

Aceptadores finitos deteministas

José de Jesús Lavalle Martínez

Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación
Lenguajes Formales y Autómatas CCOS 014

- 1 Motivación
- 2 Aceptadores Finitos Deterministas (DFAs)
- 3 Lenguajes y DFAs
- 4 Lenguajes Regulares
- 5 Ejercicios

- Nuestra introducción en el primer capítulo a los conceptos básicos de computación, particularmente la discusión de los autómatas, es breve e informal.

- Para progresar, debemos ser más precisos, brindar definiciones formales y comenzar a desarrollar resultados rigurosos.

- Comenzamos con los aceptadores finitos, que son un caso especial simple del esquema general presentado en el capítulo anterior.

- Este tipo de autómatas se caracteriza por no tener almacenamiento temporal.

- Dado que un archivo de entrada no se puede reescribir, un autómata finito está severamente limitado en su capacidad para “recordar” cosas durante el cálculo.

- Se puede retener una cantidad finita de información en la unidad de control colocando la unidad en un estado específico.

- Pero dado que el número de tales estados es finito, un autómata finito solo puede lidiar con situaciones en las que la información que se almacenará en cualquier momento está estrictamente limitada.

Definición 1

Un **aceptador finito determinista** o **dfa** se define por el quintuple $M = (Q, \Sigma, \delta, q_0, F)$, donde

Definición 1

Un **aceptador finito determinista** o **dfa** se define por el quintuple $M = (Q, \Sigma, \delta, q_0, F)$, donde

- Q es un conjunto finito de estados internos,

Definición 1

Un **aceptador finito determinista** o **dfa** se define por el quintuple $M = (Q, \Sigma, \delta, q_0, F)$, donde

- Q es un conjunto finito de estados internos,
- Σ es un conjunto finito de símbolos llamado alfabeto de entrada,

Definición 1

Un **aceptador finito determinista** o **dfa** se define por el quintuple $M = (Q, \Sigma, \delta, q_0, F)$, donde

- Q es un conjunto finito de estados internos,
- Σ es un conjunto finito de símbolos llamado alfabeto de entrada,
- $\delta : Q \times \Sigma \rightarrow Q$ es una función total llamada función de transición,

Definición 1

Un **aceptador finito determinista** o **dfa** se define por el quintuple $M = (Q, \Sigma, \delta, q_0, F)$, donde

- Q es un conjunto finito de estados internos,
- Σ es un conjunto finito de símbolos llamado alfabeto de entrada,
- $\delta : Q \times \Sigma \rightarrow Q$ es una función total llamada función de transición,
- $q_0 \in Q$ es el estado inicial,

Definición 1

Un **aceptador finito determinista** o **dfa** se define por el quintuple $M = (Q, \Sigma, \delta, q_0, F)$, donde

- Q es un conjunto finito de estados internos,
- Σ es un conjunto finito de símbolos llamado alfabeto de entrada,
- $\delta : Q \times \Sigma \rightarrow Q$ es una función total llamada función de transición,
- $q_0 \in Q$ es el estado inicial,
- $F \subseteq Q$ es un conjunto de estados finales.

Funcionamiento de un aceptador finito determinista I

- Un aceptador finito determinista opera de la siguiente manera.

Funcionamiento de un aceptador finito determinista I

- En el momento inicial, se supone que está en el estado inicial q_0 , con su mecanismo de entrada en el símbolo más a la izquierda de la cadena de entrada.

Funcionamiento de un aceptador finito determinista I

- Durante cada movimiento del autómata, el mecanismo de entrada avanza una posición hacia la derecha, por lo que cada movimiento consume un símbolo de entrada.

Funcionamiento de un aceptador finito determinista I

- Cuando se alcanza el final de la cadena, la cadena se acepta si el autómata se encuentra en uno de sus estados finales, de lo contrario, la cadena se rechaza.

Funcionamiento de un aceptador finito determinista I

- El mecanismo de entrada solo puede moverse de izquierda a derecha y lee exactamente un símbolo en cada paso.

Funcionamiento de un aceptador finito determinista I

- Las transiciones de un estado interno a otro se rigen por la función de transición δ .

- Por ejemplo, si

$$\delta(q_0, a) = q_1,$$

entonces, si el dfa está en el estado q_0 y el símbolo de entrada actual es a , el dfa pasará al estado q_1 .

- Al hablar de los autómatas, es esencial tener una imagen clara e intuitiva con la que trabajar.

- Para visualizar y representar autómatas finitos, utilizamos grafos de transición, en los que los vértices representan estados y las aristas representan transiciones.

- Las etiquetas en los vértices son los nombres de los estados, mientras que las etiquetas en las aristas son los valores actuales del símbolo de entrada.

- Por ejemplo, si q_0 y q_1 son estados internos de algún dfa M , entonces el grafo asociado con M tendrá un vértice etiquetado como q_0 y otro etiquetado como q_1 .

- Una arista etiquetada (q_0, q_1) representa la transición $\delta(q_0, a) = q_1$.

- El estado inicial será identificado por una flecha entrante sin etiqueta que no se origina en ningún vértice.

- Los estados finales se dibujan con un círculo doble.

Funcionamiento de un aceptador finito determinista IV

- Más formalmente, si $M = (Q, \Sigma, \delta, q_0, F)$ es un aceptador finito determinista, entonces su grafo de transición asociado G_M tiene exactamente $|Q|$ vértices, cada uno etiquetado con un $q_i \in Q$ diferente.

- Para cada regla de transición $\delta(q_i, a) = q_j$, el grafo tiene una arista (q_i, q_j) etiquetada a .

- El vértice asociado con q_0 se llama vértice inicial, mientras que los etiquetados con $q_f \in F$ son los vértices finales.

- Es una cuestión trivial convertir de la definición $(Q, \Sigma, \delta, q_0, F)$ de un dfa a su representación como grafo de transición y viceversa.

Ejemplo 1 I

Ejemplo 1

El grafo de la Figura 1 representa al dfa

$$M = (\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_1\}),$$

donde δ está definida mediante

$$\delta(q_0, 0) = q_0,$$

$$\delta(q_1, 0) = q_0,$$

$$\delta(q_2, 0) = q_2,$$

$$\delta(q_0, 1) = q_1,$$

$$\delta(q_1, 1) = q_2,$$

$$\delta(q_2, 1) = q_1.$$

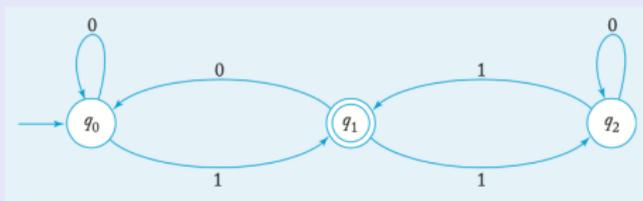
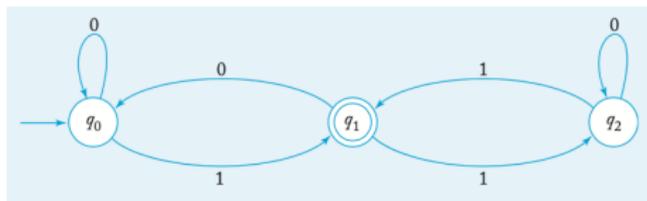


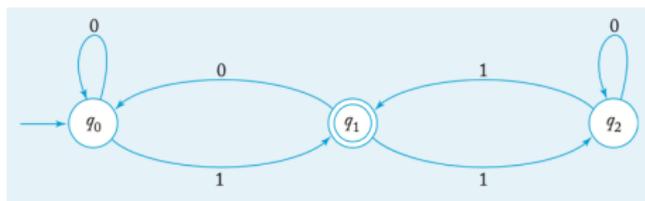
Figura 1: Grafo que representa al autómata del Ejemplo 1.

Ejemplo 1 II



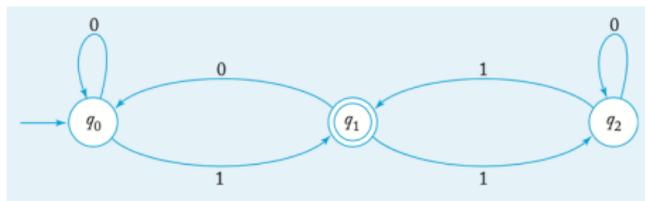
- Este dfa acepta la cadena 01.

Ejemplo 1 II



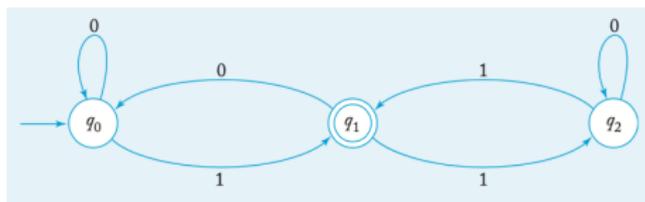
- Comenzando en el estado q_0 , se lee primero el símbolo 0. Mirando las aristas del grafo, vemos que el autómata permanece en el estado q_0 .

Ejemplo 1 II



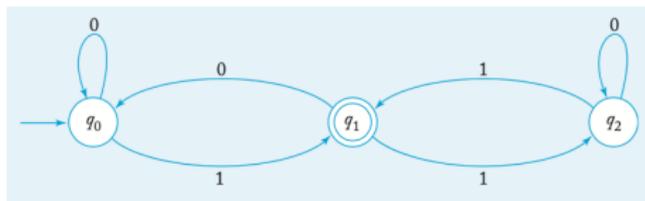
- A continuación, se lee el 1 y el autómata pasa al estado q_1 .

Ejemplo 1 II



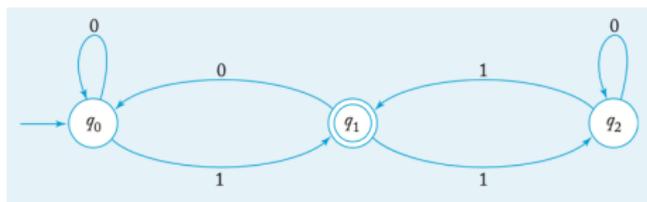
- Ahora estamos al final de la cadena y , al mismo tiempo, en un estado final q_1 .

Ejemplo 1 II



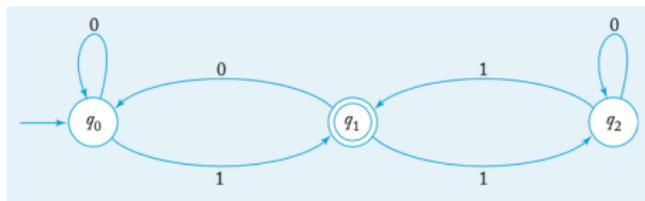
- Por tanto, se acepta la cadena 01.

Ejemplo 1 II



- El dfa no acepta la cadena 00, ya que después de leer dos ceros consecutivos, estará en el estado q_0 .

Ejemplo 1 II



- Con un razonamiento similar, vemos que el autómata aceptará las cadenas 101, 0111 y 11001, pero no 100 ni 1100. □

- Es conveniente introducir la función de transición extendida $\delta^* : Q \times \Sigma^* \rightarrow Q$.

- Es conveniente introducir la función de transición extendida $\delta^* : Q \times \Sigma^* \rightarrow Q$.
- El segundo argumento de δ^* es una cadena, en lugar de un solo símbolo, y su valor da el estado en el que se encontrará el autómata después de leer esa cadena .

- Es conveniente introducir la función de transición extendida $\delta^* : Q \times \Sigma^* \rightarrow Q$.
- El segundo argumento de δ^* es una cadena, en lugar de un solo símbolo, y su valor da el estado en el que se encontrará el autómata después de leer esa cadena .
- Por ejemplo, si

$$\delta(q_0, a) = q_1 \tag{1}$$

y

$$\delta(q_1, b) = q_2, \tag{2}$$

entonces

$$\delta^*(q_0, ab) = q_2.$$

Extensión de δ a cadenas II

- Formalmente, podemos definir recursivamente δ^* mediante

$$\delta^*(q, \lambda) = q, \quad (3)$$

$$\delta^*(q, wa) = \delta(\delta^*(q, w), a), \quad (4)$$

para todo $q \in Q$, $w \in \Sigma^*$ y $a \in \Sigma$.

Extensión de δ a cadenas II

- Formalmente, podemos definir recursivamente δ^* mediante

$$\delta^*(q, \lambda) = q, \quad (3)$$

$$\delta^*(q, wa) = \delta(\delta^*(q, w), a), \quad (4)$$

para todo $q \in Q$, $w \in \Sigma^*$ y $a \in \Sigma$.

- Para ver por qué esto es apropiado, apliquemos estas definiciones al caso simple anterior.

$$\begin{aligned} \delta^*(q_0, ab) &\stackrel{4}{=} \delta(\delta^*(q_0, a), b) \\ &\stackrel{4}{=} \delta(\delta(\delta^*(q_0, \lambda), a), b) \\ &\stackrel{3}{=} \delta(\delta(q_0, a), b) \\ &\stackrel{1}{=} \delta(q_1, b) \\ &\stackrel{2}{=} q_2, \end{aligned}$$

como se esperaba.

Definición de lenguaje aceptado por un dfa

- Una vez dada una definición precisa de un aceptador, ahora estamos listos para definir formalmente lo que entendemos por lenguaje asociado.

Definición de lenguaje aceptado por un dfa

- La asociación es obvia: el lenguaje es el conjunto de todas las cadenas aceptadas por el autómata.

Definición 2

El lenguaje aceptado por un dfa $M = (Q, \Sigma, \delta, q_0, F)$ es el conjunto de todas las cadenas en Σ aceptadas por M . En notación formal,

$$L(M) = \{w \in \Sigma^* : \delta^*(q_0, w) \in F\}.$$

- Tenga en cuenta que requerimos que δ , y en consecuencia δ^* , sean funciones totales.
- En cada paso, se define un movimiento único, por lo que se justifica llamar determinista a dicho autómata.
- Un dfa procesará cada cadena en Σ^* y la aceptará o no la aceptará.
- No aceptación significa que el dfa se detiene en un estado no final, de modo que

$$\overline{L(M)} = \{w \in \Sigma^* : \delta^*(q_0, w) \notin F\}.$$

Ejemplo 2

Considere el dfa en la Figura 2.



Figura 2: Grafo que representa al autómata del Ejemplo 2.

- Al dibujar la Figura 2 permitimos el uso de dos etiquetas en una sola arista.

Ejemplo 2

Considere el dfa en la Figura 2.



Figura 2: Grafo que representa al autómata del Ejemplo 2.

- Estas aristas con etiquetas múltiples son una forma abreviada de dos o más transiciones distintas:

Ejemplo 2

Considere el dfa en la Figura 2.



Figura 2: Grafo que representa al autómata del Ejemplo 2.

- La transición se toma siempre que el símbolo de entrada coincide con cualquiera de las etiquetas de la arista.

Ejemplo 2 II



- El autómata de la Figura 2 permanece en su estado inicial q_0 hasta que se encuentra la primera b .

Ejemplo 2 II



- Si éste es también el último símbolo de la entrada, entonces se acepta la cadena.

Ejemplo 2 II



- De lo contrario, el dfa entra en el estado q_2 , del que nunca podrá escapar.

Ejemplo 2 II



- El estado q_2 es un estado trampa.

Ejemplo 2 II



- Vemos claramente en el grafo que el autómata acepta todas las cadenas que constan de un número arbitrario de as , seguidas de una sola b .

Ejemplo 2 II



- Todas las demás cadenas de entrada se rechazan.

Ejemplo 2 II



- En notación de conjuntos, el lenguaje aceptado por el autómata es

$$L = \{a^n b : n \geq 0\}.$$

□

Correspondencia entre autómatas y grafos I

- Estos ejemplos muestran lo convenientes que son los grafos de transición para trabajar con autómatas finitos.

- Si bien es posible basar todos los argumentos estrictamente en las propiedades de la función de transición y su extensión a través de (3) y (4), los resultados son difíciles de seguir.

- En nuestra discusión, usamos grafos, que son más intuitivos, en la medida de lo posible.

- Para hacerlo, debemos, por supuesto, tener cierta seguridad de que la representación no nos engañará y de que los argumentos basados en grafos son tan válidos como los que utilizan las propiedades formales de δ .

- El siguiente resultado preliminar nos da esta seguridad.

Teorema 1

Sea $M = (Q, \Sigma, \delta, q_0, F)$ un aceptador finito determinista, y sea G_M su grafo de transición asociado. Entonces, para cada $q_i, q_j \in Q$ y $w \in \Sigma^+$, $\delta^*(q_i, w) = q_j$ si y sólo si en G_M hay una camino con etiqueta w de q_i a q_j .

Teorema 1

Sea $M = (Q, \Sigma, \delta, q_0, F)$ un aceptador finito determinista, y sea G_M su grafo de transición asociado. Entonces, para cada $q_i, q_j \in Q$ y $w \in \Sigma^+$, $\delta^*(q_i, w) = q_j$ si y sólo si en G_M hay un camino con etiqueta w de q_i a q_j .

- **Prueba:** Esta afirmación es bastante obvia al examinar casos tan simples como el Ejemplo 1.

Teorema 1

Sea $M = (Q, \Sigma, \delta, q_0, F)$ un aceptador finito determinista, y sea G_M su grafo de transición asociado. Entonces, para cada $q_i, q_j \in Q$ y $w \in \Sigma^+$, $\delta^*(q_i, w) = q_j$ si y sólo si en G_M hay un camino con etiqueta w de q_i a q_j .

- Puede demostrarse rigurosamente utilizando inducción sobre la longitud de w .

Correspondencia entre autómatas y grafos III

- Suponga que la afirmación es verdadera para todas las cadenas v con $|v| \leq n$.

Correspondencia entre autómatas y grafos III

- Suponga que la afirmación es verdadera para todas las cadenas v con $|v| \leq n$.
- Considere entonces cualquier w de longitud $n + 1$ y escríbala como

$$w = va.$$

Correspondencia entre autómatas y grafos III

- Suponga que la afirmación es verdadera para todas las cadenas v con $|v| \leq n$.
- Considere entonces cualquier w de longitud $n + 1$ y escríbala como

$$w = va.$$

- Supongamos ahora que $\delta^*(q_i, v) = q_k$.

Correspondencia entre autómatas y grafos III

- Suponga que la afirmación es verdadera para todas las cadenas v con $|v| \leq n$.
- Considere entonces cualquier w de longitud $n + 1$ y escríbala como

$$w = va.$$

- Supongamos ahora que $\delta^*(q_i, v) = q_k$.
- Dado que $|v| = n$, debe haber un camino en G_M etiquetado con v de q_i a q_k .

Correspondencia entre autómatas y grafos III

- Suponga que la afirmación es verdadera para todas las cadenas v con $|v| \leq n$.
- Considere entonces cualquier w de longitud $n + 1$ y escríbala como

$$w = va.$$

- Supongamos ahora que $\delta^*(q_i, v) = q_k$.
- Dado que $|v| = n$, debe haber un camino en G_M etiquetado con v de q_i a q_k .
- Pero si $\delta^*(q_i, w) = q_j$, entonces M debe tener una transición $\delta(q_k, a) = q_j$, de modo que por construcción G_M tiene una arista (q_k, q_j) con etiqueta a .

Correspondencia entre autómatas y grafos IV

- Por lo tanto, hay un camino en G_M etiquetado con $va = w$ entre q_i y q_j .



Correspondencia entre autómatas y grafos IV

- Por lo tanto, hay un camino en G_M etiquetado con $va = w$ entre q_i y q_j .
- Dado que el resultado es obviamente cierto para $n = 1$, podemos afirmar por inducción que, para todo $w \in \Sigma^+$,



Correspondencia entre autómatas y grafos IV

- Por lo tanto, hay un camino en G_M etiquetado con $va = w$ entre q_i y q_j .
- Dado que el resultado es obviamente cierto para $n = 1$, podemos afirmar por inducción que, para todo $w \in \Sigma^+$,

-

$$\delta^*(q_i, w) = q_j \quad (5)$$

implica que hay un camino en G_M de q_i a q_j etiquetado con w .



Correspondencia entre autómatas y grafos IV

- Por lo tanto, hay un camino en G_M etiquetado con $va = w$ entre q_i y q_j .
- Dado que el resultado es obviamente cierto para $n = 1$, podemos afirmar por inducción que, para todo $w \in \Sigma^+$,

-

$$\delta^*(q_i, w) = q_j \quad (5)$$

implica que hay un camino en G_M de q_i a q_j etiquetado con w .

- El argumento se puede invertir de una manera sencilla para mostrar que la existencia de tal camino implica (5), completando así la prueba.



- Nuevamente, el resultado del teorema es tan intuitivamente obvio que una demostración formal parece innecesaria.

- Revisamos los detalles por dos razones.

- Revisamos los detalles por dos razones.
 - La primera es que es un ejemplo simple pero típico de una prueba inductiva en relación con los autómatas.

- Revisamos los detalles por dos razones.
 - La segunda es que el resultado se usará una y otra vez, por lo que afirmarlo y demostrarlo como un teorema nos permite argumentar con bastante confianza usando grafos.

- Esto hace que nuestros ejemplos y pruebas sean más transparentes de lo que serían si usáramos las propiedades de δ^* .

Representación de autómatas mediante tablas

- Si bien los grafos son convenientes para visualizar autómatas, también son útiles otras representaciones.

- Por ejemplo, podemos representar la función δ como una tabla.

- La Tabla 1 es equivalente al grafo de la Figura 2.

	a	b
q_0	q_0	q_1
q_1	q_2	q_2
q_2	q_2	q_2

Tabla 1: Tabla equivalente al grafo de la Figura 2.

Representación de autómatas mediante tablas

- Aquí, la etiqueta de la fila es el estado actual, mientras que la etiqueta de la columna representa el símbolo de entrada actual.

	a	b
q_0	q_0	q_1
q_1	q_2	q_2
q_2	q_2	q_2

Tabla 1: Tabla equivalente al grafo de la Figura 2.

- La entrada en la tabla define el siguiente estado.

	a	b
q_0	q_0	q_1
q_1	q_2	q_2
q_2	q_2	q_2

Tabla 1: Tabla equivalente al grafo de la Figura 2.

Ejemplo 3

Encuentre un aceptador finito determinista que reconozca el conjunto de todas las cadenas en $\Sigma = \{a, b\}$ comenzando con el prefijo ab .

- El único problema aquí son los dos primeros símbolos de la cadena; una vez leídas, no se necesitan más decisiones.

Ejemplo 3

Encuentre un aceptador finito determinista que reconozca el conjunto de todas las cadenas en $\Sigma = \{a, b\}$ comenzando con el prefijo ab .

- Aún así, el autómata tiene que procesar toda la cadena antes de tomar su decisión.

Ejemplo 3

Encuentre un aceptador finito determinista que reconozca el conjunto de todas las cadenas en $\Sigma = \{a, b\}$ comenzando con el prefijo ab .

- Por lo tanto, podemos resolver el problema con un autómata que tiene cuatro estados: un estado inicial, dos estados para reconocer ab que termina en un estado de trampa final y un estado de trampa no final.

Ejemplo 3 II

- Si el primer símbolo es una a y el segundo es una b , el autómata pasa al estado de trampa final, donde permanecerá ya que el resto de la entrada no importa.

Ejemplo 3 II

- Si el primer símbolo es una a y el segundo es una b , el autómata pasa al estado de trampa final, donde permanecerá ya que el resto de la entrada no importa.
- Por otro lado, si el primer símbolo no es una a o el segundo no es una b , el autómata entra en el estado de trampa no final.

Ejemplo 3 II

- Si el primer símbolo es una a y el segundo es una b , el autómata pasa al estado de trampa final, donde permanecerá ya que el resto de la entrada no importa.
- Por otro lado, si el primer símbolo no es una a o el segundo no es una b , el autómata entra en el estado de trampa no final.
- La solución simple se muestra en la Figura 3.

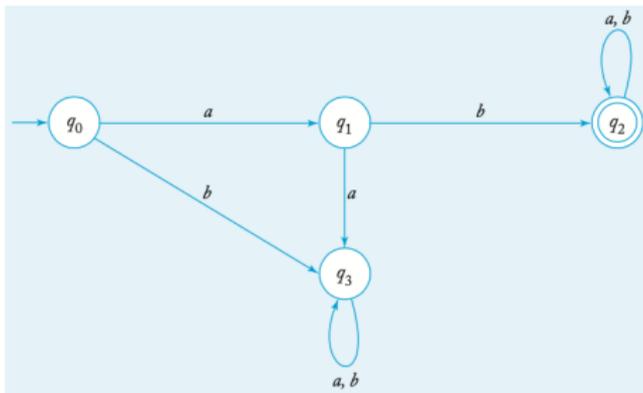


Figura 3: Grafo que representa al autómata del Ejemplo 3.

Ejemplo 4

Busque un dfa que acepte todas las cadenas en $0,1$, excepto las que contengan la subcadena 001 .

- Al decidir si se ha producido la subcadena 001 , necesitamos saber no sólo el símbolo de entrada actual, sino también recordar si ha sido o no precedido por uno o dos ceros.

Ejemplo 4

Busque un dfa que acepte todas las cadenas en $0,1$, excepto las que contengan la subcadena 001 .

- Podemos realizar un seguimiento de esto poniendo el autómata en estados específicos y etiquetándolos en consecuencia.

Ejemplo 4

Busque un dfa que acepte todas las cadenas en $0,1$, excepto las que contengan la subcadena 001 .

- Al igual que los nombres de variables en un lenguaje de programación, los nombres de estado son arbitrarios y se pueden elegir por razones mnemotécnicas.

Ejemplo 4

Busque un dfa que acepte todas las cadenas en $0,1$, excepto las que contengan la subcadena 001 .

- Por ejemplo, el estado en el que dos 0 eran los símbolos inmediatamente anteriores se puede etiquetar simplemente como 00 .

Ejemplo 4 II

- Si la cadena comienza con 001, debe rechazarse.

Ejemplo 4 II

- Esto implica que debe haber una ruta etiquetada como 001 desde el estado inicial a un estado no final.

Ejemplo 4 II

- Por conveniencia, este estado no final está etiquetado como 001.

- Este estado debe ser un estado de trampa, porque los símbolos posteriores no importan.

- Todos los demás estados son estados de aceptación.

- Esto nos da la estructura básica de la solución, pero aún debemos agregar provisiones para la subcadena 001 que ocurre en el medio de la entrada.

Ejemplo 4 II

- Debemos definir Q y δ de modo que el autómata recuerde todo lo que necesitamos para tomar la decisión correcta.

Ejemplo 4 III

- En este caso, cuando se lee un símbolo, necesitamos saber alguna parte de la cadena a la izquierda, por ejemplo, si los dos símbolos anteriores eran 00 o no.

- Si etiquetamos los estados con los símbolos relevantes, es muy fácil para ver cuáles deben ser las transiciones.

- Por ejemplo,

$$\delta(00, 0) = 00$$

porque esta situación surge sólo si hay tres ceros consecutivos.

- Sólo nos interesan los dos últimos, un hecho que recordamos al mantener el dfa en el estado 00.

- En la Figura 4 se muestra una solución completa.

Ejemplo 4 IV

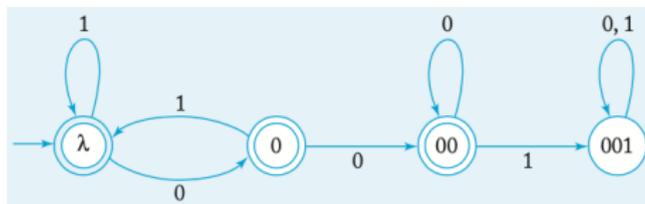


Figura 4: Grafo que representa al autómata del Ejemplo 4.

Ejemplo 4 IV

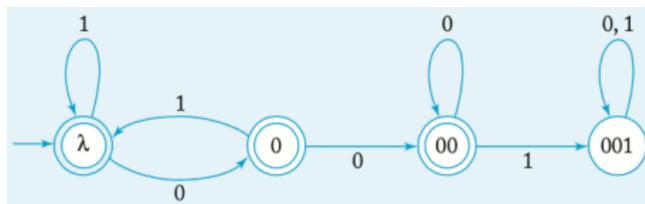


Figura 4: Grafo que representa al autómata del Ejemplo 4.

- Vemos en este ejemplo lo útiles que son las etiquetas mnemotécnicas en los estados para realizar un seguimiento de las cosas.

Ejemplo 4 IV

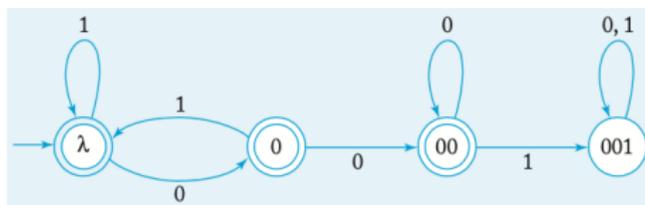


Figura 4: Grafo que representa al autómata del Ejemplo 4.

- Rastree algunas cadenas, como 100100 y 1010100, para ver que la solución sea correcta.

- Todo autómata finito acepta algún lenguaje.

- Si consideramos todos los posibles autómatas finitos, obtenemos un conjunto de lenguajes asociados con ellos.

- Llamaremos familia a ese conjunto de lenguajes.

- La familia de lenguajes que aceptan los aceptadores finitos deterministas es bastante limitada.

- La estructura y propiedades de los lenguajes de esta familia se aclararán a medida que avance nuestro estudio; por el momento simplemente asignaremos un nombre a esta familia.

Definición 3

Un lenguaje L se llama *regular* si y sólo si existe algún aceptador finito determinista M tal que

$$L = L(M).$$

Ejemplo 5

Muestre que el lenguaje

$$L = \{awa; w \in \{a, b\}^*\}$$

es regular.

- Para demostrar que este o cualquier otro lenguaje es regular, todo lo que tenemos que hacer es encontrar un dfa para ello.

Ejemplo 5

Muestre que el lenguaje

$$L = \{awa; w \in \{a, b\}^*\}$$

es regular.

- La construcción de un dfa para este lenguaje es similar al Ejemplo 3, pero un poco más complicada.

Ejemplo 5

Muestre que el lenguaje

$$L = \{awa; w \in \{a, b\}^*\}$$

es regular.

- Lo que debe hacer este dfa es verificar si una cadena comienza y termina con una a ; lo que está en medio es inmaterial.

Ejemplo 5 II

- La solución se complica por el hecho de que no hay una forma explícita de probar el final de la cadena.

Ejemplo 5 II

- Esta dificultad se supera simplemente poniendo el dfa en un estado final siempre que se encuentre la segunda a .

Ejemplo 5 II

- Si este no es el final de la cadena y se encuentra otra b , el dfa saldrá del estado final.

- El escaneo continúa de esta manera, cada uno llevando al autómata a su estado final.

- La solución completa se muestra en la Figura 5.

- Nuevamente, rastree algunos ejemplos para ver por qué esto funciona.

Ejemplo 5 II

- Después de una o dos pruebas, será obvio que la dfa acepta una cadena si y sólo si comienza y termina con una a .

- Dado que hemos construido un dfa para el lenguaje, podemos afirmar que, por definición, el lenguaje es regular.

Ejemplo 5 III

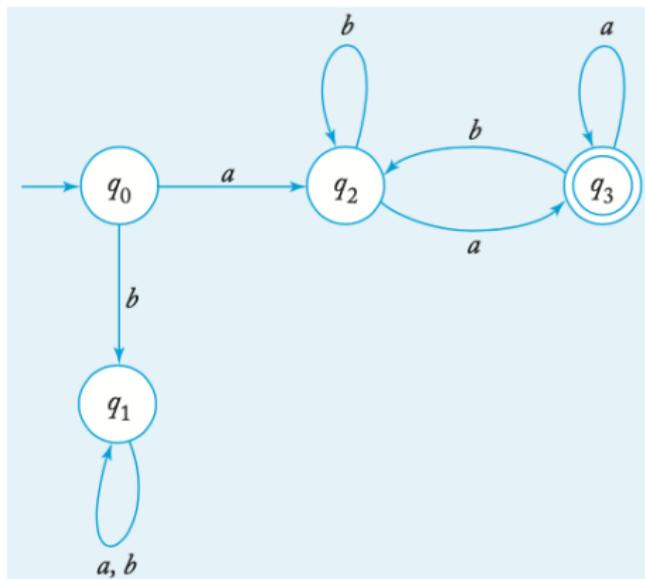


Figura 5: Grafo que representa al autómata del Ejemplo 5.

Ejemplo 6

Sea L el lenguaje del Ejemplo 5, muestre que L^2 es regular.

- Nuevamente mostramos que el lenguaje es regular construyendo un dfa para él.

Ejemplo 6

Sea L el lenguaje del Ejemplo 5, muestre que L^2 es regular.

- Podemos escribir una expresión explícita para L^2 , a saber,

$$L^2 = \{aw_1aaw_2a : w_1, w_2 \in \{a, b\}^*\}.$$

Ejemplo 6

Sea L el lenguaje del Ejemplo 5, muestre que L^2 es regular.

- Por lo tanto, necesitamos un dfa que reconozca dos cadenas consecutivas de esencialmente la misma forma (pero no necesariamente idénticas en valor).

Ejemplo 6

Sea L el lenguaje del Ejemplo 5, muestre que L^2 es regular.

- El diagrama de la Figura 5 se puede utilizar como punto de partida, pero el vértice q_3 debe modificarse.

Ejemplo 6

Sea L el lenguaje del Ejemplo 5, muestre que L^2 es regular.

- Este estado ya no puede ser final ya que, en este punto, debemos empezar a buscar una segunda subcadena de la forma awa .

Ejemplo 6 II

- Para reconocer la segunda subcadena, replicamos los estados de la primera parte (con nuevos nombres), con q_3 como comienzo de la segunda parte.

Ejemplo 6 II

- Dado que la cadena completa se puede dividir en sus partes constituyentes dondequiera que ocurra aa , permitimos que la primera aparición de dos a consecutivas sea el disparador que lleva al autómata a su segunda parte.

Ejemplo 6 II

- Podemos hacer esto haciendo $\delta(q_3, a) = q_4$.

- La solución completa está en la Figura 6.

Ejemplo 6 II

- Este dfa acepta L^2 , que por lo tanto es regular.

Ejemplo 6 III

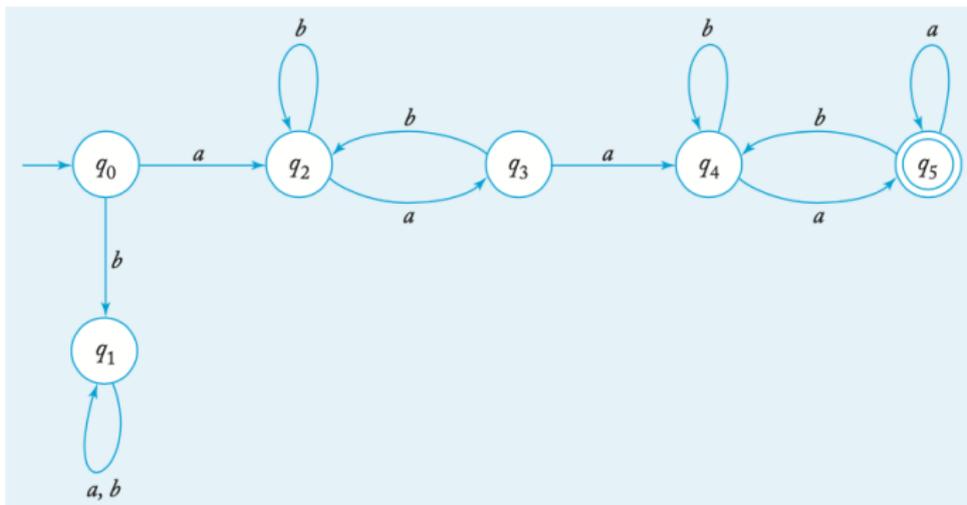


Figura 6: Grafo que representa al autómata del Ejemplo 6.

- 1 ¿Cuáles de las cadenas 0001, 01101, 00001101 son aceptadas por el dfa en la Figura 1?
- 2 Traduzca la gráfica de la Figura 4 a notación δ .
- 3 Para $\Sigma = \{a, b\}$, construya dfa's que acepten los conjuntos que consisten de
 - 1 todas las cadenas con un número par de símbolos a .
 - 2 todas las cadenas con un número par de símbolos a y un número impar de símbolos b .
 - 3 todas las cadenas con al menos un símbolo b y exactamente dos símbolos a .
 - 4 todas las cadenas con exactamente dos símbolos a y más de tres símbolos b .
- 4 Construya dfas para los siguientes lenguajes sobre $\Sigma = \{a, b\}$.
 - 1 $L = \{ab^4wb^2 : w \in \{a, b\}^*\}$.
 - 2 $L = \{ab^n a^m : n \geq 3, m \geq 2\}$.
 - 3 $L = \{w : |w| \bmod 3 \neq 0\}$.
 - 4 $L = \{w : |w| \bmod 5 = 0\}$.

- Demuestre que si cambiamos la Figura 5, haciendo q_3 un estado no final y haciendo q_0, q_1, q_2 estados finales, el dfa resultante acepta \overline{L} .
- Generalice la observación en el ejercicio previo. Específicamente, demuestre que si $M = (Q, \Sigma, \delta, q_0, F)$ y $\widehat{M} = (Q, \Sigma, \delta, q_0, Q - F)$ son dos dfas, entonces $\overline{L(M)} = L(\widehat{M})$.
- Use las ecuaciones (3) y (4) para demostrar que

$$\delta^*(q, wv) = \delta^*(\delta^*(q, w), v)$$

para toda $w, v \in \Sigma^*$.