglas de producción dadas. Sea $u \in L$, esta palabra es no vacía y empieza y termina por un símbolo terminal. Supongamos que empieza por a , luego sabemos que la regla de producción a elegir será $S \rightarrow aXb|aXa$, dependiendo de por que letra termina. Luego la palabra tendrá la siguiente forma $a.v.a.w.\{a,b\}$, la a del principio y el símbolo terminal del final nos lo hemos quitado ya aplicando la regla de producción sobre S . Ahora la regla de producción sobre X nos asegura que tendremos una a en el medio, y nos permite formar cualquier subcadena v e w ya que si nos fijamos en los símbolos terminales de v por la derecha y w por la izquierda uno a uno podemos elegir cualquiera de las combinaciones que nos facilita X , luego cualquier palabra de L se puede obtener con las reglas de producción dadas (análogo si empezase por b).

Defino la gramática generativa $G=(V,T,P,S)$ donde $T = \{a,b\}$, $V = \{S,Y\}$, P es el conjunto de las reglas de producción y S es el estado inicial. Y en el conjunto P se incluyen las siguientes reglas de producción,

$$\begin{aligned} S &\rightarrow aSb|bSa|aSa|bSb|Y \\ Y &\rightarrow aa|bb \end{aligned}$$

Esta gramática genera el lenguaje $L(G)$ de las palabras de longitud par y cuyas dos letras centrales son iguales. Que la longitud es par es claro, ya que al aplicar cualquier regla de producción se añaden dos símbolos no terminales, luego la longitud de la palabra es múltiplo de 2. Que las palabras tienen la dos letras centrales iguales se debe a que con la regla de producción sobre S solo se añaden símbolos terminales a la izquierda y la derecha del centro de la palabra, y para terminar la misma se tiene que poner en el centro un simbolo terminal Y, ya que no hay posibilidad de usar la palabra vacía, y la regla de producción sobre Y añade dos simbolos terminales iguales, dejando por tanto, dos simbolos terminales en el centro de la palabra.

Ahora tomamos una palabra del lenguaje de las palabras de longitud par y con las dos letras centrales iguales $u \in L$. Esta palabra se puede reescribir como $v.aa.w$ o $v.bb.w$. Si nos vamos fijando en los simbolos no terminales de v por la derecha y de w por la izquierda, uno a uno, podemos aplicar la regla de producción sobre S para obtener las palabras v y w, quedándonos solo los dos símbolos del centro, para lo que se aplica la regla de producción $S \rightarrow Y$ seguida de $Y \rightarrow aa|bb$, obteniendo así la palabra u por medio de las reglas de producción.

10. Ejercicio 10

Determinar si el lenguaje generado por la gramática,

$$S \rightarrow SSS \rightarrow XXXS \rightarrow aX|Xa|b$$

es regular. Justificar la respuesta.

Para que este lenguaje sea regular tiene que poder generarse por medio de una gramática regular. Este lenguaje es el lenguaje de las palabras con un numero de b multiplo de 3. La regla de producción sobre S nos permite generar o bien 3 X o bien 2 S, luego obligatoriamente el numero de X cuando desaparecen todos los simbolo S es divisible por 3. Ahora veamos que se puede generar cualquier numero de X multiplo de 3. Sea un múltiplo de 3 el que sea, basta con dividir dicho multiplo de 3 entre 3, para ver cuantas S necesitaremos. Al hacer este paso obtenemos un numero $k \geq 1$, si k es uno el caso es trivial, ahora veamos si k es mayor que 1. Vemos que podemos aumentar las S de dos en dos o de 1 en uno. Para el segundo caso aplicamos lo siguiente, $S \rightarrow SS$, y sobre una de las S hacemos $S \rightarrow XXX$, mientras que sobre la otra hacemos $S \rightarrow SS$, llegando así a $XXXSS$ luego aquí se tendría los múltiplos impares no pares de 3. Luego ya hemos visto que obligatoriamente el número de simbolos terminales X antes de aplicar la regla de producción sobre X es múltiplo de tres. Ahora por otro lado vemos que la regla de producción sobre X nos permite añadir todas las a que queramos

o una b, donde siempre se obliga a poner una b por cada simbolo no terminal X, llegando así al lenguaje de palabras con numero de b multiplo de 3.

Ahora veamos el recíproco. Tomamos una palabra del lenguaje de palabras con numero de b multiplo de 3 $u \in L$ la cual tiene un número múltiplo de 3 de b y entre las distintas b puede haber o no un número mayor que 1 de a. Lo único que tenemos que hacer es contar el número de b y generar tantos simbolos terminales X como b tenga la palabra. Ahora como X nos permite añadir tantas a como queramos metemos entre las b tantas a como sean necesarias para obtener la palabra y finalmente sustituimos por b las X que queden obteniendo así la palabra buscada.

Nuestro objetivo será encontrar una reglas de producción que generen este mismo lenguaje pero que se correspondan con una gramática regular.

Defino la gramática generativa $G=(V,T,P,S)$ donde $T = \{a,b\}$, $V = \{S, X\}$, P es el conjunto de las reglas de producción y S es el estado inicial. Y en el conjunto P se incluyen las siguientes reglas de producción,

$$\begin{aligned} S &\rightarrow XbXbXbX \\ X &\rightarrow XbXbXbX|aX|\epsilon \end{aligned}$$

Veamos ahora que las reglas de producción dadas generan la gramática pedida. Es bastante claro que la palabra obtenida tendra un numero de b múltiplo de 3 o al menos 3 b. Para ello vemos que la regla de producción sobre S nos fuerza a que como mínimo la palabra tenga 3 b, y la regla de producción sobre X nos permite añadir más b pero siempre de 3 en 3, así como añadir todas las a que queramos entre cada par de b consecutivas, o bien no poner nada. Luego la palabra que nos queda tendrá un numero de b múltiplo de 3 y por tanto pertenece al lenguaje.

Ahora tomamos una palabra del lenguaje $u \in L$ y veamos que puede ser generada por nuestras reglas de producción. u estará contendrá un número múltiplo de 3 de b en ella, para ello hacemos $N_b(u)/3 = k$ y aplicamos una vez $S \rightarrow XbXbXbX$ y k-1 veces $X \rightarrow XbXbXbX$. Ahora solo tenemos que fijarnos si la palabra empieza y termina por una secuencia de a y si entre los pares de b consecutivos hay secuencias de a y en lso casos en los que sea afirmativo aplicamos $X \rightarrow aX$ tantas veces como sea necesario. Una vez tengamos todas las a que necesitamos nos quedarán símbolos no terminales X sobre los que aplicamos la regla de producción $X \rightarrow \epsilon$. Así vemos que se puede generar cualquier palabra del lenguaje con nuestra gramática, la cual es una gramática regular y por consiguiente se trata de un lenguaje regular.

11. Ejercicio 11

12. Ejercicio 12

13. Ejercicio 13

14. Ejercicio 14

Sea $L \subset A^*$ un lenguaje arbitrario. Sea $C_0 = L$ y definamos los lenguajes S_i y C_i , para todo $i \geq 1$; por $S_i = C_{i-1}^+$ y $C_i = \overline{S_i}$.

- ¿Es S_1 siempre, nunca o a veces igual a C_2 ? Justifica la respuesta.
- Demostrar que $S_2 = C_3$, cualquiera que sea L . (Pista: Demuestra que C_3 es cerrado para la concatenación).

a) $C_0 = L$, $S_1 = L^+$, $C_1 = \overline{L^+}$, $S_2 = \overline{L^+}^+$ y $C_2 = \overline{\overline{L^+}^+}$.

15. Ejercicio 16

Dada la gramática $G = (\{S, A\}, \{a, b\}, P, S)$ donde $P = \{S \rightarrow abAS, abA \rightarrow baab, S \rightarrow a, A \rightarrow b\}$. Determinar el lenguaje que genera.

Genera el lenguaje de los números naturales. Por las reglas de producción sobre S tenemos claro que obligatoriamente las palabras generadas por esta gramática terminan en a o se componen únicamente de una a . Luego $L(G)$ es el lenguaje de las palabras formadas al concatenar abb y $baab$ tantas veces como sea necesarias (incluso ninguna) en cualquier orden y terminan en a . Veamos que esto se verifica. La regla de producción sobre S nos permite el caso de que la palabra termine en a y no contenga ninguna de las subcadenas mencionadas o bien $S \rightarrow abAS$. Si repetimos esta última regla nos queda una sucesión de símbolos terminales y no terminales con la siguiente forma $(abA)^i S$, $i > 0$. Ahora sabemos que cada una de las subcadenas abA puede ser sustituida usando la regla de producción $abA \rightarrow baab$, obteniendo una de las subcadenas mencionadas o usando la regla de producción $A \rightarrow b$ que nos da la subcadena abb , luego si la palabra no es a contiene al menos una de las subcadenas mencionadas y finalmente la regla de producción $S \rightarrow a$ nos asegura que la palabra termine por a luego toda palabra de $L(G)$ pertenece al lenguaje de palabras que se compone por las subcadenas abb y $baab$ un número cualquiera de veces y que terminan por a .

Ahora tomamos una palabra de este lenguaje $u \in L$ y comprobemos que se puede obtener por nuestra gramática. Si la palabra u se compone solo de una a es trivial pues se obtiene aplicando únicamente $S \rightarrow a$. Si la

palabra termina por a y contiene las subcadenas mencionadas simplemente contamos cuantas de estas subcadenas contiene y utilizamos la regla de producción $S \rightarrow abAS$ tantas veces como subcadenas mencionadas contenga la palabra. Luego usamos $S \rightarrow a$, para asegurarnos que termina por a . Y ahora utilizamos $abA \rightarrow baab$ o $A \rightarrow b$ en función de cual de las subcadenas queramos obtener, así hasta que no queden símbolos no terminales y finalmente obtendremos la palabra u .

16. Ejercicio 17

Sea la gramática $G = (V, T, P, S)$ donde:

- $V = \{ \langle \text{numero} \rangle, \langle \text{digito} \rangle \}$
- $T = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- $S = \langle \text{numero} \rangle$
- Las reglas de producción P son:
 - $\langle \text{numero} \rangle \rightarrow \langle \text{numero} \rangle \langle \text{digito} \rangle$
 - $\langle \text{numero} \rangle \rightarrow \langle \text{digito} \rangle$
 - $\langle \text{digito} \rangle \rightarrow 0|1|2|3|4|5|6|7|8|9$

Determinar el lenguaje que genera.

La gramática genera T^+ . Para ello tomaremos el estado inicial y veremos que posibilidades hay. Si tomamos el estado inicial tenemos dos opciones, $\langle \text{numero} \rangle \rightarrow \langle \text{digito} \rangle$ o $\langle \text{numero} \rangle \rightarrow \langle \text{numero} \rangle \langle \text{digito} \rangle$, luego por lo menos en el lenguaje generado todas las palabras tendrán al menos un dígito. Si tomamos el segundo camino y seguimos aplicando sobre número la regla de producción $\langle \text{numero} \rangle \rightarrow \langle \text{numero} \rangle \langle \text{digito} \rangle$, se seguirán añadiendo dígitos uno a uno, luego la palabra puede tener tantos dígitos como se quieran. Luego la única restricción que imponen estas reglas de producción es que la palabra vacía no pertenece al lenguaje generado, es decir, toda palabra del lenguaje $L(G)$ tiene al menos un dígito, lo que se puede representar por $L(G) = \{u | u \in T^* \setminus \{\epsilon\} = T^+\}$. Ya hemos visto que por como están definidas las reglas de producción toda palabra generada por ellas tendrán al menos un dígito, luego está claro que pertenecen al lenguaje de las palabras con al menos un dígito. Para la inversa tomamos una palabra del lenguaje $u \in L$, donde L es el lenguaje de las palabras con al menos un dígito y veamos que se puede obtener a partir de nuestra gramática. Lo único que tenemos que hacer es contar el número de símbolos no terminales de esta, si es uno aplicamos $\langle \text{numero} \rangle \rightarrow \langle \text{digito} \rangle$ y luego

$\langle \text{digito} \rangle \rightarrow 0 | \dots | 9$ donde en esta última ponemos el dígito que corresponde a la palabra. En caso de que sea de longitud mayor que uno aplicamos $\langle \text{numero} \rangle \rightarrow \langle \text{numero} \rangle \langle \text{digito} \rangle$ tantas veces sea necesario hasta que nos quede un numero de símbolos no terminales digito igual a la longitud de la palabra menos 1. Luego aplicamos $\langle \text{numero} \rangle \rightarrow \langle \text{digito} \rangle$ obteniendo así que la palabra esté compuesta de momento por un número de símbolos no terminales igual a la longitud de la palabra y finalmente aplicamos $\langle \text{digito} \rangle \rightarrow 0 | \dots | 9$ substituyendo cada simbolo no terminal $\langle \text{digito} \rangle$ por el número adecuado para obtener la palabra u . Luego hemos visto que $L(G)=L$

17. Dificultad Media

$\{uv \in \{0,1\}^* \text{ tales } u^{-1} \text{ es un prefijo de } v\}$

Defino la gramática generativa $G=(V,T,P,S)$ donde $V = \{S, X, Y\}$, $T = \{0,1\}$, P es el conjunto de las reglas de producción y S es el estado inicial. Y en el conjunto P se incluyen las siguientes reglas de producción, $S \rightarrow XY$, $X \rightarrow 0X0|1X1|\epsilon$ y $Y \rightarrow 0Y|1Y|\epsilon$.

Explicacion: La regla de producción sobre S nos permite únicamente obtener símbolos no terminales X e Y . Estos símbolos a su vez nos permiten generar la palabra vacía. Por su parte la regla de producción sobre Y nos permite generar cualquier subcadena de $\{0,1\}^*$. Por otro lado la regla de producción sobre X nos permite ir generando símbolos no terminales iguales a la derecha y la izquierda del símbolo no terminal X o bien poner la palabra vacía. Esto nos permite crear subcadenas de longitud par con la forma $a_1, \dots, a_n, a_n, \dots, a_1$, lo que se puede reescribir como $u.u^{-1}$. Juntando lo obtenido por la regla de producción X y la regla de producción Y obtenemos $u.u^{-1}.w$ que se puede reescribir $u.v$ donde $u.v \in \{0,1\}^*$ tales que u^{-1} es un prefijo de u .

$\{ucv \in \{a,b,c\}^* \text{ tales que } u \text{ y } v \text{ tienen la misma longitud}\}$

Defino la gramática generativa $G=(V,T,P,S)$ donde $V = \{S, X\}$, $T = \{a,b,c\}$, P es el conjunto de las reglas de producción y S es el estado inicial. Y en el conjunto P se incluyen las siguientes reglas de producción,

$$\begin{aligned} S &\rightarrow XSX|c \\ X &\rightarrow a|b|c. \end{aligned}$$

Explicación: La regla de producción sobre S nos permite añadir un símbolo no terminal X a la derecha y otro a la izquierda del símbolo no terminal S que se sitúa de nuevo en el centro. Si repetimos tantas veces como queramos este proceso obtenemos la cadena de símbolos no terminales, X^nSX^n , si aquí aplicamos la regla de producción sobre X podemos substituir

cada X por a, b o c en el orden que nos convenga obteniendo uSv . Finalmente aplicamos $S \rightarrow c$ y obtenemos $ucv \in \{a, b, c\}^*$ donde la longitud de u y v son iguales.

$$\{u1^n \in \{0, 1\}^* \text{ donde } |u| = n\}$$

Defino la gramática generativa $G=(V,T,P,S)$ donde $V = \{S, X, Y\}$, $T = \{0, 1\}$, P es el conjunto de las reglas de producción y S es el estado inicial. Y en el conjunto P se incluyen las siguientes reglas de producción,

$$\begin{aligned} S &\rightarrow XSY | \epsilon \\ X &\rightarrow 0 | 1 \\ Y &\rightarrow 1 \end{aligned}$$

La regla de producción sobre S nos permite colocar tantos símbolos no terminales X a la izquierda e Y a la derecha del símbolo no terminal S , como queramos dando lugar a X^nSY^n , o bien la palabra vacía. Con esto nos aseguramos que la palabra tenga longitud par pues las reglas de producción sobre X e Y solo permiten generar símbolos no terminales, no la palabra vacía. La regla de producción sobre X nos permite sustituir todas las X por 1 o 0 , según nos convenga generando una subcadena cualquiera u de $\{0, 1\}^*$ de longitud n . Por otro lado la regla de producción sobre Y nos permite únicamente sustituir las Y por 1 , dando lugar a 1^n . Si lo juntamos todo llegamos a $uS1^n$. Finalmente se aplica la regla de producción sobre $S \rightarrow \epsilon$ obteniendo una palabra con la forma $u1^n$ donde u tiene longitud n .

Ahora tomemos una palabra $w \in L$. Esta palabra tendrá la forma $u, 1^n$, donde u tiene longitud n . Aplico la regla de producción $S \rightarrow XSY$ n veces con lo que llegaremos a X^nSY^n . Seguidamente usamos la regla de producción $S \rightarrow \epsilon$ obteniendo así X^nY^n . Como la regla de producción sobre Y únicamente nos permite generar ceros llegamos a X^n1^n . Ahora finalmente usamos la regla de producción $X \rightarrow 0 | 1$ sustituyendo cada símbolo terminal X por el símbolo no terminal que más convenga hasta obtener la subcadena u , con lo que tendremos finalmente la palabra $u, 1^n$ que es una palabra cualquiera del lenguaje. Si $n=0$ simplemente se aplica al principio $S \rightarrow \epsilon$.

$\{a^n b^n a^{n+1} \in \{a, b\}^* \text{ con } n \geq 0\}$ Defino la gramática generativa $G=(V,T,P,S)$ donde $V = \{S, X, Y\}$, $T = \{a, b\}$, P es el conjunto de las reglas de producción y S es el estado inicial. Y en el conjunto P se incluyen las siguientes reglas de producción,

$$\begin{aligned}
S &\rightarrow abaa|abXa \\
bX &\rightarrow Xb \\
aX &\rightarrow aabY \\
Yb &\rightarrow bY \\
Ya &\rightarrow aaX \\
Ya &\rightarrow aaa
\end{aligned}$$

La regla de producción sobre S nos permite o bien que la palabra generada sea abaa o bien que se genere la cadena abXa. Para este segundo caso entramos en un comportamiento cíclico. Primero se haría $S \rightarrow abXa$, seguido de $bX \rightarrow Xb$, dando lugar a aXba, seguido de $aX \rightarrow aabY$ dando lugar a aabYba, seguido de $Yb \rightarrow bY$ dando lugar a aabbYa, seguido de o bien $Ya \rightarrow aaa$ que ya nos da lugar a una palabra del tipo $a^n.b^n.a^{n+1}$ o bien $Ya \rightarrow Xaa$ que nos da lugar a la cadena aabbXaa, de nuevo se volverían a aplicar las reglas en el orden dado repitiendo el ciclo así y aumentando el valor del exponente n en una unidad.

18. Dificiles

$\{u0v \in \{0,1\}^* \text{ tales que } u^{-1} \text{ es un prefijo de } v\}$

Defino la gramática generativa $G=(V,T,P,S)$ donde $V = \{S, X, Y\}$, $T = \{0, 1\}$, P es el conjunto de las reglas de producción y S es el estado inicial. Y en el conjunto P se incluyen las siguientes reglas de producción, $S \rightarrow XY|\epsilon$, $X \rightarrow 0X0|1X1|0$ y $Y \rightarrow 0Y|1Y|\epsilon$.

La regla de producción sobre S nos permite generar o bien la palabra vacía o bien la cadena de símbolos terminales XY. Las regla de producción sobre X nos permite generar cualquier subcadena de $\{0,1\}^*$ con la forma $a_1, \dots, a_n, 0, a_n, \dots, a_1$ y $n > 0$. Por último la regla de producción sobre Y nos permite generar cualquier subcadena de $\{0,1\}^*$. Si juntamos lo anterior nos quedará una palabra con la forma $a_1 \dots a_n, 0, a_n \dots a_1.w$ lo cual podemos reescribir como $u, 0.u^{-1}.w$, lo cual se puede reescribir como $u, 0.v$ con u y v en $\{0,1\}^*$