

2020-PL-1.Introducción.pdf ???

-Un intérprete es un compilador que se escribe en el mismo lenguaje que interpreta.

Falso

-Dentro del Compilador, el Procesador de Lenguaje se corresponde con la fase de:

Preparación

Preprocesamiento

Análisis - ?

Síntesis

-Los módulos principales del Procesador de Lenguajes son:

Analizador Léxico, Sintáctico y Semántico

-La cadena "HOLA2021" es una cadena válida para el alfabeto formado por las letras mayúsculas del abecedario *

Falso

-El conjunto de números hexadecimales de 3 dígitos es un lenguaje válido para el alfabeto: *

{0,1,2,3,4,5,6,7,8,9}

{0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F} -?

{0,1,2,3,4,5,6}

{1,2,3,4,5,6,7,8,9,A,B,C,D,E}

-Cual de los siguientes conjuntos de reglas puede ser parte de una gramática regular por la derecha (tipo 3): * ??????????????????????

B --> 1C - C-->2 3

B --> A C - A--> 3 - C-->2 3

B --> 2 C - A--> 3 - C-->1

B --> A - A--> 3

-Esta regla no puede ser parte de una G.Regular: A-->1 A 2

Verdadero ??????????????????????

Falso

-Toda Gramática Regular es también una gramática Independiente del Contexto *

Verdadero

-Usaremos Gramáticas de Tipo3 para An. Léxico y de Tipo2 para An. Sintáctico *

Verdadero

Analizador Léxico 2

Indica cuáles de los siguientes pares Tipo de Token-Atributo son correctos:

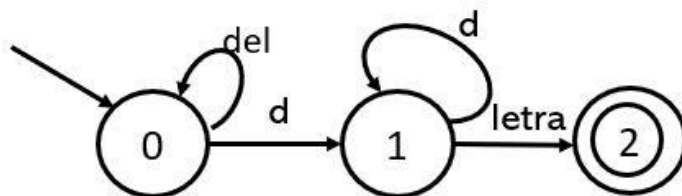
- <lexema, punteroTS> Falso
- <operador-lógico, 3> Verdadero
- <operador-relacional, lexema> Falso
- <operador-lógico, valor> Falso
- <punto-y-coma, "punto y coma"> Falso
- <identificador, punteroTS> Verdadero
- <constante-real, lexema> Falso
- <For, -> Verdadero
- <While, lexema> Falso
- <operador-relacional, 2> Verdadero

La gramática regular cuyas reglas son "S--> d A A-->d B B--> d B B-->lambda" *
Genera el conjunto de todos los números enteros que tienen al menos 2 dígitos

La gramática "S--> del S S--> letra A A--> B B--> letra"
No es una gramática válida para construir un Analizador Léxico

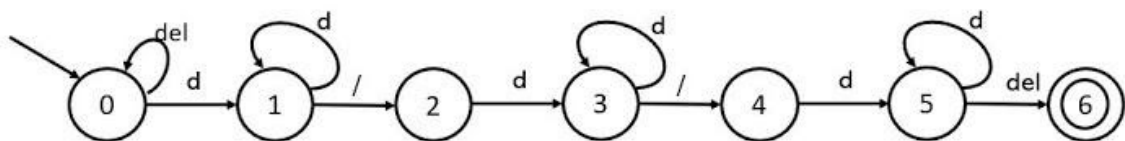
El autómata de la imagen permite reconocer los siguientes lexemas

10/10



Lexemas tipo DNI

El autómata de la imagen permite reconocer



Expresiones aritméticas con el operador división

Secuencias de un mínimo de 2 dígitos separados por "/"

Lexemas como por ejemplo del, d, d/d, d/d/d, delimitador

Cualquier secuencia de dígitos y "/"

En relación con las acciones semánticas del Analizador Léxico:

Añaden funcionalidad al reconocimiento de los lexemas de los tokens y pueden ayudar a la generación del token

Permiten ejecutar funciones en los estados del Autómata

Sirven exclusivamente para la detección de errores

Solo se pueden aplicar en las transiciones a un estado final

COMPLETAR (FALTAN 2 PREGUNTAS)

Analizador Léxico 4

-Si tenemos que procesar un fichero de texto con números de teléfono, cuál sería el formato del token correspondiente en el Analizador Léxico? *

<telefono, valor>

<telefono, lexema>

<telefono, ->

No sería un token

-Si en un lenguaje que debemos procesar nos encontramos número en formato octal (empezando por o, por ejemplo o76) y hexadecimal (p.e. ox12D4), los tokens que deberíamos definir son: *

<constante-octal, valor> y <constante-hexadecimal, valor>

<constante-octal, lexema> y <constante-hexadecimal, lexema>

<o,->, <constante-octal, valor>, <ox,-> y <constante-hexadecimal, valor>

<constante-entera, valor>

-Una Base de Datos guarda los "timestamps" como el nº de segundos transcurridos desde las 0:00:00horas. Para rellenar ese campo, voy a leer el valor en un fichero de texto donde los "timestamp"s aparecen en el formato 0:12:43. Como serían los token a definir? *

<hora, valor>, <minuto,valor>, <segundo,valor> por ejemplo, para 0:12:43, habría que enviar 3 tokens: <hora, 0>, <minuto,12>, <segundo,43>

<timestamp, puntero-a-reloj>

<timestamp, valor>, donde por ejemplo, para 0:12:43, valor seria $0*3600+12*60+43=763$

<timestamp, lexema>, por ejemplo <timestamp, 0:12:43>

Para generar los lexemas correspondientes a las cadenas de caracteres, que permiten el uso de comillas dentro de la cadena mediante el uso del carácter de escape # ("Hola comillas # " "), la gramática correcta sería (siendo c: todo carácter excepto "):

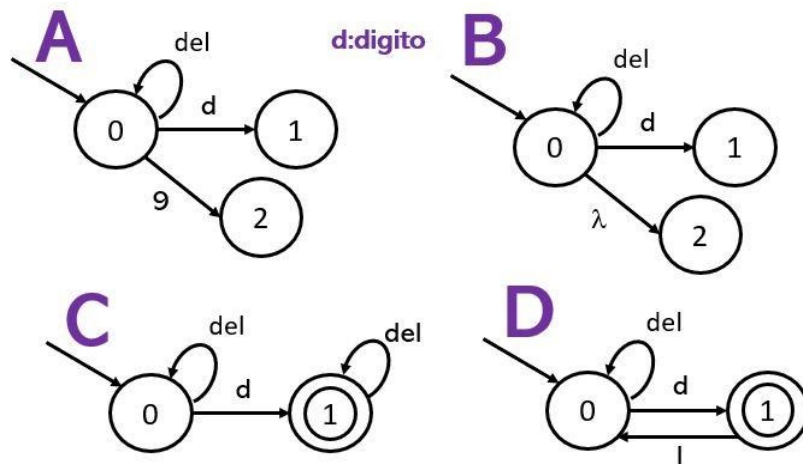
$S \rightarrow " A$ $A \rightarrow c A$ $A \rightarrow "$ $A \rightarrow \# B$ $B \rightarrow " A$

Para generar los lexemas correspondientes a las constantes de tipo hora en el formato

hh:mm:ss, cuál de las siguientes opciones es correcta?

$S \rightarrow d A$ $A \rightarrow d B$ $B \rightarrow : C$ $C \rightarrow d D$ $D \rightarrow d E$
 $E \rightarrow : F$ $F \rightarrow d G$ $G \rightarrow d H$ $H \rightarrow \lambda$

-Cuáles de los autómatas de la imagen son NO Deterministas?



Todos son no deterministas

-¿Cuáles de las siguientes afirmaciones son correctas?

Las acciones semánticas se ejecutan en las transiciones del autómata V

Las acciones semánticas pueden hacer que sea válido un lexema no reconocido por el autómata F

Las acciones semánticas que generan un token solo se pueden colocar en las transiciones a estado final V

La acción semántica Concatenar permite calcular el valor de un número F

-En relación con las acciones semánticas encargadas de detectar errores en un Analizador Léxico: *

Colocarlas en las transiciones a estado final asegura una mejor recuperación de los errores

Se pueden integrar con las demás acciones semánticas

Aseguran que no se van a reconocer lexemas inválidos cuando el formato reconocido por el autómata es demasiado flexible

Todas las opciones son correctas

-Cuales de los siguientes errores deben ser detectados por el Analizador Léxico:

Constante entera fuera de rango V

Abre-paréntesis no tiene el correspondiente Cierre-paréntesis F

Carácter inválido en el lenguaje V

Constante cadena sin comilla de cierre V

Return fuera del cuerpo de la función F

TEST SIMULACRO EXAMEN MOODLE:

-Considérese el lenguaje de programación en el que está escrito el siguiente programa:

Program ExOnlineJun20

Global integer a,b,n;

Procedure PAR (valor integer: x1_Par; referencia integer: y2_Par);
// x1_Par se pasa por valor, y2_Par por referencia

Begin

x1_Par := a * y2_Par; // ***

End

Begin

n := read ("Introduce tu número de matrícula (solo dígitos):");

a := módulo (n/10);

if módulo (n/2) = 0 then PAR(a,b) Else PAR(b,a);

End

Considerando que el lenguaje solo permite los elementos que se muestran en el programa, y que éste es correcto, ¿cuál de los siguientes tipos de token es incorrecto?

Seleccione una opción:

<punto y coma, ->

<constante, lexema>

<palabraclave, código>

Respuesta en blanco

<identificador, punteroTS>

-Queremos analizar un fichero de texto donde solo aparecen números reales, números enteros, cadenas y comentarios. Si queremos diseñar un Analizador Léxico como parte de una aplicación para trasvasar la información del fichero a una Base de Datos y teniendo en cuenta que los comentarios son anotaciones del responsable del fichero que han perdido completamente su vigencia, ¿Cuáles de los siguientes serían *tokens* de este lenguaje?

Selecciona una o más opciones:

<NÚMERO, valor>

Respuesta en blanco

<CADENA, lexema>

<ENTERO, valor>

<COMENTARIO, ->

<ID, posTS>

<COMENTARIO, lexema>

<REAL, valor>

<CADENA, posTS>

-La gramática de un analizador léxico diseñado para reconocer constantes hexadecimales (que empiezan por 0x) sería (siendo d: digito hexadecimal 0..F) (señálese la correcta):

Seleccione una opción:

a. Respuesta en blanco

b. $S \rightarrow 0 x A$ $A \rightarrow d A \mid \lambda$

c. $S \rightarrow d x A$ $A \rightarrow d B \mid \lambda$ $B \rightarrow d A \mid \lambda$

d. $S \rightarrow 0 A$ $A \rightarrow x B$ $B \rightarrow d C$ $C \rightarrow d C \mid \lambda$

e. $S \rightarrow 0 A$ $A \rightarrow x B$ $B \rightarrow d S \mid \lambda$

-Considérese el lenguaje de programación en el que está escrito el siguiente programa:

Program ExOnlineJun20

Global integer a,b,n;

Procedure PAR (valor integer: x1_Par; referencia integer: y2_Par);
// x1_Par se pasa por valor, y2_Par por referencia

Begin

x1_Par := a * y2_Par; // ***

End

Begin

n := read ("Introduce tu número de matrícula (solo dígitos):");

a := módulo (n/10);

if módulo (n/2) = 0 then PAR(a,b) Else PAR(b,a);

End

¿Cuál de los siguientes conjuntos de reglas genera correctamente los lexemas válidos para el token identificador?

Seleccione una opción:

S --> letra D D --> guión_bajo D --> lambda

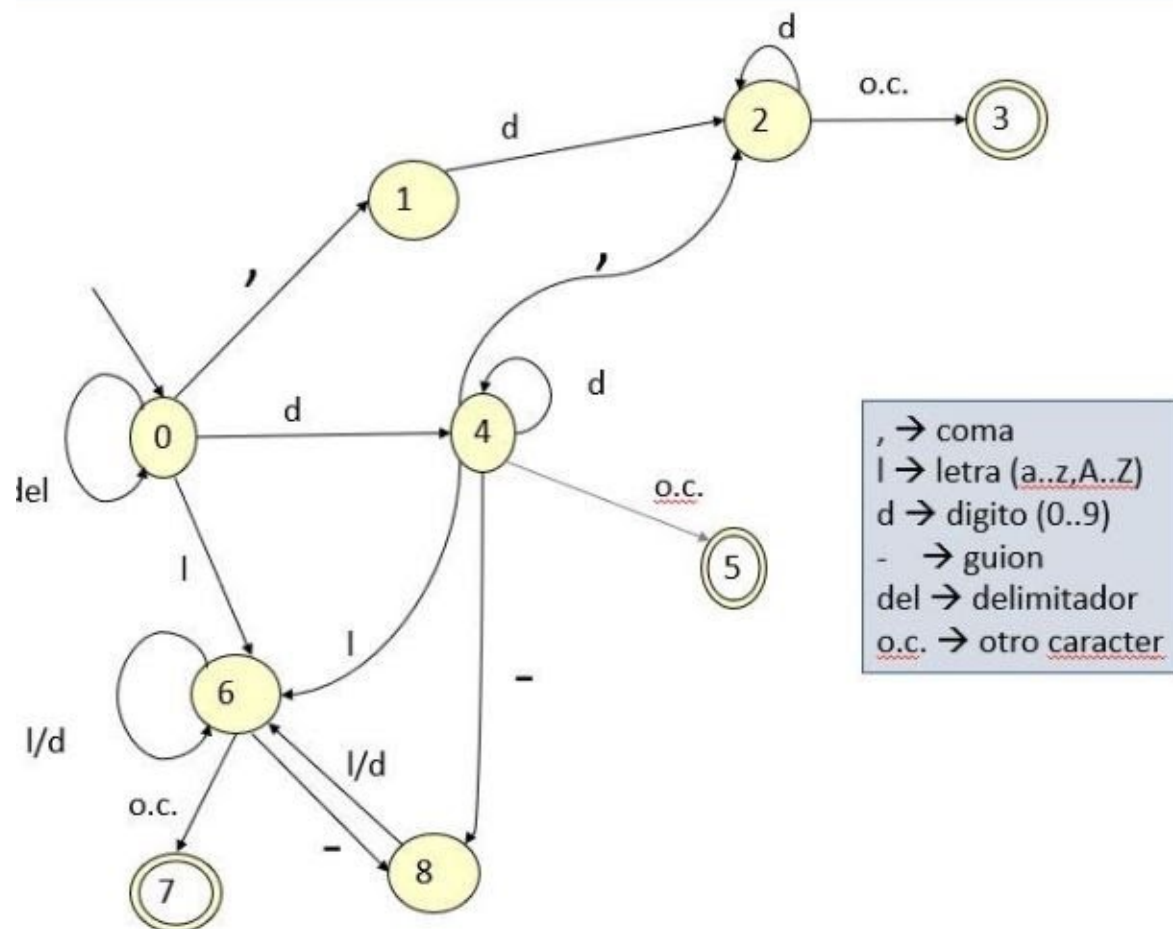
D --> letra D D --> dígito D

S --> letra D D --> letra D D --> dígito D D --> lambda

S --> letra D D --> letra D D --> dígito D D --> guión_bajo D --> lambda

S --> letra D D --> letra D D --> dígito D D --> guión_bajo E D --> lambda
E --> letra D E --> dígito D

-Considérese el autómata finito determinista mostrado en la imagen (subconjunto del autómata del Analizador Léxico de un lenguaje de programación).



¿Cuál de los siguientes tokens es posible reconocer con este autómata?

Seleccione una opción:

Un identificador con lexema "9-hola-"

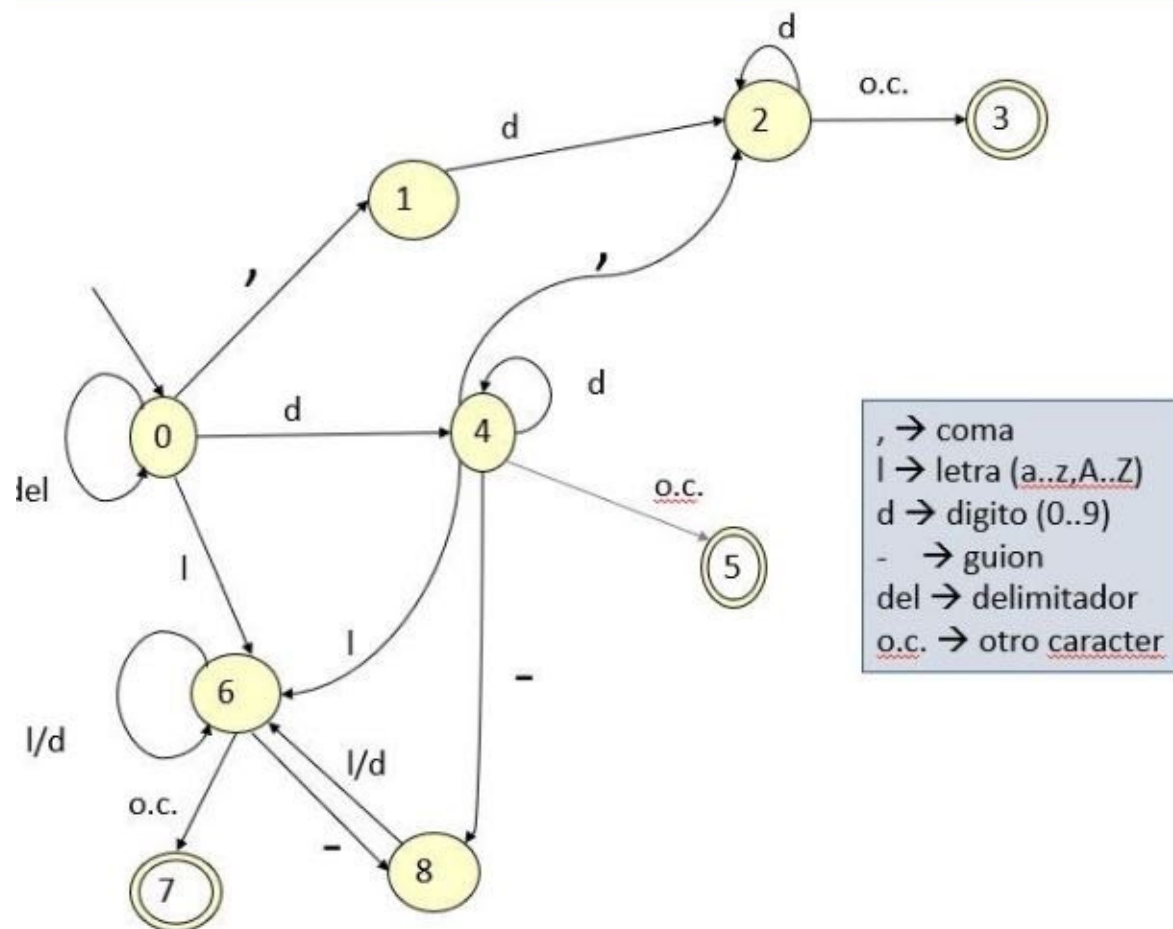
Cualquier carácter como "("

Respuesta en blanco

Una constante ",67"

Un guion "-"

-Considérese el autómata finito determinista mostrado en la imagen (subconjunto del autómata del Analizador Léxico de un lenguaje de programación).



En la transición del estado 4 al estado 5 se deben ejecutar, entre otras, las siguientes acciones semánticas (señálese la correcta):

Leer el siguiente carácter. Actualizar el valor según el último dígito leído. Emitir el token <constante-entera, valor>

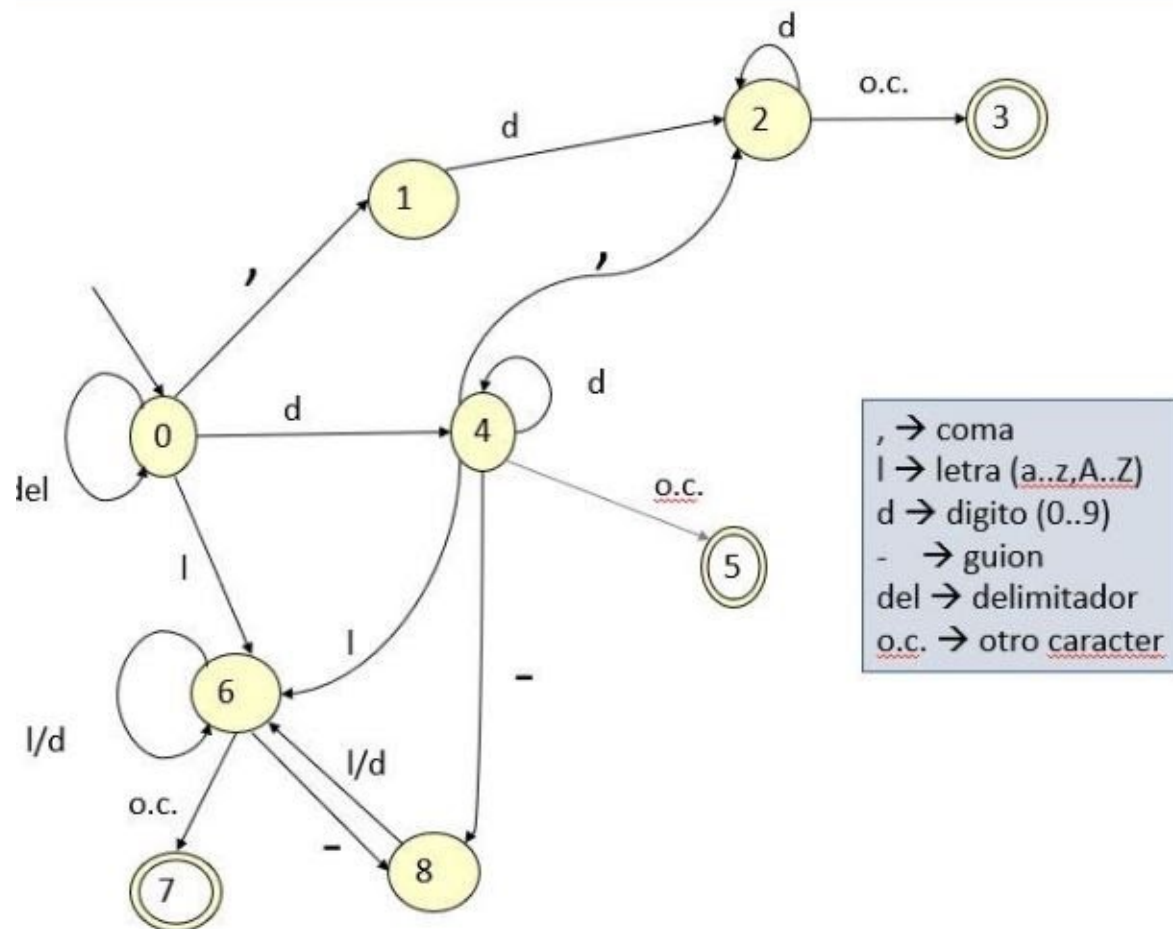
Emitir el token <constante-real, valor>

Comprobar el rango máximo de enteros. Emitir el token <constante-entera, valor>

Respuesta en blanco

Emitir el token <constante, lexema>

-Considérese el autómata finito determinista mostrado en la imagen (subconjunto del autómata del Analizador Léxico de un lenguaje de programación).



En relación con la detección de errores por las Acciones semánticas, señálese la verdadera:

Seleccione una opción:

Respuesta en blanco

En la transición 6-8 se debe comprobar que no se leen dos guiones seguidos

En la transición 6-7 se debe comprobar que el lexema tiene al menos una letra

En la transición 6-7, si se está leyendo un identificador en una zona de uso de variables, se debe comprobar que el lexema está en la Tabla de Símbolos

En la transición 2-3 se debe comprobar que la constante tiene al menos un dígito

-¿De qué tipo es la siguiente Gramática?

$S \rightarrow 1 X \mid 1$

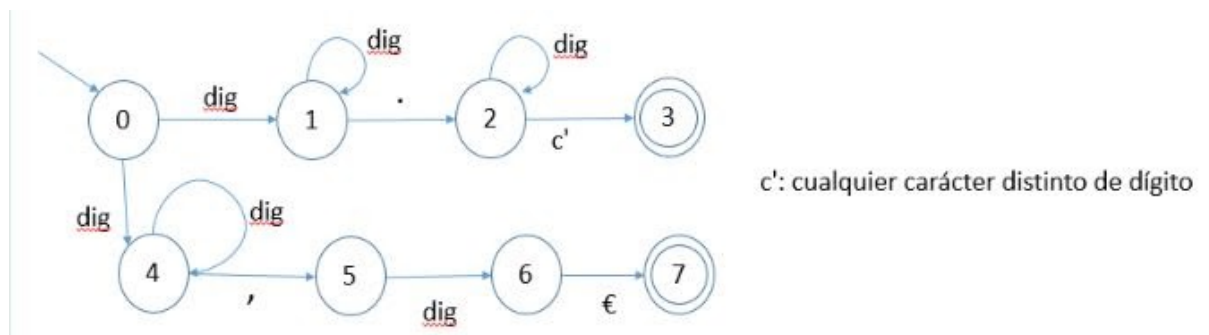
$X \rightarrow 2 B \mid B$

$B \rightarrow 2$

Seleccione una opción:

- a. Es una gramática de atributo
- b. Es una gramática regular
- c. Es válida para un sintáctico descendente
- d. Es una gramática de contexto libre**
- e. Respuesta en blanco

-Dos de los tokens de un lenguaje son número real y precio. Los números reales tienen al menos un dígito en la parte entera, un punto, y pueden o no tener parte decimal. Los precios vienen dados siempre con un dígito decimal, que viene separado por una coma, y terminan con el símbolo de euro. Si se diseña el Analizador Léxico mediante el siguiente autómata, ¿cuál de las siguientes afirmaciones es cierta?

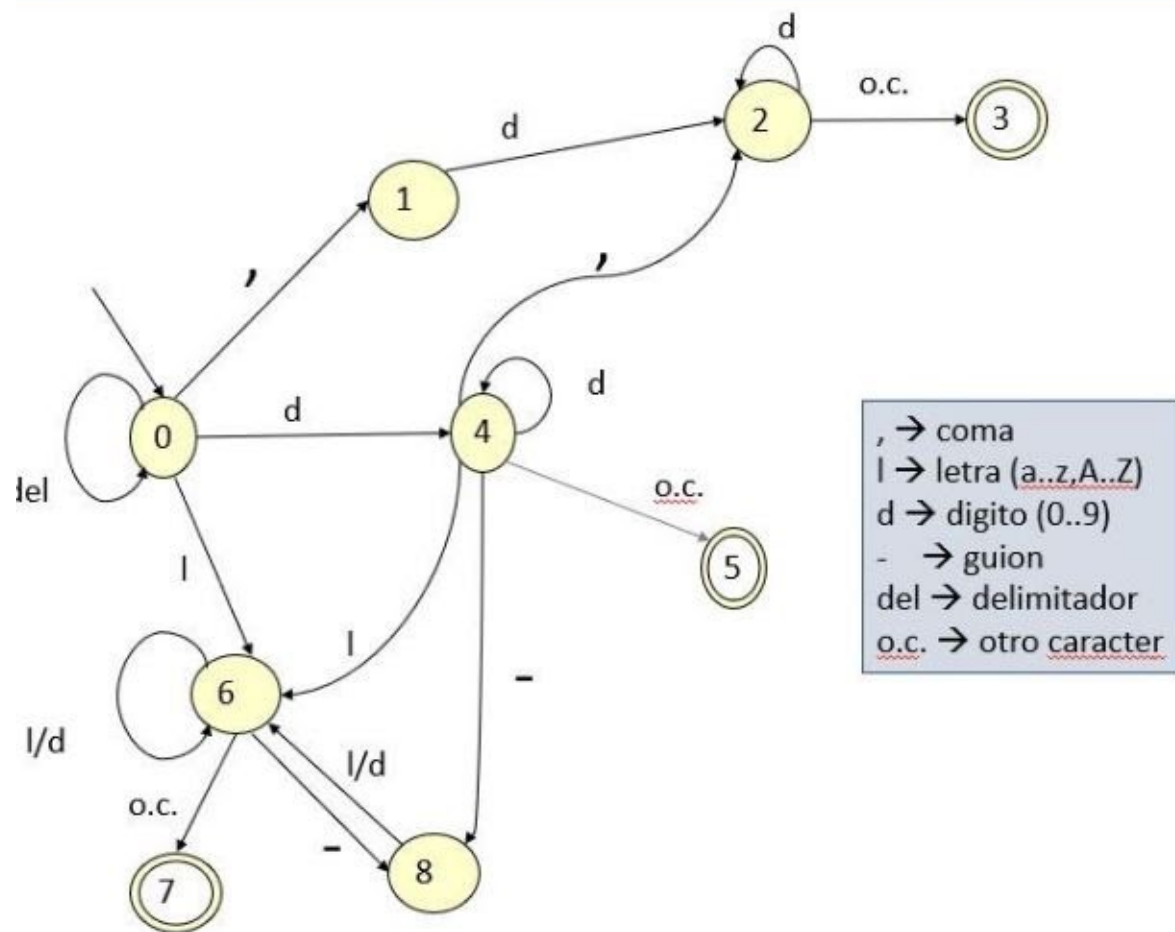


Seleccione una opción:

- a. El autómata construido es no determinista y válido para reconocer le lenguaje
- b. El lexema € se tiene que reconocer de manera independiente para que el autómata sea válido para reconocer el lenguaje
- c. El autómata construido es no determinista y no es válido para reconocer el lenguaje**
- d. El autómata construido es determinista y válido para reconocer el lenguaje
- e. Respuesta en blanco

-Considérese el autómata finito determinista mostrado en la imagen (subconjunto del autómata del Analizador Léxico de un lenguaje de programación).

¿Cuál de los siguientes lexemas sería detectado como un error por este autómata?



Seleccione una opción:

Respuesta en blanco

hola-Todos9

hola-Todos

holaTodos-

9hola9