

PRÁCTICA 1

David Muñoz Sánchez 3ºA2

Ejercicio 1:

Tenemos un lenguaje con palabras formadas por los símbolos terminales a y b . A una iteración n -ésima de a (un valor mayor o igual que 0), se le concatena una iteración m -ésima de b . El número de b tiene que estar comprendido entre el número de a y su triple.

Nota: para el caso especial en el que $n=0$, $m=0$ y por tanto se genera la palabra vacía.

Table Text Size		
LHS		RHS
S	\rightarrow	λ
S	\rightarrow	aXYY
X	\rightarrow	b
X	\rightarrow	aXZYY
Y	\rightarrow	b
Y	\rightarrow	λ
Z	\rightarrow	b

En esta imagen de JFLAP podemos ver las reglas que hacen que nuestra gramática genere el lenguaje pedido. Desde el símbolo inicial podemos ir al vacío (véase la nota) o a una secuencia formada por aXYY. X e Y sirven para lo mismo (alcanzar b), pero tienen una ligera diferencia. X nos sirve además para añadir a y sus correspondientes X e Y. Para asegurar que el número de b sea como mínimo igual que el número de a, tenemos Z (presente en XZYY), que solo alcanza b. Además, Y puede ir al vacío, puesto que alcanzar el máximo número de b en nuestra palabra no es una condición necesaria sino suficiente.

Ejercicio 2:

Tenemos un lenguaje que en resumidas cuentas, genera palabras compuestas por una iteración de a, b y c concatenadas en ese orden y donde por cada a hay una c y por cada b hay dos c. El ejercicio no pone posibles restricciones a los valores de n, m y k, por lo que he supuesto que pueden valer 0.

Table Text Size		
LHS		RHS
S	→	λ
S	→	XZ
X	→	aXZY
Z	→	bZYY
Y	→	c
X	→	λ
Z	→	λ

En esta imagen podemos ver que partimos del símbolo inicial o al vacío (por lo arriba dicho) o a XZ. Por un lado, X nos sirve para añadir a y sus correspondientes c al final de la palabra, con la b en medio (que la alcanzamos con Z). De forma análoga, Z nos sirve para añadir b, con sus correspondientes c al final. La c se alcanza únicamente desde Y.

Ejercicio 3:

Table Text Size	
LHS	RHS
S	→ pX
S	→ gZ
X	→ dX
X	→ fY
Y	→ dY
Y	→ gZ
Y	→ Q
Z	→ dZ_1
Z	→ fZ_2
Z_1	→ dZ_3
Z_1	→ fZ_4
Z_2	→ dZ_5
Z_2	→ fZ_6
Z_3	→ dV
V	→ dV
V	→ fH
H	→ fH
H	→ Q
Z_3	→ fH
Z_4	→ dW
W	→ dW
W	→ fH
W	→ Q
Z_4	→ Q
Z_5	→ dW
Z_6	→ dW
Z_5	→ Q
X	→ Q
Z_4	→ fH
Z_5	→ fH

Table Text Size	
LHS	RHS
Y	→ dY
Y	→ gZ
Y	→ Q
Z	→ dZ_1
Z	→ fZ_2
Z_1	→ dZ_3
Z_1	→ fZ_4
Z_2	→ dZ_5
Z_2	→ fZ_6
Z_3	→ dV
V	→ dV
V	→ fH
H	→ fH
H	→ Q
Z_3	→ fH
Z_4	→ dW
W	→ dW
W	→ fH
W	→ Q
Z_4	→ Q
Z_5	→ dW
Z_6	→ dW
Z_5	→ Q
X	→ Q
Z_4	→ fH
Z_5	→ fH
Q	→ r
Q	→ pX
Q	→ gZ

La gramática obtenida es de tipo 3 según la jerarquía de Chomsky, puesto que todas las reglas de producción son de la forma $X \rightarrow a$ ó $X \rightarrow aX$, siendo a símbolo terminal y X variable.

Partimos de gZ o pX . El caso más sencillo es que sea una sala pequeña (p). A partir de p , podemos poner todos los enemigos débiles que queramos (d) y como máximo uno fuerte (f). El caso más complejo es partir de g . En las salas grandes (g), tiene que haber al menos un enemigo fuerte y uno débil. Además, los dos primeros enemigos pueden aparecer en cualquier orden y los demás siempre mostrando primero los débiles y después los fuertes. Es por ello, que a partir de los dos primeros, se presentan cuatro casos que han sido tratados en las reglas de producción (ff , fd , df , dd). Además, cada vez que se puede terminar una cadena, alcanzamos Q , y mediante Q podemos alcanzar r (para finalizar la palabra) o introducir una nueva sala (gZ , pX).