

hoja1.pdf



Anónimo



Autómatas y Lenguajes



3º Grado en Ingeniería Informática



**Escuela Politécnica Superior
Universidad Autónoma de Madrid**

Ayudas hasta el 40%

MÁSTER EN

**Inteligencia Artificial
y Ciencia de Datos**

ONLINE

Estudia el máster líder en inteligencia
artificial y ciencia de datos

**¡ÚLTIMAS
PLAZAS!**

EOI Escuela de
organización
industrial

Info y descuentos



Universidad Autónoma de Madrid
Departamento de Ingeniería Informática
3^{er}. Curso 2º Cuatrimestre
Autómatas y Lenguajes

Enunciados sobre la unidad 1 modelos cómputo, familias de lenguajes Independencia del contexto Hoja 1

GRAMÁTICAS INDEPENDIENTES DEL CONTEXTO

1. Considere la gramática independiente que puede deducirse de las siguientes reglas de producción:

$$\begin{aligned} S &\rightarrow aSb \\ S &\rightarrow ab \end{aligned}$$

que genera el siguiente lenguaje:

$$\{a^n b^n \mid 0 < n\}$$

A partir de esta gramática se quiere diseñar otra que genere el lenguaje siguiente

$$a^i b^j a^j b^i \dots a^p b^p$$

Donde los superíndices pueden tomar cualquier valor. Observe que las cadenas del lenguaje realmente consisten en una secuencia de palabras del primer lenguaje explicado. Por ejemplo

aabbabaaabbb
ab
ababaabb

Soluciones:

$$\begin{aligned} X &\rightarrow S \\ X &\rightarrow SX \\ S &\rightarrow aSb \\ S &\rightarrow ab \end{aligned}$$

2. Considere el siguiente lenguaje:

$$\{0^n 1^{2n} \mid 0 \leq n\}$$

Diseñe una gramática independiente del contexto que lo pueda generar. Justifíquelo breve e informalmente.

Soluciones:

$$\begin{aligned} S &\rightarrow 0S11 \\ S &\rightarrow \lambda \end{aligned}$$



3. Considere el siguiente lenguaje:

$$\{a^{n+m}b^nc^m \mid 0 \leq n, m\}$$

Diseñe una gramática independiente del contexto que lo pueda generar. Justifíquelo breve e informalmente.

Soluciones:

$$S \rightarrow aSc$$

$$S \rightarrow X$$

$$X \rightarrow aXb$$

$$X \rightarrow \lambda$$

4. Considere el siguiente lenguaje:

$$\{a^{m-n}b^nc^m \mid 0 \leq n \leq m\}$$

Se le pide

Diseñe una gramática independiente del contexto que lo pueda generar. Justifíquelo breve e informalmente.

Soluciones:

$$S \rightarrow aSc$$

$$S \rightarrow X$$

$$X \rightarrow bXc$$

$$X \rightarrow \lambda$$

5.- Diseñar una gramática independiente del contexto que (informalmente) reconozca el siguiente lenguaje

$$L = \{w@w^{-1} \mid w \in \{a,b\}^*\}$$

Soluciones:

$$G = (S_T = \{a, b\}, S_N = \{S\}, S, P)$$

$S ::= @$
 $S ::= aSa$
 $S ::= bSb$

6.- Diseñar una gramática independiente del contexto que (informalmente) reconozca el siguiente lenguaje

$$L = \{a^n b^m \mid 0 \leq n \leq m \leq 3n\}$$

Soluciones:

$$G = (S_T = \{a, b\}, S_N = \{S\}, S, P)$$

Donde P contiene las siguientes reglas de producción

$S ::= aSb$
 $S ::= aSbb$
 $S ::= aSbbb$
 $S ::= \lambda$

7.- Diseñar una gramática independiente del contexto que (informalmente) reconozca el siguiente lenguaje. Donde $\#_x(\gamma)$ representa el número de veces que el símbolo x aparece en la cadena γ .

$$L = \{w \in \{a,b,c\}^* \mid \#_a(w) + \#_b(w) = \#_c(w)\}$$

Soluciones:

$S ::= SS$
 $S ::= OS$
 $S ::= cSO$
 $S ::= \lambda$
 $O ::= a$
 $O ::= b$

En la tienda privada de Samsung Estudiantes, te esperan **nuestras ofertas más exclusivas**

Listos
para
estudiar

Financiación
en 36 meses



Envío
gratis



Ahorra
entregando
tu antiguo
dispositivo

Escanea el
código QR,
regístrate y
consigue un
5% de dto.



8.- Diseñar una gramática a pila que (informalmente) reconozca el siguiente lenguaje

$$L = \{ab(ab)^n b(ba)^n \mid n \geq 0\}$$

Soluciones:

$$G = (S_T = \{a, b\}, S_N = \{S, X\}, S, P)$$

Donde P contiene las siguientes reglas de producción

$S ::= abX$
 $X ::= abXba$
 $X ::= b$

9.- Los lenguajes de programación de alto nivel a menudo incluyen diferentes delimitadores (paréntesis, llaves, etc...) que indican el comienzo y final de 'algo' (bloque de sentencias, subexpresiones, lista de parámetros en la definición o llamada a una función, etc...)

En ese contexto, se desea diseñar un lenguaje de programación para el 'lenguaje de los paréntesis'. Este lenguaje está formado por cadenas compuestas exclusivamente por paréntesis balanceados y todos incluidos en un paréntesis exterior.

Las siguientes cadenas pertenecen a este lenguaje:

$(\quad) \quad (\quad)$
 $(\quad (\quad) (\quad) (\quad) \quad) \quad) \quad)$

Las siguientes cadenas no pertenecen a este lenguaje:

(\quad)
 $(\quad) \quad)$

Diseña una gramática independiente del contexto que genere el lenguaje de los paréntesis.

Soluciones:

$$G = (S_T = \{ (,) \}, S_N = \{ Lista, ListaInterna \}, Lista, P)$$

Donde P contiene las siguientes reglas de producción

$Lista \rightarrow (\quad ListaInterna \quad)$
 $ListaInterna \rightarrow ListaInterna \quad (\quad ListaInterna \quad)$
 $ListaInterna \rightarrow \lambda$

10.- Diseñar una gramática independiente del contexto que genere la estructura de los bloques de sentencias de un lenguaje de programación de alto nivel. Para ello tenga en cuenta las siguientes cuestiones:

- Se van a representar las sentencias elementales (asignación, lectura, escritura, etc...) mediante el símbolo s sentencias
- Para simplificar no se necesitarán separadores entre las sentencias
- Se supondrá que las sentencias podrán aparecer una tras otra formando una secuencia
- Así mismo, se considerarán como posibles sentencias las estructuras de control de flujo de programa siguientes:
 - Condicional
 - La palabra reservada si se utilizará para indicar su comienzo.

- Tendrá una única condición que podrá representarse mediante el símbolo *condicion*. Para simplificar no se detallará más su estructura.
- La rama 'then' comenzará con la palabra reservada *entonces*
- La rama 'else' comenzará con la palabra reservada *otro_caso*
- Cualquiera de las dos ramas contendrá cualquier secuencia de sentencias delimitadas por los símbolos { y }
- Bucle o estructura repetitiva
 - La palabra reservada *mientras* se utilizará para indicar su comienzo
 - Tendrá una única condición que podrá representarse mediante el símbolo *condicion*. Para simplificar no se detallará más su estructura.
 - Su cuerpo será cualquier secuencia de sentencias delimitado por los símbolos { y }
- Cualquier secuencia de sentencias podría aparecer vacía
- No tiene que preocuparse en este momento de que la gramática que genere sea ambigua ni ninguna otra consideración adicional

Un posible 'programa' ejemplo sería el siguiente

```

s
s
si condicion
entonces
{
    s
    mientras condicion
    {
        si condicion
        entonces
        { s }
        otro_caso {}
    }
    s
}
otro_caso
{ s }
s

```

Soluciones:

$G=(S_T=\{s, si, condicion, entonces, otro_caso, mientras, condicion, \{, \}, S_N=\{Sentencias, Sentencia, Estructura, Condicional, Repetitiva\}, Sentencias, P)$

Donde *P* contiene las siguientes reglas de producción

<i>Sentencias</i>	→	<i>Sentencia Sentencias</i>
<i>Sentencias</i>	→	λ
<i>Sentencia</i>	→	<i>s</i>
<i>Sentencia</i>	→	<i>Estructura</i>
<i>Estructura</i>	→	<i>Condicional</i>
<i>Estructura</i>	→	<i>Repetitiva</i>
<i>Condicional</i>	→	<i>si condicion entonces {Sentencias } otro_caso {Sentencias }</i>
<i>Repetitiva</i>	→	<i>mientras condicion {Sentencias }</i>

SAMSUNG

Llena tu mochila de descuentos

En la tienda privada de Samsung Estudiantes,
te esperan **nuestras ofertas más exclusivas**



Listos s
para 
estudiar



AUTÓMATAS A PILA

1. Considere el siguiente lenguaje:

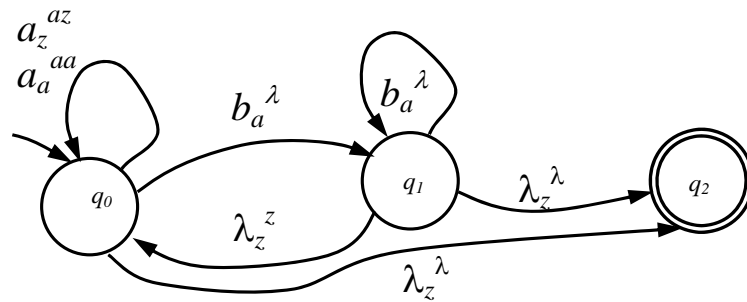
$$\{a^n b^n \mid 0 \leq n\}$$

A partir de él se quiere diseñar un autómata a pila que “informalmente” reconozca el lenguaje siguiente

$$a^i b^i a^j b^j \dots a^p b^p$$

Solución:

Las transiciones lambda hacen que sea no determinista.



En la tienda privada de Samsung Estudiantes, te esperan **nuestras ofertas más exclusivas**

Listos
para
estudiar

Financiación
en 36 meses

Envío
gratis

Ahorra
entregando
tu antiguo
dispositivo

Escanea el
código QR,
regístrate y
consigue un
5% de dto.

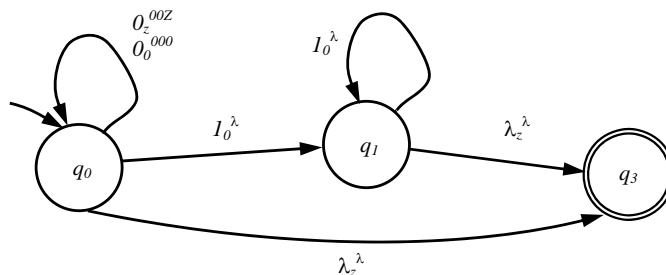


2. Diseñar un autómata a pila que (informalmente) reconozca el siguiente lenguaje:
 $\{0^n 1^{2n} \mid 0 \leq n\}$

Soluciones:

La idea de este autómata es conservar en la pila tantos símbolos 0 como el doble de los que aparecen en la entrada, no es imprescindible añadir símbolos de ningún otro tipo a la pila para completar el autómata:

- Cada 0 es guardado en la pila como dos símbolos 0
- Una vez que se llega al bloque de los símbolos 1, tiene que haber tantos como símbolos 0 hay en la pila.
- Cuando la cadena se termina, en la pila sólo debe estar el símbolo de inicio de pila.
- Hay que añadir las transiciones que aseguran que el autómata funciona de manera coherente para el valor mínimo de n (0)

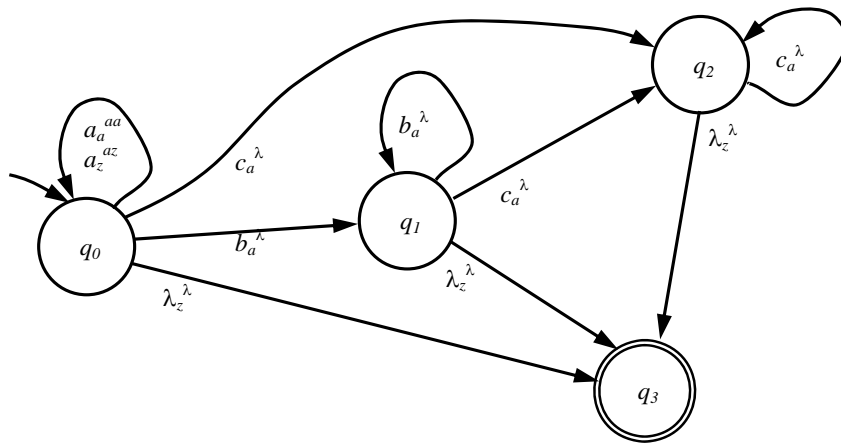


3. Diseñar un autómata a pila que (informalmente) reconozca el siguiente lenguaje:
 $\{a^{n+m}b^nc^m \mid 0 \leq n, m\}$

Soluciones:

La idea de este autómata es conservar en la pila tantos símbolos a como se necesite, no se guarda ningún símbolo adicional. Así que el funcionamiento del autómata puede esquematizarse como sigue:

- Los símbolos a del principio de la cadena se guardan en la pila tal y como aparecen en la entrada
- Para que la cadena sea correcta, debe haber después dos bloques, uno de símbolos b y otro de símbolos c (y en ese orden) con el mismo número de símbolos que en el bloque de símbolos a. Por eso, cada aparición de ese tipo de símbolos debe eliminar una a de la pila. Al final de la cadena debe quedar la pila vacía (a excepción del símbolo de inicio de pila)
- Hay que tener en cuenta añadir las transiciones coherentes para el buen funcionamiento debido a la posibilidad de que no haya algún tipo de símbolo por el número mínimo (0) de símbolos permitido.

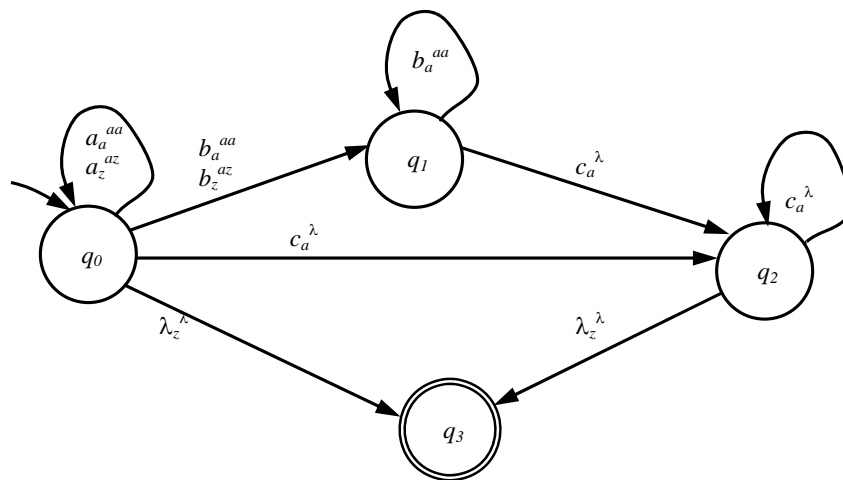


4. Diseñar un autómata a pila que (informalmente) reconozca el siguiente lenguaje:
 $\{a^m b^n c^m \mid 0 \leq n \leq m\}$

Soluciones:

La idea de este autómata es conservar en la pila tantos símbolos a como se necesite, no se guarda ningún símbolo adicional. Así que el funcionamiento del autómata puede esquematizarse como sigue:

- Los símbolos a del principio de la cadena se almacenan en la pila directamente
- Los símbolos b que siguen a los a se almacenan en la pila directamente
- Para que la cadena sea correcta, debe haber tantos símbolos c como a y b . Así pues, cuando comienza el bloque de símbolos c tiene que haber exactamente el mismo número de símbolos c que de símbolos a hay en la pila. Se eliminará de la pila un símbolo cada vez. Cuando la cadena haya terminado la pila tiene que estar vacía (a excepción del símbolo de inicio de pila)
- Hay que tener cuidado con el número mínimo, ya que puede ser 0 hay que añadir transiciones coherentes para el caso en el que no haya algún tipo de símbolo

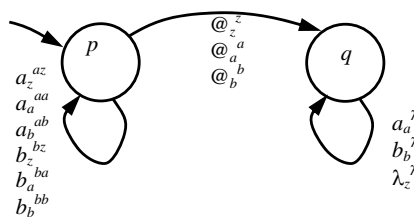


5.- Diseñar un autómata a pila que (informalmente) reconozca el siguiente lenguaje

$$L = \{w@w^{-1} \mid w \in \{a,b\}^*\}$$

Soluciones:

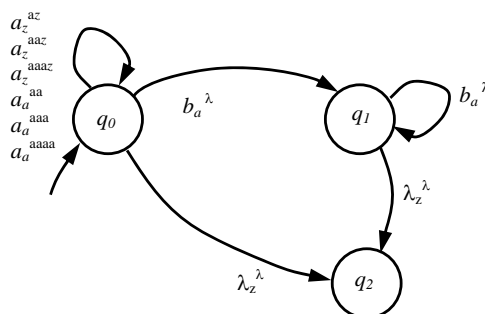
$$(\Sigma = \{a, b, @ \}, \Gamma = \{z, a, b\}, Q = \{p, q\}, z, q_0 = p, \delta, F = \emptyset)$$



6.- Diseñar un autómata a pila que (informalmente) reconozca el siguiente lenguaje

$$L = \{a^n b^m \mid 0 \leq n \leq m \leq 3n\}$$

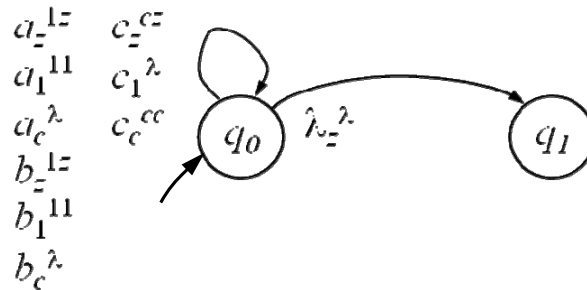
Soluciones:



7.- Diseñar un autómata a pila que (informalmente) reconozca el siguiente lenguaje. Donde $\#_x(\gamma)$ representa el número de veces que el símbolo x aparece en la cadena γ .

$$L = \{w \in \{a, b, c\}^* \mid \#_a(w) + \#_b(w) = \#_c(w)\}$$

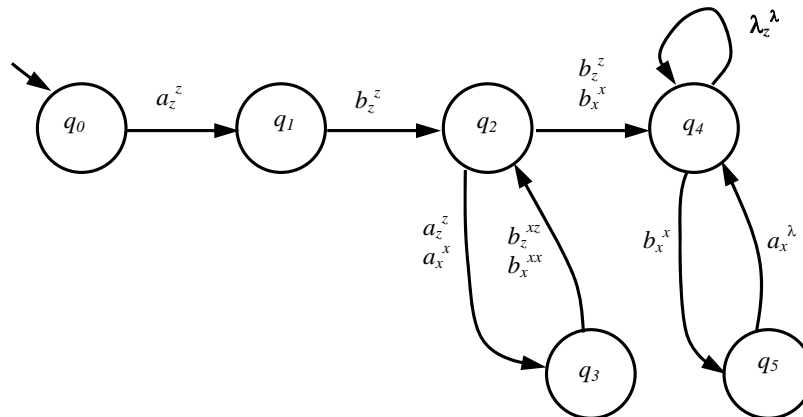
Soluciones:



8.- Diseñar un autómata a pila que (informalmente) reconozca el siguiente lenguaje

$$L = \{ab(ab)^nb(ba)^n \mid n \geq 0\}$$

Soluciones:



9.- Los lenguajes de programación de alto nivel a menudo incluyen diferentes delimitadores (paréntesis, llaves, etc...) que indican el comienzo y final de 'algo' (bloque de sentencias, subexpresiones, lista de parámetros en la definición o llamada a una función, etc...)

En ese contexto, se desea diseñar un lenguaje de programación para el 'lenguaje de los paréntesis'. Este lenguaje está formado por cadenas compuestas exclusivamente por paréntesis balanceados y todos incluidos en un paréntesis exterior.

Las siguientes cadenas pertenecen a este lenguaje:

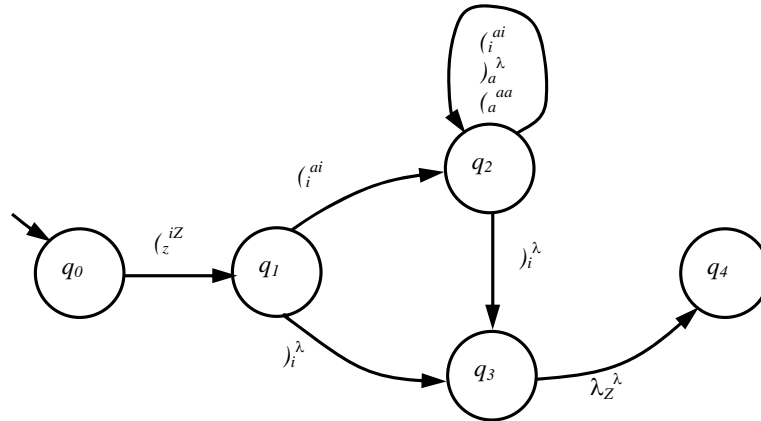
() ()
(() ())

Las siguientes cadenas no pertenecen a este lenguaje:

()
() ()

Diseñe un autómata a pila que reconozca el lenguaje de los paréntesis.

Soluciones:



10.- Se desea diseñar un autómata a pila capaz de reconocer la estructura básica de los programas escritos en cierto lenguaje de programación de alto nivel. La condición que se desea capturar simplemente es que las declaraciones deben realizarse antes de las sentencias en las que se utilicen las variables declaradas.

Tenga en cuenta también las siguientes características:

- Cada declaración individual será representada por el símbolo d .
- Cada sentencia individual será representada por el símbolo s .
- El programa completo está delimitado por los símbolos $\{$ y $\}$ que indican respectivamente el comienzo y el final del mismo.

El siguiente programa sería correcto

$\{dddss\}$

Pero el siguiente no, ya que entremezcla sentencias y declaraciones

$\{dsddss\}$

Soluciones:

