Autómatas Deterministas

Ivan Olmos Pineda

Introducción

- Los autómatas son una representación formal muy útil, que permite modelar el comportamiento de diferentes dispositivos, máquinas, programas, etc.
  - Maquinas expendedoras de refrescos
  - El comportamiento de un programa (software)
  - El comportamiento de semaforos
  - …
- La idea general consiste en modelar un sistema que
  - Recibe un conjunto de elementos de entrada (estímulos)
  - Realiza algún proceso (cómputo)
  - Se produce una salida

Definición autómata

- Formalmente, un autómata finito determinista es una quíntupla $(Q, \Sigma, \delta, q_0, F)$, donde:
  - Q: conjunto finito NO VACIO de estados
  - Σ: alfabeto de entrada
  - $\delta: Q \times \Sigma \rightarrow Q$, función de transición que especifica a qué estado pasa el autómata desde el estado actual al recibir un símbolo d entrada. Esta función se define para todas las parejas posibles de estados y de símbolos de entrada. $\delta(q,a) = q'$ significa que del estado "q" con el símbolo "a", el autómata pasa al estado q'
  - $q_0 \in Q$: estado inicial de autómata
  - $F \subseteq Q$: conjunto de estados finales

Autómatas finitos deterministas

- Ejemplo
  - Considere un sistema formado por una lámpara y un interruptor. La lámpara puede estar encendida o apagada. El sistema sólo puede recibir un estímulo exterior: pulsar el interruptor. El funcionamiento es habitual: si se pulsa el interruptor y estaba apagada la lámpara, se pasa al estado de encendido, o si esta encendida, pasa a pagada. Se desea que la bombilla este inicialmente apagada.
  - Considere que 0: encendido, 1: apagado y la única entrada posible (pulsar interruptor) es "p"
- ¿Cómo sería el autómata que representa a este sistema?

Solución

- $A = (Q, \Sigma, \delta, q_0, F)$, donde:
  - $Q = \{0,1\}$
  - $\Sigma = \{p\}$
  - $\delta(0,p) = 1$, $\delta(1,p) = 0$
  - $q_0 = 0$
  - $F = \{\}$

Representación tabular

- En ocasiones es útil representar a las transiciones de una autómata a través de una tabla
  - El número de filas es igual a |Q|
  - El número de columnas es igual a |Σ|
  - Todas las columnas son etiquetadas con los correspondientes símbolos en Σ
  - Todos los renglones son etiquetados con los estados de Q
  - La casilla en el renglón "q" y en la columna "a" contendrá la transición $\delta(q,a)$
  - El estado inicial se suele marcar con una "→"
  - Los estados finales se suelen marcar con " * "

Diagrama de estados

- A través de un diagrama de estados, se puede representar a un autómata
- Un diagrama de estados de un autómata es un grafo dirigido, donde:
  - Los nodos representan a estados (cada nodo es etiquetado con el nombre del estado)
  - Los arcos representan a transiciones entre los estados
  - Cada arco lleva una etiqueta que indica qué símbolo de entrada provoca la transición correspondiente
  - Si es necesario distinguir al edo inicial, se marca con una "→"
  - Los estados finales irán recuadrados

Reconocimiento de palabras con autómatas

- Una de las aplicaciones más importantes de los autómatas finitos deterministas, es el reconocimiento de palabras y lenguajes
- El proceso de reconocimiento es el siguiente:
  - El autómata comienza en el edo inicial cuando recibe la primera letra de la palabra y transita al edo. siguiente
  - A partir del edo. actual, se vuelve a procesar la siguiente letra para pasar de forma iterativa a los siguientes estados
  - Al finalizar de procesar la palabra, si se llega a un estado final, la palabra es aceptada por el autómata. En caso contrario, se considera rechazada

Extensión de la función de transición a palabras

- Es posible extender la función de transición de un autómata para que acepte un estado y una cadena de símbolos tomados del mismo alfabeto. A dicha función se le denomina **función de transición extendida o sobre palabras**

$$\hat{\delta}(q,x)=\begin{cases} q \ si \ x=\lambda \\ \hat{\delta}(\delta(q,a),y) \ si \ x=ay, a\in\Sigma, y\in\Sigma^* \end{cases}$$

Extensión de la función de transición a palabras

- Ejemplo: construya un AFD que acepte todas las palabras formadas por las letras del alfabeto latino que comienzan con *ep*
- $A = (Q, \Sigma, \delta, q_0, F)$, donde:
  - Q =
  - Σ =
  - δ =
  - q_0 =
  - F =
- Considere que se aplica la función $\hat{\delta}$ de este autómata a las cadenas eppe eepe. ¿Cómo se comportaría?

Extensión de la función de transición a palabras

$$\hat{\delta}(q_0, eppe) = \hat{\delta}(\delta(q_0, e), ppe) = \cdots = \hat{\delta}(q_2, \lambda) = q_2$$

Lenguaje reconocido por un AFD

- Un lenguaje L es **reconocido o aceptado** por un AFD $A = (Q, \Sigma, \delta, q_0, F)$ cuando una palabra cualquiera w es reconocida por el autómata si y sólo si $w \in$ L:

$$L(A) = \{w \in \Sigma^* \mid \hat{\delta}(q_0, w) \in F\}$$

- El lenguaje complementario al lenguaje aceptado por el autómata se representa por L^C(A)

$$L^C(A) = \{w \in \Sigma^* \mid \hat{\delta}(q_0, w) \notin F\}$$

- Se dice que dos autómatas son equivalentes cuando ambos reconocen o aceptan el mismo lenguaje L.

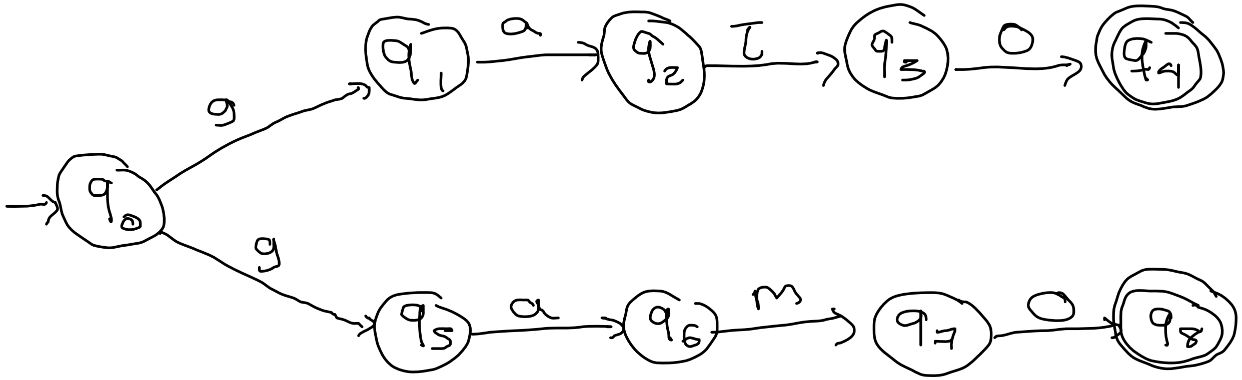
Autómatas Finitos No Deterministas

- Informalmente, un AF no determinista, es una extensión de los deterministas:
  - A partir de un estado, no es necesario que el autómata tenga prevista ninguna transición a otro estado en respuesta a todos los símbolos de entrada posibles
  - A partir de un estado concreto y ante un símbolo de entrada, se permite que el autómata transite a más de un estado distinto (transiciones no deterministas)
  - No es obligatorio consumir un símbolo de entrada para que el autómata cambie de estado (transiciones λ)

Transiciones no deterministas

- Una transición no determinista a partir de un estado q es aquella que, dado un símbolo $a \in \Sigma$, se cumple que:
  - $\delta(q,a)$ es no determinista $\Leftrightarrow |\delta(q,a)| > 1$

- Por ejemplo, considere que se desea diseñar un autómata finito no determinista que admita las palabras {gato, gamo}



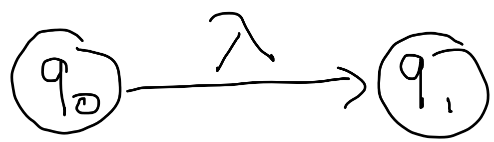
Autómatas finitos no deterministas

- Diseñar un autómata finito no determinista sobre el alfabeto {0,1} que acepte: todas las palabras que tienen al menos tres unos y el de todas las palabras que tienen un número impar de unos.

- ¿Cuál sería la solución?

Transiciones λ

- **Se llama transición λ** a aquélla que hace que el autómata cambie de estado sin consumir ningún símbolo de entrada. El símbolo λ, que representa la palabra vacía, se utiliza para representar la entrada consumida en estas transiciones



- Estas transiciones son útiles para crear un autómata que acepta palabras de otros autómatas

Autómata finito no determinista

- Definición formal: un autómata finito no determinista se define como una quíntupla A = (Q, Σ, δ, q_0, F), donde todo tiene el mismo significado que en un AFD, excepto la función de transición, donde

$$\delta : Qx(\Sigma \cup \{\lambda\} \rightarrow P(Q))$$

- Donde P(Q) es el conjunto de todos los subconjuntos del conjunto Q

Lenguaje reconocido por un autómata finito no determinista

- Se define casi de forma idéntica a un AFD

- Un lenguaje L es **reconocido o aceptado** por un AF no determinista $A = (Q, \Sigma, \delta, q_0, F)$ cuando una palabra cualquiera w es reconocida por el autómata si y sólo si $w \in$ L:

$$L(A) = \{w \in \Sigma^* \mid \hat{\delta}(q_0, w) \bigcap F \neq \phi\}$$

Operaciones entre Autómatas

Minimización de AFD

- Un AFD puede ser reducido (sin alterar el lenguaje aceptado L) a través de dos operaciones:

  - Eliminación de estados inaccesibles

  - Agrupación de estados equivalentes o indistinguibles

- Nota: aplicar dichos operadores da como resultado AFD equivalentes

Eliminación de estados inaccesibles

- Dado A = (Q, Σ, δ, q_0, F) un AFD, se dice que un estado p ∈ Q es **accesible** desde otro estado *q* si existe una palabra x ∈ Σ* tal que

$$\hat{\delta}(q, x) = p$$

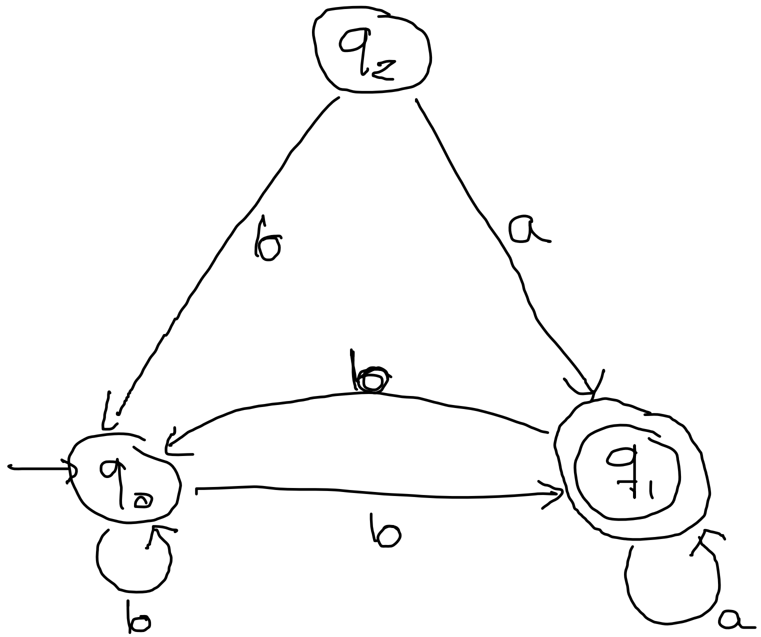
- En caso contrario, se dique que el estado p es **inaccesible** desde q.

Algoritmo para detectar estados inaccesibles en un AFD

1. Se marca q0 como accesible
2. Se inicia una cola $C = q_0$
3. Mientras C no este vacía
   1. Se extrae de C el primer estado q
   2. Para cada transición en el autómata desde q hasta algún estado p, si $p \notin C$, entonces se marca como accesible y p se introduce a C
4. Todo q no marcado es un estado inaccesible del AFD y puede eliminarse

Ejemplo

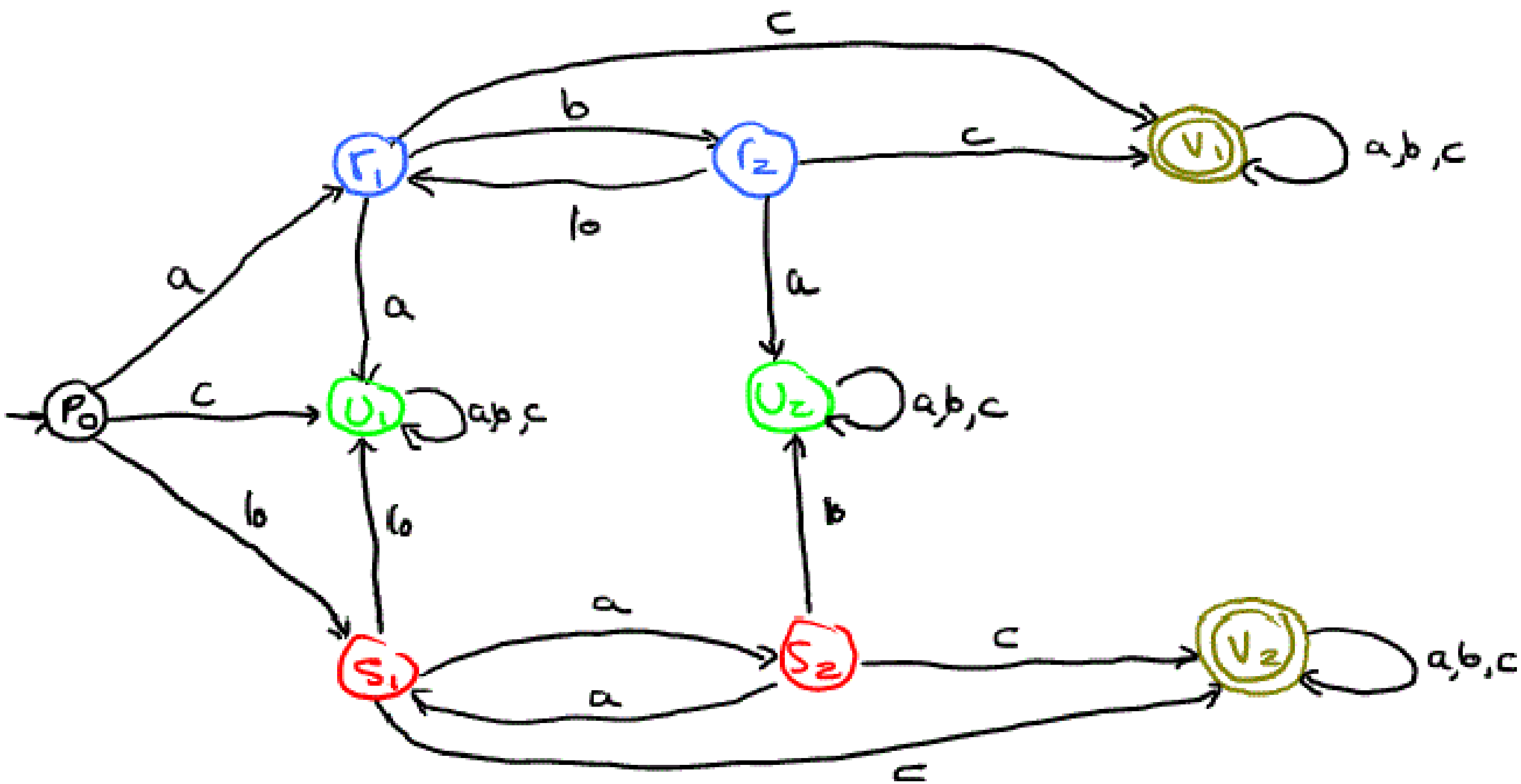
- Del siguiente autómata, eliminar los estados inaccesibles



Agrupación de estados indistinguibles

- Esta reducción se basa en la idea de identificar grupos de estados, los cuales se puedan reducir a un único estado equivalente
  - Dado un estado cualquiera “p”, Lp es el lenguaje formado por las palabras que llevan desde “p” a un estado final
  - Se dice que dos estados p y q de un autómata son equivalentes (se denota pEq) si Lp = Lq

Ejemplo visual



Comentarios

- Visualmente se pode observar lo siguiente:
  - Dos estados serán compatibles si todos los arcos que salen de dichos estados llegan a estados del mismo grupo
  - Que no se mezclen en un mismo grupo estados finale sy no finales del autómata original.

Algoritmo de Minimización

1. Para cada par de estados (p,q), si p es un estado final y q no lo es, o al revés, se marca (p, q) como no equivalentes
2. Marcar que ha habido cambios en el último ciclo
3. Mientras haya cambios en el último ciclo:
    1. Marcar que no ha habido cambios en el último ciclo
    2. Iterar sobre todos los caracteres $\sigma \in \Sigma$
        1. Si (p,q) no están marcados como no equivalentes y $(\delta(p, \sigma), \delta(q, \sigma))$ están marcados como no equivalentes, marcar (p,q) como no equivalentes y marcar que ha habido cambios en el último ciclo

Ejercicios

- Encontrar autómatas finitos deterministas mínimos que reconozcan los siguientes lenguajes
  - $L = \{a^n b^m: \ n >= 2, m >= 1\}$
  - $L = \{a^n b: \ n >= 0\} \cup \{b^n a: \ n >= 1\}$
  - $L = \{a^n: \ n >= 0 \ n \neq 3\}$
  - L = {todas las cadenas del alfabeto {a,b} con exactamente una a}
  - L = {todas las cadenas del alfabeto con no más de tres "a"s}
  - $L = \{ab^5 wb^4: \ w \in \{a,b\}^*\}$