Descripción básica

En la primera parte de la clase de hoy, mostraremos un método para simplificar los autómatas deterministas. Este método es importante, porque los autómatas deterministas representan programas, y los programas asociados a autómatas pequeños son más eficientes que los programas asociados a los autómatas grandes.

A continuación, definiremos los autómatas indeterministas, que son modelos computacionales equivalentes a los autómatas deterministas, pero más fáciles de deseñar.

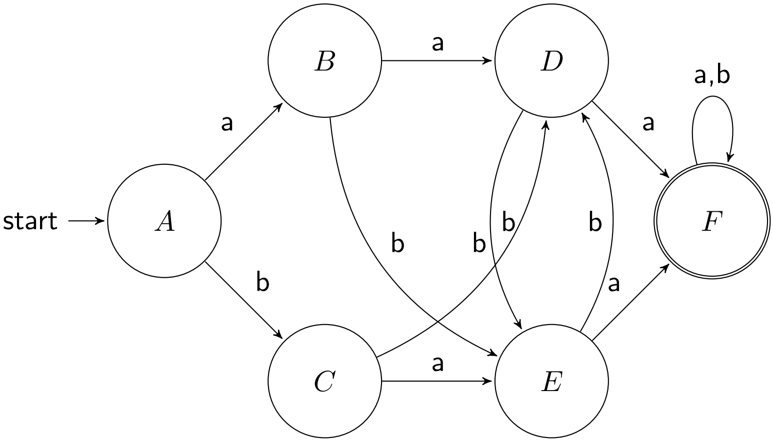
Finalmente, introduciremos un algoritmo que nos permite transformar un autómata indeterminista en un autómata determinista equivalente.

Para simplificar un autómata determinista, la idea es eliminar un estado del autómata determinista que estemos considerando cuando haya otro estado que haga la misma función. Y, dicho de manera intuitiva, dos estados hacen la misma función, si para cualquier entrada que consideremos, se obtiene el mismo resultado tanto si se parte de un estado como del otro.

Formalmente, la definición es la siguiente. Sea $M = (K, \Sigma, \delta, q_0, F)$ un autómata determinista. Sean $p, q \in K$. Decimos que p, q son equivalentes (o indistingibles), si para todo $x \in \Sigma^*$, si $px \vdash_M^* p'$ y $qx \vdash_M^* q'$ entonces $p', q' \in F$ o $p', q' \notin F$.

Ejemplo

Consideremos el siguiente autómata determinista M:



Ejemplo

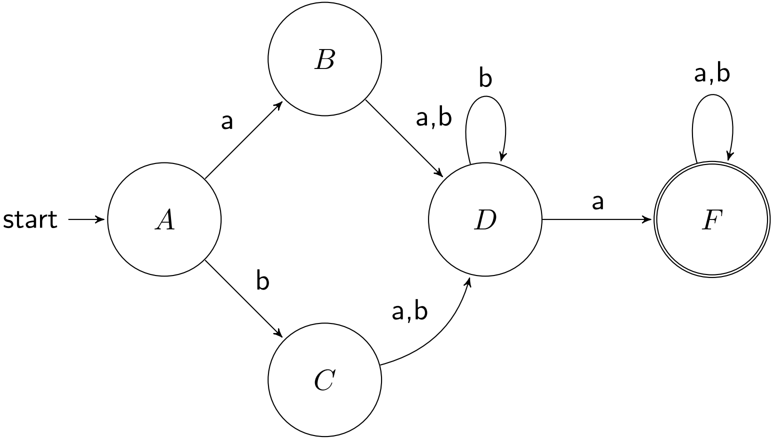
Tenemos que los estados D y E son equivalentes, ya que para toda palabra $x \in \{a, b\}^*$ se tiene que:

$$Dx \vdash_M^* F \Longleftrightarrow x \in L(b^*a(a \cup b)^*),$$

$$Ex \vdash_M^* F \Longleftrightarrow x \in L(b^*a(a \cup b)^*).$$

Por tanto, podemos juntar los estados D y E, obteniendo el siguiente autómata M' equivalente a M:

Ejemplo



Ejemplo

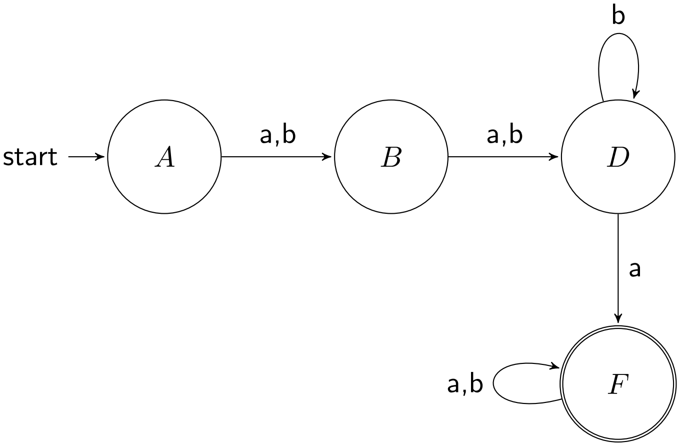
Ahora, observamos que los estados B y C son equivalentes, ya que para toda palabra $x \in \{a, b\}^*$ se tiene que:

$$Bx \vdash^*_{M'} F \Longleftrightarrow x \in L((a \cup b)b^*a(a \cup b)^*),$$

$$Cx \vdash^*_{M'} F \Longleftrightarrow x \in L((a \cup b)b^*a(a \cup b)^*).$$

Por tanto, obtenemos el siguiente autómata M'' equivalente a M:

Ejemplo



Se observa entonces que este autómata M'' ya no se puede simplificar.

Autómatas indeterministas

Los autómatas indeterministas son modelos computacionales equivalentes a los autómatas deterministas, pero más fáciles de construir, porque tienen generalmente menos estados y menos transiciones. La construcción de autómatas deterministas para diseñar programas se va haciendo más difícil a medida que crece el número de estados involucrados. Entonces, el modelo de autómata indeterminista nos ayuda a resolver este problema. Además, siempre se puede recurrir a autómatas indeterministas, ya que veremos que existe un algoritmo que nos permite transformar un autómata indeterminista en un autómata determinista equivalente. Entonces, cuando resulta complicado construir un autómata determinista para el diseño de un programa, podemos construir en primer lugar un autómata indeterminista, y a partir de él, mediante el algoritmo que veremos, construir el autómata determinista equivalente. El programa asociado a dicho autómata determinista será entonces el programa que buscamos.

Definición de autómata indeterminista

Un autómata indeterminista es una estructura $M = (K, \Sigma, \Delta, q_0, F)$ donde K, Σ, q_0, F son como en la definición de autómata determinista y Δ es un subconjunto de $K \times (\Sigma \cup \{\lambda\}) \times K$.

La diferencia entre los autómatas deterministas y los indeterministas radica en que en el caso de los autómatas deterministas se ha de realizar exactamente una transición desde un estado para cada símbolo de la entrada, mientras que en el caso de los autómatas indeterministas se permite que desde un estado se realicen cero, una o más transiciones para cada símbolo de la entrada, y se permite además que se tengan transiciones con la palabra vacía, es decir, se permite pasar de un estado a otro sin leer ningún símbolo de la entrada.

Por tanto, en un autómata determinista el cómputo es único para cualquier entrada que sumistremos al autómata, mientras que un autómata indeterminista habrá en general varios cómputos para una entrada que le suministremos.

Noción de cómputo en un autómata indeterminista

Supongamos que $M = (K, \Sigma, \Delta, q_0, F)$ es un autómata indeterminista. A los elementos de Δ los llamaremos transiciones. Al igual que en el caso determinista, los cómputos de M se realizan aplicando transiciones de Δ.

Definimos una configuración de M como una palabra $px \in K\Sigma^*$. Si en un paso de cómputo del autómata nos encontramos en la configuración px, eso significa que el autómata se encuentra en el estado p y la x es la parte de la palabra de entrada aún no leída.

Si px, qy son configuraciones de M, decimos que px produce qy en un paso de cómputo , lo que representamos por $px \vdash_M qy$, si en M podemos pasar de px a qy aplicando una transición de Δ.

Si px, qy son configuraciones de M, decimos que px produce qy, lo que representamos por $px \vdash_M^* qy$, si en M podemos pasar de px a qy aplicando un número finito de transiciones de Δ.

Noción de lenguaje asociado a un autómata indeterminista

Supongamos que $M = (K, \Sigma, \Delta, q_0, F)$ es un autómata indeterminista. Decimos que una palabra $x \in \Sigma^*$ es reconocida (o aceptada) por M, si existe $q \in F$ tal que $q_0 x \vdash_M^* q$.

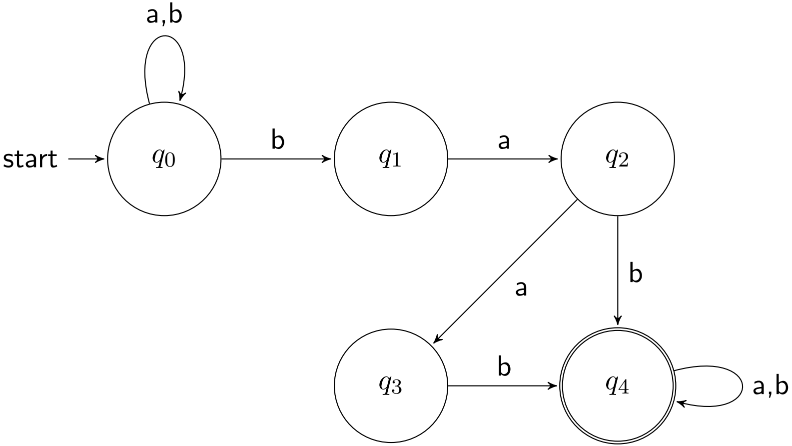
Por tanto, para que una palabra $x \in \Sigma^*$ sea reconocida por M tiene que existir un cómputo en el autómata que lea toda la palabra x y termine en un estado aceptador.

Definimos entonces el lenguaje reconocido (o aceptado) por M por

$$L(M) = \{x \in \Sigma^* : x \text{ es reconocida por } M\}.$$

Ejemplo 1

M:



Ejemplo 1

Consideremos la entrada $x = abbbabaabba$ para el autómata M. Vemos que hay varios cómputos de M para x. Por ejemplo, los dos siguientes cómputos reconocen la palabra x:

(1) $q_0abbabaabba \vdash_M q_0bbabaabba \vdash_M q_0babaabba \vdash_M q_1abaabba \vdash_M q_2baabba \vdash_M q_4aabba \vdash_M^* q_4\lambda$.

(2) $q_0abbabaabba \vdash_M q_0bbabaabba \vdash_M q_0babaabba \vdash_M q_0abaabba \vdash_M q_0baabba \vdash_M q_1aabba \vdash_M q_2abba \vdash_M q_3bba \vdash_M q_4ba \vdash_M q_4a \vdash_M q_4\lambda$.

Y hay también cómputos que no reconocen la palabra x. Por ejemplo, el cómputo que va leyendo todos los símbolos de x permaneciendo siempre en el estado q_0.

Normalmente, en un autómata indeterminista habrá computos que reconozcan una entrada y cómputos que no la reconozcan. Pero para que una palabra esté en el lenguaje de un autómata indeterminista, basta con que haya un cómputo que reconozca la palabra.

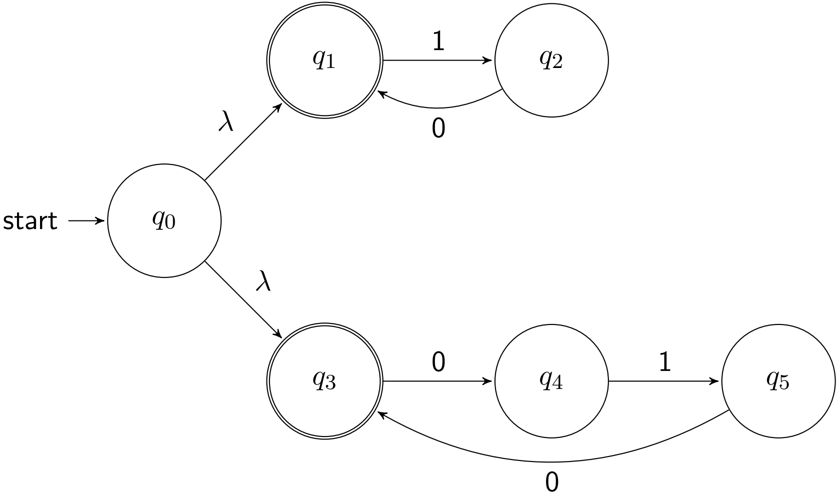
Ejemplo 1

En este autómata M, vemos que para que una palabra x sea reconocida, tiene quen haber un cómputo para esa palabra que parta del estado inicial q_0 y termine en el estado aceptador q_4. Y eso sucederá cuando o bien la palabra x contenga bab o bien contenga baab. Por tanto,

$L(M) = \{x \in \{a, b\}^*$: en x aparece la palabra bab o la palabra baab$\}$.

Ejemplo 2

M:



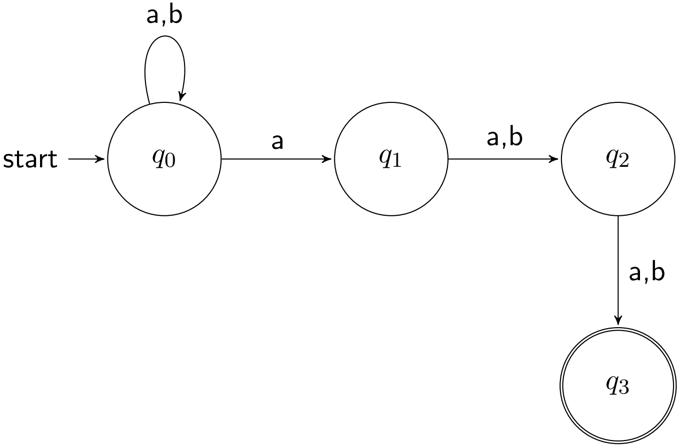
Ejemplo 2

Los estados aceptadores de M' son los estados q_1 y q_3. Si un cómputo del autómata termina en el estado q_1, se habrá reconocido una palabra de la forma $(10)^n$ donde $n \geq 0$. Y si un cómputo del autómata termina en el estado q_3, se habrá reconocido una palabra de la forma $(010)^n$ donde $n \geq 0$. Por tanto,

$L(M) = L(\alpha)$ donde α es la expresión regular $(10)^* \cup (010)^*$.

Ejemplo 3

M:



Se tiene que $L(M)$ es el lenguaje de las palabras $x \in \{a, b\}^*$ tales que a es el antepenúltimo símbolo de x.

Equivalencia entre autómatas deterministas e indeterministas

Dos autómatas M y M' son equivalentes, si reconocen el mismo lenguaje, es decir, si $L(M) = L(M')$.

La equivalencia entre los autómatas deterministas y los autómatas indeterministas viene dada por el siguiente teorema.

Teorema

Para todo lenguaje L, existe un autómata determinista M tal que $L = L(M)$ si y sólo si existe un autómata indeterminista M tal que $L(M) = L$.

Por tanto, para todo autómata indeterminista M existe un autómata determinista M' que es equivalente a M.

Equivalencia entre autómatas deterministas e indeterministas

El siguiente concepto será utilizado en el algoritmo para transformar un autómata indeterminista en un autómata determinista equivalente.

Si $M = (K, \Sigma, \Delta, q_0, F)$ es un autómata indeterminista y $p \in K$ definimos el λ-cierre de p por

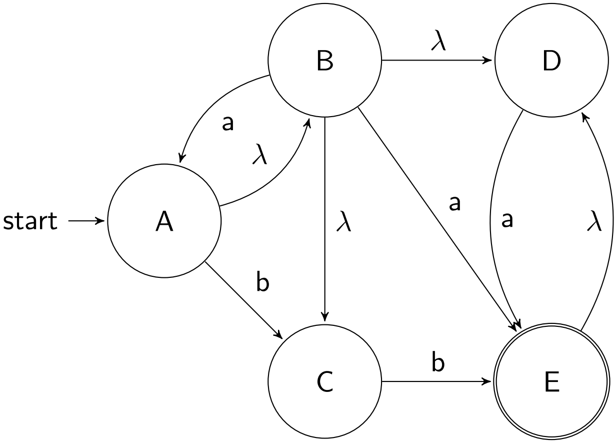
$$\Lambda(p) = \{q \in K : p\lambda \vdash_M^* q\}.$$

Por tanto, $\Lambda(p)$ es el conjunto de aquellos estados a los que podemos llegar desde el estado p sin leer ningún símbolo de la entrada.

Obsérvese que todo estado pertenece a su λ-cierre, es decir, para todo estado p se tiene que $p \in \Lambda(p)$.

Ejemplo

Consideremos el siguiente autómata indeterminista:



Ejemplo

Para simplificar la notación, representamos a un conjunto de estados $\{X_1, \ldots X_n\}$ por la secuencia $X_1 \ldots X_n$.

Tenemos entonces:

$\Lambda(A) = \{A, B, C, D\} = ABCD$,

$\Lambda(B) = BCD$,

$\Lambda(C) = C$,

$\Lambda(D) = D$,

$\Lambda(E) = DE$.

Algoritmo para transformar un autómata indeterminista en un autómata determinista equivalente

Sea $M = (K, \Sigma, \Delta, q_0, F)$ un autómata indeterminista. Definimos el autómata determinista $M' = (K', \Sigma, \delta', q_0', F')$ de la siguiente manera:

(1) $K' = P(K) =$ conjunto de subconjuntos de K,

(2) $q_0' = \Lambda(q_0)$,

(3) $F' = \{X \in K' : X \cap F \neq \emptyset\}$,

(4) Si $X \in K'$ y $a \in \Sigma$, definimos

$$\delta'(X, a) = \bigcup \{\Lambda(q) : \text{ existe un estado } p \in X \text{ tal que } (p, a, q) \in \Delta\}.$$

Por tanto, para calcular $\delta'(X, a)$ se ha de hacer lo siguiente:

Algoritmo para transformar un autómata indeterminista en un autómata determinista equivalente

(1) Obtener todos los estados $q \in K$ para los cuales existe un estado $p \in X$ de manera que $(p, a, q) \in \Delta$.

(2) Computar $\Lambda(q)$ para todo estado q obtenido en la etapa (1).

(3) Tomar la unión de los conjuntos $\Lambda(q)$ computados en (2).

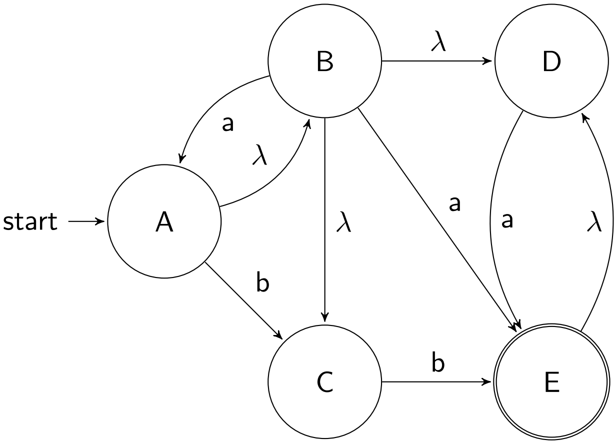
Se puede demostrar entonces que $L(M) = L(M')$, es decir, M y M' son equivalentes.

Algoritmo para transformar un autómata indeterminista en un autómata determinista equivalente

Por tanto, para obtener un autómata determinista equivalente a M, consideramos en primer lugar el estado inicial $q_0' = \Lambda(q_0)$, y computamos $\delta'(q_0', a)$ para cada símbolo a de Σ. A continuación, computamos de nuevo la función δ' para los estados nuevos que hayan salido. Y así continuamos, hasta que no salgan estados nuevos. Marcaremos entonces como estados aceptadores del autómata determinista M' aquellos estados que contengan algún estado aceptador del autómata indeterminista M.

Ejemplo

Consideremos el siguiente autómata indeterminista visto anteriormente:



Ejemplo

Recordemos que $\Lambda(A) = ABCD$, $\Lambda(B) = BCD$, $\Lambda(C) = C$, $\Lambda(D) = D$ y $\Lambda(E) = DE$.

El estado inicial del auómata determinista M es $\Lambda(A) == ABCD$. Tenemos entonces:

$\delta'(ABCD, a) = \Lambda(A) \cup \Lambda(E) = ABCDE$,

$\delta'(ABCD, b) = \Lambda(C) \cup \Lambda(E) = CDE$,

$\delta'(ABCDE, a) = \Lambda(A) \cup \Lambda(E) = ABCDE$,

$\delta'(ABCDE, b) = \Lambda(C) \cup \Lambda(E) = CDE$,

$\delta'(CDE, a) = \Lambda(E) = DE$,

$\delta'(CDE, b) = \Lambda(E) = DE$,

$\delta'(DE, a) = \Lambda(E) = DE$,

$\delta'(DE, b) = \emptyset$,

$\delta'(\emptyset, a) = \delta'(\emptyset, b) = \emptyset$.

Ejemplo

Por tanto, los estados de M' son
$q_0 = ABCD, q_1 = ABCDE, q_2 = CDE, q_3 = DE$ y $q_4 = \emptyset$. Y como E es el único estado aceptador de M, los estados aceptadores de M' son q_1, q_2 y q_3. El grafo del autómata determinista obtenido es entonces el siguiente: