TECNOLÓGICO NACIONAL DE MEXICO

INSTITUTO TECNOLÓGICO DE IZTAPALAPA

INTEGRANTE:

PEREZ ARMAS FAUSTO ISAAC

181080037

ISC-6AM

LENGUAJES Y AUTOMATAS I

M.C. ABIEL TOMÁS PARRA HERNÁNDEZ

SEP 2020 / FEB 2021

ACTIVIDAD SEMANA 7

NFA

Un NFA puede estar en varios estados a la vez o se puede ver que “adivina” a qué estado ir.

NFA Formal

Formalmente, un NFA es una quíntupla *A*=(*Q* ,*S* ,*d* , *q0*,*F*), donde todo es un **DFA**,

pero

· *d* (*q,a*) es un conjunto de estados en lugar de un solo estado. De hecho puede ser vacío, tener un solo estado o tener más estados.

Un NFA acepta, al igual que un DFA, lenguajes regulares

Por ejemplo, para el NFA que acepta cadenas que acaban en 01 su función de

	0	1
$\rightarrow q_0$	$\{q_0, q_1\}$	$\{q_0\}$
q_1	$\emptyset$	$\{q_2\}$
$\star q_2$	$\emptyset$	$\emptyset$

Como puede observarse, todo se especifica en conjuntos.

Lenguajes de un NFA

El lenguaje aceptado por un NFA, A, es: L(A)={w: (q0,w)∩F≠Ø}

Equivalencia entre un DFA y un NFA

Un NFA es normalmente más fácil de definir, aunque al mismo tiempo, para cualquier NFA N existe un DFA D tal que L(D)=L(N) y viceversa.

Para esto se usa la construcción de subconjunto que muestra un ejemplo de cómo un autómata se puede construir a partir de otro.

Construcción de Subconjunto

1. Para cada **NFA** existe un **DFA** equivalente (acepta el mismo lenguaje).
2. Pero el **DFA** puede tener un número exponencial de estados.

Sea *N* = (*QN*, *SN*, *dN*, *q0*, *FN*) un **NFA**. El DFA equivalente construido a partir del subconjunto de construcción es |QN| *D* = (Q^D, *S*, *d* ^{D}, {*q* ^{0}}, *F* ^{D}}, donde:

1. |*Q D* |= 2 ; i.e., *Q D* es el conjunto de todos los subconjuntos de *QN* .
2. *FD* es el conjunto de conjuntos *S* en *QD* tal que *S* Ç *FN* ¹ Æ.
3. Para cualquier y ,

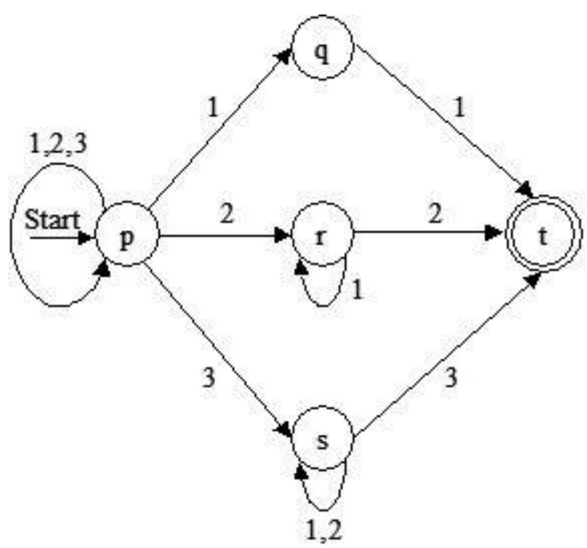
O sea, la unión de todos los estados a partir de con entrada *dD*({*q1*,*q* *2*,...,*qk*},*a*) = *dN*(*p1*,*a*) È *dN*(*p2*,*a*) È ... È *dN*(*pk*,*a*).

La función de transición *dD* del NFA anterior es:

	0	1
$\emptyset$	$\emptyset$	$\emptyset$
$\rightarrow \{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_1\}$	$\emptyset$	$\{q_2\}$
$\star\{q_2\}$	$\emptyset$	$\emptyset$
$\{q_0, q_1\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$
$\star\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0\}$
$\star\{q_1, q_2\}$	$\emptyset$	$\{q_2\}$
$\star\{q_0, q_1, q_2\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$

Al existir 3 estados, tenemos 8 subconjuntos. Esto mismo lo podemos poner como:

	0	1
A	A	A
$\rightarrow B$	E	B
C	A	D
$\star D$	A	A
E	E	F
$\star F$	E	B
$\star G$	A	D
$\star H$	E	F



Ejemplo: Subconjunto de Construcción del NFA Previo

Un truco práctico importante utilizado por analizadores léxicos y otros procesadores de texto es ignorar los (frecuentemente muchos) estados que no son accesibles desde el estado de inicio (i.e., no hay ruta que lleve a ellos). Para el ejemplo anterior de NFA, de los 32 subconjuntos posibles, solo 15 son

accesibles. Calculando las transiciones "por demanda" obtenemos el siguiente *d D*:

DFA

Con la propiedad de no-determinismo se agrega eficiencia al describir una aplicación o Permite programar soluciones en un lenguaje de más alto nivel o Hay un algoritmo para compilar un N-DFA en un DFA y poder ser ejecutado en una computadora convencional

Extensión del N-DFA para hacer saltos de un estado a otro

espontáneamente o Con la cadena vacía como entrada

o ÎN-DFA

o Estos autómatas también aceptan lenguajes regulares

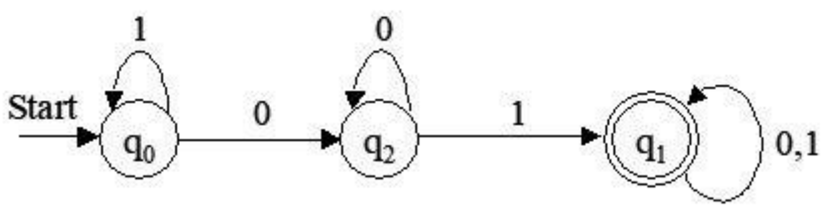
Ejemplo: Un Autómata *A* que acepta *L* = {*x* 01*y* : *x* , *y* Î {0,1}*}

- El DFA acepta cadenas que tienen 01 en alguna parte de la cadena
- El lenguaje del DFA es el conjunto de cadenas que acepta
- {w **|** w tiene la forma *x* 01*y* para algunas cadenas *x* y *y* que consisten sólo de 0's y 1's}

El Autómata *A* = ({*q* 0, *q* 1, *q* 2}, {0,1}, d, *q* 0, {*q* 1})

Autómata representado con una tabla de transiciones:

Autómata representado con un diagrama de transiciones:



Teorema de Myhill-Nerode

En informática teórica, y en particular en teoría de lenguajes formales y autómatas, el teorema de Myhill-Nerode da una condición necesaria y suficiente para que un lenguaje formal sea un lenguaje racional, es decir, reconocible por un autómata finito. . Este teorema lleva el nombre de John Myhill y Anil Nerode, quienes lo demostraron en 1958 (Nerode 1958).

El interés de esta afirmación es que la condición necesaria y suficiente se relaciona con el lenguaje mismo y no requiere la construcción de un autómata. La prueba, por otra parte, es constructiva y permite obtener de forma eficaz un autómata que además resulta mínimo.

Si L es un lenguaje regular, entonces, por definición, hay un DFA A que lo reconoce, con solo un número finito de estados. Si hay n estados, entonces particione el conjunto de todas las cadenas finitas en n subconjuntos, donde el subconjunto S i es el conjunto de cadenas que, cuando se dan como entrada al autómata A , hacen que termine en el estado i . Por cada dos cadenas X y Y que pertenecen al mismo subconjunto, y para cada opción de una tercera cadena z , autómata A alcanza el mismo estado en la entrada XZ , ya que llega en la entrada yz , y por lo tanto debe aceptar ya sea tanto de las entradas xz e yz o rechazar ambos. Por lo tanto, ninguna cadena z puede ser una extensión distintiva de x y y , así que deben estar relacionados por R L . Por lo tanto, s i es un subconjunto de una clase de equivalencia de R L . Combinando este hecho con el hecho de que cada miembro de una de estas clases de equivalencia pertenece a uno de los conjuntos S i , esto da una relación de muchos a uno de los estados de A a las clases de equivalencia, lo que implica que el número de clases de equivalencia es finito y como máximo n . En la otra dirección, suponga que R L tiene un número finito de clases de equivalencia. En este caso, es posible diseñar un autómata finito determinista que tenga un estado para cada clase de equivalencia. El estado de inicio del autómata corresponde a la clase de equivalencia que contiene la cadena vacía, y la función de transición de un estado X en el símbolo de entrada y lleva al autómata a un nuevo estado, el estado correspondiente a la clase de equivalencia que contiene la cadena xy , donde x es una cadena arbitrariamente elegido en la clase de equivalencia de X . La definición de la relación Myhill-Nerode implica que la función de transición está bien definida: no importa qué cadena representativa x se elija para el estado X , se obtendrá el mismo valor de función de transición. Un estado de este autómata es aceptable si la clase de equivalencia correspondiente contiene una cadena en L ; en este caso, nuevamente, la definición de la relación implica que todas las cadenas de la misma clase de equivalencia también deben pertenecer a L , porque de lo contrario la cadena vacía sería una cadena distintiva para algunos pares de cadenas de la clase. Así, la existencia de un autómata finito que reconoce L implica que la relación Myhill-Nerode tiene un número finito de clases de equivalencia, a lo sumo igual al número de estados del autómata, y la existencia de un número finito de clases de equivalencia implica la existencia de un autómata con tantos estados.

MINIMIZACIÓN

Minimización de autómatas

Construcción de un AFDt con un número de estados mínimo que sea equivalente a un AFDt dado.

Definiciones previas:

- Estados accesibles: $q0$ es accesible
  q accesible $\Rightarrow \forall s \in \Sigma$, $\delta(q, s)$ es accesible
- Estados k-equivalentes o k-indistinguibles: $p \equiv_k q$

  $\forall x \in \Sigma^{\leq k}$ ($\delta^*(p, x) \in F \leftrightarrow \delta^*(q, x) \in F$)
- Estados equivalentes o indistinguibles: $p \equiv q$

  $\forall k\ \forall x \in \Sigma^{\leq k}$ ($\delta^*(p, x) \in F \leftrightarrow \delta^*(q, x) \in F$), es decir,
  $\forall x \in \Sigma^*$ ($\delta^*(p, x) \in F \leftrightarrow \delta^*(q, x) \in F$)

Tema 2: Lenguajes regulares 45

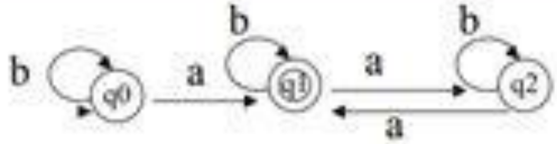
Minimización de autómatas

Construcción del AFDt **mínimo N** a partir del AFDt **M = ($Q, \Sigma, \delta, q_0, F$)**

1) Eliminar estados inaccesibles
2) Determinar las clases de estados equivalentes:
   $p \equiv_0 q \iff (p \in F \leftrightarrow q \in F)$
   $p \equiv_{k+1} q \iff (p \equiv_k q \land \forall s \in \Sigma\ \delta(p, s) \equiv_k \delta(q, s))$
3) Construcción del AFD **N = ($P, \Sigma, \gamma, p_0, G$)** con
   $P = Q/\equiv$ siendo $\equiv$ es la menor $\equiv_k$ tal que $\equiv_k$ coincide con $\equiv_{k+1}$
   $p_0 = [q_0]$
   $\gamma([p], s) = [\delta(p, s)]$
   $G = \{[p]: p \in F\}$

Tema 2: Lenguajes regulares 46

Ejemplo 1

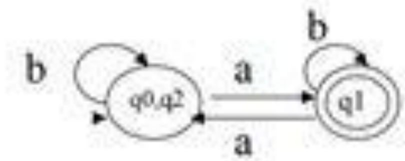


Etapa 0 : $q0 \equiv_0 q2$ (ambos $\notin$ F), pero q0 y q1 no son equivalentes
Clases a nivel 0: [q0,q2] [q1]

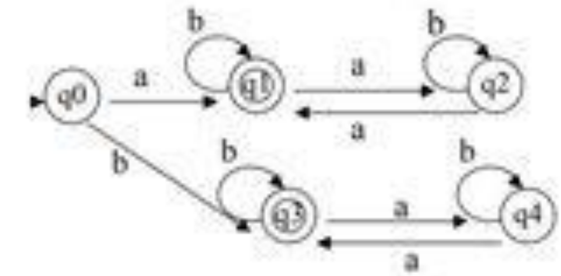
Etapa 1 : $q0 \equiv_1 q2$ porque $\delta(q0, a) \equiv_0 \delta(q2, a)$ y $\delta(q0, b) \equiv_0 \delta(q2, b)$
Clases a nivel 1: [q0,q2] [q1]

Por tanto $\equiv\, = \equiv_0$

AFD mínimo:



Ejemplo 2



q0,q2	a	q1
	b	q3, q2 **no** equiv.
q0,q4	a	q1,q3
	b	q3, q4 **no** equiv.
q2,q4	a	q1,q3
	b	q2,q4
q1,q3	a	q2,q4
	b	q1,q3

Etapa 0 : [q0,q2,q4] [q1,q3]
Etapa 1 : [q0] [q2,q4] [q1,q3]
Etapa 2 : [q0] [q2,q4] [q1,q3]

Por tanto $\equiv = \equiv_1$

AFD mínimo :