Parte III: Lenguajes y Autómatas

2 de noviembre de 2018

Autómatas

Los autómatas son un modelo matemático de dispositivos que *computan*. Los problemas que puede resolver un autómatas dependen de su sofisticación.

Algunas clases de autómatas

Autómatas finitos Problemas que sólo requieren una cantidad *finita* de estados. (Expresiones regulares para encontrar texto con patrones).

Autómatas a pila Problemas que requieren memoria con estructura de pila (¿Están los paréntesis balanceados?).

Máquinas de Turing Problemas que requieren memoria con acceso aleatorio (Cualquier problema resoluble).

Lenguajes y ...

¿Por qué hablamos de lenguajes?

Un problema puede *codificarse* como un lenguaje.

Ejemplo: Sea L_p el conjunto de secuencias de números binarios que representan primos ($01 \in L_p, 101 \in L_p, 100 \notin L_p$).

Decidir si n es primo se reduce a comprobar si su representación binaria está o no en L_p.

Aplicaciones

1. Diseño y verificación de circuitos digitales.
2. Analizador *léxico* de un compilador: descomponer el texto inicial (lo que escribe quien programa) en unidades lógicas (identificador, palabra reservada, puntuación).
3. Exploración de grandes corpus de texto para encontrar patrones.

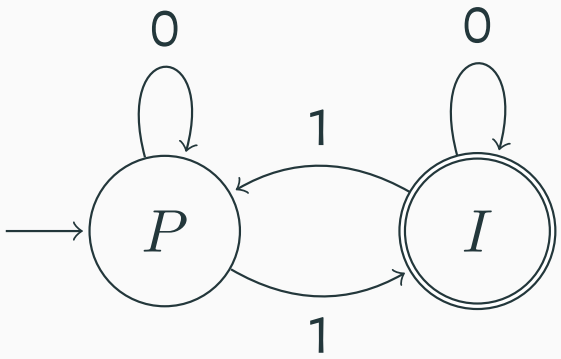
Ejemplo

Ejercicio 9 [Org. Comp.]

Diseñar un circuito secuencial chequeador de paridad de una señal de entrada (IN) de un bit. Funcionamiento: en cada flanco de clk ingresa un nuevo bit, y la salida (OUT) pasa a 1 cuando la cantidad de 1s ingresados desde el inicio de la secuencia es impar, caso contrario es 0.

El modelo de autómatas finitos busca abstraer los detalles que no hacen a la escencia del problema: decidir si una secuencia de 0 y 1 tiene una cantidad impar de unos.

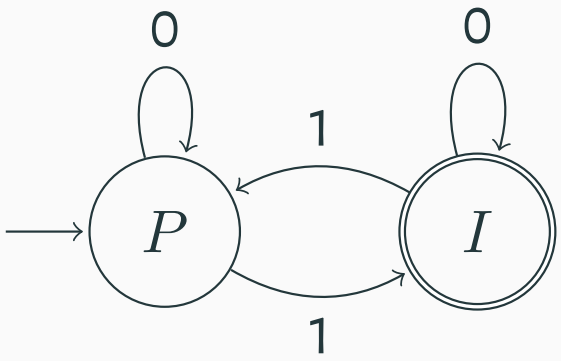
Autómata para el ej. 9



Vocabulario

Estados	Son los nodos del grafo.
Transiciones	Son las aristas, tienen un *símbolo* como etiqueta.
Estado inicial	Se indica por la flecha que no tiene origen.
Estados finales	Se indican por un doble círculo.
Símbolos	Acciones que conoce el autómata.

Autómata para el ej. 9

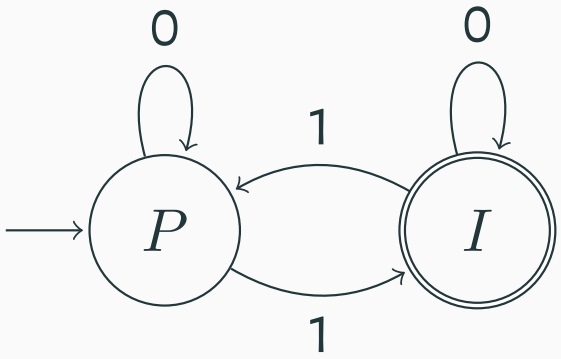


Estados
Cada estado representa cierta información *relevante* de lo que sucedió para lo que estamos computando.

Con el estado P representamos que hemos recibido una cantidad par de unos; mientras que I indica un número impar de unos.

Etiquetas
Señalan las acciones que pueden producir un cambio de estado. Por ejemplo, que llegue un 0 no altera la paridad de unos que recibimos.

Autómata para el ej. 9



Ejecución
Una secuencia de acciones (es decir una palabra de símbolos) produce una secuencia de cambios de estado.

Aceptación
Una palabra es aceptada si la ejecución desde el estado inicial nos lleva a algún estado final.

Demasiados dibujitos, queremos matemática!

Conceptos más formales

Alfabeto Conjunto finito no vacío, habitualmente usamos Σ para referirnos a alfabetos. En el ejemplo anterior $\Sigma = \{0, 1\}$.

Cadena (fijado el alfabeto) Secuencia finita de símbolos de Σ.

A la secuencia de longitud 0 la denotamos con ϵ.

Dado una cadena α y un símbolo $x\alpha$ es la cadena que tiene como primer símbolo x y luego α.

Potencias de Σ Definimos por recursión en $k \in \mathbb{N}$.

$$\Sigma^0 = \{\epsilon\}$$
$$\Sigma^{k+1} = \{x\alpha \mid \alpha \in \Sigma^k, x \in \Sigma\}$$
$$\Sigma^* = \bigcup_{n \in \mathbb{N}} \Sigma^n$$

Notar que Σ^* es siempre infinito. Si $\alpha \in \Sigma^k$, entonces $|\alpha| = k$.

Lenguaje

Un *lenguaje* es un subconjunto de Σ^*. Los podemos describir más o menos formalmente:

Ejemplos

L_p secuencias binarias que representan números primos.

L_{id} cadenas ASCII que son nombres de variables válidas en Python.

L_9 Cadenas con una cantidad impar de unos.

$L_=$ Cadenas con igual cantidad de ceros que de unos.

Casos especiales

$\emptyset$ el lenguaje vacío.

$\{\epsilon\}$

Σ considerado a los símbolos como cadenas.

Σ^* todas las palabras del alfabeto Σ.

Más vocabulario

Concatenación Informalmente: pegar dos palabras del mismo alfabeto.

Formalmente por recursión en una de las palabras:

$$\epsilon\beta = \beta$$
$$(x\alpha)\beta = x(\alpha\beta)$$

Subcadena α es *subcadena* de β si existen γ y γ' tales que $\beta = \gamma\alpha\gamma'$.

Prefijo α es *prefijo* de β si $\beta = \alpha\gamma'$.

Sufijo α es *sufijo* de β si $\beta = \gamma\alpha$.

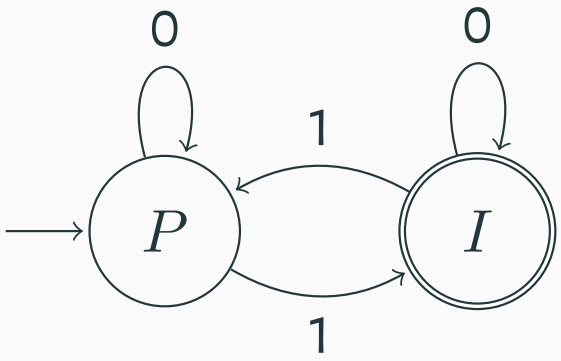
Autómatas finitos deterministas (AFD)

Un *autómata finito determinista* A consta (está determinado por) los siguientes componentes:

1. Un conjunto finito de estados Q.
2. Un alfabeto Σ.
3. Una función de transición (que codifica las aristas de la representación gráfica) $\delta : Q \times \Sigma \to Q$.
4. Un estado inicial $q_0 \in Q$.
5. Un conjunto de estados finales $F \subseteq Q$.

Usaremos la notación $A = (Q, \Sigma, \delta, q_0, F)$ para indicar los componentes del autómata A.

Formalizando el autómata anterior



$$Q = \{P, I\}$$
$$\Sigma = \{0, 1\}$$
$$q_0 = P$$
$$F = \{I\}$$

δ	0	1
P	P	I
I	I	P

	0	1
$\rightarrow P$	P	I
$*I$	I	P

Observaciones

El tipo de la función de transición δ fuerza a que de cada estado salga una (y sólo una) arista por cada símbolo del alfabeto.

Por eso hablamos de autómatas *deterministas*: si el autómata está en un estado sabemos que dado un símbolo de entrada pasará a un único estado siguiente.

Generalizando: dada una palabra existe un único camino desde el estado inicial que consume toda la palabra.

Lenguaje de un autómata y Lenguajes Regulares

La función de transición δ puede ser extendida a $\hat{\delta}$ para que acepte palabras:

$$\hat{\delta} \colon Q \times \Sigma^* \to Q$$
$$\hat{\delta}(q, \epsilon) = q$$
$$\hat{\delta}(q, x\alpha) = \hat{\delta}(\delta(q, x), \alpha)$$

El lenguaje del autómata A son las palabras que comenzando en el estado inicial terminan (según $\hat{\delta}$) en uno final:

$$L_A = \{\alpha \in \Sigma^* \mid \hat{\delta}(q_0, \alpha) \in F\}$$

Dado un lenguaje cualquiera L, diremos que L es *regular* si existe un autómata finito determinista A tal que $L = L_A$.

Cómo probar que L_A es el que pienso que es

¡Caracterizar el lenguaje de cada estado!

Cómo probar que L_A es el que pienso que es

Definimos $L_q = \{\alpha \in \Sigma^* \mid \hat{\delta}(q_0, \alpha) = q\}$

Probamos $\alpha \in L_q$ si y sólo si α cumple cierta propiedad, que está determinada por la información que representa cada estado.
Probamos simultáneamente todos los casos por inducción en el largo de la palabra.

Volviendo al ejemplo
$\alpha \in L_P$ si y sólo si α tiene una cantidad par de unos.
$\alpha \in L_I$ si y sólo si α tiene una cantidad impar de unos.

La caracterización del lenguaje del autómata se deduce de la caracterización de los estados finales.

Equivalencia de Autómatas

Dos autómatas A, A' son *equivalentes* si $L_A = L_{A'}$.

A continuación veremos otra clase de autómatas tales que todo AFD es equivalente a uno de nuestra clase (y viceversa).

Autómatas Finitos No-deterministas (AFN)

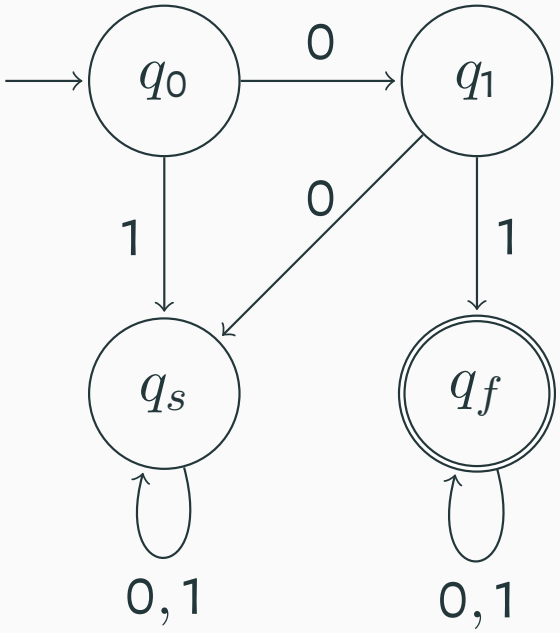
Los autómatas no-deterministas son más permisivos que los AFD: permite que para estado haya cero, una o más transiciones por cada símbolo.

Dada una palabra, puede haber más de un camino desde el estado inicial. Por ello, cambiamos la condición de aceptación.

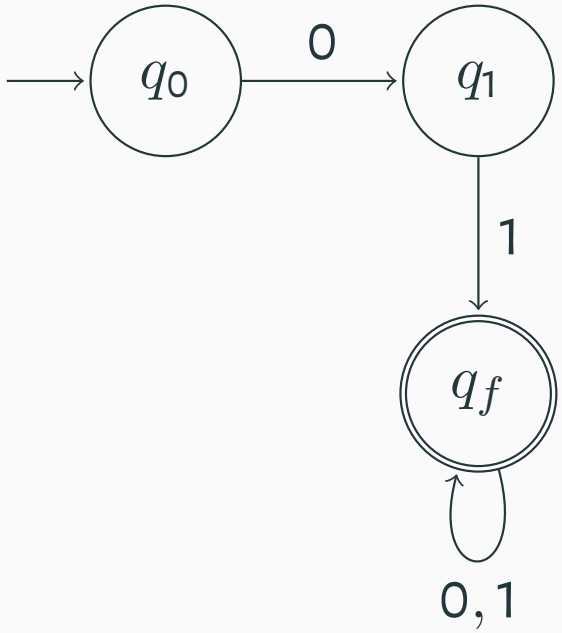
Una palabra es aceptada por un AFN si *existe un camino* que la consume y termina en un estado final.

Ejemplo: palabras que empiezan con 01

Determinista



No-determinista



¡Basta de dibujitos!

Formalizando NFA

Un *autómata finito no-determinista* A consta (está determinado por) los siguientes componentes:

1. Un conjunto finito de estados Q.
2. Un alfabeto Σ.
3. Una función de transición (que codifica las aristas de la representación gráfica) $\delta : Q \times \Sigma \to \mathcal{P}(Q)$.
4. Un estado inicial $q_0 \in Q$.
5. Un conjunto de estados finales $F \subseteq Q$.

El único cambio está en la función de transición.

Aceptación para NFA

Informalmente la aceptación para un NFA está dada por la existencia de un camino que consume la palabra y termina en un estado final.

Como $\delta(p, q)$ es un conjunto de estados no podemos definir la extensión de δ tan sencillamente.

$$\hat{\delta} \colon Q \times \Sigma^* \to \mathcal{P}(Q)$$

$$\hat{\delta}(q, \epsilon) = \{q\}$$

$$\hat{\delta}(q, x\alpha) = \bigcup_{q' \in \delta(q,x)} \hat{\delta}(q', \alpha)$$

El lenguaje del autómata A son las palabras que comenzando en el estado inicial terminan (según $\hat{\delta}$) en uno final:

$$L_A = \{\alpha \in \Sigma^* \mid \hat{\delta}(q_0, \alpha) \cap F \neq \emptyset\}$$