Equivalencia DFAs-NFAs y Expresiones Regulares

José de Jesús Lavalle Martínez

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

Primavera 2021

Contenido

Motivación

- Llegamos ahora a una cuestión fundamental.

Motivación

- ¿En qué sentido son diferentes los dfas y nfas?

Motivación

- Evidentemente, existe una diferencia en su definición, pero esto no implica que exista una distinción esencial entre ellos.

Motivación

- Para explorar esta pregunta, presentamos el concepto de equivalencia entre autómatas.

Equivalencia entre aceptadores

Definición 1

Se dice que dos aceptadores finitos, M_1 y M_2, son equivalentes si

$$L(M_1) = L(M_2),$$

es decir, si ambos aceptan el mismo lenguaje

Ejemplo 1

Ejemplo 1

El dfa que se muestra en la Figura 1 es equivalente al nfa en la Figura 2 ya que ambos aceptan el lenguaje $\{(10)^n : n \geq 0\}$.

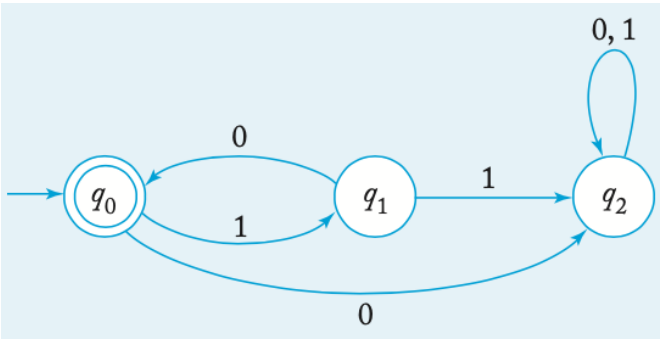

Figura 1: dfa equivalente al nfa de la Figura 2.

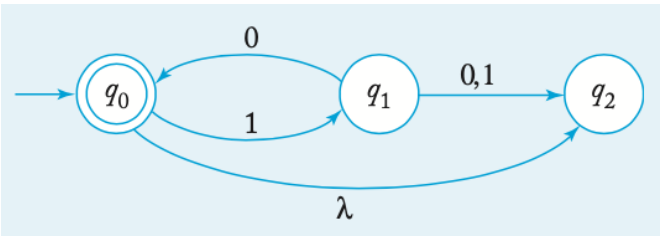

Figura 2: Grafo que representa un nfa que reconoce $\{(10)^n : n \geq 0\}$.

Equivalencia entre dfas y nfas I

- Cuando comparamos diferentes clases de autómatas, invariablemente surge la pregunta de si una clase es más poderosa que la otra.

Equivalencia entre dfas y nfas I

- Por "más poderoso" queremos decir que un autómata de un tipo puede lograr algo que ningún autómata del otro tipo puede hacer.

Equivalencia entre dfas y nfas I

- Examinemos esta pregunta para los aceptadores finitos.

Equivalencia entre dfas y nfas I

- Dado que un dfa es en esencia un tipo restringido de nfa, está claro que cualquier lenguaje que sea aceptado por un dfa también lo es por algunos nfa.

Equivalencia entre dfas y nfas I

- Pero lo contrario no es tan obvio.

Equivalencia entre dfas y nfas II

- Hemos añadido el no determinismo, por lo que es al menos concebible que haya un lenguaje aceptado por alguna nfa para el que, en principio, no podemos encontrar un dfa.

Equivalencia entre dfas y nfas II

- Pero resulta que no es así.

Equivalencia entre dfas y nfas II

- Las clases de dfa y nfa son igualmente poderosas: para cada lenguaje aceptado por algún nfa, hay un dfa que acepta el mismo lenguaje.

Equivalencia entre dfas y nfas II

- Este resultado no es obvio y ciertamente debe demostrarse.

Equivalencia entre dfas y nfas II

- El argumento, como la mayoría de los argumentos de este libro, será constructivo.

Equivalencia entre dfas y nfas II

- Esto significa que en realidad podemos ofrecer una forma de convertir cualquier nfa en un dfa equivalente.

Equivalencia entre dfas y nfas III

- La construcción no es difícil de entender; una vez que la idea es clara, se convierte en el punto de partida para un argumento riguroso.

Equivalencia entre dfas y nfas III

- El fundamento de la construcción es el siguiente.

Equivalencia entre dfas y nfas III

- Después de que una nfa ha leído una cadena w, es posible que no sepamos exactamente en qué estado estará, pero podemos decir que debe estar en un estado de un conjunto de estados posibles, digamos $\{qi, qj, ..., qk\}$.

Equivalencia entre dfas y nfas III

- Un dfa equivalente después de leer la misma cadena debe estar en algún estado definido.

Equivalencia entre dfas y nfas III

- ¿Cómo podemos hacer que estas dos situaciones se correspondan?

Equivalencia entre dfas y nfas IV

- La respuesta es un buen truco: etiquete los estados del dfa con un conjunto de estados de tal manera que, después de leer w, el dfa equivalente estará en un solo estado etiquetado $\{qi, qj, ..., qk\}$.

Equivalencia entre dfas y nfas IV

- Dado que para un conjunto de $|Q|$ estados hay exactamente $2^{|Q|}$ subconjuntos, el correspondiente dfa tendrá un número finito de estados.

Equivalencia entre dfas y nfas IV

- La mayor parte del trabajo en esta construcción sugerida radica en el análisis del nfa para obtener la correspondencia entre los posibles estados y las entradas.

Equivalencia entre dfas y nfas IV

- Antes de llegar a la descripción formal de esto, ilustremos con un ejemplo sencillo.

Ejemplo 2 I

Ejemplo 2

Convierta el nfa de la Figura 3 en un dfa equivalente.

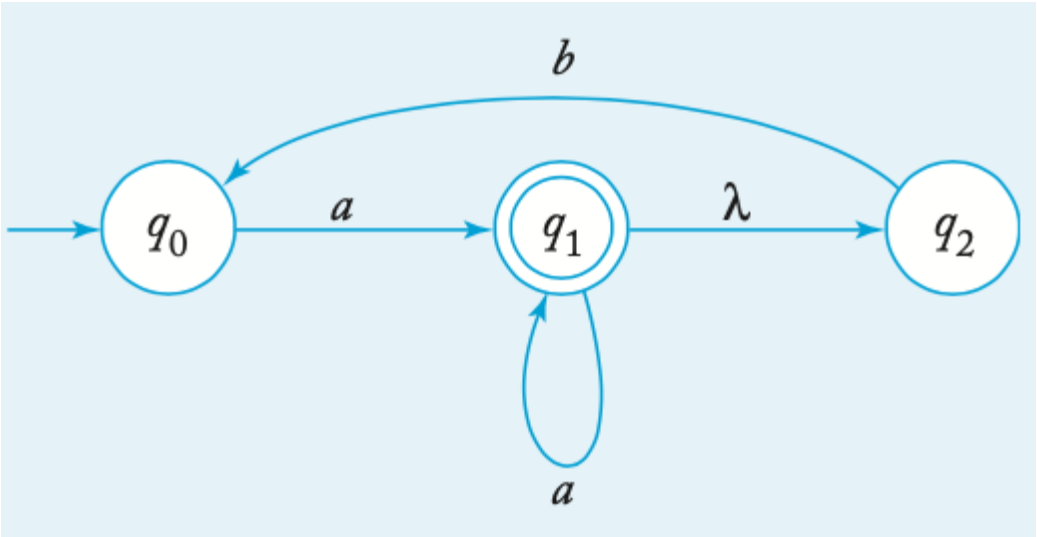

Figura 3: Autómata para el Ejemplo 2.

Ejemplo 2 II

- El nfa comienza en el estado q_0, por lo que el estado inicial del dfa se etiquetará como $\{q_0\}$.

Ejemplo 2 II

- Después de leer una a, el nfa puede estar en el estado q_1 o, al hacer una transición λ, en el estado q_2.

Ejemplo 2 II

- Por lo tanto, el dfa correspondiente debe tener un estado etiquetado $\{q_1, q_2\}$ y una transición

$$\delta(\{q_0\}, a) = \{q_1, q_2\}.$$

Ejemplo 2 II

- Ahora hemos introducido en el dfa el estado $\{q_1, q_2\}$, por lo que necesitamos encontrar las transiciones fuera de este estado.

Ejemplo 2 II

- Recuerde que este estado del dfa corresponde a dos posibles estados del nfa, por lo que debemos volver a referirnos al nfa.

Ejemplo 2 III

- Si el nfa está en el estado q_1 y lee una a, puede ir a q_1.

Ejemplo 2 III

- Además, desde q_1 el nfa puede hacer una transición λ a q_2.

Ejemplo 2 III

- Si, para la misma entrada, el nfa está en el estado q_2, entonces no hay una transición especificada.

Ejemplo 2 III

- Por lo tanto,

$$\delta(\{q_1, q2\}, a) = \{q_1, q_2\}.$$

Ejemplo 2 III

- Similarmente,

$$\delta(\{q_1, q2\}, b) = \{q_0\}.$$

Ejemplo 2 IV

- En este punto, cada estado tiene todas las transiciones definidas.

Ejemplo 2 IV

- El resultado, que se muestra en la Figura 4, es un dfa, equivalente al nfa con el que comenzamos.

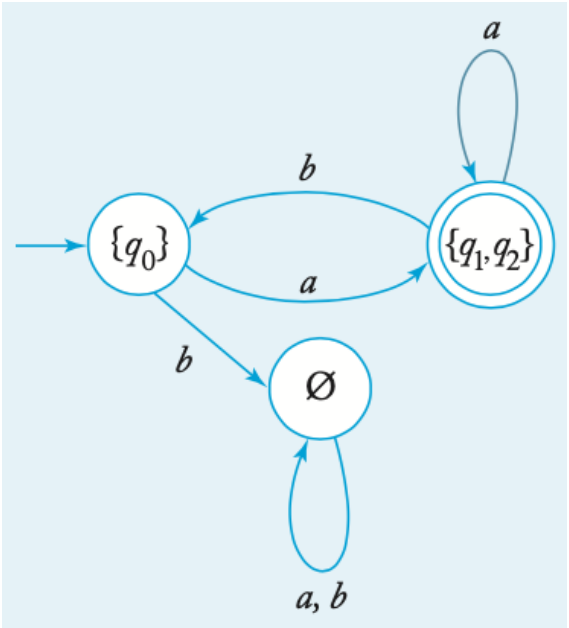

Figura 4: dfa equivalente al nfa de la Figura 3.

Ejemplo 2 IV

- El nfa de la Figura 3 acepta cualquier cadena para la que $\delta^*(q_0, w)$ contenga q_1.

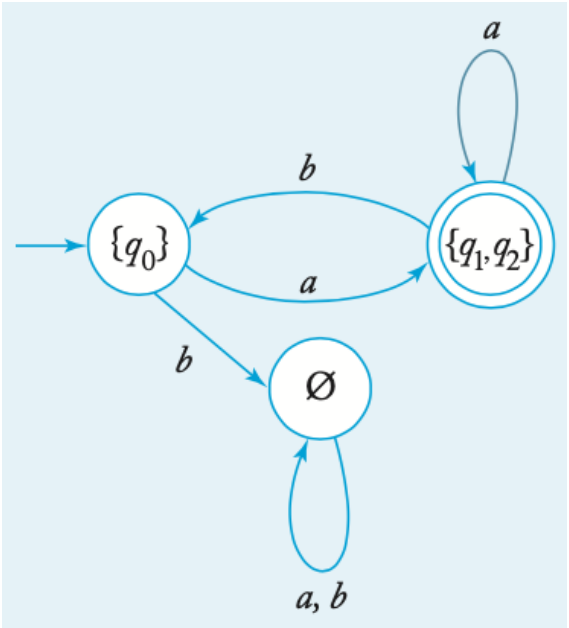

Figura 4: dfa equivalente al nfa de la Figura 3.

Ejemplo 2 IV

- Para que el dfa correspondiente acepte cada w, cualquier estado cuya etiqueta incluya q_1 debe convertirse en un estado final.

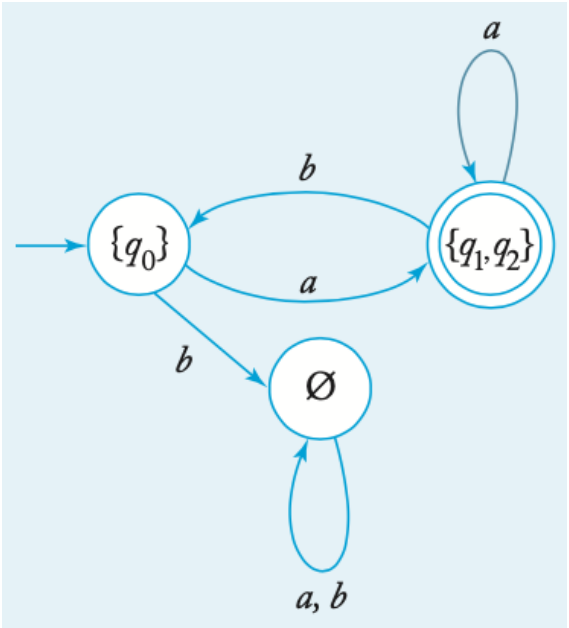

Figura 4: dfa equivalente al nfa de la Figura 3.

Todo nfa tiene un dfa equivalente I

Teorema 1

Sea L el lenguaje aceptado por un aceptador finito no determinista $M_N = (Q_N, \Sigma, \delta_N, q_0, F_N)$. Entonces existe un aceptador finito determinista $M_D = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$ tal que

$$L = L(M_D).$$

Todo nfa tiene un dfa equivalente I

Teorema 1

Sea L el lenguaje aceptado por un aceptador finito no determinista $M_N = (Q_N, \Sigma, \delta_N, q_0, F_N)$. Entonces existe un aceptador finito determinista $M_D = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$ tal que

$$L = L(M_D).$$

Demostración:

Todo nfa tiene un dfa equivalente I

Teorema 1

Sea L el lenguaje aceptado por un aceptador finito no determinista $M_N = (Q_N, \Sigma, \delta_N, q_0, F_N)$. Entonces existe un aceptador finito determinista $M_D = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$ tal que

$$L = L(M_D).$$

Demostración:

- Dado M_N, usamos el procedimiento nfa-to-dfa, mostrado a continuación, para construir el grafo de transición G_D para M_D.

Todo nfa tiene un dfa equivalente I

Teorema 1

Sea L el lenguaje aceptado por un aceptador finito no determinista $M_N = (Q_N, \Sigma, \delta_N, q_0, F_N)$. Entonces existe un aceptador finito determinista $M_D = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$ tal que

$$L = L(M_D).$$

Demostración:

- Para comprender la construcción, recuerde que G_D debe tener ciertas propiedades.

Todo nfa tiene un dfa equivalente I

Teorema 1

Sea L el lenguaje aceptado por un aceptador finito no determinista $M_N = (Q_N, \Sigma, \delta_N, q_0, F_N)$. Entonces existe un aceptador finito determinista $M_D = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$ tal que

$$L = L(M_D).$$

Demostración:

- Cada vértice debe tener exactamente $|\Sigma|$ aristas salientes, cada una etiquetada con un elemento diferente de Σ.

Todo nfa tiene un dfa equivalente I

Teorema 1

Sea L el lenguaje aceptado por un aceptador finito no determinista $M_N = (Q_N, \Sigma, \delta_N, q_0, F_N)$. Entonces existe un aceptador finito determinista $M_D = (Q_D, \Sigma, \delta_D, \{q_0\}, F_D)$ tal que

$$L = L(M_D).$$

Demostración:

- Durante la construcción, es posible que falten algunas de las aristas, pero el procedimiento continúa hasta que estén todas allí.

Todo nfa tiene un dfa equivalente II

procedimiento nfa-a-dfa

1. Cree un grafo G_D con vértice $\{q_0\}$. Identifique este vértice como el vértice inicial.
2. Repita los siguientes pasos hasta que no falten más aristas.
   1. Tome cualquier vértice $\{q_i, q_j, \ldots, q_k\}$ de G_D que no tenga aristas salientes para $a \in \Sigma$. Calcule $\delta_N^*(q_i, a), \delta_N^*(q_j, a), \ldots, \delta_N^*(q_k, a)$. Si

$$\delta_N^*(q_i, a) \cup \delta_N^*(q_j, a) \cup \cdots \cup \delta_N^*(q_k, a) = \{q_l, q_m, \ldots, q_n\},$$

      cree un vértice para G_D etiquetado $\{q_l, q_m, \ldots, q_n\}$ si aún no existe.
   2. Agregue a G_D una arista de $\{q_i, q_j, \ldots, q_k\}$ a $\{q_l, q_m, \ldots, q_n\}$ y etiquételo con a.
3. Cada estado de G_D cuya etiqueta contiene cualquier $q_f \in F_N$ se identifica como un vértice final.
4. Si M_N acepta λ, el vértice $\{q_0\}$ en G_D también se convierte en un vértice final.

Todo nfa tiene un dfa equivalente III

- Está claro que este procedimiento siempre termina.

Todo nfa tiene un dfa equivalente III

- Cada paso a través del ciclo en el paso 2 agrega una arista a G_D.

Todo nfa tiene un dfa equivalente III

- Pero G_D tiene como máximo $2^{|Q_N|}|\Sigma|$ aristas, de modo que el bucle finalmente se detiene.

Todo nfa tiene un dfa equivalente III

- Para mostrar que la construcción también da la respuesta correcta, argumentamos por inducción sobre la longitud de la cadena de entrada.

Todo nfa tiene un dfa equivalente III

- Suponga que para cada v de longitud menor o igual a n, la presencia en G_N de un camino etiquetado v de q_0 a q_i implica que en G_D hay un camino etiquetado v de $\{q_0\}$ a un estado $Q_i = \{\ldots, q_i, \ldots\}$.

Todo nfa tiene un dfa equivalente III

- Considere ahora cualquier $w = va$ y observe un camino en G_N etiquetado como w de q_0 a q_l.

Todo nfa tiene un dfa equivalente III

- Entonces debe haber un camino etiquetado v de q_0 a q_i y una arista (o una secuencia de aristas) etiquetada a de q_i a q_l.

Todo nfa tiene un dfa equivalente IV

- Según el supuesto inductivo, en G_D habrá un camino etiquetado v desde $\{q_0\}$ a Q_i.

Todo nfa tiene un dfa equivalente IV

- Pero por construcción, habrá una arista de Q_i a algún estado cuya etiqueta contenga q_l.

Todo nfa tiene un dfa equivalente IV

- Por lo tanto, la suposición inductiva se cumple para todas las cadenas de longitud $n + 1$.

Todo nfa tiene un dfa equivalente IV

- Como es obviamente cierto para $n = 1$, lo es para todos n.

Todo nfa tiene un dfa equivalente IV

- El resultado entonces es que siempre que $\delta_N^*(q_0, w)$ contiene un estado final q_f, también lo hace la etiqueta de $\delta_D^*(q_0, w)$.

Todo nfa tiene un dfa equivalente IV

- Para completar la demostración, invertimos el argumento para mostrar que si la etiqueta de $\delta_D^*(q_0, w)$ contiene q_f, entonces $\delta_N^*(q_0, w)$ también la debe contener. ■

Ejemplo 3 I

Ejemplo 3

Convierta el nfa de la Figura 5 en una máquina determinista equivalente.

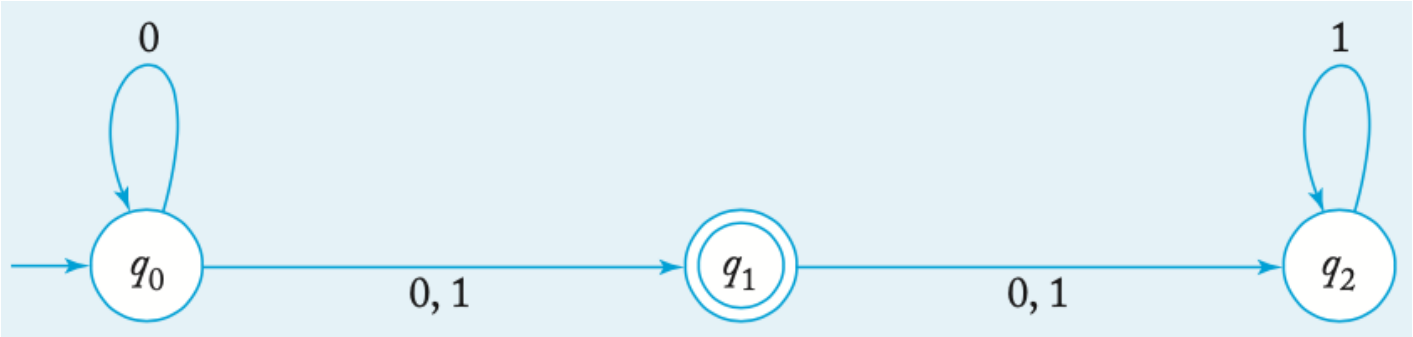

Figura 5: nfa para el Ejemplo 3.

Ejemplo 3 I

Ejemplo 3

Convierta el nfa de la Figura 5 en una máquina determinista equivalente.

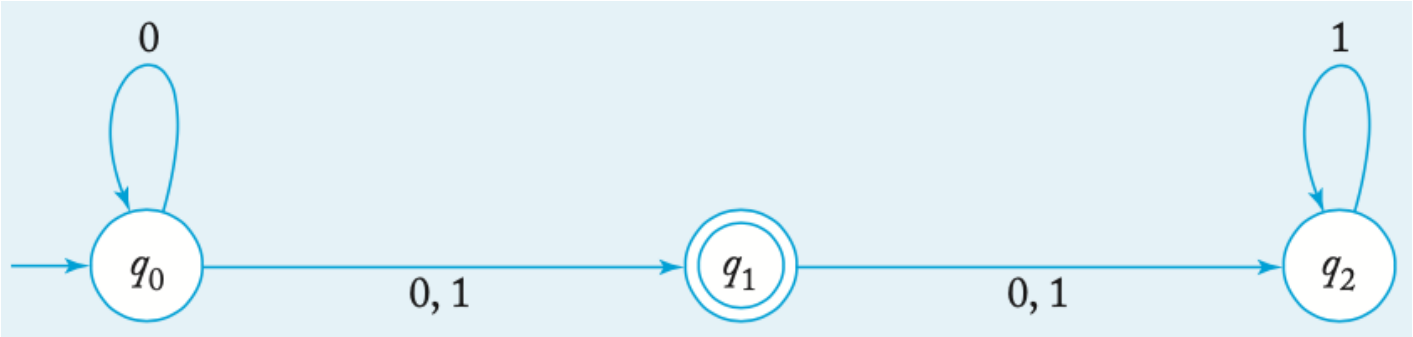

Figura 5: nfa para el Ejemplo 3.

- Como $\delta_N(q_0, 0) = \{q_0, q_1\}$, introducimos el estado $\{q_0, q_1\}$ en G_D y agregamos una arista etiquetada como 0 entre $\{q_0\}$ y $\{q_0, q_1\}$.

Ejemplo 3 I

Ejemplo 3

Convierta el nfa de la Figura 5 en una máquina determinista equivalente.

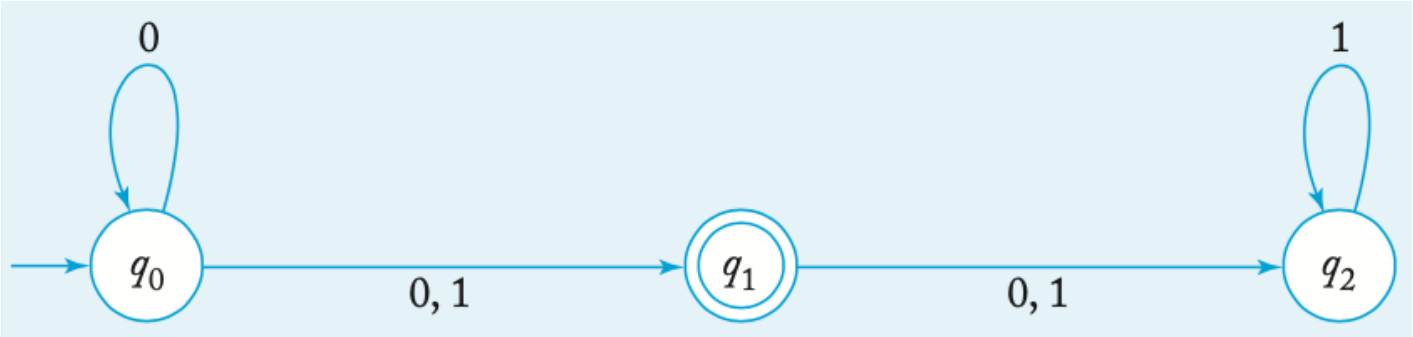

Figura 5: nfa para el Ejemplo 3.

- De la misma manera, considerando $\delta_N(q_0, 1) = \{q_1\}$ nos da el nuevo estado $\{q_1\}$ y una arista etiquetada 1 entre él y $\{q_0\}$.

Ejemplo 3 II

- Ahora hay varias aristas faltantes, por lo que continuamos usando la construcción del Teorema 1.

Ejemplo 3 II

- Al observar el estado $\{q_0, q_1\}$, vemos que no hay una arista saliente etiquetada como 0, por lo que calculamos

$$\delta_N^*(q_0, 0) \cup \delta_N^*(q_1, 0) = \{q_0, q_1, q_2\}.$$

Ejemplo 3 II

- Esto nos da el nuevo estado $\{q_0, q_1, q_2\}$ y la transición $\delta_D(\{q_0, q_1\}, 0) = \{q_0, q_1, q_2\}$.

Ejemplo 3 II

- Entonces, usando $a = 1, i = 0, j = 1, k = 2$,

$$\delta_N^*(q_0, 1) \cup \delta_N^*(q_1, 1) \cup \delta_N^*(q_2, 1) = \{q_1, q_2\}$$

es necesario introducir otro estado más $\{q_1, q_2\}$.

Ejemplo 3 II

- En este punto, tenemos el autómata parcialmente construido que se muestra en la Figura 6.

Ejemplo 3 II

- Dado que todavía faltan algunas aristas, continuamos hasta obtener la solución completa en la Figura 7.

Ejemplo 3 III

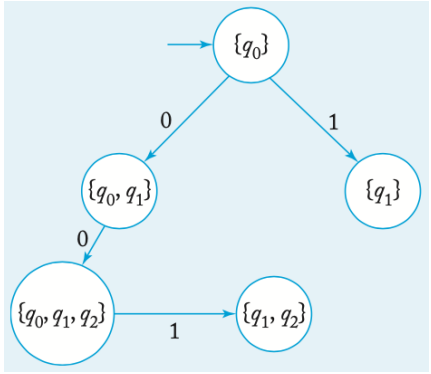

Figura 6: dfa parcial equivalente al nfa de la Figura 5.

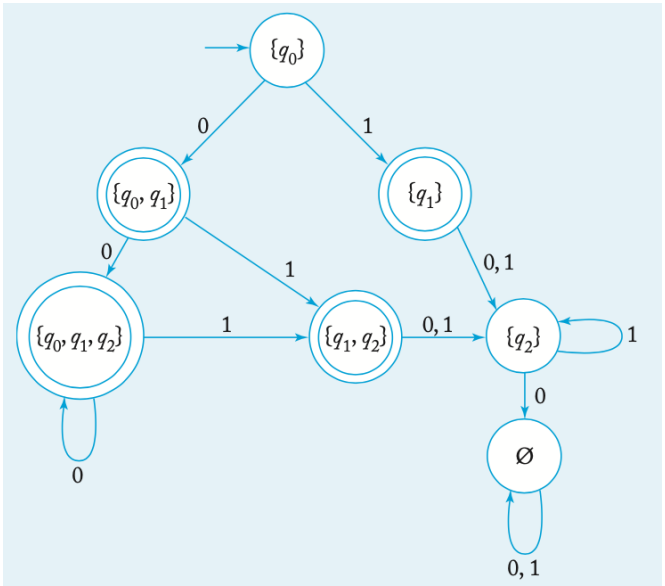

Figura 7: dfa completo equivalente al nfa de la Figura 5.

Ejercicios I

1. Convierta los siguientes nfas a sus equivalentes dfas, donde en cada caso q_0 es el estado inicial y q_2 el estado final.
   1.
   $$\begin{aligned}\delta(q_0, a) &= \{q_0, q_1\}\\ \delta(q_1, b) &= \{q_1, q_2\}\\ \delta(q_2, a) &= \{q_2\}\end{aligned}$$
   2.
   $$\begin{aligned}\delta(q_0, a) &= \{q_0, q_1\}\\ \delta(q_0, \lambda) &= \{q_2\}\\ \delta(q_1, b) &= \{q_1, q_2\}\\ \delta(q_2, a) &= \{q_2\}\end{aligned}$$

Ejercicios II

3.

$$\delta(q_0, a) = \{q_0, q_1\}$$
$$\delta(q_1, b) = \{q_1, q_2\}$$
$$\delta(q_1, \lambda) = \{q_1, q_2\}$$
$$\delta(q_2, a) = \{q_2\}$$

2. ¿Es cierto que para cada nfa $M = (Q, \Sigma, \delta, q_0, F)$, el complemento de $L(M)$ es igual al conjunto $\{w \in \Sigma^* : \delta^*(q_0, w) \cap (Q - F) \neq \emptyset\}$? Si es así, pruébelo; en caso contrario, dé un contraejemplo.
3. Demuestre que todos los lenguajes finitos son regulares.
4. Demuestre que si L es regular, también lo es L^R.

Motivación

- Según nuestra definición, un lenguaje es regular si existe un aceptador finito que lo reconoce.

Motivación

- Por lo tanto, cada lenguaje regular puede ser descrito por algún dfa o algún nfa.

Motivación

- Esta descripción puede resultar muy útil, por ejemplo, si queremos mostrar la lógica mediante la cual decidimos si una cadena determinada está en un lenguaje determinado.

Motivación

- Pero en muchos casos, necesitamos formas más concisas de describir los lenguajes regulares.

Motivación

- En este capítulo, analizamos otras formas de representar lenguajes regulares.

Definición formal de expresiones regulares

- Construimos expresiones regulares a partir de constituyentes primitivos aplicando repetidamente ciertas reglas recursivas.

Definición formal de expresiones regulares

- Esto es similar a la forma en que construimos expresiones aritméticas familiares.

Definición formal de expresiones regulares

Definición 2

Sea Σ un alfabeto. Entonces

Definición formal de expresiones regulares

Definición 2

Sea Σ un alfabeto. Entonces

1. $\emptyset, \lambda$ y a son expresiones regulares. Se les llama **expresiones regulares primitivas**.

Definición formal de expresiones regulares

Definición 2

Sea Σ un alfabeto. Entonces

1. $\emptyset, \lambda$ y a son expresiones regulares. Se les llama **expresiones regulares primitivas**.
2. Si r_1 y r_2 son expresiones regulares, también lo son $r_1 + r_2, r_1 \cdot r_2, r_1^*$ y (r_1).

Definición formal de expresiones regulares

Definición 2

Sea Σ un alfabeto. Entonces

1. $\emptyset, \lambda$ y a son expresiones regulares. Se les llama **expresiones regulares primitivas**.
2. Si r_1 y r_2 son expresiones regulares, también lo son $r_1 + r_2, r_1 \cdot r_2, r_1^*$ y (r_1).
3. Una cadena es una expresión regular si y sólo si puede derivarse de las expresiones regulares primitivas mediante un número finito de aplicaciones de las reglas en 2.

Ejemplo 4

Para $\Sigma = \{a, b, c\}$, la cadena

$$(a + b \cdot c)^* \cdot (c + \emptyset)$$

es una expresión regular, ya que se construye mediante la aplicación de las reglas anteriores.

Ejemplo 4

Para $\Sigma = \{a, b, c\}$, la cadena

$$(a + b \cdot c)^* \cdot (c + \emptyset)$$

es una expresión regular, ya que se construye mediante la aplicación de las reglas anteriores.

- Por ejemplo, si tomamos $r_1 = c$ y $r_2 = \emptyset$, encontramos que $c + \emptyset$ y $(c + \emptyset)$ también son expresiones regulares.

Ejemplo 4

Para $\Sigma = \{a, b, c\}$, la cadena

$$(a + b \cdot c)^* \cdot (c + \emptyset)$$

es una expresión regular, ya que se construye mediante la aplicación de las reglas anteriores.

- Repitiendo esto, eventualmente generamos la cadena completa.

Ejemplo 4

Para $\Sigma = \{a, b, c\}$, la cadena

$$(a + b \cdot c)^* \cdot (c + \emptyset)$$

es una expresión regular, ya que se construye mediante la aplicación de las reglas anteriores.

- Por otro lado, $(a + b+)$ no es una expresión regular, ya que no hay forma de que se pueda construir a partir de las expresiones regulares primitivas.

□

Definición 3

El lenguaje $L(r)$ denotado por cualquier expresión regular r está definido por las siguientes reglas, donde r_1 y r_2 son expresiones regulares.

Definición 3

El lenguaje $L(r)$ denotado por cualquier expresión regular r está definido por las siguientes reglas, donde r_1 y r_2 son expresiones regulares.

1. $\emptyset$ es una expresión regular que denota el conjunto vacío,

Definición 3

El lenguaje $L(r)$ denotado por cualquier expresión regular r está definido por las siguientes reglas, donde r_1 y r_2 son expresiones regulares.

1. $\emptyset$ es una expresión regular que denota el conjunto vacío,
2. λ es una expresión regular que denota $\{\lambda\}$,

Definición 3

El lenguaje $L(r)$ denotado por cualquier expresión regular r está definido por las siguientes reglas, donde r_1 y r_2 son expresiones regulares.

1. $\emptyset$ es una expresión regular que denota el conjunto vacío,
2. λ es una expresión regular que denota $\{\lambda\}$,
3. Para cada $a \in \Sigma$, a es una expresión regular que denota $\{a\}$.

Definición 3

El lenguaje $L(r)$ denotado por cualquier expresión regular r está definido por las siguientes reglas, donde r_1 y r_2 son expresiones regulares.

1. $\emptyset$ es una expresión regular que denota el conjunto vacío,
2. λ es una expresión regular que denota $\{\lambda\}$,
3. Para cada $a \in \Sigma$, a es una expresión regular que denota $\{a\}$.
4. $L(r_1 + r_2) = L(r_1) \cup L(r_2)$,

Definición 3

El lenguaje $L(r)$ denotado por cualquier expresión regular r está definido por las siguientes reglas, donde r_1 y r_2 son expresiones regulares.

1. $\emptyset$ es una expresión regular que denota el conjunto vacío,
2. λ es una expresión regular que denota $\{\lambda\}$,
3. Para cada $a \in \Sigma$, a es una expresión regular que denota $\{a\}$.
4. $L(r_1 + r_2) = L(r_1) \cup L(r_2)$,
5. $L(r_1 \cdot r_2) = L(r_1)L(r_2)$,

Definición 3

El lenguaje $L(r)$ denotado por cualquier expresión regular r está definido por las siguientes reglas, donde r_1 y r_2 son expresiones regulares.

1. $\emptyset$ es una expresión regular que denota el conjunto vacío,
2. λ es una expresión regular que denota $\{\lambda\}$,
3. Para cada $a \in \Sigma$, a es una expresión regular que denota $\{a\}$.
4. $L(r_1 + r_2) = L(r_1) \cup L(r_2)$,
5. $L(r_1 \cdot r_2) = L(r_1)L(r_2)$,
6. $L((r_1)) = L(r_1)$,

Definición 3

El lenguaje $L(r)$ denotado por cualquier expresión regular r está definido por las siguientes reglas, donde r_1 y r_2 son expresiones regulares.

1. $\emptyset$ es una expresión regular que denota el conjunto vacío,
2. λ es una expresión regular que denota $\{\lambda\}$,
3. Para cada $a \in \Sigma$, a es una expresión regular que denota $\{a\}$.
4. $L(r_1 + r_2) = L(r_1) \cup L(r_2)$,
5. $L(r_1 \cdot r_2) = L(r_1)L(r_2)$,
6. $L((r_1)) = L(r_1)$,
7. $L(r_1^*) = (L(r_1))^*$.

Ejemplo 5

Ejemplo 5

Exhiba el lenguaje $L(a^* \cdot (a + b))$ en notación de conjuntos.

Ejemplo 5

Exhiba el lenguaje $L(a^* \cdot (a + b))$ en notación de conjuntos.

$$L(a^* \cdot (a + b)) =$$

Ejemplo 5

Exhiba el lenguaje $L(a^* \cdot (a+b))$ en notación de conjuntos.

$$L(a^* \cdot (a+b)) = L(a^*)L(a+b)$$

Ejemplo 5

Exhiba el lenguaje $L(a^* \cdot (a+b))$ en notación de conjuntos.

$$\begin{aligned} L(a^* \cdot (a+b)) &= L(a^*)L(a+b) \\ &= (L(a))^*(L(a) \cup L(b)) \end{aligned}$$

Ejemplo 5

Exhiba el lenguaje $L(a^* \cdot (a+b))$ en notación de conjuntos.

$$\begin{aligned} L(a^* \cdot (a+b)) &= L(a^*)L(a+b) \\ &= (L(a))^*(L(a) \cup L(b)) \\ &= \{\lambda, a, aa, aaa, \ldots\}\{a, b\} \end{aligned}$$

Ejemplo 5

Exhiba el lenguaje $L(a^* \cdot (a+b))$ en notación de conjuntos.

$$\begin{aligned} L(a^* \cdot (a+b)) &= L(a^*)L(a+b) \\ &= (L(a))^*(L(a) \cup L(b)) \\ &= \{\lambda, a, aa, aaa, \ldots\}\{a, b\} \\ &= \{a, aa, aaa, \ldots, b, ab, aab, aaab, \ldots\}. \end{aligned}$$

□

Reglas de precedencia I

- Hay un problema con las reglas 4 a 7 en la Definición 3.

Reglas de precedencia I

- Definen un lenguaje precisamente si se dan r_1 y r_2, pero puede haber alguna ambigüedad al dividir una expresión complicada en partes.

Reglas de precedencia I

- Considere, por ejemplo, la expresión regular $a \cdot b + c$.

Reglas de precedencia I

- Considere, por ejemplo, la expresión regular $a \cdot b + c$.
- Podemos considerar que está formado por $r_1 = a \cdot b$ y $r_2 = c$.

Reglas de precedencia I

- Considere, por ejemplo, la expresión regular $a \cdot b + c$.
- Podemos considerar que está formado por $r_1 = a \cdot b$ y $r_2 = c$.
- En este caso, encontramos $L(a \cdot b + c) = \{ab, c\}$.

Reglas de precedencia I

- Considere, por ejemplo, la expresión regular $a \cdot b + c$.
- Podemos considerar que está formado por $r_1 = a \cdot b$ y $r_2 = c$.
- En este caso, encontramos $L(a \cdot b + c) = \{ab, c\}$.
- Pero no hay nada en la Definición 3 que nos impida tomar $r_1 = a$ y $r_2 = b + c$.

Reglas de precedencia I

- Considere, por ejemplo, la expresión regular $a \cdot b + c$.
- Podemos considerar que está formado por $r_1 = a \cdot b$ y $r_2 = c$.
- En este caso, encontramos $L(a \cdot b + c) = \{ab, c\}$.
- Pero no hay nada en la Definición 3 que nos impida tomar $r_1 = a$ y $r_2 = b + c$.
- Ahora obtenemos un resultado diferente, $L(a \cdot b + c) = \{ab, ac\}$.

Reglas de precedencia I

- Para superar esto, podríamos requerir que todas las expresiones estén entre paréntesis, pero esto da resultados engorrosos.

Reglas de precedencia II

- En su lugar, usamos una convención familiar de las matemáticas y los lenguajes de programación.

Reglas de precedencia II

- Establecemos un conjunto de reglas de precedencia para la evaluación en las que la cerradura estrella precede a la concatenación y la concatenación precede a la unión.

Reglas de precedencia II

- Además, el símbolo para la concatenación puede omitirse, por lo que podemos escribir $r_1 r_2$ en lugar de $r_1 \cdot r_2$.

Reglas de precedencia II

- Con un poco de práctica, podemos ver rápidamente qué lenguaje denota una expresión regular en particular.

Ejemplo 6

Para $\Sigma = \{a, b\}$, la expresión

$$r = (a + b)^*(a + bb)$$

es regular. Denota el lenguaje

$$L(r) = \{a, bb, aa, abb, ba, bbb, \ldots\}.$$

Ejemplo 6

Para $\Sigma = \{a, b\}$, la expresión

$$r = (a + b)^*(a + bb)$$

es regular. Denota el lenguaje

$$L(r) = \{a, bb, aa, abb, ba, bbb, \ldots\}.$$

- Podemos ver esto considerando las diversas partes de r.

Ejemplo 6

Para $\Sigma = \{a, b\}$, la expresión

$$r = (a + b)^*(a + bb)$$

es regular. Denota el lenguaje

$$L(r) = \{a, bb, aa, abb, ba, bbb, \ldots\}.$$

- La primera parte, $(a + b)^*$, representa cualquier cadena de as y bs.

Ejemplo 6

Para $\Sigma = \{a, b\}$, la expresión

$$r = (a + b)^*(a + bb)$$

es regular. Denota el lenguaje

$$L(r) = \{a, bb, aa, abb, ba, bbb, \ldots\}.$$

- La primera parte, $(a + b)^*$, representa cualquier cadena de as y bs.
- La segunda parte, $(a + bb)$ representa una a o una doble b.

Ejemplo 6

Para $\Sigma = \{a, b\}$, la expresión

$$r = (a + b)^*(a + bb)$$

es regular. Denota el lenguaje

$$L(r) = \{a, bb, aa, abb, ba, bbb, \ldots\}.$$

- La primera parte, $(a + b)^*$, representa cualquier cadena de as y bs.
- La segunda parte, $(a + bb)$ representa una a o una doble b.
- En consecuencia, $L(r)$ es el conjunto de todas las cadenas en $\{a, b\}$, terminadas por una a o una bb.

□

Ejemplo 7

La expresión

$$r = (aa)^*(bb)^*b$$

denota el conjunto de todas las cadenas con un número par de as seguida por un número impar de bs; esto es,

$$L(r) = \{a^{2n}b^{2m+1} : n \geq 0, m \geq 0\}.$$

□

Ejemplo 8

Para $\Sigma = \{0, 1\}$, dé una expresión regular r tal que

$$L(r) = \{w \in \Sigma^* : w \text{ tiene al menos un par de ceros consecutivos}\}.$$

Ejemplo 8

Para $\Sigma = \{0, 1\}$, dé una expresión regular r tal que

$$L(r) = \{w \in \Sigma^* : w \text{ tiene al menos un par de ceros consecutivos}\}.$$

- Se puede llegar a una respuesta razonando algo como esto: cada cadena en $L(r)$ debe contener 00 en algún lugar, pero lo que viene antes y lo que sigue es completamente arbitrario.

Ejemplo 8

Para $\Sigma = \{0, 1\}$, dé una expresión regular r tal que

$$L(r) = \{w \in \Sigma^* : w \text{ tiene al menos un par de ceros consecutivos}\}.$$

- Se puede llegar a una respuesta razonando algo como esto: cada cadena en $L(r)$ debe contener 00 en algún lugar, pero lo que viene antes y lo que sigue es completamente arbitrario.
- Una cadena arbitraria en $\{0, 1\}$ se puede denotar por $(0 + 1)^*$.

Ejemplo 8

Para $\Sigma = \{0, 1\}$, dé una expresión regular r tal que

$$L(r) = \{w \in \Sigma^* : w \text{ tiene al menos un par de ceros consecutivos}\}.$$

- Se puede llegar a una respuesta razonando algo como esto: cada cadena en $L(r)$ debe contener 00 en algún lugar, pero lo que viene antes y lo que sigue es completamente arbitrario.
- Una cadena arbitraria en $\{0, 1\}$ se puede denotar por $(0 + 1)^*$.
- Juntando estas observaciones, llegamos a la solución

$$r = (0 + 1)^* 00 (0 + 1)^*.$$

□

Ejemplo 9 I

Ejemplo 9

Encuentre una expresión regular para el lenguaje

$$L = \{w \in \{0,1\}^* : w \text{ no tiene un par de ceros consecutivos}\}.$$

Ejemplo 9 I

Ejemplo 9

Encuentre una expresión regular para el lenguaje

$$L = \{w \in \{0,1\}^* : w \text{ no tiene un par de ceros consecutivos}\}.$$

- Aunque parece similar al ejemplo 8, la respuesta es más difícil de construir.

Ejemplo 9 I

Ejemplo 9

Encuentre una expresión regular para el lenguaje

$$L = \{w \in \{0,1\}^* : w \text{ no tiene un par de ceros consecutivos}\}.$$

- Una observación útil es que siempre que aparece un 0, debe ir seguido inmediatamente por un 1.

Ejemplo 9 I

Ejemplo 9

Encuentre una expresión regular para el lenguaje

$$L = \{w \in \{0,1\}^* : w \text{ no tiene un par de ceros consecutivos}\}.$$

- Dicha subcadena puede ir precedida y seguida de un número arbitrario de unos.

Ejemplo 9 I

Ejemplo 9

Encuentre una expresión regular para el lenguaje

$$L = \{w \in \{0,1\}^* : w \text{ no tiene un par de ceros consecutivos}\}.$$

- Esto sugiere que la respuesta implica la repetición de cadenas de la forma $1 \cdots 101 \cdots 1$, es decir, el lenguaje denotado por la expresión regular $(1^*011^*)^*$.

Ejemplo 9 II

- Sin embargo, la respuesta sigue siendo incompleta, ya que no se tienen en cuenta las cadenas que terminan en 0 o que constan de 1s.

Ejemplo 9 II

- Después de atender estos casos especiales llegamos a la respuesta

$$r = (1^*011^*)^*(0 + \lambda) + 1^*(0 + \lambda).$$

Ejemplo 9 II

- Si razonamos de manera ligeramente diferente, podríamos encontrar otra respuesta.

Ejemplo 9 II

- Después de atender estos casos especiales llegamos a la respuesta

$$r = (1^*011^*)^*(0 + \lambda) + 1^*(0 + \lambda).$$

- Si vemos L como la repetición de las cadenas 1 y 01, la expresión más corta

$$r = (1 + 01)^*(0 + \lambda)$$

podría ser propuesta.

Ejemplo 9 III

- Aunque las dos expresiones se ven diferentes, ambas respuestas son correctas, ya que denotan el mismo lenguaje.

Ejemplo 9 III

- Generalmente, hay un número ilimitado de expresiones regulares para cualquier lenguaje.

Ejemplo 9 III

- Tenga en cuenta que este lenguaje es el complemento del lenguaje del Ejemplo 8.

Ejemplo 9 III

- Sin embargo, las expresiones regulares no son muy similares y no sugieren claramente la estrecha relación entre los dos lenguajes.

□

Ejercicios

1. Encuentre una expresión regular para el conjunto

$$\{a^n b^m : n \geq 3, m \text{ es impar}\}.$$

2. Encuentre una expresión regular para el conjunto

$$\{a^n b^m : (n+m) \text{ es impar}\}.$$

3. Dé expresiones regulares para los siguientes lenguajes.
   1. $L_1 = \{a^n b^m : n \geq 3, m \leq 4\}$.
   2. $L_2 = \{a^n b^m : n < 4, m \leq 4\}$.
   3. El complemento de L_1.
   4. El complemento de L_2.
4. Encuentre una expresión regular para

$$L = \{ab^n w : n \geq 4, w \in \{a,b\}^+\}.$$

5. Encuentre una expresión regular para

$$L = \{vwv : v, w \in \{a,b\}^*, |v| = 2\}.$$