3.3.1 Lema de iteración (o de bombeo) para LIC

3.3 Propiedades de LIC

Lema de iteración (o de bombeo)

- Resultado útil para demostrar que algunos lenguajes no son independientes del contexto
  - Como en el caso de los regulares con la versión que ya conocemos (para REG)
- Interesa tener presentes todas las observaciones y recomendaciones hechas para la aplicación de la versión anterior
  - La versión para REG era más sencilla…

Lema de iteración para LICs

- Sea L un LIC. Entonces:
  - Existe una constante $n \geq 1$ tal que
  - Para toda cadena $z \in L$ con $|z| \geq n$
  - Existen cadenas $u, v, w, x, y \in \Sigma^*$ tales que
    1. $z = uvwxy$
    2. $|vwx| \leq n$
    3. $|vx| \geq 1$ $(vx \neq \varepsilon)$
    4. Para todo $i \geq 0$: $uv^iwx^iy \in L$
       - ¡Se itera en dos sitios a la vez!

Enunciado

Ejemplo 1: $L_1 = \{a^m b^m c^m \mid m > 0\}$

- L_1 no es un LIC
  - Lo demostraremos probando que no cumple la propiedad que todos los LIC cumplen
  - Es decir, demostraremos que cumple su negación
    - $\forall\, n \geq 1$
    - $\exists\, z \in L_1$ con $|z| \geq n$
    - $\forall\, u, v, w, x, y \in \Sigma^*$
      $(z = uvwxy \wedge 1 \leq |vx| \leq |vwx| \leq n \rightarrow \exists\, i \geq 0\text{: } uv^iwx^iy \notin L_1)$
  - Y concluiremos así que $L_1 \notin$ LIC

Uso

Ejemplo 1: $L_1 = \{a^m b^m c^m \mid m > 0\}$

- L_1 no es un LIC
  - $\forall\ n \geq 1$
  - $\exists\ z \in L_1$ con $|z| \geq n$
    - **$z = a^n b^n c^n$**
  - $\forall\ u, v, w, x, y \in \Sigma^*: z = uvwxy \wedge 1 \leq |vx| \leq |vwx| \leq n$
    - v, x no pueden contener apariciones de los 3 símbolos
      - Máximo de 2 símbolos consecutivos (a's y c's no puede ser)
      - Esto garantiza que uno de los 3 grupos nunca se va a alterar
  - $\exists\ i \geq 0: uv^iwx^iy \notin L_1$
    - Entonces $L_1 \notin$ LIC

Ejemplo 1: $L_1 = \{a^m b^m c^m \mid m > 0\}$

- **$z = a^n b^n c^n$**

- $\forall$ u, v, w, x, y $\in \Sigma^*$: $z = uvwxy \wedge 1 \leq |vx| \leq |vwx| \leq n$
  1. vx tiene apariciones de un único símbolo
  2. vx tiene apariciones de dos símbolos (más es imposible)
- $\exists\, i \geq 0$: $uv^iwx^iy \notin L_1$

Distinción de casos

  - **i = 0**
    1. $uv^0wx^0y = uwy \notin L_1$ pues tienen menos apariciones de uno de los grupos de símbolos que de los otros dos
    2. $uv^0wx^0y = uwy \notin L_1$ pues tienen menos apariciones de dos de los grupos de símbolos que del otro
- Entonces $L_1 \notin$ LIC

Nada obliga a elegir la misma iteración para cada caso

Ejemplo 1: cadenas iteradas

- **$z = a^n b^n c^n$**

1. <u>vx tiene apariciones de un único símbolo</u>
   - Sea $|vx| = j$ (sabemos que $1 \leq j \leq n$)

- **$i = 0$**

Pinta de las cadenas iteradas (en cada caso)

1. $uv^0wx^0y = uwy \notin L_1$
   » $a^{n-j}\, b^n\, c^n \notin L_1$
   » $a^n\, b^{n-j}\, c^n \notin L_1$
   » $a^n\, b^n\, c^{n-j} \notin L_1$

   pues tienen menos apariciones de uno de los grupos de símbolos que de los otros dos

Ejemplo 1: cadenas iteradas

- **$z = a^n b^n c^n$**

2. <u>vx tiene apariciones de dos símbolos</u> (más es imposible)
   - Sean $|vx|_a = s_a$, $|vx|_b = s_b$, $|vx|_c = s_c$
   - Sabemos que:
     - » O bien $s_c = 0$ y $1 \leq s_a, s_b < n$
     - » o bien $s_a = 0$ y $1 \leq s_b, s_c < n$

Pinta de las cadenas iteradas (en cada caso)

- **$i = 0$**

  2. $uv^0wx^0y = uwy \notin L_1$
     - » $a^{n-s_a} b^{n-s_b} c^n \notin L_1$
     - » $a^n b^{n-s_b} c^{n-s_c} \notin L_1$

     pues tienen menos apariciones de dos de los grupos de símbolos que del otro

Ejemplo 1: $L_1 = \{a^m b^m c^m \mid m > 0\}$

- Conclusión:
  - Al haber demostrado todos los casos: $L_1 \notin$ LIC
  - Aunque los lenguajes independientes del contexto pueden emparejar 2 grupos de símbolos para establecer si son iguales o no
    - $\{a^m b^m \mid m > 0\} \in$ LIC
  - Los LIC no pueden emparejar tres grupos de símbolos para establecer si son iguales o no
    - $\{a^m b^m c^m \mid m > 0\} \notin$ LIC

Ejemplo 1: eligiendo la iteración 2

- Si tuviésemos que elegir la iteración 2, habría que afinar más en la distinción de casos para poder describir la pinta de las cadenas iteradas
  1. vx tiene apariciones de un único símbolo
     - » $vx \in \{a\}^+$ o $vx \in \{b\}^+$ o $vx \in \{c\}^+$
  2. vx tiene apariciones de dos símbolos (más es imposible)
     1. v contiene apariciones de un símbolo y x de otro distinto
        - » $v \in \{a\}^+, x \in \{b\}^+$ o $v \in \{b\}^+, x \in \{c\}^+$
     2. una de las dos cadenas contiene apariciones de dos símbolos distintos
        - » $v \in \{a\}^+ \{b\}^+, x \in \{b\}^*$ o $v \in \{b\}^+ \{c\}^+, x \in \{c\}^*$
          $x \in \{a\}^+ \{b\}^+, v \in \{a\}^*$ o $x \in \{b\}^+ \{c\}^+, v \in \{b\}^*$

Ejemplo 1: eligiendo la iteración 2

- Los primeros casos son igual de fáciles:
  1. <u>vx tiene apariciones de un único símbolo</u>
     - $vx \in \{a\}^+$ o $vx \in \{b\}^+$ o $vx \in \{c\}^+$
       - » Sea $|vx| = j$ (sabemos que $1 \leq j \leq n$)
- **i = 2**
  1. $uv^2wx^2y = uvvwxxy \notin L_1$
     - » $a^{n+j}\, b^n\, c^n \notin L_1$
     - » $a^n\, b^{n+j}\, c^n \notin L_1$
     - » $a^n\, b^n\, c^{n+j} \notin L_1$

     pues tienen más apariciones de uno de los grupos de símbolos que de los otros dos

Ejemplo 1: eligiendo la iteración 2

- Los primeros casos son igual de fáciles:
  2.1. <u>v contiene apariciones de un símbolo y x de otro distinto</u>
    - » $v \in \{a\}^+, x \in \{b\}^+$ o $v \in \{b\}^+, x \in \{c\}^+$
    - » Sean $|vx|_a = s_a$, $|vx|_b = s_b$, $|vx|_c = s_c$
    - » Sabemos que:
      - O bien $s_c = 0$ y $1 \leq s_a, s_b < n$
      - o bien $s_a = 0$ y $1 \leq s_b, s_c < n$
- $uv^2wx^2y = uvvwxxy \notin L_1$
    - » $a^{n+s_a} b^{n+s_b} c^n \notin L_1$
    - » $a^n b^{n+s_b} c^{n+s_c} \notin L_1$

  pues tienen más apariciones de dos de los grupos de símbolos que del otro

Ejemplo 1: eligiendo la iteración 2

- El último se complica:
  - 2.2. <u>una de las dos cadenas contiene apariciones de dos símbolos distintos</u>
    - » <u>$v \in \{a\}^+ \{b\}^+, x \in \{b\}^*$</u> o $v \in \{b\}^+ \{c\}^+, x \in \{c\}^*$
      $x \in \{a\}^+ \{b\}^+, v \in \{a\}^*$ o $x \in \{b\}^+ \{c\}^+, v \in \{b\}^*$
  - Haremos sólo el primer caso detallado **(faltaría el resto)**
    - » Sean $v = a^{s_a} b^{s_{bv}}$, $x = b^{s_{bx}}$
    - » Sabemos que:
      - $1 \leq s_a, s_{bv} < n$ y $0 \leq s_{bx}$
    - » Entonces podemos deducir que:
      - $u = a^{n-s_a}$ $w = b^{s_{bw}}$ $y = b^{n - (s_{bv} + s_{bx} + s_{bw})} c^n$
  - Con todo esto deduciremos la pinta de $uv^2wx^2y = uvvwxxy \notin L_1$

Ejemplo 1: eligiendo la iteración 2

- El último se complica:
  - Recordamos:
    - $v = a^{s_a}\, b^{s_{bv}}$ $\quad x = b^{s_{bx}}$ $\qquad (1 \leq s_a, s_{bv} < n$ y $0 \leq s_{bx})$
    - $u = a^{n-s_a}$ $\quad w = b^{s_{bw}}$ $\qquad y = b^{n-(s_{bv}+s_{bx}+s_{bw})}\, c^n$
  - La pinta de $uv^2wx^2y = uvvwxxy$ es

$$uv^2wx^2y = \underbrace{a^{n-s_a}}_{u}\ \underbrace{a^{s_a}\, b^{s_{bv}}}_{v}\ \underbrace{a^{s_a}\, b^{s_{bv}}}_{v}\ \underbrace{b^{s_{bw}}}_{w}\ \underbrace{b^{s_{bx}}}_{x}\ \underbrace{b^{s_{bx}}}_{x}\ \underbrace{b^{n-(s_{bv}+s_{bx}+s_{bw})}\, c^n}_{y}$$

$$= a^n\, b^{s_{bv}}\, a^{s_a}\, b^{n+s_{bx}}\, c^n \notin L_1 \qquad (\text{por ser } 1 \leq s_a, s_{bv})$$

pues tienen más apariciones de dos de los grupos de símbolos que del otro (o porque aparece una ‘a’ tras una ‘b’, y eso rompe la estructura de las cadenas de L_1)

Moraleja

- Siempre que sea posible, elegiremos la iteración 0
  - Es más sencillo describir las cadenas iteradas cuando "se quita" que cuando "se pone"
  - Hay menos distinciones de casos que considerar

Ejemplo 2: $L_2 = \{ww \mid w \in \{0, 1\}^*\}$

- Otra restricción:
  - Los LIC tampoco pueden emparejar dos cadenas de longitud arbitraria, si las cadenas se forman con más de un símbolo
- Implicación:
  - Las GICs no son un mecanismo adecuado para forzar determinadas restricciones “semánticas” en los LPs como el requisito habitual de que un identificador tenga que declararse antes de usarlo

Ejemplo 2: $L_2 = \{ww \mid w \in \{0, 1\}^*\}$

- L_2 no es un LIC
  - $\forall\ n \geq 1$
  - $\exists\ z \in L_2$ con $|z| \geq n$
    - Cadenas que puedan partirse por la mitad en 2 iguales
    - $z = 0^n0^n$
      - ¿Vale?
    - $z = (01)^{2n}$ $z = 0^n10^n1$
      - ¿Vale?
    - $z = 0^n10^n1$
      - ¿Vale?

Ejemplo 2: $L_2 = \{ww \mid w \in \{0, 1\}^*\}$

- ¿Por qué elegir estas z no funciona?
  - $z = (01)^{2n}$ $z = 0^n0^n$ $z = 0^n10^n1$
  - Podría ocurrir que justo lo que ponemos o quitamos en la iteración i no impida la pertenencia de la cadena iterada uv^iwx^iy a L_2
  - Bastaría con que
    - v o x = $(01)^{2j}$ (y la otra ε)
    - Entonces $uv^iwx^iy = (01)^{2(n + (i-1)j)} \in L_2\ \forall\ i \geq 0$
      - Por ser de la forma $(01)^{2s} = (01)^s\,(01)^s$

Ejemplo 2: $L_2 = \{ww \mid w \in \{0, 1\}^*\}$

- L_2 no es un LIC
  - $\forall\ n \geq 1$
  - $\exists\ z \in L_2$ con $|z| \geq n$
    - Cadenas que puedan partirse por la mitad en 2 iguales
    - $z = 0^n 1^n 0^n 1^n$
      - ¿Vale?

Elección de la cadena:
su pertenencia al lenguaje debe depender de la coincidencia de grupos de n símbolos separados por, al menos, n símbolos

Ejemplo 2: $L_2 = \{ww \mid w \in \{0, 1\}^*\}$

- L_2 no es un LIC
  - $\forall\, n \geq 1$
  - $\exists\, z \in L_2$ con $|z| \geq n$
    - **$z = 0^n1^n0^n1^n$**
  - $\forall\, u, v, w, x, y \in \Sigma^*: z = uvwxy \wedge 1 \leq |vx| \leq |vwx| \leq n$
    - vwx, como mucho, puede solapar 2 de los 4 grupos de símbolos contiguos
      - Esto garantiza que 2 de los 4 grupos nunca se van a alterar
  - $\exists\, i \geq 0: uv^iwx^iy \notin L_2$
    - Entonces $L_2 \notin$ LIC

Ejemplo 2: $L_2 = \{w'w' \mid w' \in \{0, 1\}^*\}$

- **$z = 0^n1^n0^n1^n$**

- $\forall\, u, v, w, x, y \in \Sigma^*: z = uvwxy \wedge 1 \leq |vx| \leq |vwx| \leq n$
  1. vx tiene apariciones de un único símbolo (de un mismo grupo)
  2. vx tiene apariciones de dos símbolos (de grupos consecutivos)
- $\exists\, i \geq 0: uv^iwx^iy \notin L_2$
  - **$i = 0$**

    $uv^0wx^0y = uwy \notin L_2$ pues no existe w' tal que uwy = w'w'
- Entonces $L_2 \notin$ LIC

Ejemplo 2: cadenas iteradas

- **$z = 0^n1^n0^n1^n$**

1. vx tiene apariciones de un único símbolo (del mismo grupo)
   - Sea $|vx| = j$ (sabemos que $1 \le j \le n$)

Pinta de las cadenas iteradas (en cada caso)

- **$i = 0$**
   1. $uv^0wx^0y = uwy \notin L_2$
      » $0^{n-j}\,1^n\,0^n\,1^n \notin L_2$
      » $0^n\,1^{n-j}\,0^n\,1^n \notin L_2$
      » $0^n\,1^n\,0^{n-j}\,1^n \notin L_2$
      » $0^n\,1^n\,0^n\,1^{n-j} \notin L_2$

   pues no existe w' tal que uwy = w'w' al ser $j \ge 1$

Ejemplo 2: cadenas iteradas

- **$z = 0^n 1^n 0^n 1^n$**

2. vx tiene apariciones de dos símbolos (de grupos consecutivos)
   - Sean $|vx|_0 = s$, $|vx|_1 = t$ (sabemos que $1 \leq s, t < n$)

- **$i = 0$**

  2. $uv^0wx^0y = uwy \notin L_2$
     - $0^{n-s}\, 1^{n-t}\, 0^n\, 1^n \notin L_2$
     - $0^n\, 1^{n-t}\, 0^{n-s}\, 1^n \notin L_2$
     - $0^n\, 1^n\, 0^{n-t}\, 1^{n-s} \notin L_2$

     pues no existe w' tal que uwy = w'w' al ser $s, t \geq 1$

Pinta de las cadenas iteradas (en cada caso)

- Por lo tanto $L_2 \notin$ LIC (demostrados todos los casos)

Lenguajes parecidos

- A L_1
  - $\{a^i b^j c^k \mid i \leq j \leq k\}$
  - $\{x \in \{a, b, c\}^* \mid |x|_a = |x|_b = |x|_c\}$
- A L_2
  - $\{a^i b^i a^i b^i \mid i \geq 0\}$
  - $\{a^i b^j a^i b^j \mid i, j \geq 0\}$
  - $\{wcw \mid w \in \{a, b\}^*\}$
- Más ejemplos…
  - $\{0^{k^2} \mid k \geq 0\}$
  - $\{a^i b^j c^k \mid i < j < k\}$

Últimas consideraciones

- No siempre es posible que todos los casos se traten con la misma iteración
  - Puede elegirse una distinta para cada caso
    - El objetivo es demostrar que existe al menos una iteración para cada caso
- Ejemplo de lenguaje no LIC que cumple la propiedad del LI
  - $\{a^i b^j c^k d^l \mid i = 0 \text{ o } j = k = l\}$
    $= \{b^j c^k d^l \mid j, k, l \geq 0\} \cup \{a^i b^j c^j d^j \mid i > 0, j \geq 0\}$