Introducción a la Lógica y la Computación

Mariana Badano Héctor Gramaglia
Pedro Sánchez Terraf Mauricio Tellechea
Guido Ivetta

FaMAF, 29 de octubre de 2021

UNC Universidad Nacional de Córdoba 1613 - 2013 400 AÑOS

Contenidos estimados para hoy

1. Repaso
   - Expresiones regulares
   - Operaciones con lenguajes
   - $Regex$ definen lenguajes regulares
2. Teorema de Kleene
   - Algoritmo

Operaciones con lenguajes $L \subseteq \Sigma^*$

vacío	$\emptyset$	$\emptyset$
singulete	$\{x\}$ $\quad (x \in \Sigma)$	x
complemento	$L^c = \Sigma^* \setminus L$	
intersección	$L \cap L' = \{\alpha \mid \alpha \in L \text{ y } \alpha \in L'\}$	
unión	$L \cup L' = \{\alpha \mid \alpha \in L \text{ ó } \alpha \in L'\}$	$r + r'$
concatenación	$L\,L' = \{\alpha\beta \mid \alpha \in L \text{ y } \beta \in L'\}$	$r\,r'$
potencias	$L^n = \begin{cases} \{\epsilon\} & \text{si } n = 0 \\ L\,L^k & \text{si } n = k+1 \end{cases}$	$r^0 = \epsilon$ r^n
clausura	$L^* = \bigcup_{n \in \mathbb{N}} L^n$	r^*

Expresiones regulares

Fijado un alfabeto Σ, son el menor conjunto *Regex* que cumple:

vacío	$\emptyset \in Regex.$
épsilon	$\epsilon \in Regex.$
símbolo	$x \in \Sigma \implies \boldsymbol{x} \in Regex.$
unión	$r_1, r_2 \in Regex \implies (r_1 + r_2) \in Regex.$
concatenación	$r_1, r_2 \in Regex \implies r_1\, r_2 \in Regex.$
clausura	$r \in Regex \implies (r)^* \in Regex.$

Expresiones regulares

Fijado un alfabeto Σ, son el menor conjunto $Regex$ que cumple:

vacío	$\emptyset \in Regex$.
épsilon	$\epsilon \in Regex$.
símbolo	$x \in \Sigma \implies \boldsymbol{x} \in Regex$.
unión	$r_1, r_2 \in Regex \implies r_1 + r_2 \in Regex$.
concatenación	$r_1, r_2 \in Regex \implies r_1\, r_2 \in Regex$.
clausura	$r \in Regex \implies r^* \in Regex$.

$$
\begin{aligned}
L(\emptyset) &:= \emptyset \\
L(\epsilon) &:= \{\epsilon\} \\
L(\boldsymbol{x}) &:= \{x\} \\
L(r_1 + r_2) &:= L(r_1) \cup L(r_2) \\
L(r_1\, r_2) &:= L(r_1)\, L(r_2) \\
L(r^*) &:= (L(r))^*
\end{aligned}
$$

concatenación de strings

operación de concatenación de lenguajes

Regex definen lenguajes regulares

Teorema

Para toda expresión regular r, existe un ϵ-NFA $\mathbb{A}(r)$ con un único estado final tal que $L(r) = L(\mathbb{A}(r))$.

Regex definen lenguajes regulares

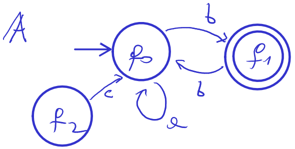
Teorema

Para toda expresión regular r, existe un ϵ-NFA $\mathbb{A}(r)$ con un único estado final tal que $L(r) = L(\mathbb{A}(r))$.

Ahora, vemos la recíproca.

Teorema (Kleene)

Todo lenguaje regular es de la forma $L(r)$ para alguna $r \in Regex$.



Palabra aceptada α camino

$$q_0 \xRightarrow{\alpha} q_1$$

$aabbab \in L(A)$

$$q_0 \xrightarrow{a} q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_1 \xrightarrow{b} q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_1$$

Regex definen lenguajes regulares

Teorema

Para toda expresión regular r, existe un ϵ-NFA $\mathbb{A}(r)$ con un único estado final tal que $L(r) = L(\mathbb{A}(r))$.

Ahora, vemos la recíproca.

Teorema (Kleene)

Todo lenguaje regular es de la forma $L(r)$ para alguna $r \in Regex$.

Daremos un método recursivo para obtener a partir de un ϵ-NFA $\mathbb{A} = (\{q_0, \ldots, q_r\}, \Sigma, \delta, q_0, \{q_f\})$ con un único estado final, una expresión regular que denota el lenguaje $L(\mathbb{A})$.

$F = \{p_f, p_f'\}$.

$L(\mathbb{A}) = \{\alpha : p_0 \xRightarrow{\alpha} p_f$ ó $p_0 \xRightarrow{\alpha} p_f'\}$

$= \{\alpha : p_0 \xRightarrow{\alpha} p_f\} \cup \{\text{——}\}$

Teorema

Para toda expresión regular r, existe un ϵ-NFA $\mathbb{A}(r)$ con un único estado final tal que $L(r) = L(\mathbb{A}(r))$.

Ahora, vemos la recíproca. omitir.

Teorema (Kleene)

Todo lenguaje regular es de la forma $L(r)$ para alguna $r \in Regex$.

Daremos un método recursivo para obtener a partir de un ϵ-NFA $\mathbb{A} = (\{q_0, \ldots, q_r\}, \Sigma, \delta, q_0, \{q_f\})$ con un único estado final, una expresión regular que denota el lenguaje $L(\mathbb{A})$.

La recursión será en $|Q| = |\{q_0, \ldots, q_r\}|$, así que iremos quitando estados para pasar a autómatas más chicos.

Advertencia: A veces voy a llamar "n" al estado q_n

Algoritmo del Teorema de Kleene

- $\mathbb{A} = (\overbrace{\{q_0, \ldots, q_r\}}^{Q}, \Sigma, \delta, q_0, \{q_f\})$.
- $L(\mathbb{A}) =: L_{0f}(Q)$

símbolo extra!

$q \xrightarrow{d} q'$

$d \in \Sigma \cup \{\varepsilon\}$

Algoritmo del Teorema de Kleene

- $\mathbb{A} = (\{q_0, \ldots, q_r\}, \Sigma, \delta, q_0, \{q_f\})$.
- $L(\mathbb{A}) =: L_{0f}(Q)$ $\rightsquigarrow$ palabras que arrancan en q_0 y terminan en q_f.

- $\mathbb{A} = (\{q_0, \ldots, q_r\}, \Sigma, \delta, q_0, \{q_f\})$.
- $L(\mathbb{A}) =: L_{0f}(Q)$ $\rightsquigarrow$ palabras que arrancan en q_0 y terminan en q_f.

En general, supongamos $R \subseteq Q$.

Definición

$L_{nm}(R) :=$ palabras que arrancan en q_n y terminan en q_m, involucrando sólo estados en R.

En part., $q_n, q_m \in R$!

Algoritmo del Teorema de Kleene

- $\mathbb{A} = (\{q_0, \ldots, q_r\}, \Sigma, \delta, q_0, \{q_f\})$.
- $L(\mathbb{A}) =: L_{0f}(Q) \rightsquigarrow$ palabras que arrancan en q_0 y terminan en q_f.

En general, supongamos $R \subseteq Q$.

Definición

$L_{nm}(R) :=$ palabras que arrancan en q_n y terminan en q_m, involucrando sólo estados en R.

Pueden pasar por q_n varias veces

$$\alpha = \alpha_1\alpha_2\alpha_3 \in L_{nm}(R) \rightsquigarrow q_n \stackrel{\alpha_1}{\Longrightarrow} q_n \stackrel{\alpha_2}{\Longrightarrow} q_n \stackrel{\alpha_3}{\Longrightarrow} q_m$$

$aabbab \in L_{01}(Q)$

$$\underbrace{q_0 \xrightarrow{a} q_0}_{\alpha_1}\underbrace{\xrightarrow{a} q_0}_{\alpha_2}\underbrace{\xrightarrow{b} q_1 \xrightarrow{b} q_0}_{\alpha_2'}\underbrace{\xrightarrow{a} q_0}_{\alpha_2''}\underbrace{\xrightarrow{b} q_1}_{\alpha_3}$$

$b \in F_{01}(Q)$

$a, bb \in I_0(Q)$

Algoritmo del Teorema de Kleene

- $\mathbb{A} = (\{q_0, \ldots, q_r\}, \Sigma, \delta, q_0, \{q_f\})$.
- $L(\mathbb{A}) =: L_{0f}(Q) \quad \rightsquigarrow$ palabras que arrancan en q_0 y terminan en q_f.

En general, supongamos $R \subseteq Q$.

Definición

$L_{nm}(R) :=$ palabras que arrancan en q_n y terminan en q_m, involucrando sólo estados en R.

Pueden pasar por q_n varias veces

$$\alpha = \alpha_1\alpha_2\alpha_3 \in L_{nm}(R) \quad \rightsquigarrow \quad q_n \xRightarrow{\alpha_1} q_n \xRightarrow{\alpha_2} q_n \xRightarrow{\alpha_3} q_m$$

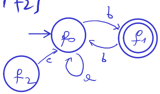
Definición

$I_n(R) :=$ palabras que salen y vuelven a q_n sin pasar en el medio por él (e involucrando sólo estados en R).

$F_{nm}(R) :=$ palabras que salen de q_n y llegan a q_m sin pasar nuevamente por q_n (e involucrando sólo estados en R).

Algoritmo del Teorema de Kleene, primera capa

$Q = \{q_0, q_1, q_2\}$



- $L_{nm}(R) :=$ de q_n a q_m (sólo estados en R).
- $I_n(R) :=$ de q_n a sí mismo sin repetirlo (sólo estados en R).
- $F_{nm}(R) :=$ de q_n a q_m sin repetir q_n (sólo estados en R).

$b \in F_{01}(Q)$

$$\underbrace{q_0 \xrightarrow{a} q_0}_{\alpha_1} \underbrace{\xrightarrow{a} q_0}_{\alpha_2} \underbrace{\xrightarrow{b} q_1 \xrightarrow{b} q_0}_{\alpha_2'} \underbrace{\xrightarrow{a} q_0}_{\alpha_2''} \underbrace{\xrightarrow{b} q_1}_{\alpha_3}$$

$a, bb \in I_0(Q)$

$a \in I_0(\{q_0\}) \not\ni bb$

Algoritmo del Teorema de Kleene, primera capa

- $L_{nm}(R) :=$ de q_n a q_m (sólo estados en R).
- $I_n(R) :=$ de q_n a sí mismo sin repetirlo (sólo estados en R).
- $F_{nm}(R) :=$ de q_n a q_m sin repetir q_n (sólo estados en R).

$$\overbrace{q_n \stackrel{\alpha_1}{\Longrightarrow}}^{I_n(R)} \overbrace{q_n \stackrel{\alpha_2}{\Longrightarrow}}^{I_n(R)} \overbrace{q_n \stackrel{\alpha_3}{\Longrightarrow} q_m}^{F_{nm}(R)}$$

Algoritmo del Teorema de Kleene, primera capa

- $L_{nm}(R)$:= de q_n a q_m (sólo estados en R).
- $I_n(R)$:= de q_n a sí mismo sin repetirlo (sólo estados en R).
- $F_{nm}(R)$:= de q_n a q_m sin repetir q_n (sólo estados en R).

$$\overbrace{q_n \xRightarrow{\alpha_1}}^{I_n(R)} \overbrace{q_n \xRightarrow{\alpha_2}}^{I_n(R)} \overbrace{q_n \xRightarrow{\alpha_3} q_m}^{F_{nm}(R)}$$

Luego

$$L_{nn}(R) := I_n(R)^*$$

$$L_{nm}(R) := I_n(R)^* F_{nm}(R) \qquad \text{si } n \neq m$$

Algoritmo del Teorema de Kleene, ciclos iniciales

- $L_{nm}(R) :=$ de q_n a q_m (sólo estados en R).
- $I_n(R) :=$ de q_n a sí mismo sin repetirlo (sólo estados en R).
- $F_{nm}(R) :=$ de q_n a q_m sin repetir q_n (sólo estados en R).

Algoritmo del Teorema de Kleene, ciclos iniciales

- $L_{nm}(R) :=$ de q_n a q_m (sólo estados en R).
- $I_n(R) :=$ de q_n a sí mismo sin repetirlo (sólo estados en R).
- $F_{nm}(R) :=$ de q_n a q_m sin repetir q_n (sólo estados en R).

$$q_n \underset{\cancel{q_n}}{\overset{a\,\beta\,b}{\Longrightarrow}} q_n$$

Algoritmo del Teorema de Kleene, ciclos iniciales

- $L_{nm}(R) :=$ de q_n a q_m (sólo estados en R).
- $I_n(R) :=$ de q_n a sí mismo sin repetirlo (sólo estados en R).
- $F_{nm}(R) :=$ de q_n a q_m sin repetir q_n (sólo estados en R).

$$q_n \overset{a\,\beta\,b}{\underset{\cancel{q_n}}{\Longrightarrow}} q_n \quad \rightsquigarrow \quad q_n \xrightarrow{a} q_t \overset{\beta}{\Longrightarrow} q_s \xrightarrow{b} q_n$$

Algoritmo del Teorema de Kleene, ciclos iniciales

- $L_{nm}(R) :=$ de q_n a q_m (sólo estados en R).
- $I_n(R) :=$ de q_n a sí mismo sin repetirlo (sólo estados en R).
- $F_{nm}(R) :=$ de q_n a q_m sin repetir q_n (sólo estados en R).

$$q_n \overset{a\,\beta\,b}{\underset{\cancel{q_n}}{\Longrightarrow}} q_n \quad \rightsquigarrow \quad q_n \xrightarrow{a} \underbrace{q_t \overset{\beta}{\Longrightarrow} q_s}_{L_{ts}(R \setminus \{q_n\})} \xrightarrow{b} q_n$$

- $L_{nm}(R)$:= de q_n a q_m (sólo estados en R).
- $I_n(R)$:= de q_n a sí mismo sin repetirlo (sólo estados en R).
- $F_{nm}(R)$:= de q_n a q_m sin repetir q_n (sólo estados en R).

$$q_n \xRightarrow[\cancel{q_n}]{a\,\beta\,b} q_n \rightsquigarrow q_n \xrightarrow{a} \underbrace{q_t \xRightarrow{\beta} q_s}_{L_{ts}(R\setminus\{q_n\})} \xrightarrow{b} q_n$$

¡ojo! sólo contempla trozos de long. 2 o más.

Luego

$$I_n(R) := \sum_{\substack{q_n \xrightarrow{a} q_t \\ q_s \xrightarrow{b} q_n}} a\,L_{ts}(R \setminus \{q_n\})\,b$$

- $L_{nm}(R)$:= de q_n a q_m (sólo estados en R).
- $I_n(R)$:= de q_n a sí mismo sin repetirlo (sólo estados en R).
- $F_{nm}(R)$:= de q_n a q_m sin repetir q_n (sólo estados en R).

$$q_n \xRightarrow[\cancel{q_n}]{a\,\beta\,b} q_n \quad \rightsquigarrow \quad q_n \xrightarrow{a} \underbrace{q_t \xRightarrow{\beta} q_s}_{L_{ts}(R\setminus\{q_n\})} \xrightarrow{b} q_n$$

Luego

$$I_n(R) := \sum_{\substack{q_n \xrightarrow{a} q_t \\ q_s \xrightarrow{b} q_n}} a\, L_{ts}(R \setminus \{q_n\})\, b \;+\; \sum_{q_n \xrightarrow{c} q_n} c \qquad (n \neq t, s)$$

$a, b, c \in \Sigma \cup \{\varepsilon\}$

Algoritmo del Teorema de Kleene, camino al final

- $L_{nm}(R) :=$ de q_n a q_m (sólo estados en R).
- $I_n(R) :=$ de q_n a sí mismo sin repetirlo (sólo estados en R).
- $F_{nm}(R) :=$ de q_n a q_m sin repetir q_n (sólo estados en R).

Algoritmo del Teorema de Kleene, camino al final

- $L_{nm}(R) :=$ de q_n a q_m (sólo estados en R).
- $I_n(R) :=$ de q_n a sí mismo sin repetirlo (sólo estados en R).
- $F_{nm}(R) :=$ de q_n a q_m sin repetir q_n (sólo estados en R).

$$q_n \overset{a\,\beta}{\underset{\cancel{q_n}}{\Longrightarrow}} q_m$$

Algoritmo del Teorema de Kleene, camino al final

- $L_{nm}(R) :=$ de q_n a q_m (sólo estados en R).
- $I_n(R) :=$ de q_n a sí mismo sin repetirlo (sólo estados en R).
- $F_{nm}(R) :=$ de q_n a q_m sin repetir q_n (sólo estados en R).

$$q_n \overset{a\,\beta}{\underset{\cancel{q_n}}{\Longrightarrow}} q_m \quad \rightsquigarrow \quad q_n \xrightarrow{a} q_t \overset{\beta}{\Longrightarrow} q_m$$

Algoritmo del Teorema de Kleene, camino al final

- $L_{nm}(R) :=$ de q_n a q_m (sólo estados en R).
- $I_n(R) :=$ de q_n a sí mismo sin repetirlo (sólo estados en R).
- $F_{nm}(R) :=$ de q_n a q_m sin repetir q_n (sólo estados en R).

$$q_n \overset{a\,\beta}{\underset{\cancel{q_n}}{\Longrightarrow}} q_m \quad \rightsquigarrow \quad q_n \xrightarrow{a} \underbrace{q_t \overset{\beta}{\Longrightarrow} q_m}_{L_{tm}(R \smallsetminus \{q_n\})}$$

Algoritmo del Teorema de Kleene, camino al final

- $L_{nm}(R)$:= de q_n a q_m (sólo estados en R).
- $I_n(R)$:= de q_n a sí mismo sin repetirlo (sólo estados en R).
- $F_{nm}(R)$:= de q_n a q_m sin repetir q_n (sólo estados en R).

$$q_n \overset{a\,\beta}{\underset{\cancel{q_n}}{\Longrightarrow}} q_m \quad \rightsquigarrow \quad q_n \xrightarrow{a} \underbrace{q_t \overset{\beta}{\Longrightarrow} q_m}_{L_{tm}(R\setminus\{q_n\})}$$

De igual modo,

$$F_{nm}(R) := \sum_{q_n \xrightarrow{a} q_t} a\,L_{tm}(R \setminus \{q_n\}) \qquad (n \neq t, m)$$

$a \in \Sigma \cup \{\varepsilon\}$

No olvidar los casos base

Algoritmo recursivo de Kleene

$$
\begin{aligned}
L_{nm}(R) &:= \emptyset && \text{si } q_n \text{ ó } q_m \text{ no están en } R \\
L_{nn}(R) &:= I_n(R)^* \\
L_{nm}(R) &:= I_n(R)^* F_{nm}(R) && \text{si } n \neq m \\
I_n(R) &:= \sum_{\substack{q_n \xrightarrow{a} q_t \\ q_s \xrightarrow{b} q_n}} a\, L_{ts}(R \setminus \{q_n\})\, b \;+\; \sum_{q_n \xrightarrow{c} q_n} c && (n \neq t, s) \\
F_{nm}(R) &:= \sum_{q_n \xrightarrow{a} q_t} a\, L_{tm}(R \setminus \{q_n\}) && (n \neq t, m)
\end{aligned}
$$

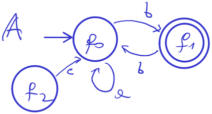
$$L_{nm}(R) := \emptyset \quad \text{si } \{q_n, q_m\} \not\subseteq R$$

$$L_{nn}(R) := I_n(R)^*$$

$$L_{nm}(R) := I_n(R)^* F_{nm}(R) \quad \text{si } n \neq m$$

$$I_n(R) := \sum_{\substack{q_n \xrightarrow{a} q_t \\ q_s \xrightarrow{b} q_n}} a\, L_{ts}(R \setminus \{q_n\})\, b \;+\; \sum_{q_n \xrightarrow{c} q_n} c$$

$$F_{nm}(R) := \sum_{q_n \xrightarrow{a} q_t} a\, L_{tm}(R \setminus \{q_n\}) \qquad (n \neq t, s, m)$$



$$L(A) = L_{01}(Q) = I_0(Q)^* F_{01}(Q)$$

$q_0 \circlearrowleft$	$q_0 \longrightarrow$	$\longrightarrow q_0$
$q_0 \xrightarrow{a} q_0$	A $q_0 \xrightarrow{b} q_1$	$q_1 \xrightarrow{b} q_0$ B $q_2 \xrightarrow{c} q_0$ C

$$I_0(Q) = \overbrace{\underset{A}{b}\, L_{11}(\{q_1, q_2\})\, \underset{B}{b}}^{bb} + \underset{A}{b}\, L_{12}(\{q_1, q_2\})\, \underset{C}{c} + a$$

$$L_{11}(\{q_1, q_2\}) = I_1(\{q_1, q_2\})^* = \emptyset^* = \varepsilon$$

$$I_1(\{q_1, q_2\}) = \emptyset + \emptyset = \emptyset$$

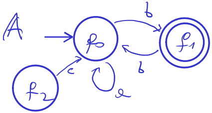
$L_{nm}(R) := \emptyset$ si $\{q_n, q_m\} \not\subseteq R$

$L_{nn}(R) := I_n(R)^*$

$L_{nm}(R) := I_n(R)^* F_{nm}(R)$ si $n \neq m$

$$I_n(R) := \sum_{\substack{q_n \xrightarrow{a} q_t \\ q_s \xrightarrow{b} q_n}} a\, L_{ts}(R \setminus \{q_n\})\, b \;+\; \sum_{q_n \xrightarrow{c} q_n} c$$

$$F_{nm}(R) := \sum_{q_n \xrightarrow{a} q_t} a\, L_{tm}(R \setminus \{q_n\}) \qquad (n \neq t, s, m)$$



$L(A) = L_{01}(Q) = I_0(Q)^* F_{01}(Q)$

$I_0(Q) = bb + b\,L_{12}(\{p_1, p_2\})\,c + a$

$L_{12}(p_1, p_2) = I_1(p_1, p_2)^* F_{12}(p_1, p_2) = \emptyset^* F_{12}(p_1, p_2)$

$I_1(\{p_1, p_2\}) = \emptyset$

$= \varepsilon\, F_{12}(p_1, p_2)$

$= F_{12}(p_1, p_2)$

$F_{12}(p_1, p_2) = \emptyset$

$L_{12}(p_1, p_2) = \emptyset$

$I_0(Q) = bb + b\emptyset c + a = bb + a.$

$L_{nm}(R) := \emptyset$ si $\{q_n, q_m\} \not\subseteq R$

$L_{nn}(R) := I_n(R)^*$

$L_{nm}(R) := I_n(R)^* F_{nm}(R)$ si $n \neq m$

$$I_n(R) := \sum_{\substack{q_n \xrightarrow{a} q_t \\ q_s \xrightarrow{b} q_n}} a\, L_{ts}(R \setminus \{q_n\})\, b \;+\; \sum_{q_n \xrightarrow{c} q_n} c$$

$$F_{nm}(R) := \sum_{q_n \xrightarrow{a} q_t} a\, L_{tm}(R \setminus \{q_n\}) \qquad (n \neq t, s, m)$$

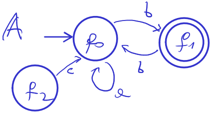
$L(A) = L_{01}(Q) = I_0(Q)^* F_{01}(Q)$

$I_0(Q) = bb + a$

$L_{01}(Q) = (bb+a)^* F_{01}(Q)$

una sola forma

$q_0 \xrightarrow{b} q_1$



$F_{01}(Q) = b\, L_{11}(\{q_1, q_2\}) = b\varepsilon = b.$

$$\boxed{L_{01}(Q) = (bb+a)^* b}$$

" ~ " = "mismo L(.)"

Simplificaciones (lenguajes denotados)

$\varepsilon r \sim r \varepsilon \sim r$

$\emptyset + r \sim r + \emptyset \sim r$

$\emptyset^* \sim \varepsilon$

$$L_{nm}(R) := \emptyset \quad \text{si } \{q_n, q_m\} \nsubseteq R$$

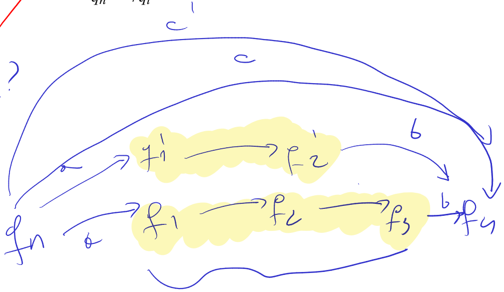
$$L_{nn}(R) := I_n(R)^*$$

$$L_{nm}(R) := I_n(R)^* F_{nm}(R) \quad \text{si } n \neq m$$

$$I_n(R) := \sum_{\substack{q_n \xrightarrow{a} q_t \\ q_s \xrightarrow{b} q_n}} a\, L_{ts}(R \setminus \{q_n\})\, b \;+ \sum_{q_n \xrightarrow{c} q_n} c$$

$$F_{nm}(R) := \sum_{q_n \xrightarrow{a} q_t} a\, L_{tm}(R \setminus \{q_n\}) \qquad (n \neq t, s, m)$$

Consulta fuera de grabación: ¿por qué hay Σ?



Respuesta: hay que considerar todos los caminos de q_n a q_n.

Pueden involucrar diversas combinaciones de salir/llegar

($q_n \xrightarrow{a} q_1'$ con $q_2' \xrightarrow{a} q_n$
$q_n \xrightarrow{a} q_1'$ con $q_2 \xrightarrow{b} q_n$
etc.) q_n