

Autómatas y Lenguajes Formales 2016-1

Maestría en Ciencia e Ingeniería de la Computación UNAM
Tema 6: Propiedades avanzadas de cerradura de lenguajes
regulares

Dr. Favio Ezequiel Miranda Perea
favio@ciencias.unam.mx

Facultad de Ciencias UNAM

3 de febrero de 2020



Minimal de L , $\text{min}(L)$

- Sea L un lenguaje, definimos el lenguaje $\text{min}(L)$ como sigue:

$$\text{min}(L) = \{x \in \Sigma^* \mid \text{ningún prefijo propio de } x \text{ pertenece a } L\}$$

- Si L es regular entonces $\text{min}(L)$ es regular
- Idea de la prueba:
 - ▶ Sea M es un AFD que acepta a L
 - ▶ Sea N el AFN obtenido a partir de M de tal forma que todos los estados finales de M ahora están bloqueados.
 - ▶ Es decir, se eliminan todas las aristas que salen de estados finales en M .
 - ▶ $\text{min}(L)$ es aceptado por N



Cociente de dos lenguajes L_1/L_2

- Sean L_1, L_2 lenguajes, definimos al lenguaje cociente L_1/L_2 como sigue:

$$L_1/L_2 = \{x \in \Sigma^* \mid \exists z \in L_2 (xz \in L_1)\}$$

- Si L_1 es regular entonces L_1/L_2 es regular
- Idea de la prueba:
 - Sea M un AFD que reconoce a L_1 .
 - Si $x \in L_1/L_2$ entonces existe $z \in L_2$ tal que $\delta^*(q_0, xz) = \delta^*(\delta^*(q_0, x), z) \in F$
 - Para reconocer a x basta modificar M a un AFD M' tal que

$$F' = \{q \in Q \mid \exists z \in L_2 (\delta^*(q, z) \in F)\}$$

- Obsérvese que L_2 no tiene porqué ser regular.



Conjugado o desplazamiento cíclico de un lenguaje $cyc(L)$

- Sea L un lenguaje, definimos el lenguaje $cyc(L)$ como sigue:

$$cyc(L) = \{x_1x_2 \in \Sigma^* \mid x_2x_1 \in L\}$$

- Si L es regular entonces $cyc(L)$ es regular
- Idea de la prueba:
 - ▶ Dividir de forma no determinista una cadena $w \in \Sigma^*$ en dos partes x_1x_2 .
 - ▶ Saltar de forma no determinista a q_i y simular x_1 en M desde q_i
 - ▶ Si $\delta^*(q_i, x_1) = q_j$ y $q_j \notin F$ la simulación falla.
 - ▶ En otro caso $q_j \in F$ y simulamos M en x_2 , aceptando si $\delta^*(q_0, x_2) = q_i$



Mitad de un lenguaje $L_{1/2}$

- Sea L un lenguaje, definimos el lenguaje $L_{1/2}$ como sigue:

$$L_{1/2} = \{x \in \Sigma^* \mid \exists z \in \Sigma^* (|x| = |z| \text{ y } xz \in L)\}$$

- Si L es regular entonces $L_{1/2}$ es regular
- Idea de la prueba:
 - ▶ Adivinar de forma no determinista una cadena z y un estado $q_j \in F$ con $|x| = |z|$
 - ▶ Simular M , en paralelo, procesando x desde q_0 y z hacia atrás desde q_j .
 - ▶ Aceptamos si las dos simulaciones anteriores terminan en un mismo estado q_k .



Particiones racionales de L , L_q con $q \in \mathbb{Q}$

- Sean L un lenguaje y $q \in \mathbb{Q}$, definimos el lenguaje L_q como sigue:

$$L_q = \{x \in \Sigma^* \mid \exists z \in \Sigma^* (|x| = q|xz| \text{ y } xz \in L)\}$$

- Si L es regular entonces L_q es regular
- La idea de la prueba consiste en generalizar el caso de $q = 1/2$.



Derivada de un lenguaje L con respecto a una cadena u , $\partial_u(L)$

- Sean L es un lenguaje y $u \in \Sigma^*$. La derivada de L con respecto a u se define como

$$\partial_u(L) = \{v \in \Sigma^* \mid uv \in L\}$$

- $\partial_u(L)$ es el lenguaje de aquellas cadenas v tales que al agregar u como prefijo a v , el resultado pertenece a L .
- Si L es regular entonces $\partial_u(L)$ es regular.



Derivada de un lenguaje L con respecto a una cadena u , $\partial_u(L)$

- Si L es regular entonces $\partial_u(L)$ es regular.
- Idea de la prueba:
 - ▶ Sea M un AFD que reconoce a L .
 - ▶ Si $uv \in L$ entonces $\delta^*(q_0, uv) = \delta^*(\delta^*(q_0, u), v) \in L$
 - ▶ Para reconocer a $\partial_u(L)$, basta definir M' únicamente cambiando el estado inicial de M a $q'_0 = \delta^*(q_0, u)$
- El concepto de derivada permite definir un algoritmo para el problema $ER \Rightarrow AFD$ sin pasar por un AFN .

