

Autómatas

Escuela Nacional de Antropología e Historia (ENAH)

Agosto – diciembre de 2015

Cadena de texto

- Es una secuencia de caracteres alfanuméricos (letras, números, espacios, tabuladores y signos de puntuación).

Expresiones regulares

- Lenguaje para especificar búsqueda de cadenas de texto
- Uso práctico
- Herramienta teórica para las ciencias de la computación y la lingüística

Expresiones regulares

- Es una fórmula escrita en un lenguaje especial usada para especificar clases de cadenas
- Es una notación algebraica para caracterizar un conjunto de cadenas

Expresiones regulares

- Puede especificar una cadena de búsqueda así como definir un lenguaje de manera formal

Expresiones regulares

- Patrón de búsqueda
- Corpus dónde buscar
- Una función de búsqueda de una expresión regular busca a través de un corpus y regresa todas las cadenas que contienen el patrón

Expresiones regulares

- Patrón de búsqueda
- Corpus dónde buscar
- Una función de búsqueda de una expresión regular busca a través de un corpus y regresa todas las cadenas que contienen el patrón

Expresiones regulares

Cadena de texto

- `/bix/`
- `/u/`

Sensible a mayúsculas y minúsculas

- `/Bix/`

Caracteres opcionales []

- `/[bB]ix/`
- `/[1234567890]/`
- `/[ABCDEFGHIJKLMNOPQRSTUVWXYZ]/`

Expresiones regulares

Rango [-]

- `/[0-9]/`
- `/[a-z]`
- `/[A-Z]/`
- `/[A-Za-z0-9]/`

Negación

- `/[^A-Z]/`
- `/[^A-Za-z0-9]/`
- `/[^A-Za-z0-9\']/`

Expresiones regulares

Repeticiones de caracteres

- /uu/
- /uuu/
- /uuuu/

Expresiones regulares

Contadores

- ? = Cero o una vez
- /u?/
- * = Cero o más veces
- /u*/
- + = Una o más veces
- /u+/

Expresiones regulares

Contadores

- $/uu?/$ = u, uu
- $/uu^*/$ = u, uu, uuu, uuuu
- $/uu+ /$ = uu, uuu, uuuu

Expresiones regulares

Contadores

$\{n\}$ = n ocurrencias del carácter anterior

$\{n,m\}$ = de n a m ocurrencias del carácter anterior

$\{n,\}$ = al menos n ocurrencias del carácter anterior

- $/u\{2}/$
- $/u\{2,3}/$
- $/u\{2,}/$

Expresiones regulares

Comodín (wildcard)

- `/./` = cualquier carácter
- `/u.u/`
- `/.*/` = cualquier cadena de caracteres

Anclas

- `^` inicio
- `$` final
- `/^u/`
- `/u$/`

Expresiones regulares

Escapar caracteres especiales

- `\.`, `\?`, `*`, `\+`
- `/u\.u/`

Disyunción |

- `/ya|yo/`

Expresiones regulares

Alias

- `\d` = cualquier dígito `[0-9]`
- `\D` = cualquier no dígito `[^0-9]`
- `\w` = cualquier carácter alfanumérico y guion bajo `[a-zA-Z0-9_]`
- `\W` = cualquier no carácter alfanumérico ni guion bajo `[^\w]`
- `\s` = espacio `[]`
- `\S` = cualquier no espacio `[^]`

TAREA

- ¿Cuál es la diferencia entre $/^ka|ku\$/$ y $/^ka\|^ku\$/$
- Obtener $/u/$ como palabra completa
- Obtener palabras con más de una $/a/$
- Obtener palabras con prefijo $/mee/$
- Obtener palabras con sufijo $/tik/$
- Escriba las expresiones para comprobar si existen palabras con consonantes a, e, i, o y u repetidas dos o más veces
- Obtener palabras donde tik no sea ni sufijo ni prefijo. Ayuda: la palabra debe tener caracteres al principio y al caracteres al final

TAREA

- Escriba la expresión para buscar si existen palabras con las secuencias de caracteres s_k , l_k y r_k .

Autómatas de estados finitos

- Finite-state automaton (FSA)
- Toda expresión regular puede implementarse como un FSA
- Describe un lenguaje regular

Autómatas de estados finitos

- baa!
- baaa!
- baaaa!
- baaaaa!

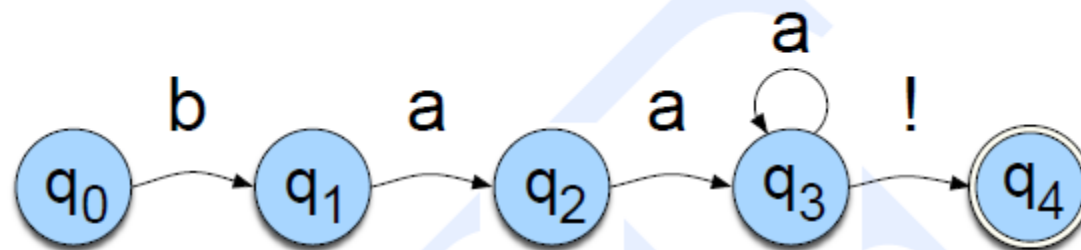


Figure 2.10 A finite-state automaton for talking sheep.

Autómatas de estados finitos

Grafo dirigido

- Vértices o nodos = círculos
- Arcos = flechas

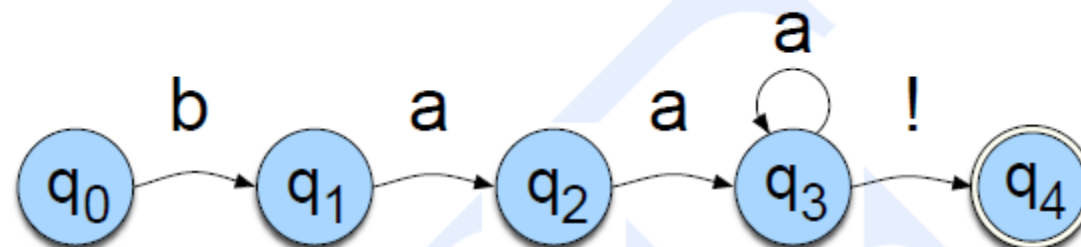


Figure 2.10 A finite-state automaton for talking sheep.

Autómatas de estados finitos

- Estados representados por nodos
- q_0 = estado inicial
- q_4 = estado final (estado de aceptación)
- Transiciones representadas por los arcos

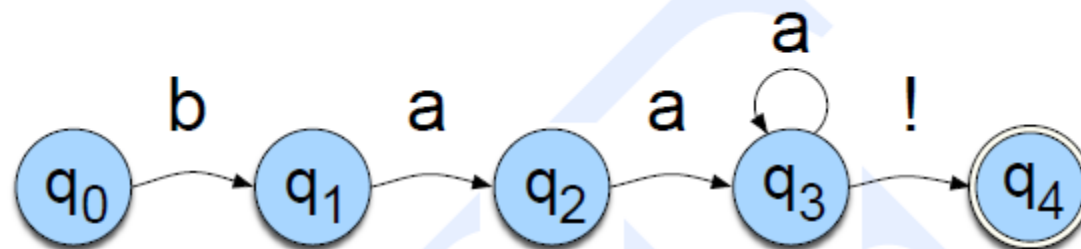


Figure 2.10 A finite-state automaton for talking sheep.

Autómatas de estados finitos

Autómata reconocedor o aceptador: llegar al estado de aceptación

- aba!b
- ba!
- bbaa!

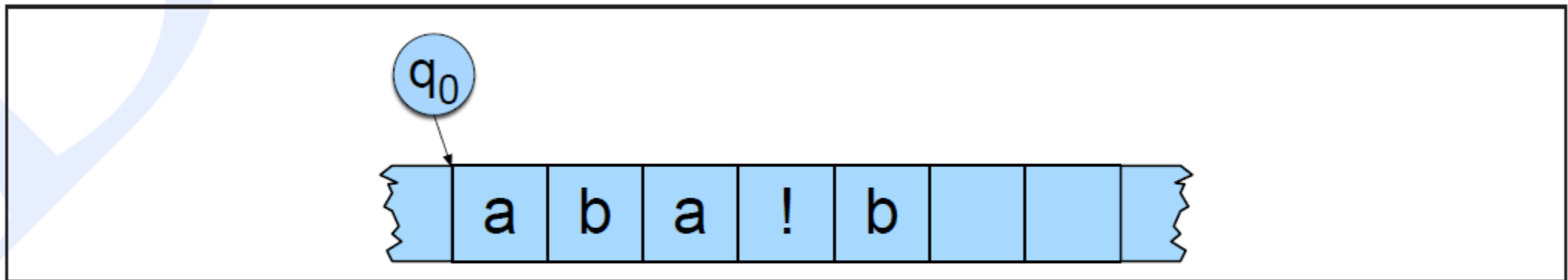


Figure 2.11 A tape with cells.

Autómatas de estados finitos

Autómata reconocedor o aceptador: llegar al estado de aceptación

- baaa!

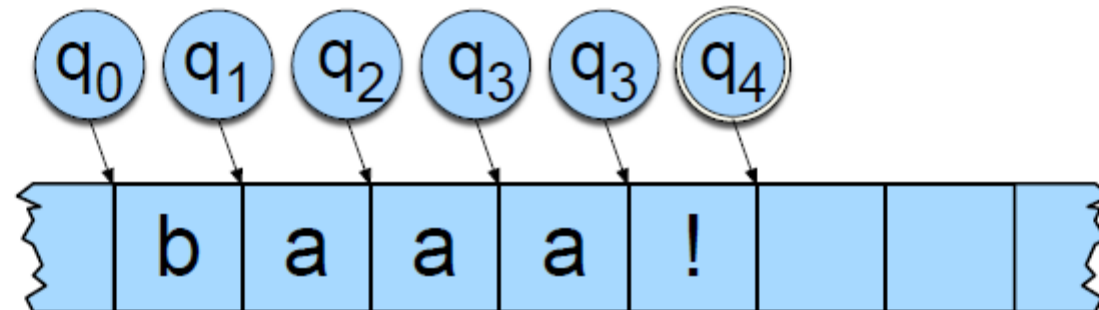


Figure 2.14 Tracing the execution of FSA #1 on some sheeptalk.

Autómatas de estados finitos

Definición formal: conjunto de cinco elementos

$Q = q_0 q_1 q_2 \dots q_{N-1}$		Conjunto de N estados
Σ		Alfabeto de símbolos de entrada
q_0		Un estado de entrada
F		Un conjunto de estados finales, $F \subseteq Q$
$\delta(q, i)$		Funciones de transición. Dado un estado y un símbolo, regresan un nuevo estado

Autómatas de estados finitos

$Q = q_0 q_1 q_2 \dots q_{N-1}$		$Q = \{q_0, q_1, q_2, q_3, q_4\}$
Σ		$\Sigma = \{a, b, !\}$
q_0		q_0
F		$F = \{q_4\}$
$\delta(q, i)$		$\delta(q_0, b) = q_1$ $\delta(q_1, a) = q_2$ $\delta(q_2, a) = q_3$ $\delta(q_3, a) = q_3$ $\delta(q_3, !) = q_4$

Ejercicio

- xooko'
- xook
- xooki'
- xooke'
- xoot
- xooko'obo'
- xooknak
- xookna'ako'on
- xooka'
- xooko'obe'

TAREA

- wiliko'
- wilike'ex
- wila'aj
- wilej
- wilike'
- wilik
- wilibe'
- wilaj
- wil
- wili'
- wile'
- wile'ex
- wilmaj
- wilajo'
- wilmaji'
- wilmi'
- wili'i
- wilo'ob
- wilike'exe'
- wilaje'
- wilme'exi'
- wilmaje'ex
- wilmaje'exi'
- wilike'exo'
- wila'e'

Fin