Lenguajes y gramáticas

Prof. Enrique Vílchez Quesada

Universidad Nacional de Costa Rica

Introducción

Los lenguajes naturales utilizados por los seres humanos para comunicarse entre sí, tienen la característica de recurrir a un conjunto de reglas muy complejas que determinan la sintaxis correcta de las oraciones o frases comprendidas por quienes dominan dicho lenguaje. Estas reglas estructuran una gramática que define las oraciones permitidas y no permitidas, en el contexto de un conjunto de composiciones o producciones base, por medio de las cuales es posible deducir cualquier otra expresión que pertenezca al lenguaje.
En este capítulo se estudiará el concepto de lenguaje formal como un modelo teórico que ha servido para el desarrollo de los lenguajes de programación actuales y algunas otras formas de representación de los lenguajes naturales.

Introducción

En informática estos principios resultan fundamentales, pues a través de estos modelos, se han creado distintos esquemas de caracterización que han permitido a las computadoras modernas, interactuar con el usuario mediante flujos de información que tratan de simular la comunicación humana.

Se iniciará el tema con algunos conceptos básicos.

Lenguajes formales y gramáticas

Prof. Enrique Vílchez Quesada

Universidad Nacional de Costa Rica

Definición 8.1. Alfabeto del lenguaje

Un lenguaje formal es un conjunto de arreglos, cadenas de caracteres o “strings” obtenidos de un conjunto finito de símbolos A, usualmente llamado “alfabeto”.

Definition (8.1)

Sea A un conjunto finito no vacío. Un lenguaje formal sobre A, denotado $L(A)$, es un subconjunto de A^*, donde A^* está constituido por todas las cadenas de caracteres que es posible construir con cada uno de los elementos de A. Al conjunto A se le denomina “alfabeto” o “abecedario” del lenguaje.

- Los lenguajes de los autómatas de estado finito determinísticos diseñados en los ejemplos 7.15, 7.16, 7.17, 7.18, 7.19 y 7.20 del capítulo anterior, son lenguajes formales sobre el conjunto τ de acuerdo con la definición 1.
- Los lenguajes formales son el resultado de un conjunto de reglas que determinan cuáles elementos pertenecen y no pertenecen a este conjunto. Estas reglas estructuran lo que se denomina una "gramática".

Definición 8.2. Gramática de un lenguaje

Definition (8.2)

Una gramática G en su forma normal es una 4−tupla $G = (\sigma, \tau, P, \sigma^*)$ con:

1. σ un conjunto de símbolos denominados: "símbolos no terminales".
2. τ una colección de elementos llamados: "símbolos terminales".

Definición 8.2. Continuación

3. P un subconjunto finito de $\left[(\sigma \cup \tau)^* - \tau^*\right] \times (\sigma \cup \tau)^*$, denominado "conjunto de composiciones" o "producciones" de la gramática. Una composición, por lo tanto, es un par ordenado donde la primera componente es una hilera de símbolos no terminales y/o terminales, con la restricción de contener al menos un símbolo no terminal y la segunda es una hilera de símbolos terminales y/o no terminales sin ninguna condición adicional. Usualmente una producción (α, β), se denota: $\alpha \to \beta$ y se lee "α deriva a β".
4. σ^* es un símbolo no terminal inicial. De este elemento parte cualquier derivación producto del conjunto P.

Comentario sobre la definición 2

La definición 2 parece en principio un poco complicada, sin embargo, en esencia quiere decir que: los símbolos no terminales representan las palabras que usa la gramática G para construir cada uno de los elementos del lenguaje formal, mediante una serie de sustituciones permitidas por las producciones de G. Por otro lado, los símbolos terminales, son los elementos a través de los cuales, se elaboran los "strings" que forman parte del lenguaje de la gramática. Por una convención en este texto, se utilizarán casi siempre, letras mayúsculas del abecedario para representar elementos de σ y letras minúsculas para denotar elementos de τ. Además, se tomará a $\sigma^* = S$, en todos los ejemplos a exponer en este capítulo. Estas consideraciones son primordiales para tener una correspondencia adecuada con el funcionamiento interno de las distintas instrucciones integradas en el paquete **VilCretas**, que se relacionan con este tema.

Example (8.1)

Sea la gramática $G = (\sigma, \tau, P, \sigma^*)$ con: $\sigma = \{S, A, B, C, D, E\}$, $\tau = \{a, b, c\}$, $P = \{S \to aAB,\ S \to aB,\ A \to aAC,\ A \to aC,\ B \to Dc,\ AD \to Ab,\ C \to b,\ CB \to CE,\ CE \to DE,\ DE \to Dc\}$ y $\sigma^* = S$. Verifique que la hilera "$aaabbc$" es un elemento del lenguaje de G.

Solución del ejemplo 8.1

Para deducir si el "string" "*aaabbc*" forma parte del lenguaje de G, se inicia con cualquier composición que comience con el símbolo no terminal inicial $\sigma^* = S$. En este sentido, este símbolo es similar al estado inicial en una *MEF*, o bien, en un autómata, cuando se procesa una hilera de símbolos de entrada. En este caso, se utilizará de base:

$$S \to aAB \tag{1}$$

Luego, $A \to aAC$, que se lee: "A deriva a aAC", lo anterior significa que podemos sustituir A por aAC en cualquier expresión que la contenga. Si realizamos esta sustitución en 1, se obtiene:

$$aAB \to aaACB$$

Solución del ejemplo 8.1

A su vez en P, $CB \to CE$, $CE \to DE$ y $DE \to Dc$, es decir, es válido reemplazar CB por Dc, entonces:

$$aaACB \to aaADc$$

También, $AD \to Ab$, por consiguiente:

$$aaADc \to aaAbc$$

Además, $A \to aC$ y $C \to b$, finalmente:

$$aaAbc \to aaaCbc \to aaabbc$$

En conclusión, S deriva a la hilera "$aaabbc$", por lo tanto, este "string" es un elemento del lenguaje de G.

Solución del ejemplo 8.1

El alumno puede cuestionarse por qué iniciar con la producción $S \rightarrow aAB$ y no $S \rightarrow aB$, la razón obedece a la conveniencia acarreada por "*aaabbc*", pues si se comienza con la composición $S \rightarrow aB$, no es posible obtener dicha hilera.

En un proceso de derivación como el descrito en este ejercicio, siempre es necesario justificar paso a paso cada una de las sustituciones llevadas a cabo. Con el objetivo de facilitar esta tarea, se pueden enumerar las reglas de composición del conjunto P :

1.	$S \rightarrow aAB$	6.	$AD \rightarrow Ab$
2.	$S \rightarrow aB$	7.	$C \rightarrow b$
3.	$A \rightarrow aAC$	8.	$CB \rightarrow CE$
4.	$A \rightarrow aC$	9.	$CE \rightarrow DE$
5.	$B \rightarrow Dc$	10.	$DE \rightarrow Dc$

Solución del ejemplo 8.1

Luego, el proceso de derivación en "$aaabbc$" se podría también, expresar así:

$S \rightarrow aAB$ por 1

$\rightarrow aaACB$ por 3

$\rightarrow aaADc$ por 8, 9 y 10

$\rightarrow aaAbc$ por 6

$\rightarrow aaaCbc$ por 4

$\rightarrow aaabbc$ por 7

Nota

El método empleado para derivar el "string" "*aaabbc*" ha consistido en obtener la hilera al experimentar previamente una serie de "pruebas y errores". Si el estudiante entabla la solución de este ejercicio por cuenta propia sin consultar su desarrollo, muy problemente no llegará de manera directa en un solo intento, a las derivaciones requeridas. Esto es normal y no debe ocasionar ningún tipo de frustración al respecto, pues en un inicio no se conocen cuáles son las reglas de composición que permitirán construir la hilera de interés. En este capítulo se mencionará en una sección posterior, una estrategia de trabajo más sistemática para determinar si un "string" pertenece al lenguaje, aplicable a cierto tipo de gramáticas mediante el uso de un autómata de estado finito equivalente.

Definición 8.3. Lenguaje derivado

Todos los arreglos sobre el conjunto τ en una gramática G, que se han obtenido de las producciones contenidas en P partiendo de $\sigma^* = S$, determinan el lenguaje derivado por G. Veamos la siguiente definición.

Definition (8.3)

Sea $G = (\sigma, \tau, P, \sigma^*)$ una gramática. Si $\alpha \to \beta \in P$ y γ, $\delta \in (\sigma \cup \tau)^*$ entonces se dice que "$\gamma\alpha\delta$ deriva a $\gamma\beta\delta$" y se denota: $\gamma\alpha\delta \to \gamma\beta\delta$. Además, cualquier hilera de símbolos de $(\sigma \cup \tau)^*$ que se derive de $\gamma\beta\delta$ se considera también, una derivación de $\gamma\alpha\delta$. El lenguaje generado por G, representado como $L(G)$, es el conjunto de todos los "strings" de símbolos terminales τ, que se derivan de $\sigma^* = S$, a través de las composiciones de G.

Comentario sobre la definición 4

El alumno debe tener claro según la definición 4, que el lenguaje asociado a una grámatica G $(L(G))$ es el conjunto de todas las hileras de símbolos terminales (conjunto τ) que se pueden construir mediante las derivaciones implicadas por las reglas de composición establecidas en P, comenzando siempre con el símbolo no terminal $\sigma^* = S$. Esto quiere decir, que nunca una hilera perteneciente a $L(G)$ podrá contener un símbolo no terminal (conjunto σ).

Como el alumno comprobará, encontrar el lenguaje generado por una gramática G en algunas ocasiones no es una tarea simple. Consideremos otro ejemplo.

Example (8.2)

Dada la gramática $G = (\sigma, \tau, P, \sigma^*)$ con: $\sigma = \{S, A, B, C, D\}$, $\tau = \{a, b, c\}$, $P = \{S \to aA,\ S \to aB,\ A \to aC,\ B \to Dc,\ C \to bC,\ C \to b,\ D \to c,\ D \to Dc\}$ y $\sigma^* = S$, establezca si las hileras "*aabbbbbb*", "*acccccc*" y "*aabbbccc*" son elementos de $L(G)$.

Solución del ejemplo 8.2

En primera instancia se enumerarán las reglas de composición:

$$\begin{array}{llll} 1. & S \to aA & 5. & C \to bC \\ 2. & S \to aB & 6. & C \to b \\ 3. & A \to aC & 7. & D \to c \\ 4. & B \to Dc & 8. & D \to Dc \end{array} \tag{2}$$

Solución del ejemplo 8.2

- Con relación a la hilera "*aabbbbbb*", se tiene:
  $S \rightarrow aA$ por 1 $\rightarrow aaC$ por 3 $\rightarrow aabC$ por 5 $\rightarrow aabbbbbC$ usando 5 cuatro veces $\rightarrow aabbbbbb$ por 6
  El "string" "*aabbbbbb*" sí pertenece al lenguaje derivado por la gramática G.
- Luego, en "*acccccccc*":
  $S \rightarrow aB$ por 2 $\rightarrow aDc$ por 4 $\rightarrow aDcc$ por 8 $\rightarrow aDcccccc$ usando 8 cuatro veces $\rightarrow acccccccc$ por 7
  La hilera "*acccccccc*" sí pertenece al conjunto $L(G)$.

Solución del ejemplo 8.2

- Finalmente, con respecto a "*aabbbccc*" se observa por lo ya resuelto, que si se comienza con $S \rightarrow aA$ se construye sin problemas "*aabbb*", sin embargo, a partir de ese punto, no es plausible agregar las tres c faltantes. Si se inicia mediante la regla $S \rightarrow aB$ no es viable el camino, dado que la hilera nunca iniciaría con dos a. Por este motivo, "*aabbbccc*" no es consecuencia de las producciones pertenecientes a P y por esta razón, no forma parte del lenguaje de la gramática G.

En conclusión, "*aabbbbbb*" $\in L(G)$, "*acccccc*" $\in L(G)$ y "*aabbbccc*" $\notin L(G)$.

Solución del ejemplo 8.2

Por otra parte, la librería **VilCretas** provee el comando `ElementoLenguajeQ`, una función booleana encargada de determinar dada una hilera de símbolos terminales, si ésta pertenece o no a una gramática clasificada como "libre del contexto". La definición 8 como se aboradará después, justifica que la gramática G de este ejercicio es precisamente de ese tipo. El ejemplo, por lo tanto, puede ser resuelto con apoyo de software, veamos:

In[ ] :=

```
Hileras = {"aabbbbbb", "acccccc", "aabbbccc"};
Table[ElementoLenguajeQ[{"S->aA|aB", "A->aC", "B->Dc",
"C->bC|b", "D->c|Dc"}, α], {α, Hileras}]
```

Out[ ] =

{True, True, False}

Solución del ejemplo 8.2

En el código se pasa a la instrucción `ElementoLenguajeQ` las hileras de interés empleando un `Table`. Además, se observa que `ElementoLenguajeQ` recibe la gramática incluyendo en un vector cada una de las reglas de composición como un "string". Si existen producciones con el mismo inicio se utiliza "|" con el propósito de agregarlas en una sola expresión. Así por ejemplo, `S->aA|aB` explicita en *Wolfram* las reglas 1 y 2 compartidas en 2. La sentencia `ElementoLenguajeQ` siempre presupone a "*S*" como el símbolo no terminal inicial de la gramática. La sintaxis descrita con anterioridad, para crear a G en el software *Mathematica* es compartida por todas las funciones del paquete **VilCretas** donde se requiere como argumento una gramática en su forma normal.

Solución del ejemplo 8.2

También se aclara que `ElementoLenguajeQ` posee dos opciones: `derivaciones -> True` y `pruebas -> entero`. La primera facilita visualizar las sustituciones internas que usa el comando para concluir si la hilera recibida es parte o no del lenguaje de G. La segunda, mediante el número "`entero`" precisado, aumenta o disminuye una cantidad de pruebas internas (por defecto su valor es igual a 30) que ejecuta la instrucción para comprobar el valor lógico de salida. Se sugiere al estudiante ser cauteloso con el uso de `ElementoLenguajeQ` cuando retorne un **Out[ ]** igual a False, a consecuencia de que el comando podría devolver ese valor lógico, aunque la hilera en cuestión sí pertenezca al lenguaje de la gramática. De allí, la necesidad de utilizar la opción `pruebas` con distintos valores de "`entero`" mayores a 30, emprendiendo con ello, varias verificaciones del resultado False.

Descargue un archivo

https://www.escinf.una.ac.cr/discretas/Archivos/Lenguajes/File-186.zip

Uso de la sentencia `ElementoLenguajeQ`.

Explicación en video

https://youtu.be/wI2sx3rk4_Q

Definición 8.4. Notación de *Backus-Naur*

Además de la "forma normal" de una gramática G, existe otro mecanismo notacional para poder representarla: la notación de "*Backus-Naur*". Algunos lenguajes de programación en sus inicios fueron diseñados mediante el uso de este tipo de simbología.

Definition (8.4)

Sea $G = (\sigma, \tau, P, \sigma^*)$ una gramática. Si cada composición $\alpha \to \beta_i$, $1 \leq i \leq n$, $n \in \mathbb{N}$ es sustituida por $\alpha ::= \beta_1 \mid \beta_2 \mid \cdots \mid \beta_n$ y los símbolos no terminales se encierran en $<>$ se dice que la gramática se ha expresado en notación de *Backus-Naur*, cuyo acrónimo es **BNF**.

Example (8.3)

Exprese la gramática del ejemplo 5 en notación **BNF**.

Solución del ejemplo 8.3

Empleando la definición 6 y el enunciado del ejemplo 5, se obtiene:

$$
\begin{aligned}
&< S >::= a < A >|\ a < B >\\
&< A >::= a < C >\\
&< B >::=< D > c\\
&< C >::= b < C >|\ b\\
&< D >::= c\ |< D > c\\
&\sigma^* =< S >
\end{aligned}
$$

Solución del ejemplo 8.3

La librería **VilCretas** incluye además, la sentencia `NotationBNF` que al recibir una gramática (tal y como se explicó en el ejemplo 5 para el comando `ElementoLenguajeQ`) la devuelve en notación de *Backus-Naur*. En este ejercicio, `NotationBNF` retorna:

In[ ] :=

```
NotationBNF[{"S->aA|aB", "A->aC", "B->Dc", "C->bC|b",
"D->c|Dc"}]
```

Out[ ] =

<S>::=a<A>|a<B>
<A>::=a<C>
<B>::=<D>c
<C>::=b<C>|b
<D>::=c|<D>c
σ*=S

Solución del ejemplo 8.3

Adicionalmente, **VilCretas** cuenta con el comando `NotationFNormal` encargado de expresar una gramática G en su forma normal, es decir, mediante la 4−tupla $G = (\sigma, \tau, P, \sigma^*)$ señalada en la definición 2. Al recurrir a `NotationFNormal` en el presente ejemplo:

In[ ] :=

```
NotationFNormal[{"S->aA|aB", "A->aC", "B->Dc", "C->bC|b",
"D->c|Dc"}]
```

Out[ ] =

Símbolos no terminales: {A,B,C,D,S}
Símbolos terminales: {a,b,c}
Producciones o reglas de composición: {A->aC, B->Dc, C->b, C->bC, D->c, D->Dc, S->aA, S->aB}
σ*=S

Descargue un archivo

https://www.escinf.una.ac.cr/discretas/Archivos/Lenguajes/File-187.zip

Utilización de la instrucción `NotationBNF`.

Explicación en video

https://youtu.be/P-0YI4R95KQ

Empleo de la sentencia `NotationFNormal`.

Explicación en video

https://youtu.be/gLmobddtY8w

Definición 8.5. Tipos de gramáticas

Las gramáticas se pueden clasificar de acuerdo con la forma de las composiciones que las caracterizan. La definición que prosigue establece tres tipos distintos de gramáticas.

Definition (8.5)

Sea $G = (\sigma, \tau, P, \sigma^*)$ una gramática y ε el arreglo de caracteres vacío o nulo, entonces:

1. Si toda producción de P tiene la forma: $\gamma\alpha\delta \to \gamma\beta\delta$, con $\alpha \in \sigma$, $\beta \in (\sigma \cup \tau)^* - \{\varepsilon\}$ y γ, $\delta \in (\sigma \cup \tau)^*$, se dice que G y su lenguaje son sensibles al contexto.
2. Si toda composición de P tiene la forma: $\alpha \to \beta$, con $\alpha \in \sigma$ y $\beta \in (\sigma \cup \tau)^*$, G y su lenguaje son libres del contexto.
3. Si toda producción de P tiene la forma: $\alpha \to x\beta$, $\alpha \to x$, $\alpha \to \varepsilon$, con α, $\beta \in \sigma$ y $x \in \tau$, se dice que G y su lenguaje son regulares.

Comentario sobre la definición 8

Una gramática G es sensible al contexto cuando todas sus reglas de composición tienen la estructura $\gamma\alpha\delta \to \gamma\beta\delta$, donde α es un símbolo no terminal, β es una hilera de símbolos terminales y/o no terminales, no pudiendo ser igual a vacía y tanto γ como δ son hileras de símbolos terminales y/o no terminales incluyendo la hilera vacía.

Una gramática G es libre del contexto si todas las reglas de composición tienen la forma $\alpha \to \beta$, siendo α un símbolo no terminal y β una hilera de símbolos terminales y/o no terminales, cabiendo la posibilidad de ser β igual a vacía. Es decir, una gramática es libre del contexto si todas sus producciones parten de un símbolo no terminal (solo uno). Si en una gramática libre del contexto nunca ocurre una producción donde $\beta = \phi$ entonces se aprecia que G cumple también con la definición de sensible al contexto. Las gramáticas libres del contexto es usual que se representen por medio de la notación **BNF**.

Comentario sobre la definición 8

Finalmente, una gramática G es regular si todas sus producciones van de un símbolo no terminal (NT) a una hilera con dos elementos: uno terminal (T) y otro no terminal (NT) es ese orden estricto, o bien, si las reglas de composición van de un símbolo no terminal (NT) a otro terminal (T). Es decir, solo existen dos tipos de estructura en las reglas gramaticales sin arreglos nulos: $NT \to T\ NT$, o bien, $NT \to T$. También, la definición admite $NT \to \varepsilon$ siendo ε la hilera vacía. En función de ello, el alumno debe notar que toda gramática regular satisface además, la definición de libre del contexto.

Comentario sobre la definición 8

Como consecuencia de todo lo anterior, si se desea clasificar una gramática G en alguno de los tipos señalados en la definición 8 es indispensable analizar primero si cumple con el concepto de gramática regular, si no es así, se estudia la condición de libre del contexto y sino, la propiedad de sensible al contexto.

No toda gramática G será clasificable en alguno de los tipos mencionados en la definición 8, existen gramáticas que no son sensibles al contexto, ni libres y tampoco regulares.

Example (8.4)

Clasifique usando la definición 8, las gramáticas de los ejemplos 3 y 5.

Solución del ejemplo 8.4

La gramática G del ejemplo 3 no es regular pues hay reglas gramaticales que rompen con las estructuras $NT \to T\ NT$, o bien, $NT \to T$. Un contrajemplo se tiene en la composición $CB \to CE$, donde CE representa una hilera con dos símbolos no terminales. También, la gramática no es libre del contexto dado que hay reglas gramaticales que no parten de un símbolo no terminal (solo uno), la producción $CB \to CE$ en ese sentido, parte de dos símbolos no terminales.

Solución del ejemplo 8.4

La gramática sí se puede clasificar como sensible al contexto, interpretando:

1. $\underbrace{S}_{\alpha} \to \underbrace{aAB}_{\beta}, \gamma = \delta = \phi$
2. $\underbrace{S}_{\alpha} \to \underbrace{aB}_{\beta}, \gamma = \delta = \phi$
3. $\underbrace{A}_{\alpha} \to \underbrace{aAC}_{\beta}, \gamma = \delta = \phi$
4. $\underbrace{A}_{\alpha} \to \underbrace{aC}_{\beta}, \gamma = \delta = \phi$
5. $\underbrace{B}_{\alpha} \to \underbrace{Dc}_{\beta}, \gamma = \delta = \phi$
6. $\underbrace{A}_{\gamma}\underbrace{D}_{\alpha} \to \underbrace{A}_{\gamma}\underbrace{b}_{\beta}, \delta = \phi$
7. $\underbrace{C}_{\alpha} \to \underbrace{b}_{\beta}, \gamma = \delta = \phi$
8. $\underbrace{C}_{\gamma}\underbrace{B}_{\alpha} \to \underbrace{C}_{\gamma}\underbrace{E}_{\beta}, \delta = \phi$
9. $\underbrace{C}_{\alpha}\underbrace{E}_{\delta} \to \underbrace{D}_{\beta}\underbrace{E}_{\delta}, \gamma = \phi$
10. $\underbrace{D}_{\gamma}\underbrace{E}_{\alpha} \to \underbrace{D}_{\gamma}\underbrace{c}_{\beta}, \delta = \phi$

Solución del ejemplo 8.4

Por otro lado, la gramática G del ejemplo 5 tampoco es regular, la producción $D \to Dc$ tiene la estructura $NT \to NT\ T$ no cumpliéndose con el orden establecido en el concepto de regular para las reglas de composición. Esta gramática sí cumple con la definición de libre del contexto pues todas sus reglas gramaticales parten de un símbolo no terminal (solo uno).

- A continuación se socializarán dos interesantes gramáticas libres del contexto. La primera se basa en las reglas gramaticales del idioma español y la segunda tiene como lenguaje un conjunto numérico.

Example (8.5)

Encuentre el lenguaje asociado a la gramática libre del contexto G :

$$
\begin{aligned}
&< \text{oración} >::=< \text{sujeto} > b < \text{predicado} > .\\
&< \text{sujeto} >::= \text{Roberto} \mid \text{María}\\
&< \text{predicado} >::=< \text{verbo} > b < \text{complemento} > b < \text{adverbio} >\\
&< \text{verbo} >::= \text{estudia} \mid \text{examina}\\
&< \text{complemento} >::= \text{problemas}\\
&< \text{adverbio} >::= \text{siempre} \mid \text{frecuentemente}\\
&\sigma^* =< \text{oración} >
\end{aligned}
$$

donde b simboliza el espacio en blanco.

Solución del ejemplo 8.5

Con la finalidad de facilitar los procesos de derivación, se modificará la notación de los símbolos no terminales y terminales de la gramática original:

Símbolo no terminal	**Símbolo terminal**
oración: S	Roberto: r
sujeto: A	María: m
predicado: B	estudia: e
verbo: C	examina: x
complemento: D	problemas: p
adverbio: E	siempre: s
	frecuentemente: f

Solución del ejemplo 8.5

Luego, la gramática libre del contexto G de este ejemplo, se reescribiría así:

1. $< S >::=< A > b < B >$.
2. $< A >::= r \mid m$
3. $< B >::=< C > b < D > b < E >$
4. $< C >::= e \mid x$
5. $< D >::= p$
6. $< E >::= s \mid f$

   $\sigma^* =< S >$

Solución del ejemplo 8.5

Un elemento del lenguaje de esta gramática puede ser obtenido de la siguiente manera:
$S \to AbB.$ por 1 $\to rbB.$ por 2 $\to rbCbDbE.$ por 3 $\to rbebpbf.$ por 4, 5 y 6.
La hilera de símbolos terminales "*rbebpbf*." se traduce como: "Roberto estudia problemas frecuentemente.". Aquí es interesante notar, que cualquier elemento del lenguanje de G es una hilera con un significado dentro del idioma español, pues G está fundamentada en las reglas gramaticales de esa lengua.

Solución del ejemplo 8.5

Realizando otros procesos de derivación similares al ya expuesto, se concluye que los elementos del lenguaje de la gramática G son las oraciones:

"$mbebpbf$.": "María estudia problemas frecuentemente."
"$mbebpbs$.": "María estudia problemas siempre."
"$mbxbpbf$.": "María examina problemas frecuentemente."
"$mbxbpbs$.": "María examina problemas siempre."
"$rbebpbf$.": "Roberto estudia problemas frecuentemente."
"$rbebpbs$.": "Roberto estudia problemas siempre."
"$rbxbpbf$.": "Roberto examina problemas frecuentemente."
"$rbxbpbs$.": "Roberto examina problemas siempre."

Solución del ejemplo 8.5

En este ejercicio el conjunto $L(G)$ es finito y tiene cardinalidad igual a 8. El paquete **VilCretas** posee la instrucción `ElementosLenguaje[`G`, `n`, `m`]` cuya funcionalidad reside en determinar en una gramática libre del contexto G, todas las palabras del lenguaje $L(G)$ de tamaño máximo m, que pueden ser generadas por n o menos derivaciones. La sentencia cuenta con la opción `derivaciones -> True`, muestra todas las derivaciones necesarias durante el proceso.

Solución del ejemplo 8.5

En este ejemplo, `ElementosLenguaje` resulta de utilidad para encontrar las ocho oraciones que caracterizan a $L(G)$:

In[ ] :=

```
ElementosLenguaje[{"S->AbB.", "A->r|m", "B->CbDbE",
"C->e|x", "D->p", "E->s|f"}, 6, 8]
```

Out[ ] =

{mbebpbf., mbebpbs., mbxbpbf., mbxbpbs., rbebpbf., rbebpbs., rbxbpbf., rbxbpbs.}

Solución del ejemplo 8.5

Se recomienda al estudiante, ensayar con los valores de n y m en *Wolfram Mathematica* al utilizar `ElementosLenguaje`, hasta obtener un **Out[ ]** satisfactorio.

Descargue un archivo

https://www.escinf.una.ac.cr/discretas/Archivos/Lenguajes/File-188.zip

Uso del comando `ElementosLenguaje`.

Explicación en video

https://youtu.be/wQ6VCuUPHiw

Example (8.6)

Verifique que el número $-123{,}654$ pertenece al conjunto $L(G)$ de la gramática G libre del contexto, dada a continuación. Conjeture cuál es su lenguaje.

$$
\begin{array}{l}
<\text{real}>::=<\text{real con signo}>|<\text{real sin signo}> \\
<\text{real con signo}>::=-<\text{parte real}> \\
<\text{real sin signo}>::=<\text{parte real}> \\
<\text{parte real}>::=<\text{entero sin signo}>|\ 0<\text{parte decimal}> \\
|<\text{entero sin signo}><\text{parte decimal}> \\
<\text{parte decimal}>::=\ .<\text{entero sin signo}> \\
<\text{entero sin signo}>::=<\text{dígito}>|<\text{dígito}><\text{entero sin signo}> \\
<\text{dígito}>::=0\ |\ 1\ |\ 2\ |\ 3\ |\ 4\ |\ 5\ |\ 6\ |\ 7\ |\ 8\ |\ 9 \\
\sigma^*=<\text{real}>
\end{array}
$$

Solución del ejemplo 8.6

En este ejercicio, buscando facilitar la derivación de $-123{,}654$, se modificarán los símbolos no terminales de la gramática G, tal y como se detalla a continuación:

Símbolo no terminal
real: S
real con signo: A
real sin signo: B
parte real: C
entero sin signo: D
parte decimal: E
dígito: F

Solución del ejemplo 8.6

Por lo tanto, G quedaría de la forma:

1. $< S >::=< A >|< B >$
2. $< A >::= - < C >$
3. $< B >::=< C >$
4. $< C >::=< D >|\ 0 < E >|< D >< E >$
5. $< E >::= . < D >$
6. $< D >::=< F >|< F >< D >$
7. $< F >::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$

$\sigma^* =< S >$

Solución del ejemplo 8.6

Luego:
$S \to A$ por 1 $\to -C$ por 2 $\to -DE$ por 4 $\to -D.D$ por 5 $\to -FD.FD$ por 6 $\to -FFD.FFD$ por 6 $\to -FFF.FFF$ por 6 $\to -123{,}654$ por 7
En cuyo caso, $-123{,}654 \in L(G)$.
Al analizar el proceso de derivación implicado por $-123{,}654$, se intuye que cualquier número real puede ser obtenido mediante las reglas gramaticales de G, es decir, se conjetura que $L(G) = \mathbb{R}$. Al respecto, en *Mathematica* el siguiente experimento genera 20 números reales pseudoaleatorios con 5 dígitos y comprueba que cada uno es un elemento del lenguaje de la gramática G.

Solución del ejemplo 8.6

Veamos:

In[ ] :=

```
Prueba[n_] := Module[{bandera = 0, i, real, valorLogico,
valorReal, vectorReales = {}}, For[i = 100, i <= 100 + n -
1, SeedRandom[i + RandomInteger[1000]]; real = Random[Real,
{0, 1000}, 5]; vectorReales = Append[vectorReales, real];
valorLogico = ElementoLenguajeQ[{"S->A|B", "A->-C",
"B->C", "C->D|0E|DE", "E->.D", "D->F|FD",
"F->0|1|2|3|4|5|6|7|8|9"}, ToString[real]]; If[valorLogico ==
False, bandera = 1; valorReal = real; Break[]]; i++];
If[bandera == 0, Print["La prueba fue exitosa en: ",
vectorReales], Print["La prueba no fue exitosa en el número
real: ", valorReal]]]
Prueba[20]
```

Solución del ejemplo 8.6

Out[ ] $=$

La prueba fue exitosa en: $\{121.54, 484.10, 613.46, 338.55, 51.456, 903.79, 600.99, 346.29, 474.99, 590.03, 440.65, 702.70, 297.29, 475.91, 613.12, 982.63, 763.76, 295.54, 737.79, 889.43\}$

En cualquier otra ejecución que corra el lector, tal y como se visualiza en este **Out[ ]**, se verificará que los veinte números reales pseudoaleatorios generados pertenecen al conjunto $L(G)$.

Solución del ejemplo 8.6

También, el comando `ElementosLenguaje` podría contribuir con el respaldo de la conjetura. Una prueba consistiría en:

In[ ] :=

```
ElementosLenguaje[{"S->A|B", "A->-C", "B->C",
"C->D|0E|DE", "E->.D", "D->F|FD", "F->0|1|2|3|4|5|6|7|8|9"},
7, 3]
```

Solución del ejemplo 8.6

Out[ ] =

{-0, 0, -00, 0.0, 00, -01, 0.1, 01, -02, 0.2, 02, -03, 0.3, 03, -04, 0.4, 04, -05, 0.5, 05, -06, 0.6, 06, -07, 0.7, 07, -08, 0.8, 08, -09, 0.9, 09, -1, 1, -10, 10, -11, 11, -12, 12, -13, 13, -14, 14, -15, 15, -16, 16, -17, 17, -18, 18, -19, 19, -2, 2, -20, 20, -21, 21, -22, 22, -23, 23, -24, 24, -25, 25, -26, 26, -27, 27, -28, 28, -29, 29, -3, 3, -30, 30, -31, 31, -32, 32, -33, 33, -34, 34, -35, 35, -36, 36, -37, 37, -38, 38, -39, 39, -4, 4, -40, 40, -41, 41, -42, 42, -43, 43, -44, 44, -45, 45, -46, 46, -47, 47, -48, 48, -49, 49, -5, 5, -50, 50, -51, 51, -52, 52, -53, 53, -54, 54, -55, 55, -56, 56, -57, 57, -58, 58, -59, 59, -6, 6, -60, 60, -61, 61, -62, 62, -63, 63, -64, 64, -65, 65, -66, 66, -67, 67, -68, 68, -69, 69, -7, 7, -70, 70, -71, 71, -72, 72, -73, 73, -74, 74, -75, 75, -76, 76, -77, 77, -78, 78, -79, 79, -8, 8, -80, 80, -81, 81, -82, 82, -83, 83, -84, 84, -85, 85, -86, 86, -87, 87, -88, 88, -89, 89, -9, 9, -90, 90, -91, 91, -92, 92, -93, 93, -94, 94, -95, 95, -96, 96, -97, 97, -98, 98, -99, 99}

Solución del ejemplo 8.6

En la salida, se aprecian distintos valores reales de longitud menor o igual a 3, contando el "." decimal, o bien, el signo "$-$", hallados mediante un número máximo de 7 derivaciones.

Descargue un archivo

https://www.escinf.una.ac.cr/discretas/Archivos/Lenguajes/File-189.zip

- El ejecicio propuesto a continuación, ejemplifica cómo hallar usando derivaciones, el conjunto $L(G)$ en una gramática. No siempre es factible determinar el lenguaje vinculado a una gramática, dado que la tarea podría resultar muy extenuante, pese a ello, algunas veces sí es posible hacerlo.

Example (8.7)

Sea la gramática $G = (\sigma, \tau, P, \sigma^*)$ con: $\sigma = \{S, A\}$, $\tau = \{a, b\}$, $P = \{S \to bS,\ S \to aA,\ A \to bA,\ A \to b\}$ y $\sigma^* = S$. ¿Qué tipo de gramática es G? Determine cuál es su lenguaje.

Solución del ejemplo 8.7

De acuerdo con la definición 8, G es una gramática regular (y por lo tanto, libre del contexto) pues todas sus produccíónes tienen la forma: $NT \to T$ NT, o bien, $NT \to T$.

Por otro lado, para deducir el conjunto $L(G)$, se enumerarán las reglas de composición de G :

1. $S \to bS$
2. $S \to aA$
3. $A \to bA$
4. $A \to b$

Solución del ejemplo 8.7

Por la naturaleza del enunciado, donde lo que se pretente es hallar el lenguaje de la gramática G, el objetivo al realizar las derivaciones consistirá en hallar una o varias hileras de caracter general, pertenecientes a $L(G)$. Luego:
$S \to bS$ por 1 $\to b^nS$ usando 1, $n-1$ veces
Cabe aclarar que la expresión b^n representa una hilera con un número de $b's$ igual a n. Continuando con el proceso:
b^nS usando 1, $n-1$ veces $\to b^naA$ por 2 $\to b^nabA$ por 3 $\to b^nab^m$ usando 3, $m-2$ veces y 4 una vez.
Esto permite concluir que:

$$L(G) = \{b^nab^m \mid n, m \in \mathbb{N} \cup \{0\}, n \geq 0, m \geq 1\} \tag{3}$$

Solución del ejemplo 8.7

Aquí, $n \geq 0$ indica que una hilera de $L(G)$ podría no iniciar con b, sino directamente en a y la $m \geq 1$ señala que un "string" del lenguaje de G tendrá que finalizar en al menos una b.

En este ejemplo, el empleo de la instrucción `ElementosLenguaje` verifica lo planteado en 3:

In[ ] :=

```
ElementosLenguaje[{"S->bS|aA", "A->bA", "A->b"}, 10, 10]
```

Solución del ejemplo 8.7

Out[ ] =

{ab, abb, abbb, abbbb, abbbbb, abbbbbb, abbbbbbb, abbbbbbbb, abbbbbbbbb, bab, babb, babbb, babbbb, babbbbb, babbbbbb, babbbbbbb, babbbbbbbb, bbab, bbabb, bbabbb, bbabbbb, bbabbbbb, bbabbbbbb, bbabbbbbbb, bbbab, bbbabb, bbbabbb, bbbabbbb, bbbabbbbb, bbbabbbbbb, bbbbab, bbbbabb, bbbbabbb, bbbbabbbb, bbbbabbbbb, bbbbbab, bbbbbabb, bbbbbabbb, bbbbbabbbb, bbbbbbab, bbbbbbabb, bbbbbbabbb, bbbbbbbab, bbbbbbbabb, bbbbbbbbab}

Todas las hileras de la salida anterior, son "strings" de longitud menor o igual a 10, hallados mediante un número máximo de 10 derivaciones y cada uno tiene la estructura $b^n a b^m$ establecida en 3.

Solución del ejemplo 8.7

Finalmente, se insiste al estudiante que las funciones del paquete **VilCretas** sobre el tema de lenguajes y gramáticas, funcionan siempre y cuando, la gramática G de interés sea libre del contexto, como ocurre en este ejemplo, pues G es regular. De allí, que se hace viable la utilización del comando `ElementosLenguaje`.

Descargue un archivo

`https://www.escinf.una.ac.cr/discretas/Archivos/Lenguajes/File-190.zip`

- Las gramáticas regulares tienen una propiedad muy significativa, a través de ellas es posible construir un autómata de estado finito no determinístico, cuyas hileras aceptadas corresponden al lenguaje de la gramática. Este autómata a su vez, es equivalente a otro de caracter determinístico, tal y como se expuso en el teorema 7.1 del capítulo anterior. Por lo tanto, de una gramática regular siempre es admisible obtener un autómata de estado finito determinístico, donde ambos, poseen el mismo lenguaje. Recíprocamente, todo autómata de estado finito determinístico genera una gramática regular. En la siguiente sección se enuncian formalmente estos resultados.

Gramáticas regulares y autómatas

Prof. Enrique Vílchez Quesada

Universidad Nacional de Costa Rica

Teorema 8.1. ADFN a partir de una gramática regular

Se inicia con el teorema que permite construir un autómata de estado finito no determinístico, a partir de una gramática regular G, donde $L(G)$ es igual al lenguaje del autómata. Bajo esta perspectiva, la gramática y el autómata se consideran equivalentes.

Theorem (8.1)

Sea $G = (\sigma, \tau, P, \sigma^*)$ *una gramática regular. El autómata de estado finito no determinístico* $A = \left(\sigma \cup \{EA\}, \tau, \sigma^*, \Delta, \widehat{A}\right)$ *es tal que* $A^o = L(G)$, *con:* $EA \notin \sigma \cup \tau$, $\widehat{A} = \{EA\}$ *y, siendo* $B \in \sigma \cup \{EA\}$ *y* $x \in \tau$, $\Delta(B, x) = \{D \in \sigma \mid B \to xD \in P\} \cup C$, *donde:*

$$C = \begin{cases} \phi \text{ si } B \to x \notin P \\ \{EA\} \text{ si } B \to x \in P \end{cases}$$

Comentario sobre el teorema 13

La importancia del teorema 13 radica en la metodología de trabajo que introduce con el propósito de encontrar el lenguaje asociado a una gramática regular G. Para hallar $L(G)$, se construye el autómata equivalente a la gramática y luego mediante su diagrama de transición se infieren las hileras generales que el autómata puede aceptar y con ello, los elementos del lenguaje de la gramática. Este autómata se crea tomando sus estados como el conjunto de símbolos no terminales de la gramática regular, agregando un nuevo símbolo EA, correspondiente al único estado aceptado del autómata. De hecho, en este contexto las siglas "EA" hacen referencia a la palabra "estado aceptado". Los símbolos terminales de G conforman los símbolos de entrada del autómata.

Comentario sobre el teorema 13

El símbolo no terminal inicial de G es el estado inicial del autómata. Y finalmente, la función de transición de estados Δ se elabora observando las reglas gramaticales. Si en P, se tienen composiciones de la forma: $B \to xD_1$, $B \to xD_2$, ..., $B \to xD_n$, $n \in \mathbb{N}$, entonces $\Delta(B, x) = \{D_1, D_2, \ldots, D_n\}$ y en ese conjunto se incluye el estado aceptado EA, si además, P contiene la regla $B \to x$. De manera opcional, usando el autómata no determinístico descrito es realizable la elaboración de otro autómata determinístico equivalente.

- Se reitera que la equivalencia entre una gramática regular y un autómata de estado finito, puede ser útil para encontrar el lenguaje asociado a la gramática a partir del diagrama de transición del autómata. Consideremos algunos ejemplos.

Example (8.8)

Determine el lenguaje de la gramática regular del ejemplo 12, a través de un autómata de estado finito.

Solución del ejemplo 8.8

Utilizando el teorema 13, el autómata de interés se construye así:

- Estados: $\{A, EA, S\}$.
- Símbolos de entrada: $\{a, b\}$.
- Estado inicial: S.
- Estados aceptados: $\{EA\}$.

Solución del ejemplo 8.8

- La función Δ es tal que:
  $\Delta(A, a) = \phi$
  $\Delta(A, b) = \{A, EA\}$ por las producciones $A \to bA$ y $A \to b$
  $\Delta(EA, a) = \phi$
  $\Delta(EA, b) = \phi$
  $\Delta(S, a) = \{A\}$ por la regla $S \to aA$
  $\Delta(S, b) = \{S\}$ por la composición $S \to bS$
  La imagen de Δ cuando se tiene el estado EA y cualquier símbolo de entrada del conjunto τ, siempre será igual a vacío, en general, a razón de no existir en las reglas gramaticales ninguna que involucre el símbolo EA ($EA \notin \sigma \cup \tau$).

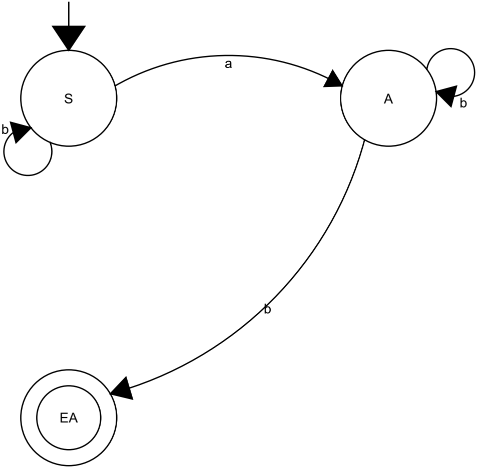
Solución del ejemplo 8.8

- En resumen:

Δ	a	b
A	ϕ	$\{A, EA\}$
EA	ϕ	ϕ
S	$\{A\}$	$\{S\}$

Solución del ejemplo 8.8

El diagrama de transición de este autómata es:



Solución del ejemplo 8.8

Al ser un digrafo relativamente sencillo, no se hace necesario hallar su autómata determinístico equivalente. Al analizar sobre él, las hileras generales que está en capacidad de aceptar, se concluye:

$$L(G) = \{b^n a b^m \mid n, m \in \mathbb{N} \cup \{0\}, n \geq 0, m \geq 1\}$$

Lo cual corrobora la respuesta del ejemplo 12.

El comando `RegularToAutomata` de la librería **VilCretas** automatiza el algoritmo planteado por el teorema 13. La sentencia recibe como argumento una gramática regular y retorna las cinco componentes del autómata no determinístico equivalente y su diagrama de transición. También, integra dos opciones: `dfa -> True` y `colores -> True`. La primera devuelve adicionalmente, el autómata determinístico equivalente al *ADFN* y la segunda agrega color a los estados y a las aristas en los digrafos de salida.

Solución del ejemplo 8.8

De esta manera, la solución ya compartida en este ejemplo, se puede ratificar en *Wolfram* así:

In[ ] :=

```
RegularToAutomata[{"S->bS|aA", "A->bA", "A->b"}]
```

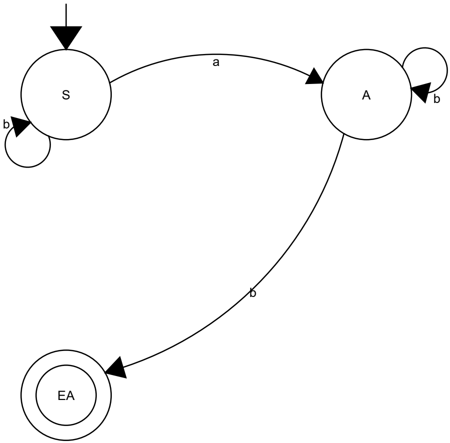
Out[ ] =

Estados: {A, EA, S}
Símbolos de entrada: {a, b}
Estado inicial: S
Estados aceptados: {EA}

$$\begin{pmatrix} & a & b \\ A & \{\} & \{A, EA\} \\ EA & \{\} & \{\} \\ S & \{A\} & \{S\} \end{pmatrix}$$

Solución del ejemplo 8.8

Función de transición de estados con otro formato: {{A, a, {}}, {A, b, {A, EA}}, {EA, a, {}}, {EA, b, {}}, {S, a, {A}}, {S, b, {S}}}



Solución del ejemplo 8.8

Se advierte al alumno emplear de forma apropiada el comando `RegularToAutomata` como un recurso exclusivo de verificación de resultados previos.

Descargue un archivo

https://www.escinf.una.ac.cr/discretas/Archivos/Lenguajes/File-191.zip

Empleo de la instrucción `RegularToAutomata`.

Explicación en video

https://youtu.be/nj-OQ_VxzeM

Example (8.9)

Sea la gramática regular $G = (\sigma, \tau, P, \sigma^*)$ con: $\sigma = \{S, A, B\}$, $\tau = \{a, b\}$, $P = \{S \to aS,\ S \to bA,\ S \to b,\ A \to aA,\ A \to a,\ A \to bB,\ B \to aS,\ B \to bA,\ B \to b\}$ y $\sigma^* = S$. Halle la forma de los elementos del conjunto $L(G)$.

Solución del ejemplo 8.9

Recurriendo al teorema 13, un autómata no determinístico equivalente a la gramática G, corresponde a:

- Estados: $\{A, B, EA, S\}$.
- Símbolos de entrada: $\{a, b\}$.
- Estado inicial: S.
- Estados aceptados: $\{EA\}$.

Solución del ejemplo 8.9

- La función Δ es tal que:
  $\Delta(A,a) = \{A, EA\}$ por las reglas gramaticales $A \to aA$ y $A \to a$
  $\Delta(A,b) = \{B\}$ por la composición $A \to bB$
  $\Delta(B,a) = \{S\}$ por la producción $B \to aS$
  $\Delta(B,b) = \{A, EA\}$ según las reglas $B \to bA$ y $B \to b$
  $\Delta(EA,a) = \phi$
  $\Delta(EA,b) = \phi$
  $\Delta(S,a) = \{S\}$ por la producción $S \to aS$
  $\Delta(S,b) = \{A, EA\}$ por las reglas de composición $S \to bA$ y $S \to b$

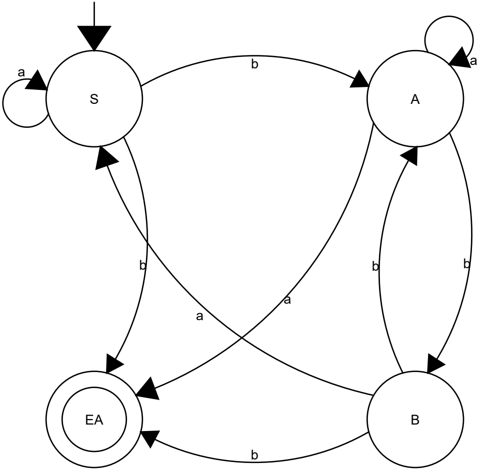
Solución del ejemplo 8.9

- En resumen:

Δ	a	b
A	$\{A, EA\}$	$\{B\}$
B	$\{S\}$	$\{A, EA\}$
EA	ϕ	ϕ
S	$\{S\}$	$\{A, EA\}$

Solución del ejemplo 8.9

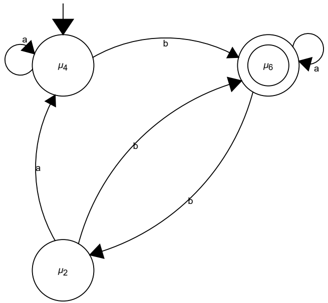
El digrafo que representa este autómata es el siguiente:



(4)

Solución del ejemplo 8.9

En este diagrama de transición, interpretar cuáles son las hileras generales que acepta el *ADFN* es una labor relativamente compleja. Por este motivo, se recomienda encontrar mediante el teorema 7.1 del capítulo anterior, un autómata determinístico equivalente. No se detallará en esta solución su desarrollo paso a paso, sin embargo, al ejecutar el procedimiento ya conocido por el lector, se obtiene el diagrama reducido que prosigue:



Solución del ejemplo 8.9

Este digrafo es claramente más sencillo que el mostrado en 4. Al realizar una lectura minuciosa sobre él, analizando las clases de hileras de símbolos de entrada que acepta, se concluye dos tipos de estructura para los elementos del conjunto $L(G)$: hileras que finalizan en $b^n a^m$, $n, m \in \mathbb{N} \cup \{0\}$, n impar, $m \geq 0$, o bien, "strings" que terminan en $ba^n b^m a^h$, $n, m, h \in \mathbb{N} \cup \{0\}$, $n \geq 0$, m par, $h \geq 0$.
En *Mathematica* el *ADFN* equivalente a G, se halla así:
In[ ] :=

```
RegularToAutomata[{"S->aS|bA|b", "A->aA|a|bB",
"B->aS|bA|b"}]
```

Solución del ejemplo 8.9

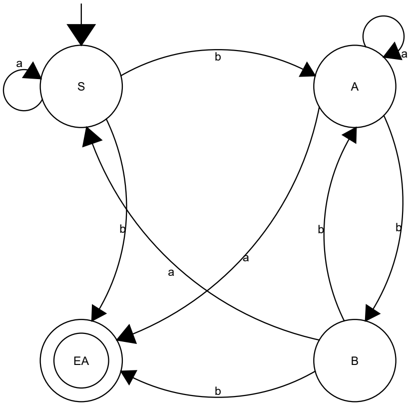
Out[ ] =

Estados: {A, B, EA, S}
Símbolos de entrada: {a, b}
Estado inicial: S
Estados aceptados: {EA}

$$\left(\begin{array}{ccc} & a & b \\ A & \{A, EA\} & \{B\} \\ B & \{S\} & \{A, EA\} \\ EA & \{\} & \{\} \\ S & \{S\} & \{A, EA\} \end{array}\right)$$

Solución del ejemplo 8.9

Función de transición de estados con otro formato: {{A, a, {A, EA}}, {A, b, {B}}, {B, a, {S}}, {B, b, {A, EA}}, {EA, a, {}}, {EA, b, {}}, {S, a, {S}}, {S, b, {A, EA}}}



Solución del ejemplo 8.9

Si en `RegularToAutomata` se añade la opción `dfa -> True`, el estudiante comprobará, además, los pasos requeridos para la elaboración del autómata determinístico equivalente al *ADFN*.

Descargue un archivo

https://www.escinf.una.ac.cr/discretas/Archivos/Lenguajes/File-192.zip

Teorema 8.2. Gramática regular a partir de un ADF

Otro teorema interesante reside en encontrar una gramática regular tomando como base un autómata de estado finito determinístico, donde de igual manera, $L(G) = A^o$ (ambos tienen el mismo lenguaje).

Theorem (8.2)

Sea $A = \left(\sigma, \tau, \sigma^*, \Delta, \widehat{A}\right)$ un autómata de estado finito determinístico. La gramática $G = (\sigma, \tau, P, \sigma^*)$ es regular y tal que: $L(G) = A^o$, con:

$$P = \{B \to xD \mid \Delta(B, x) = D\} \cup C \text{ y } C = \begin{cases} \phi \text{ si } D \notin \widehat{A} \\ \{B \to x\} \text{ si } D \in \widehat{A} \end{cases}$$

Comentario sobre el teorema 16

El teorema 16 es importante cuando se desea diseñar una gramática G con ciertas características, es decir, cuando G tenga como elementos de su lenguaje, hileras con una forma específica. De esta manera, conociendo anticipadamente cuál es el conjunto $L(G)$, se procedería a desarrollar el diagrama de transición de un autómata de estado finito determinístico donde su lenguaje corresponda a $L(G)$ (tal y como se explicó en los ejemplos: 7.15, 7.16, 7.17, 7.18, 7.19 y 7.20) y posteriormente, usando ese autómata y el algoritmo descrito en el teorema 16, se obtendría la gramática regular de interés.

El ejemplo siguiente muestra varios casos de diseño de una gramática regular donde se sabe de previo cuál es su lenguaje.

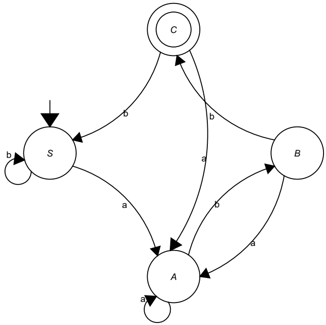
Example (8.10)

Diseñe una gramática regular G que posea el lenguaje indicado.

- $L(G) = \{$hileras que finalicen en "*abb*"$\}$.
- $L(G) = \{$hileras con al menos dos letras "*a*"$\}$.
- $L(G) = \{$hileras que inicien con "*ab*" y terminen con "*ba*"$\}$.

Solución del ejemplo 8.10

Comenzando con $L(G)$ formado por hileras que finalicen en "*abb*". Por la notación utilizada en este capítulo, los estados de este autómata se renombrarán así: $\sigma_0 = S$, $\sigma_1 = A$, $\sigma_2 = B$ y $\sigma_3 = C$, quedando su diagrama de transición como sigue:



(5)

Solución del ejemplo 8.10

Ahora, lo que resta es encontrar la gramática G equivalente a este ADF. De acuerdo con lo enunciado por el teorema 16, se tiene para G las componentes:

- Símbolos no terminales: $\{S, A, B, C\}$.
- Símbolos terminales: $\{a, b\}$.

Solución del ejemplo 8.10

- El conjunto P se determina observando las aristas salientes de cada estado en el digrafo 5. En el estado S hay un lado saliente con a que llega a A, es decir, se agrega al conjunto P, la regla gramatical $S \to aA$. En S hay otra arista saliente con b hacia el mismo estado S, lo que conduce a agregar al conjunto P, la composición $S \to bS$. En el estado A, el lazo con símbolo a permite incluir en P, la producción $A \to aA$ y la arista que va de A a B con el símbolo de entrada b, indica la existencia en P, de la regla gramatical $A \to bB$. En el estado B, el lado con símbolo a que llega a A, permite incluir en P, la composición $B \to aA$ y la arista que va de B al estado C con símbolo de entrada b, señala dos nuevas producciones en P, $B \to bC$ y $B \to b$, la regla $B \to b$ se incorpora en P, a razón de ser el estado de llegada C un estado aceptado en el autómata.

Solución del ejemplo 8.10

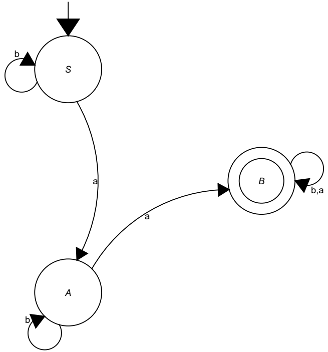
- Finalmente, los lados salientes de C con símbolos de entrada a y b que llegan a los estados A y S, respectivamente, asienten agregar en el conjunto de composiciones P, las reglas gramaticales $C \to aA$ y $C \to bS$. En resumen:

$$P = \{A \to aA, A \to bB, B \to aA, B \to b, B \to bC, C \to aA,$$
$$C \to bS, S \to aA, S \to bS\}$$

- $\sigma^* = S$.

Solución del ejemplo 8.10

Nuevamente, los estados se renombrarán para establecer una correspondencia con la simbología del presente capítulo, tal y como se detalla a continuación: $\sigma_0 = S$, $\sigma_1 = A$ y $\sigma_2 = B$. Luego, el diagrama de transición de este ADF es:



(6)

Solución del ejemplo 8.10

Recurriendo al teorema 16, una gramática regular G con el mismo lenguaje del autómata anterior, se define así:

- Símbolos no terminales: $\{S, A, B\}$.
- Símbolos terminales: $\{a, b\}$.
- Observando el diagrama de transición 6 se construye el conjunto de producciones P. En S hay una arista saliente al estado A y otra al mismo estado S con símbolos de entrada a y b, respectivamente, esto indica que las reglas $S \to aA$ y $S \to bS$ pertenecen a P. En el estado A el lazo con b, agrega la composición $A \to bA$ y el lado saliente al estado B con a, permite incluir las producciones $A \to aB$ y $A \to a$ al conjunto P. Se aclara que la regla $A \to a$ es considerada un elemento de P, pues el estado de llegada B es aceptado en el ADF.

Solución del ejemplo 8.10

- Finalmente, los dos lazos en B, implican añadir las reglas gramaticales $B \rightarrow aB$, $B \rightarrow a$, $B \rightarrow bB$ y $B \rightarrow b$, donde $B \rightarrow a$ y $B \rightarrow b$ se incorporan a P, dado que el estado de llegada B es aceptado. En resumen:

$$P = \{A \rightarrow a, A \rightarrow aB, A \rightarrow bA, B \rightarrow a, B \rightarrow aB, B \rightarrow b, B \rightarrow bB,$$

$$S \rightarrow aA, S \rightarrow bS\}$$

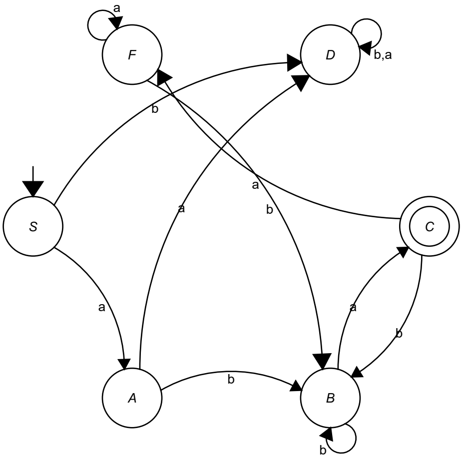
- $\sigma^* = S$.

Solución del ejemplo 8.10

Para terminar los estados, en este caso, se renombrarán así: $\sigma_0 = S$, $\sigma_1 = A$, $\sigma_2 = B$, $\sigma_3 = C$, $\sigma_4 = D$ y $\sigma_5 = F$. No se ha utililizado la letra E para reetiquetar el estado σ_5 puesto que en *Wolfram Mathematica* ese caracter corresponde al número neperiano $e \approx 2{,}71$ y si se usa en el software, éste automáticamente sustituirá la E por ese valor. De momento, no se ha hecho nada en *Wolfram*, sin embargo, al final de esta solución se mencionará una sentencia que facilita la aplicación automática del teorema 16.

Solución del ejemplo 8.10

El digrafo de representación del autómata renombrado en sus estados es el siguiente:



(7)

Solución del ejemplo 8.10

Luego, por el teorema 16, una gramática regular G equivalente al autómata 7, tiene como sus cuatro componentes:

- Símbolos no terminales: $\{S, A, B, C, D, F\}$.
- Símbolos terminales: $\{a, b\}$.
- El diagrama de transición 7 permite inferir el conjunto P. Las aristas salientes en el estado S agregan a P, las reglas gramaticales $S \rightarrow aA$ y $S \rightarrow bD$. En el estado A sus dos lados salientes incluyen en P, las producciones $A \rightarrow aD$ y $A \rightarrow bB$. En B, el lazo añade al conjunto P, la composición $B \rightarrow bB$ y la arista saliente con a incorpora en P, las reglas $B \rightarrow aC$ y $B \rightarrow a$, esta última al ser C un estado aceptado en el autómata. En el estado C sus dos lados salientes permiten incluir en P, las producciones $C \rightarrow aF$ y $C \rightarrow bB$. Ahora, en el estado D los dos lazos señalan que $D \rightarrow aD$ y $D \rightarrow bD$ pertenecen a P. En F, $F \rightarrow aF$ y $F \rightarrow bB$ se vinculan con sus dos lados salientes y por lo tanto, serían las dos últimas reglas gramaticales de P.

Solución del ejemplo 8.10

- En resumen:

$$P = \{A \to aD, A \to bB, B \to a, B \to aC, B \to bB, C \to aF,$$
$$C \to bB,\ D \to aD, D \to bD, F \to aF, F \to bB, S \to aA, S \to bD\}$$

- $\sigma^* = S$.

El algoritmo propuesto en el teorema 16 para encontrar una gramática regular equivalente a un autómata de estado finito determinístico, se ha programado en la instrucción `AutomataToRegular` del paquete **VilCretas**. `AutomataToRegular` recibe un autómata determinístico creado con la sentencia `Automata` y devuelve como salida una gramática regular en notación forma normal (mediante una 4−tupla) que posee su mismo lenguaje.

Solución del ejemplo 8.10

El archivo de descarga compartido a continuación, muestra el empleo de `AutomataToRegular` para resolver los tres ejercicios planteados en este ejemplo.

Descargue un archivo

https://www.escinf.una.ac.cr/discretas/Archivos/Lenguajes/File-193.zip

Uso del comando AutomataToRegular.

Explicación en video

https://youtu.be/0BDpA4c2ljo

Cuaderno interactivo sobre los contenidos del capítulo.

Descargue un archivo

https://www.escinf.una.ac.cr/discretas/Archivos/Cuadernos/Lenguajes.pdf.rar

Quiz interactivo de repaso sobre los contenidos del capítulo.

Descargue un archivo

https://www.escinf.una.ac.cr/discretas/Archivos/Lenguajes/Quiz_lenguajes.rar

Webmix sobre el empleo de los comandos del paquete **VilCretas** estudiados en el capítulo.

Abra un sitio web

https://www.symbaloo.com/mix/vilcretaslenguajes

¡Recuerde resolver los ejercicios asignados!

Descargue un archivo

https://www.escinf.una.ac.cr/discretas/Archivos/Lenguajes/Exercises.zip

enrique.vilchez.quesada@una.cr
http://www.escinf.una.ac.cr/discretas