Sesión 13

Lema de bombeo para *LR*

¿Cómo podemos saber is un lenguaje es regular?

- Teorema de Kleene: Si hay una *ER* que describa *L* o un FA que acepte *L*, entonces *L* es regular
- Pero, ¿qué tal si el lenguaje se decribe por otros medios?:
  - $L = \{0^n1^n \in \Sigma^* \mid n > 0\}$
  - ¿Es regular?
- Estrategia: Encontrar una propiedad que:
  - Todos los *LR* tengan
  - Sea fácil de verificar
  - Si el lenguaje no tiene dicha propiedad, no es regular!

Ya conocemos algunas propiedades…

- El Teorema de Myhill & Nerode: *L* es regular si y solo sí se puede particionar en un conjunto finito de clases de equivalencia
- ¿Qué tan fácil es verificar esta propiedad para un lenguaje dado?
  - ¿El lenguaje de las cadenas terminadas en 10?
  - ¿El lenguaje de todas las palíndromes?
  - ¿Cualquier otro lenguaje?

Otra propiedad más...

- Si $L \subseteq \Sigma^*$ y para algún *n*, hay *n* cadenas en Σ^*, tales que cualquier par de cadenas son distinguibles con respecto a *L*
- Entonces, todo FA que reconozca a *L* debe tener cuando menos *n* estados.

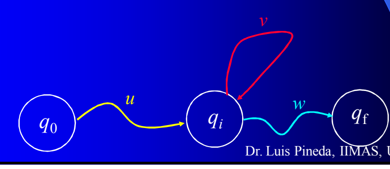
Otra propiedad: El ciclo!

- Sea $M = (Q, \Sigma, q_0, A, \delta)$ un FA con *n* estados & *x* una cadena de longitud cuando menos *n*
  - $x = a_1a_2\ldots a_ny$
- Entonces, toda secuencia de longitud $n + 1$:
  - $q_0 = \delta^*(q_0, \Lambda)$
  - $q_1 = \delta^*(q_0, a_1)$
  - $q_2 = \delta^*(q_0, a_1a_2)$
  - ….
  - $q_n = \delta^*(q_0, a_1a_2\ldots a_n)$

  contiene un ciclo necesariamente!

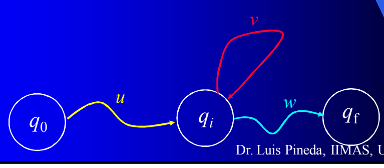
Cadena con un ciclo!

- Si $x \in L$ es suficientemente larga, la forma de *x* is uvw



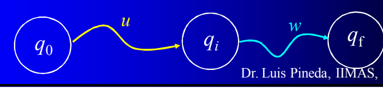
¡Un ciclo es un ciclo!

- El ciclo se puede repetir muchas veces: $uv^m w \in L$
- El ciclo corresponde a la cerradura v^m
- El ciclo puede ocurrir 0 o más veces!



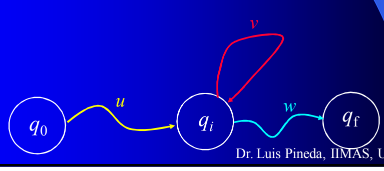
¡Bombeando el ciclo!

- El ciclo ocurre 0 veces: $uv^0 w \in L$



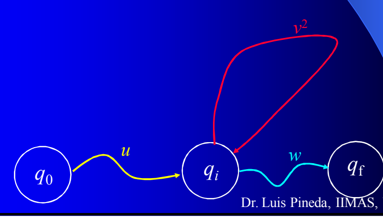
¡Bombeando el ciclo!

- El ciclo ocurre 1 vez: $uv^1 w = uvw \in L$



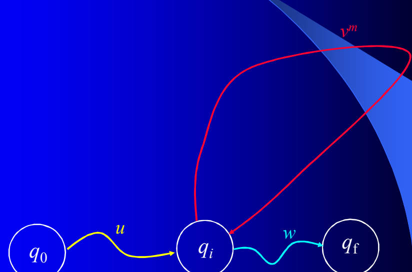
¡Bombeando el ciclo!

- El ciclo ocurre 2 veces: $uv^2 w \in L$



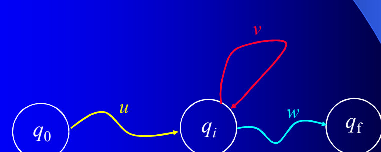
¡Bombeando el ciclo!

- El ciclo ocurre m veces: $uv^m w \in L$



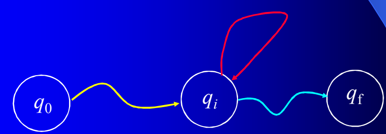
El ciclo & δ^*

- $\delta^*(q_i, v) = q_i$
- $\delta^*(q_i, v^m) = q_i$ para todo $m \geq 0$ &
- $\delta^*(q_0, uv^m w) = q_f$ para todo $m \geq 0$



Las condiciones para el ciclo

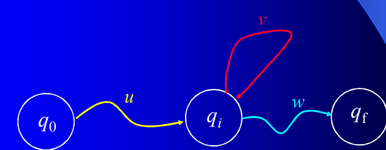
- Supongamos que $q_i = q_{i+p,}$ tal que $0 \le i \le p \le n$ done n es el número de estado:
  - $\delta^*(q_0, a_1a_2\ldots a_i) = q_i$
  - $\delta^*(q_i, a_{i+1}a_{i+2}\ldots a_{i+p}) = q_i$
  - $\delta^*(q_i, a_{i+p+1}a_{i+p+2}\ldots a_ny) = q_f \in A$



  - ¿Qué condiciones garantizan que ocurra el ciclo?

¡Simplicando la notación!

- $u = a_1a_2\ldots a_i$
- $v = a_{i+1}a_{i+2}\ldots a_{i+p}$
- $w = a_{i+p+1}a_{i+p+2}\ldots a_ny$



¿Cuándo se da el loop?

- Sólo sabemos que el FA tiene n estados:

aquí!

$$a_1a_2\ldots a_ia_{i+1}a_{i+2}\ldots a_{i+p}a_{i+p+1}a_{i+p+2}\ldots a_ny$$

- Después de leer n símbolos, se necesitan $n + 1$ estados (empezando en q_0) cuando menos
- Para asegurarnos que se da el ciclo: $n \ge i + p$

¿Cuándo se da el loop?

- Sólo sabemos que el FA tiene n estados:

Valor mínimo para n

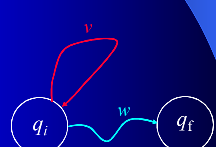
$$a_1a_2\ldots a_ia_{i+1}a_{i+2}\ldots a_{i+p}a_{i+p+1}a_{i+p+2}\ldots a_ny$$

- Para asegurarnos que hay un ciclo: $n \ge i + p$
- p debe ser cuando menos 1 (¡de otro modo no hay ciclo!)

¿Cuándo se da el ciclo?

- u puede ser vacía (pero no lo podemos saber):

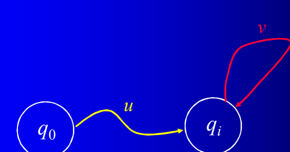
$$\Lambda a_{i+1}a_{i+2}\ldots a_{i+p}a_{i+p+1}a_{i+p+2}\ldots a_ny$$



¿Cuándo se da el ciclo?

- w puede ser vacía (pero no lo podemos saber):

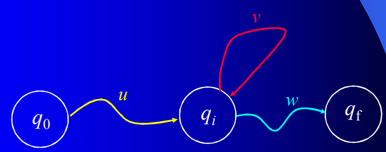
$$a_1a_2\ldots a_ia_{i+1}a_{i+2}\ldots a_{i+p}\Lambda$$



¿Cuándo se da el ciclo?

- $|v|$ tiene que ser cuando menos 1 (pero no lo podemos saber):

$a_1a_2\ldots a_i a_{i+1} a_{i+p+1} a_{i+p+2}\ldots a_n y$



El lema del bombeo para *LR*

- Sea L un LR que se reconoce por un FA con n estados. Para todo $x \in L$ con $|x| \geq n$, x se puede reescribir como $x = uvw$ para las cadenas u, v & w que satisfacen:
  - $|uv| \leq n$: Existe un segmento inicial que incluye el ciclo
  - $|v| > 0$: La longitud del ciclo es cuando menos 1
  - Para todo $m \geq 0$, $uv^mw \in L$: El ciclo se puede bombear!

¿Y si *L* no es regular?

- Si no sabemos si L es regular:
  - ¡No sabemos si hay un FA que acepte a L!
  - No sabemos cuántos estados el supuesto FA tendría (¡en el caso de que L resultara ser regular!)

El lema del bombeo

- Supongamos que L es un LR; entonces existe un n tal que para todo $x \in L$ con $|x| \geq n$, existen las cadenas u, v & w tales que:
  - $x = uvw$
  - $|uv| \leq n$: Existe un segmento inicial que incluye el ciclo
  - $|v| > 0$: La longitud del ciclo es cuando menos 1
  - Para todo $m \geq 0$, $uv^mw \in L$: El ciclo se puede bombear!

Pertenencia a una clase

- Condiciones necesarias y suficientes:
  - Necesarias: todo miembro de la clase tiene la propiedad
  - Suficientes: Tener la propiedad es suficiente para pertencer a la clase
  - ¿Qué tipo de propiedad de un lenguaje es la propiedad de satisfacer el lema del bombeo?

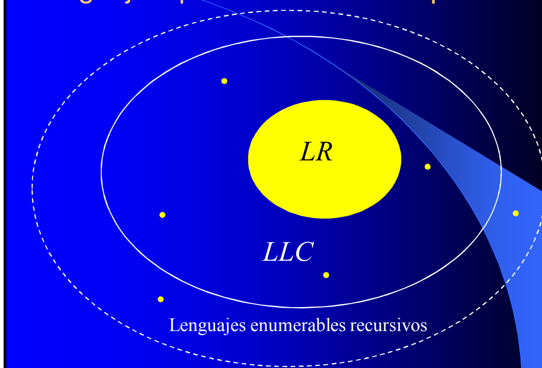
Uso del lema del bombeo

- Teorema Myhill & Nerode: Especifica una condición necesaria y suficiente de todo lenguaje regular
- Lema de bombeo: sólo una condición necesaria pero no suficiente: Hay *Lenguajes libres del contexto* (*LLC*) que satisfacen el lema del bombeo para LR!

Uso del lema del bombeo

- El lema del bombeo sólo puede usarse para mostrar que un lenguaje no es regular (por reducción al absurdo), <u>pero no puede usarse para mostrar que un lenguaje regular</u>

Lenguajes que satisfacen el *LB* para *LR*



Uso del lema del bombeo

- Si tenemos un lenguaje descrito por otros medios:
  - $L = \{0^n 1^n \in \Sigma^* \mid n > 0\}$
  - ¿Es este lenguaje regular?
- Estrategia:
  - Asumir que el lema de bombeo se satisface
  - Si de esta hipótesis se sigue una contradicción el lenguaje no es regular!

Uso del lema del bombeo

- El juego del bombeo (del lema):
  - Jugador 1 selecciona un lenguje L
  - Jugador 2 selecciona un n
  - Jugador 1 selecciona una cadena x en L tal que $|x| \geq n$
  - Jugador 2 divide x en uvw tal que $|uv| \leq n$ & $|v| > 0$ (pero no se la dice al jugador 1)
  - Jugador 1 gana encontrando una $m \geq 0$ tal que $uv^m w \notin L$

$L = \{0^n 1^n \in \Sigma^* \mid n > 0\}$ no es regular

- Probar que $L = \{0^n 1^n \in \Sigma^* \mid n > 0\}$ no es regular
- El jugador 1 selecciona al lenguaje L
- El jugador 2 selecciona un n
  - El parámetro n está bien.
- El jugador 1 selecciona una cadena x en L tal que $|x| \geq n$
  - La cadena es $x = 0^n 1^n$ está bien: $|x| = 2n \geq n$

$L = \{0^n 1^n \in \Sigma^* \mid n > 0\}$ no es regular

- El jugador 2 divide a x en uvw tal que $|uv| \leq n$ & $|v| > 0$:
  - $u = 0^i$ & $v = 0^j$ donde $i \geq 0,\ j > 0$, so $|uv| = |0^i 0^j| \leq n$ and $|v| = j > 0$
- El Jugador 1 gana encontrando a $m \geq 0$ tal que $uv^m w \notin L$
  - $m = 0$: $uv^m w = 0^i 1^n \notin L$ ya que $i < n$

L_{pr} = Los números primos

- El jugador 1 selecciona el L
  - $L_{pr} = \{x \in \Sigma^* \mid$ la longitud de x es un número primo en notación monádica$\}$
- El jugador 2 selecciona n
  - El parámetro n tal que $p \geq n + 2$ para cierto p (i.e. siempre habrá un número primo mayor que n)

L_{pr} = Los números primos

- El jugador 1 selecciona una cadena x en L tal que $|x| \geq n$
  - La cadena $x = 1^p$ está bien:

    $|x| = p \geq n + 2 \geq n$
- El jugador 2 divide a x en uvw tal que $|uv| \leq n$ & $|v| > 0$:
  - Sea $|v| = k$ para $k > 0$
  - Asumimos que $|uv| \leq n$

L_{pr} = Los números primos

- El jugador 1 gana encontrando una $m \geq 0$ tal que $uv^mw \notin L$
  - Número primo p: $|uvw| = p$
  - Sea $m = p - k$
  - Si L_{pr} es regular entonces $uv^{p-k}w$ debe estar en L_{pr}
    - $|v| = k$ para $k > 0$
    - $|uw| = p - k$

L_{pr} = Los números primos

- $|uv^{p-k}w| = |uw| + (p - k)|v|$

  $= (p - k) + (p - k)k$

  $= (p - k)(1 + k)$
- $|uv^{p-k}w|$ tiene dos factores, por lo que no puede ser un número primo a menos que uno de los factores sea 1

L_{pr} = Los número primos

- El jugador 1 gana encontrando una $m \geq 0$ tal que $uv^mw \notin L$
- Verificamos que los dos factores sean diferentes de 1:
  - $(1 + k) > 1$ dado que $k > 0$
  - $k \leq n$ dado que $k = |v| \leq |uv| \leq n$ (hipótesis original)
  - $(p - k) > 1$ dado que $p \geq n + 2$ & $k \leq n$ por lo que $p \geq k + 2$

L_{pr} = Los número primos

- Consecuentemente,
  - $|uv^{p-k}w| = (p - k)(1 + k)$ donde
    - $(p - k) > 1$
    - $(1 + k) > 1$
  - $uv^{p-k}w \notin L$
- No podemos bombear una cadena de 1's en la representación monádica de un número primo, y obtener otro primo!

El poder de las *ER* y los FA

- Un *problema* consiste en decidir si una cadena pertenece a un lenguaje
- Los FA permiten resolver problemas de decisión simples (i.e. si x es un número par, pero no si x es un palíndromo).

Dr. Luis Pineda, IIMAS, UNAM, 2010

El poder de las *ER* y los FA

- Problema de decisión genérico para un FA: dada una cadena x y un FA M decidir si x se acepta por M:

$$(M, x) \longrightarrow \boxed{\text{Problemas de decisión para } RL} \longrightarrow \text{Si/No}$$

Dr. Luis Pineda, IIMAS, UNAM, 2010

Problemas de decisión de *RL*

- Dada una *RE* r & una cadena x, $x \subseteq L(r)$?
- Dado un FA M, es $L(M) = \Phi$?
- Dado un FA M, es $L(M)$ finito?
- Dado M_1 & M_2, es $L(M_1) \cap L(M_2) \neq \Phi$?

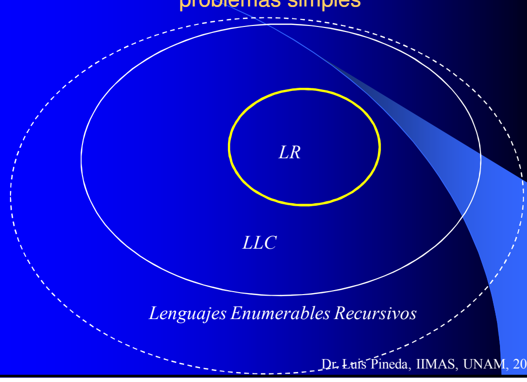
Dr. Luis Pineda, IIMAS, UNAM, 2010

Problemas de decisión de *RL*

- Dado M_1 & M_2, es $L(M_1) = L(M_2)$?
- Dado M_1 & M_2, es $L(M_1) \subseteq L(M_2)$?
- Dadas dos *REs* r_1 and r_2, es $L(r_1) = L(r_2)$?
- Dado un FA M es mínimo?

Dr. Luis Pineda, IIMAS, UNAM, 2010

Los LR ofrecen un método sólido y eficiente para resolve problemas simples



Dr. Luis Pineda, IIMAS, UNAM, 2010